Chapter 14

Optimization Transfer Methods

ch,ox

Contents

14.1 Introduction (s,ox,intro)

s,ox,intro

Many of the algorithms described in this *book* were either derived originally using the **optimization transfer** principle, or can be explained retrospectively using that principle. The concept of optimization transfer is very useful for deriving algorithms that monotonically decrease the cost function. In particular, the optimization transfer approach facilitates the design of algorithms that are tailored to specific types of cost functions of interest, in contrast to the "one size fits all" approach of general-purpose optimization methods such as **gradient descent**. The optimization transfer approach also accommodates easily constraints such as nonnegativity.

This chapter first describes optimization transfer methods in general. Then we describe the **expectation maximization** (**EM**) method and its generalizations, which are special cases of the optimization transfer approach. Probably the most commonly used optimization transfer methods are those based on **quadratic** majorizers, particularly separable quadratic surrogates, so this chapter emphasizes these methods. For convergence theory of optimization transfer methods, see [1, 2].

14.1.1 History

The history of **optimization transfer** methods is somewhat diffuse. See [3] for a discussion paper with considerable debate about this topic! The classic text by Ortega and Rheinbolt described briefly a "**majorization principle**" in the limited context of 1D line searches [4, p. 253]. Huber [5, p. 184] referred to a "**comparison function**" in the context of an algorithm for robust linear regression (see §14.3.1.2). The EM family of methods [6] uses a statistical form of optimization transfer (see §14.9). In some areas of statistics, the approach has been called **iterative majorization** [7, 8]. In the context of imaging and inverse problems, the utility of optimization transfer was clearly demonstrated in the relatively recent work of De Pierro [9–12], with extensions by Lange [13–16] and others [17–20]. (For a nice tutorial, see [21], in which the term **MM algorithm** is advocated, meaning "majorize, minimize" or "minorize, maximize.") An entire book on this topic is [22]. Such methods have also been applied to computer vision problems such as **deformable models**, *e.g.*, [23]. Allain et al. [24] cite a 1937 paper by Weiszfeld [25, 26] as an early example of majorization. Regardless of who was first, the principles are very useful and form the foundation for many algorithms.

14.1.2 Optimization transfer principle

Fig. 14.1.1 illustrates the basic idea of optimization transfer. When faced with a cost function $\Psi : \mathcal{X} \subset \mathbb{R}^{n_{\mathrm{p}}} \to \mathbb{R}$ that is difficult to minimize, at the nth iteration we replace Ψ with a **surrogate function** or **majorizer** $\phi^{(n)}(\boldsymbol{x})$ that is easier to minimize. (Usually $\phi^{(n)}$ will depend on $\boldsymbol{x}^{(n)}$, but we leave this dependence implicit.) Usually, minimizing $\phi^{(n)}$ will not yield the global minimizer $\hat{\boldsymbol{x}}$ of Ψ in one step, so we must repeat the process. We alternate between the "S-step:" choosing a surrogate function $\phi^{(n)}$, and the "M-step:" finding the minimizer of $\phi^{(n)}$. This approach can be summarized in general as follows[1].

Optimization transfer or MM method

S-step (majorize): choose a surrogate $\phi^{(n)}$ satisfying (14.1.2) below

$$\text{M-step (minimize): } \boldsymbol{x}^{(n+1)} \triangleq \arg\min_{\boldsymbol{x} \in \mathcal{X}} \phi^{(n)}(\boldsymbol{x}) \,. \tag{14.1.1}$$

e,ox,alg

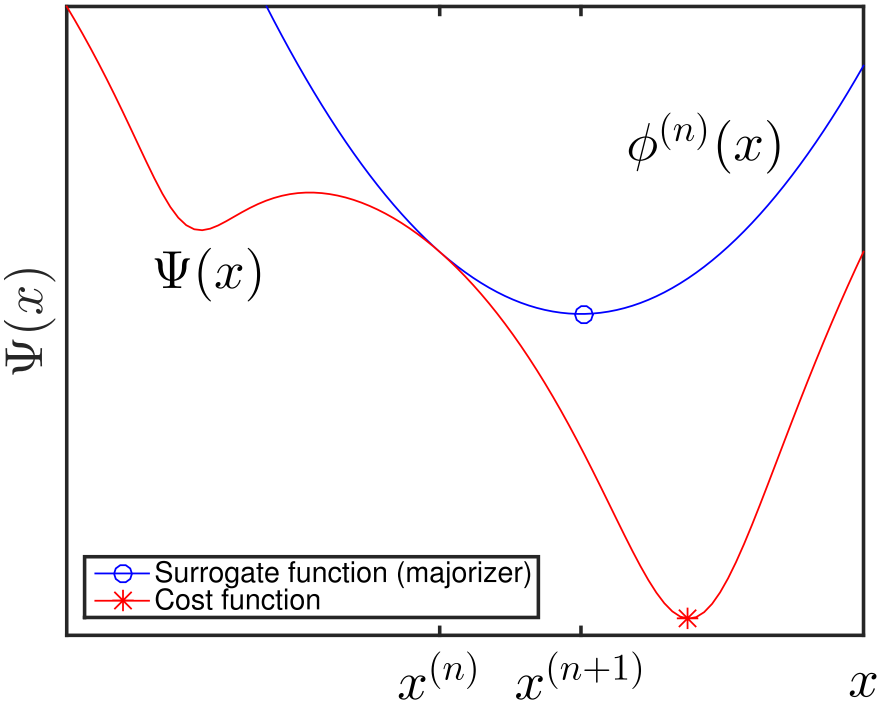
The minimization step[2] is restricted to the valid parameter space (*e.g.*, $\boldsymbol{x} \succeq \boldsymbol{0}$ for problems with nonnegative constraints). If we choose the surrogate functions and initializer $\boldsymbol{x}^{(0)}$ appropriately, then the sequence $\{\boldsymbol{x}^{(n)}\}$ is well defined and should converge to a minimizer $\hat{\boldsymbol{x}}$ eventually (see §14.5) under suitable regularity conditions.

The family of **proximal point methods** *e.g.*, [27] [28, p. 270] [29], could also be considered to be optimization transfer methods, although the surrogate functions used in some proximal point literature are often less easily optimized than the types of surrogates used in the optimization transfer literature.

Fig. 14.1.1 does not do full justice to the problem, because 1D functions are usually fairly easy to minimize. The optimization transfer principle is particularly compelling for problems where the dimension of $\boldsymbol{x}$ is large, such as in **inverse problems** like tomography. Fig. 14.1.2 shows an example of a cost function and a quadratic surrogate function where $n_{\mathrm{p}} = 2$.

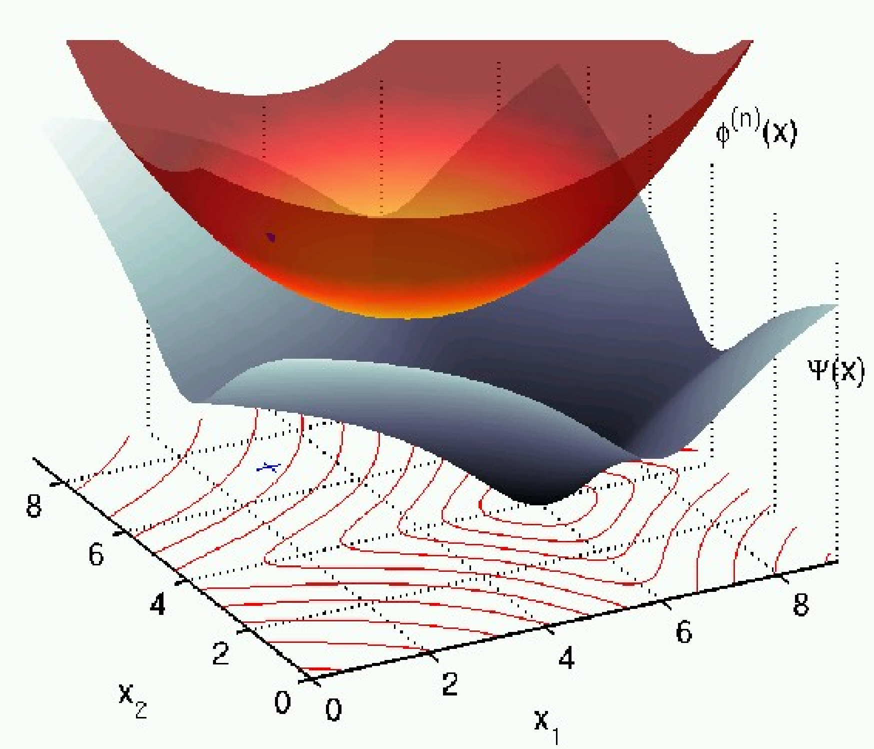
[1] In his colorful discussion of [3], Meng proposed calling this the **SM method**, essentially as an acronym for "surrogate substitution / minimization."

[2] If the surrogate $\phi^{(n)}$ has multiple minimizers, then the "argmin" in the minimization step (14.1.1) should be interpreted as giving *any* of those minimizers, *i.e.*, $\boldsymbol{x}^{(n+1)} \in \arg\min_{\boldsymbol{x}} \phi^{(n)}(\boldsymbol{x})$.



fig_ox_surr_1d_min

Figure 14.1.1: Illustration of a surrogate function $\phi^{(n)}(x)$ for optimization transfer in 1D.



fig_ox_surr_2d_min

Figure 14.1.2: Illustration of a cost function Ψ and a quadratic surrogate function $\phi^{(n)}$ for $n_{\mathrm{p}} = 2$.

14.1.3 Monotonicity

If we choose surrogate functions[3] that satisfy the following condition

e,ox,mono

$$\Psi(\boldsymbol{x}^{(n)}) - \Psi(\boldsymbol{x}) \geq \phi^{(n)}(\boldsymbol{x}^{(n)}) - \phi^{(n)}(\boldsymbol{x}), \qquad \forall \boldsymbol{x}, \boldsymbol{x}^{(n)} \in \mathcal{X}, \tag{14.1.2}$$

then one can see immediately that the update (14.1.1) will monotonically decrease Ψ as in (11.1.9). Readers familiar with EM methods (§14.9) will recognize that the above approach is a natural generalization.

The various algorithms described in the sections that follow are all based on different choices of the surrogate function $\phi^{(n)}$, and on different procedures for the minimization in (14.1.1).

To ensure monotonicity, it is not essential to find the exact minimizer in (14.1.1). It suffices to find a value $\boldsymbol{x}^{(n+1)} \in \mathcal{X}$ such that $\phi^{(n)}\big(\boldsymbol{x}^{(n+1)}\big) \leq \phi^{(n)}(\boldsymbol{x}^{(n)})$, because that alone will ensure that $\Psi\big(\boldsymbol{x}^{(n+1)}\big) \leq \Psi(\boldsymbol{x}^{(n)})$ by (14.1.2). However, such incomplete minimization in the M-step complicates convergence analysis [31].

Rather than working directly with the monotonicity condition (14.1.2), all the surrogate functions we present are designed to satisfy the following **surrogate conditions**:

e,ox,sur

$$\phi^{(n)}(\boldsymbol{x}^{(n)}) = \Psi(\boldsymbol{x}^{(n)}) \qquad \text{“matched } \Psi \text{ value”} \tag{14.1.3}$$

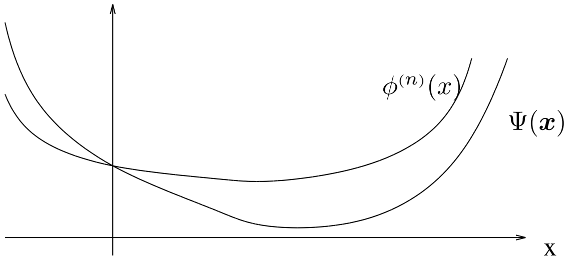
$$\phi^{(n)}(\boldsymbol{x}) \geq \Psi(\boldsymbol{x}), \; \forall \boldsymbol{x} \in \mathcal{X} \qquad \text{“lies above”.} \tag{14.1.4}$$

Any surrogate function that satisfies these conditions will satisfy (14.1.2). The conditions (14.1.4) are sometimes called **majorization** conditions. When Ψ and $\phi^{(n)}$ are differentiable, one can show that the following “**matched tangent**” condition holds for $\boldsymbol{x}^{(n)}$ in the **interior** of $\mathcal{X}$ (see Problem 14.2) due to (14.1.4):

e,ox,sur,tangent

$$\nabla_{\boldsymbol{x}} \, \phi^{(n)}(\boldsymbol{x}) \Big|_{\boldsymbol{x}=\boldsymbol{x}^{(n)}} = \nabla \, \Psi(\boldsymbol{x}) \Big|_{\boldsymbol{x}=\boldsymbol{x}^{(n)}} . \tag{14.1.5}$$

Fig. 14.1.3 illustrates why consideration of the interior is needed.



fig_ox_surr_grad

Figure 14.1.3: Illustration of a majorizing surrogate function $\phi^{(n)}(x)$ on $[0, \infty)$ that does not satisfy (14.1.5).

All of the optimization transfer methods considered in this *book* have surrogate functions that satisfy (14.1.2). However, it is interesting to consider that (14.1.2) is unnecessarily restrictive. Any sensible optimization transfer method will always find $\boldsymbol{x}^{(n+1)}$ that decreases $\phi^{(n)}$, so the condition “$\forall \boldsymbol{x}, \boldsymbol{x}^{(n)}$” in (14.1.2) could be relaxed to

$$\forall \boldsymbol{x} \in \left\{\boldsymbol{x} : \, \phi^{(n)}(\boldsymbol{x}) \leq \phi^{(n)}(\boldsymbol{x}^{(n)})\right\}.$$

Equivalently, one could simply define $\phi^{(n)}$ to be equal to Ψ outside of this set. This generalization could lead to faster converging algorithms because it enlarges the space of admissible surrogates.

The convergence results in Chapter 14 apply broadly to algorithms based on the optimization transfer principle, thanks to the monotonicity property.

14.1.4 Optimization transfer versus augmentation

A close cousin of the optimization transfer approach is the method of **augmentation** [3, 32, 33], in which one finds a function Φ for which

$$\Psi(\boldsymbol{x}) = \min_{\boldsymbol{z} \in \mathcal{Z}} \Phi(\boldsymbol{x}, \boldsymbol{z}),$$

[3] The term surrogate function has also been used to describe mere approximations to the cost function [30]. Here we use the term only when describing functions that satisfy (14.1.2) or (14.1.4)

where $\mathcal{Z}$ need not be $\mathbb{R}^{n_{\mathrm{p}}}$. When such a construction is available, a natural minimization approach is the following "**block relaxation**" method

e,ox,augmentation

$$\begin{aligned} \boldsymbol{z}^{(n)} &= \arg\min_{\boldsymbol{z}\in\mathcal{Z}} \Phi(\boldsymbol{x}^{(n)}, \boldsymbol{z}) \\ \boldsymbol{x}^{(n+1)} &= \arg\min_{\boldsymbol{x}\in\mathcal{X}} \Phi(\boldsymbol{x}, \boldsymbol{z}^{(n)}). \end{aligned} \tag{14.1.6}$$

If $\phi^{(n)}(\boldsymbol{x}) = \phi(\boldsymbol{x}; \boldsymbol{x}^{(n)})$ is a surrogate for Ψ, then by (14.1.4): $\Psi(\boldsymbol{x}) = \min_{\bar{\boldsymbol{x}}\in\mathcal{X}} \phi(\boldsymbol{x}; \bar{\boldsymbol{x}})$, so $\Phi = \phi$ provides an augmentation method with $\mathcal{Z} = \mathbb{R}^{n_{\mathrm{p}}}$. Because $\mathcal{Z}$ need not be $\mathbb{R}^{n_{\mathrm{p}}}$ for a general augmentation method, the above references have concluded that augmentation is more general than optimization transfer. However, if Φ is any augmentation function, then we can define the following surrogate function

$$\phi^{(n)}(\boldsymbol{x}) \triangleq \Phi\left(\boldsymbol{x}, \arg\min_{\boldsymbol{z}\in\mathcal{Z}} \Phi(\boldsymbol{x}^{(n)}, \boldsymbol{z})\right),$$

so the two approaches are equivalent.

s,ox,rate

14.1.5 Asymptotic convergence rate (s,ox,rate)

Postponing the analysis of convergence until §14.5, we presume convergence here and consider the **convergence rate**. For iterative algorithms based on the optimization transfer principle, we can discuss the convergence rate qualitatively by considering Figs. 14.1.1 and 14.1.2. If the surrogate function $\phi^{(n)}$ has low curvature, then it appears as a "broad" graph in Figs. 14.1.1 and 14.1.2, so the algorithm can take large steps ($\left\|\boldsymbol{x}^{(n+1)} - \boldsymbol{x}^{(n)}\right\|$ can be large) so $\{\boldsymbol{x}^{(n)}\}$ should approach a minimizer quickly. Conversely, if the surrogate function has high curvature, then it appears as a "skinny" graph, the steps are small, slowing convergence. In general we would like to find low-curvature surrogate functions, with the caveat that we want to maintain the majorization condition $\phi^{(n)} \geq \Psi$ to ensure monotonicity [34]. And of course we would also like the surrogate $\phi^{(n)}$ to be easy to minimize for (14.1.1). Unfortunately, the criteria "low curvature" and "easy to minimize" are often incompatible, so we must compromise. This trade-off will be a recurring theme throughout this *book*.

To analyze the **asymptotic convergence rate** of a general optimization transfer method (14.1.1), we consider the usual case where $\phi^{(n)}(\cdot) = \phi(\cdot, \boldsymbol{x}^{(n)})$ for every iteration, *i.e.*, the dependence on iteration is only through $\boldsymbol{x}^{(n)}$, as opposed to the more complicated case where the *form* of the surrogate function ϕ also changes with iteration n. Assuming that $\phi^{(n)}$ is twice differentiable, consider the following second-order Taylor expansion of $\phi^{(n)}$ about $\boldsymbol{x}^{(n)}$:

$$\phi^{(n)}(\boldsymbol{x}) \approx \phi^{(n)}(\boldsymbol{x}^{(n)}) + (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \nabla \phi^{(n)}(\boldsymbol{x}^{(n)}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{(n)})' \nabla^2 \phi^{(n)}(\boldsymbol{x}^{(n)})(\boldsymbol{x} - \boldsymbol{x}^{(n)}).$$

Thus

$$\nabla \phi^{(n)}(\boldsymbol{x}) \approx \nabla \phi^{(n)}(\boldsymbol{x}^{(n)}) + \nabla^2 \phi^{(n)}(\boldsymbol{x}^{(n)})(\boldsymbol{x} - \boldsymbol{x}^{(n)}).$$

Equating this approximation to zero and solving yields the following approximation for the optimization transfer update (14.1.1):

e,ox,step1

$$\boldsymbol{x}^{(n+1)} \approx \boldsymbol{x}^{(n)} - \left[\nabla^2 \phi^{(n)}(\boldsymbol{x}^{(n)})\right]^{-1} \nabla \phi^{(n)}(\boldsymbol{x}^{(n)}). \tag{14.1.7}$$

This approximation should be reasonably accurate near convergence when $\boldsymbol{x}^{(n)}$ is changing relatively little. (Of course we are ignoring any constraints on the parameter vector $\boldsymbol{x}$ in this analysis.)

To simplify further, we likewise differentiate a second-order Taylor expansion about $\hat{\boldsymbol{x}}$ of the cost function (*cf.* (11.3.11)) yielding

$$\nabla \Psi(\boldsymbol{x}^{(n)}) \approx \nabla \Psi(\hat{\boldsymbol{x}}) + \nabla^2 \Psi(\hat{\boldsymbol{x}})(\boldsymbol{x}^{(n)} - \hat{\boldsymbol{x}}) = \nabla^2 \Psi(\hat{\boldsymbol{x}})(\boldsymbol{x}^{(n)} - \hat{\boldsymbol{x}});$$

combining with (14.1.5) yields

$$\nabla \phi^{(n)}(\boldsymbol{x}^{(n)}) = \nabla\Psi(\boldsymbol{x}^{(n)}) \approx \nabla^2\Psi(\hat{\boldsymbol{x}})(\boldsymbol{x}^{(n)} - \hat{\boldsymbol{x}}).$$

Combining with (14.1.7) yields the following approximation:

e,ox,rate

$$\boldsymbol{x}^{(n+1)} - \hat{\boldsymbol{x}} \approx \left(\boldsymbol{I} - \left[\nabla^2 \phi^{(n)}(\boldsymbol{x}^{(n)})\right]^{-1} \nabla^2\Psi(\hat{\boldsymbol{x}})\right)(\boldsymbol{x}^{(n)} - \hat{\boldsymbol{x}}). \tag{14.1.8}$$

Assuming that $\nabla^2 \phi^{(n)}(\boldsymbol{x})$ is continuous in *both* $\boldsymbol{x}$ and $\boldsymbol{x}^{(n)}$, it will converge to some limit $\nabla^2 \phi(\hat{\boldsymbol{x}})$ as $\boldsymbol{x}^{(n)} \to \hat{\boldsymbol{x}}$. So the **asymptotic convergence rate** of an optimization transfer algorithm (as defined by the **root convergence factor** in §28.14) is governed by the **spectral radius** of

e,ox,rate,matrix

$$\boldsymbol{I} - \left[\nabla^2 \phi(\hat{\boldsymbol{x}})\right]^{-1} \nabla^2 \Psi(\hat{\boldsymbol{x}}) . \tag{14.1.9}$$

From this expression, we see that if the curvature of the surrogate function greatly exceeds that of the original cost function, *i.e.*, if $\nabla^2 \phi \gg \nabla^2 \Psi$, then the resulting optimization transfer algorithm will converge *very* slowly. This relationship is examined quantitatively in [34] (*cf.* §15.5.3).

The above arguments can be refined to show **local convergence** in norm of $\boldsymbol{x}^{(n)}$ to $\hat{\boldsymbol{x}}$ under various regularity conditions on Ψ and ϕ [35].

Using the **Loewner partial order** properties of **positive semi-definite matrices** (*cf.* §27.2.2), one can show that if $\boldsymbol{0} \prec \nabla^2 \Psi \preceq \nabla^2 \phi$, then $\boldsymbol{0} \prec \left[\nabla^2 \phi\right]^{-1/2} \left(\nabla^2 \Psi\right) \left[\nabla^2 \phi\right]^{-1/2} \preceq \boldsymbol{I}$, so the eigenvalues of $\left[\nabla^2 \phi\right]^{-1/2} \left(\nabla^2 \Psi\right) \left[\nabla^2 \phi\right]^{-1/2}$ lie in the interval $(0, 1]$ and thus by (27.1.1) so do the eigenvalues of $\left[\nabla^2 \phi\right]^{-1} \left(\nabla^2 \Psi\right)$. Thus the eigenvalues of (14.1.9) lie in the interval $[0, 1)$ so its spectral radius is less than unity:

e,ox,rate,rho<1

$$\rho\Big(\boldsymbol{I} - \left[\nabla^2 \phi(\hat{\boldsymbol{x}})\right]^{-1} \nabla^2 \Psi(\hat{\boldsymbol{x}})\Big) < 1. \tag{14.1.10}$$

14.2 Surrogate minimization methods (s,ox,min)

s,ox,min

After one chooses a surrogate function $\phi^{(n)}$, the next step is to minimize it per (14.1.1). For some choices of surrogates, the M-step is trivial because there is an easily implemented expression for $\boldsymbol{x}^{(n+1)}$. For other choices, there may not be a closed-form expression for the minimizer $\boldsymbol{x}^{(n+1)}$, so an iterative approach to the M-step would seem necessary, and in principle any of the general-purpose minimization methods of Chapter 11 could be applied. However, because $\phi^{(n)}$ is only a surrogate for the actual cost function Ψ, it may be inefficient to try to minimize precisely the surrogate for the M-step. It may be more reasonable to simply descend $\phi^{(n)}$, perhaps using just one iteration of some iterative minimization method, and then to find a new surrogate. The remainder of this section summarizes a few strategies.

14.2.1 Nested surrogates

If the initial surrogate $\phi^{(n)}$ is too difficult to minimize directly, then in many cases one can find a second surrogate $\phi_2^{(n)}$ that is a surrogate for $\phi^{(n)}$, *i.e.*, that satisfies the following conditions which are analogous to (14.1.4):

$$\begin{aligned} \phi_2^{(n)}(\boldsymbol{x}^{(n)}) &= \phi^{(n)}(\boldsymbol{x}^{(n)}) \\ \phi_2^{(n)}(\boldsymbol{x}) &\geq \phi^{(n)}(\boldsymbol{x}) . \end{aligned}$$

These conditions ensure that if we minimize (or simply decrease) $\phi_2^{(n)}$, then $\phi^{(n)}$ will decrease, and hence the cost function Ψ will also monotonically decrease. Several of the algorithms in this *book* for complicated cost functions use such **nested surrogate** functions.

14.2.2 One Newton step for the surrogate

Rather than iteratively minimizing the surrogate function, a simple approach would be to apply one step of Newton's method to the surrogate, assuming that $\phi^{(n)}$ is twice differentiable. This leads to the following update:

e,ox,min,newton

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \left[\nabla^2 \phi^{(n)}(\boldsymbol{x}^{(n)})\right]^{-1} \nabla \Psi(\boldsymbol{x}^{(n)}) . \tag{14.2.1}$$

In general this approach is not guaranteed to be monotonic. Interestingly, despite the "incomplete" minimization, this method has the same asymptotic convergence rate as analyzed in §14.1.5 [16, p. 146].

For nonseparable surrogate functions, the above method would be implemented in practice by the update $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \boldsymbol{\delta}^{(n)}$, where $\boldsymbol{\delta}^{(n)}$ is the solution (or an approximate solution) to the linear system of equations:

$$\left[\nabla^2 \phi^{(n)}(\boldsymbol{x}^{(n)})\right] \boldsymbol{\delta}^{(n)} = \nabla \Psi(\boldsymbol{x}^{(n)}) .$$

For imaging problems, it is usually very expensive to solve this system of equations unless the surrogate is separable (in which case its Hessian is diagonal).

If the majorizer $\phi^{(n)}$ is **quadratic** and strictly convex, then (14.2.1) provides the *exact* minimizer of $\phi^{(n)}$. In particular, if the Hessian of $\phi^{(n)}$ is $L\boldsymbol{I}$, where L denotes the Lipschitz constant of $\nabla\Psi$, then (14.2.1) simplifies to

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \frac{1}{L}\nabla\Psi(\boldsymbol{x}^{(n)}),$$

which is exactly the **gradient descent** (**GD**) method described in §11.3. Optimization transfer methods generalize GD methods by allowing other majorizers.

14.2.3 Surrogate minimization by quasi-Newton

If the surrogate is separable, so its Hessian is diagonal, then the update (14.2.1) will converge slowly when the surrogate Hessian poorly approximates the Hessian of Ψ. Convergence can be accelerated by applying a quasi-Newton update to form an improved Hessian approximation using the Hessian of the surrogate as a starting point [16, p. 148] [36, 37]. This acceleration method could be useful in imaging problems where constraints such as nonnegativity are unimportant.

s,ox,min,psd

14.2.4 Surrogate preconditioned steepest descent

The M-step (14.1.1) is a n_{p}-dimensional minimization problem. If this is impractical, then an alternative is to define a search direction based on the gradient of the cost function (assuming it is differentiable), and then search for the minimizer *of the surrogate function* $\phi^{(n)}$ in that direction. More generally, we can apply a positive definite preconditioning matrix to gradient to define the search direction:

e,ox,min,psd,dirn

$$\boldsymbol{d}^{(n)} = -\boldsymbol{P}\,\nabla\Psi(\boldsymbol{x}^{(n)})\,. \tag{14.2.2}$$

We then perform a 1D search in that direction as follows (*cf.* §11.5):

e,ox,min,psd,alfn

$$\alpha_n = \arg\min_{\alpha} \phi^{(n)}(\boldsymbol{x}^{(n)} + \alpha \boldsymbol{d}^{(n)}), \tag{14.2.3}$$

leading to the following update

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \alpha_n \boldsymbol{d}^{(n)}.$$

Because $\phi^{(n)}$ decreases, this process ensures that Ψ also decreases. This approach is reasonable in cases where 1D minimization of the cost function would be expensive yet 1D minimization of the surrogate function can be performed analytically, such as for quadratic surrogates. (See §14.5.4.)

s,ox,min,pcg

14.2.5 Surrogate minimization by preconditioned "conjugate" gradients

As described in §11.8, the preconditioned gradient vector (14.2.2) is a somewhat inefficient choice of search direction. Following §11.8, it is natural to try to modify the search directions to ensure that they are approximately **conjugate**. However, there is a subtlety here because the **line search** (14.2.3) minimizes the *surrogate* rather than the original cost function. To apply the **Polak-Ribiere** form of the **PCG** method, from (11.8.2), we would choose the following search direction

$$\boldsymbol{d}^{(n)} = -\boldsymbol{P}\,\boldsymbol{g}^{(n)} + \gamma_n \boldsymbol{d}^{(n-1)},$$

where from (11.8.7)

$$\gamma_n \approx \frac{\langle \boldsymbol{P}\,\boldsymbol{g}^{(n)},\, \boldsymbol{g}^{(n)} - \boldsymbol{g}^{(n-1)} \rangle}{\langle \boldsymbol{P}\,\boldsymbol{g}^{(n-1)},\, \boldsymbol{g}^{(n-1)} \rangle}.$$

See §14.5 for an example.

14.2.6 Multiple search direction methods

As discussed in §11.8, CG methods with multiple search directions have been investigated [38–40]. Likewise, one can use multiple search directions for the minimization step in an MM method [41]. Let $\boldsymbol{D}^{(n)} \in \mathbb{C}^{n_{\mathrm{p}} \times R}$ denote a set of R search direction vectors, then the multidimensional generalization of (14.2.3) is

e,ox,min,3mg

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \boldsymbol{D}^{(n)} \boldsymbol{\alpha}_n, \qquad \boldsymbol{\alpha}_n = \arg\min_{\boldsymbol{\alpha} \in \mathbb{C}^R} \phi^{(n)}(\boldsymbol{x}^{(n)} + \boldsymbol{D}^{(n)} \boldsymbol{\alpha})\,. \tag{14.2.4}$$

The choice $\boldsymbol{D}^{(n)} = \left[-\nabla\Psi(\boldsymbol{x}^{(n)}), \boldsymbol{x}^{(n)} - \boldsymbol{x}^{(n-1)}\right]$ is particularly effective and reduces to **CG** when Ψ is quadratic and $\phi^{(n)} = \Psi$ [41]. That approach is called the **MM memory gradient** method. When $\phi^{(n)}$ is quadratic, there is a closed-form solution for $\boldsymbol{\alpha}_n$ and an efficient recursive implementation [41]. However the method does not easily accommodate constraints like nonnegativity.

14.2.7 Acceleration methods

An alternative PCG approach is to first compute the minimizer $\boldsymbol{x}^{(n+1)}$ of the surrogate, and then consider $\boldsymbol{d}^{(n)} = \boldsymbol{x}^{(n+1)} - \boldsymbol{x}^{(n)}$ as a search direction, and then perform a line-search for Ψ along that direction by $\min_\alpha \Psi(\boldsymbol{x}^{(n)} + \alpha \boldsymbol{d}^{(n)})$ [42]. This approach was found to be useful in some statistical applications, but has not been investigated for imaging problems to my knowledge.

A variety of other acceleration methods for the EM algorithm have been proposed *e.g.*, [37]. Virtually all such methods could also be applied in the more general context of optimization transfer. However, most such methods do not allow for parameter constraints, and many of the methods destroy the intrinsic monotonicity property of optimization transfer, thus losing one of its primary practical benefits.

s,ox,design

14.3 Surrogate design for general cost functions (s,ox,design)

Although the optimization transfer method (14.1.1) has some desirable properties such as monotonicity, it can hardly be described as an “algorithm” because the first step requires the algorithm designer to choose surrogate functions that satisfy (14.1.2), and this choice is something of an art. This section summarizes some tools that may be helpful in designing surrogates. In some applications we use more than one of these tools in a sequence, *i.e.*, we find a first surrogate ϕ_1 for Ψ, and then find a surrogate ϕ_2 for ϕ_1, etc.

The methods described here are not exhaustive. See the discussion paper by Lange et al. [3] for further examples.

s,ox,design2

14.3.1 Surrogates for twice-differentiable cost functions (s,ox,design2)

One consideration in the design of surrogate functions is how differentiable the cost function Ψ is. We consider first the case where Ψ is continuously twice differentiable.

14.3.1.1 Convex surrogates

In some applications the cost function $\Psi(\boldsymbol{x})$ is not convex, so often a natural first step is to find a surrogate function that *is* convex, in hopes of simplifying the M-step. For a twice differentiable function $\Psi(\boldsymbol{x})$, its **2nd-order Taylor series** expansion (28.8.4) about the current estimate $\boldsymbol{x}^{(n)}$ is

$$\begin{aligned}\Psi(\boldsymbol{x}) = {} & \Psi(\boldsymbol{x}^{(n)}) + (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \,\nabla\Psi(\boldsymbol{x}^{(n)}) \\ & + (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \left[\int_0^1 (1-\alpha)\,\nabla^2\Psi(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{x}^{(n)})\,\mathrm{d}\alpha\right](\boldsymbol{x} - \boldsymbol{x}^{(n)}).\end{aligned}$$

To construct a convex surrogate, it is sufficient to first find a (matrix) function $\boldsymbol{J} : \mathbb{R}^{n_\mathrm{p}} \to \mathbb{R}^{n_\mathrm{p} \times n_\mathrm{p}}$ that satisfies the following conditions:

e,ox,surr,majorize
e,ox,surr,J,geq,0

$$\begin{aligned}\boldsymbol{J}(\boldsymbol{x}) &\succeq \nabla^2\Psi(\boldsymbol{x}), \qquad \forall \boldsymbol{x} && (14.3.1)\\ \boldsymbol{J}(\boldsymbol{x}) &\succeq \boldsymbol{0}, \qquad \forall \boldsymbol{x}, && (14.3.2)\end{aligned}$$

where the inequalities are interpreted in the matrix sense (see §27). For example, one choice would be

e,ox,J,max0

$$\boldsymbol{J}(\boldsymbol{x}) = \begin{cases} \nabla^2\Psi(\boldsymbol{x}), & \nabla^2\Psi(\boldsymbol{x}) \succeq \boldsymbol{0} \\ \boldsymbol{0}, & \nabla^2\Psi(\boldsymbol{x}) \prec \boldsymbol{0} \\ \boldsymbol{V}(\boldsymbol{x})\,\mathsf{diag}\{\max\{\lambda_j(\boldsymbol{x}),\, 0\}\}\,\boldsymbol{V}'(\boldsymbol{x}) & \text{otherwise,} \end{cases} \qquad (14.3.3)$$

where $\boldsymbol{V}(\boldsymbol{x})$ denotes the matrix of (orthonormal) eigenvectors of $\nabla^2\Psi(\boldsymbol{x})$, and $\lambda_j(\boldsymbol{x})$ denotes the jth corresponding eigenvalue. Having found such a $\boldsymbol{J}$, define the following function:

$$\phi^{(n)}(\boldsymbol{x}) \triangleq \Psi(\boldsymbol{x}^{(n)}) + (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \,\nabla\Psi(\boldsymbol{x}^{(n)})$$

$$+ (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \left[\int_0^1 (1-\alpha) \boldsymbol{J}(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{x}^{(n)}) \, \mathrm{d}\alpha \right] (\boldsymbol{x} - \boldsymbol{x}^{(n)}). \tag{14.3.4}$$

e,ox,surr,convex

It follows from (14.3.1) that this function majorizes $\Psi(\boldsymbol{x})$, *i.e.*, it satisfies (14.1.4). Therefore $\phi^{(n)}$ is indeed a surrogate for $\Psi(\boldsymbol{x})$. Furthermore, $\nabla^2 \phi^{(n)}(\boldsymbol{x}) = \boldsymbol{J}(\boldsymbol{x})$ by construction (*cf.* (28.8.4)), so the condition (14.3.2) ensures that $\phi^{(n)}$ is **convex** (*cf.* §28.9). The utility of this convex surrogate function depends in part on how easily one can perform the integral in (14.3.4). (It may not always be essential to perform that integral. For example, if one were to apply the Newton-Raphson method to $\phi^{(n)}$ for the M-step, then the integral is not needed. However, Newton-Raphson is not guaranteed to be monotonic, so its use would seem to defeat the motivation for using optimization transfer in the first place.) Fortunately, in many cases we can avoid the integral (14.3.4) by finding Hessians $\boldsymbol{J}$ that are independent of $\boldsymbol{x}$, as described next.

s,ox,huber

14.3.1.2 Huber's algorithm for quadratic surrogates (s,ox,huber)

In several of the applications considered later in this *book*, it is possible to find *quadratic* surrogate functions $\phi^{(n)}$ that satisfy the monotonicity condition (14.1.2), or equivalently (14.1.4). In particular, if the $\boldsymbol{J}$ in (14.3.4) is independent of $\boldsymbol{x}$ (but possibly dependent on $\boldsymbol{x}^{(n)}$), then $\phi^{(n)}$ is a **quadratic surrogate**, and (14.3.4) simplifies to the following form:

$$\phi^{(n)}(\boldsymbol{x}) = \Psi(\boldsymbol{x}^{(n)}) + (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \nabla \Psi(\boldsymbol{x}^{(n)}) + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \boldsymbol{J}_n (\boldsymbol{x} - \boldsymbol{x}^{(n)}), \tag{14.3.5}$$

e,ox,quad

where $\boldsymbol{J}_n = \boldsymbol{J}_n(\boldsymbol{x}^{(n)}) \triangleq \nabla^2 \phi^{(n)}$ is the Hessian of the surrogate $\phi^{(n)}$. Quadratic forms are particularly appealing as surrogate functions because there is a simple closed form solution for the M-step (14.1.1) in the absence of constraints. In many optimization papers [43], a quadratic surrogate of the form (14.3.5) is called a **linearization** of the cost function Ψ, even though (14.3.5) is quadratic, not linear.

Huber considered this type of algorithm [5] in the context of robust linear regression, and proposed the following iteration that follows logically from substituting (14.3.5) into (14.1.1).

$$\boxed{\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \left[\nabla^2 \phi^{(n)}\right]^{-1} \nabla \Psi(\boldsymbol{x}^{(n)}) \,. \qquad (14.3.6)}$$

e,ox,huber

If the quadratic surrogate $\phi^{(n)}$ in (14.3.5) satisfies the majorization condition (14.1.4), then Huber's algorithm will monotonically decrease Ψ.

It is interesting that Huber's algorithm is monotone yet Newton's method (11.4.3) is not, even though their general forms are fairly similar. Replacing the Hessian of Ψ in (11.4.3) with the Hessian of $\phi^{(n)}$ in (14.3.6) leads essentially to a "step size" that is sufficiently small to ensure monotonicity. Furthermore, Newton's method requires the cost function Ψ to be twice differentiable, whereas Huber's algorithm (14.3.6) need not, as seen from the sufficient condition (14.3.10) below.

Unfortunately, for imaging problems, if $\phi^{(n)}$ is nonseparable then Huber's algorithm is impractical due to the size of the required matrix inverse. It also does not accommodate the nonnegativity constraint. However, by using a *separable* quadratic surrogate $\phi^{(n)}$, we overcome both of these limitations; see §14.5.7, §15.6.5, §18.7.2, and §19.5.

In some cases one can find a quadratic surrogate function $\phi^{(n)}$ having a Hessian matrix $\boldsymbol{J}_0 = \nabla^2 \phi^{(n)}$ that is functionally independent of $\boldsymbol{x}^{(n)}$. Böhning and Lindsay [44] and Huber [5, p. 190] analyzed such problems, noting that an advantage of such a surrogate is that one can precompute $\boldsymbol{J}_0^{-1}$ (if storage permits) thereby avoiding new matrix inversions each iteration. In particular, if one can find a "maximal curvature" matrix $\boldsymbol{J}_0$ satisfying (14.3.1) and (14.3.2), then the quadratic form (14.3.5) with $\boldsymbol{J}_0$ is a valid surrogate, and the following **lower bound algorithm** of Böhning and Lindsay will monotonically decrease Ψ and will converge to a stationary point of Ψ if Ψ is bounded below [44, p. 652]:

$$\boxed{\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \boldsymbol{J}_0^{-1} \nabla \Psi(\boldsymbol{x}^{(n)}) \,. \qquad (14.3.7)}$$

e,ox,bohning

Wright et al. [45] proposed an approach in which $\boldsymbol{J}_0$ is simply constant times the identity matrix $\boldsymbol{I}$, and the constant is updated *adaptively* during the iterations. This is an intriguing approach that avoids having to find $\boldsymbol{J}_0$ analytically and may converge faster than using a fixed step size.

14.3.1.3 Existence of quadratic surrogate

In 1D problems we can define $J_0 \triangleq \max_x \frac{\partial^2}{\partial x^2} \Psi(x)$, and if this J_0 is finite then the algorithm (14.3.7) is applicable. Unfortunately, in higher dimensions we cannot define "$\max_{\boldsymbol{x}} \nabla^2 \Psi(\boldsymbol{x})$" because there is no ordering for matrices. However, under a slightly stronger assumption about Ψ we can still show that a $\boldsymbol{J}_0$ exists.

t,ox,exist **Theorem 14.3.1** *Assume that $\Psi(\boldsymbol{x})$ is twice* continuously *differentiable, and that $\boldsymbol{x}$ lies in a bounded set,* i.e., $\|\boldsymbol{x}\| \leq c_0$ *for some constant c_0. (Such an upper bound can usually be found for imaging problems in practice.) Then $\boldsymbol{J}_0 = c_1 \boldsymbol{I}$ satisfies (14.3.1) for $\|\boldsymbol{x}\| \leq c_0$, where*

$$c_1 \triangleq \max_{\boldsymbol{x} \,:\, \|\boldsymbol{x}\| \leq c_0} \left\| \nabla^2 \Psi(\boldsymbol{x}) \right\|_{\mathrm{Frob}},$$

where $\|\cdot\|_{\mathrm{Frob}}$ denotes the ***Frobenius*** *norm (27.5.6). (This maximum exists by the Weierstrass theorem [46, p. 40].)*
Proof:
For any matrix $\boldsymbol{J}$:

$$\boldsymbol{x}' \boldsymbol{J} \boldsymbol{x} = \sum_{j,k} x_j x_k' J_{jk} \leq \sqrt{\sum_{j,k} |J_{jk}|^2} \sqrt{\sum_{j,k} |x_j x_k|^2} = \|\boldsymbol{J}\|_{\mathrm{Frob}} \|\boldsymbol{x}\|^2$$

by the Cauchy-Schwarz inequality (27.4.2). Thus, if $\|\boldsymbol{x}\| \leq c_0$, then

$$\boldsymbol{x}' \nabla^2 \Psi(\boldsymbol{x}) \, \boldsymbol{x} \leq \|\boldsymbol{J}\|_{\mathrm{Frob}} \|\boldsymbol{x}\|^2 \leq c_1 \|\boldsymbol{x}\|^2,$$

so $c_1 \boldsymbol{I} \succeq \nabla^2 \Psi(\boldsymbol{x})$ for $\|\boldsymbol{x}\| \leq c_0$. □

Although this theorem shows that in principle we can find a majorizing $\boldsymbol{J}_0$ of the form $c_1 \boldsymbol{I}$, the convergence rate of (14.3.7) for this choice may be quite slow.

s,ox,qs,opt 14.3.1.4 Optimal curvature matrices (s,ox,qs,opt)

When designing a multidimensional quadratic surrogate of the form (14.3.5), one would like to choose its Hessian $\boldsymbol{J}_n$ "as small as possible" in light of the convergence rate analysis in §14.1.5. A typical approach to designing $\boldsymbol{J}_n$ consists of the following steps.

1. Select a family of matrices $\mathcal{J}_0$ having some desired structure. Typically one will choose matrices that are relatively easy to invert *e.g.*, diagonal or circulant matrices.
2. Identify a subset of that family that majorizes the cost function, *i.e.*,

$$\mathcal{J}_1 = \{\boldsymbol{J}_n \in \mathcal{J}_0 : \phi^{(n)}(\boldsymbol{x}; \boldsymbol{J}_n) \geq \Psi(\boldsymbol{x}), \forall \boldsymbol{x}\},$$

   where $\phi^{(n)}$ was defined in (14.3.5).
3. Select a specific Hessian from $\mathcal{J}_1$ using a criterion related to convergence rate. For example, based on (14.1.5) a desirable choice would be

$$\boldsymbol{J}_n = \arg\min_{\boldsymbol{J} \in \mathcal{J}_1} \rho\big(\boldsymbol{I} - \boldsymbol{J}^{-1} \nabla^2 \Psi(\boldsymbol{x}^{(n)})\big).$$

   A Hessian $\boldsymbol{J}_n$ chosen by such a method can be called optimal, meaning optimal over $\mathcal{J}_0$ with respect to the selected criterion.

   In 1D, optimal quadratic surrogates are known for a variety of potential functions; see *e.g.*, §14.4.4.4. Unfortunately, in higher dimensions the above recipe can be quite challenging except perhaps when very simple structures are assumed, such as $\mathcal{J}_0 = \{\alpha \boldsymbol{I} : \alpha \geq 0\}$.

s,ox,design1 14.3.2 Surrogates for once-differentiable cost functions (s,ox,design1)

We now turn to surrogates for differentiable cost functions that need not be twice differentiable.

14.3.2.1 Surrogates for concave terms

A large class of cost functions can be written in the form

$$\Psi(\boldsymbol{x}) = \Psi_1(\boldsymbol{x}) + \Psi_2(\boldsymbol{x})$$

where $\Psi_2(\boldsymbol{x})$ is a *concave* function [3, 47]. As described in (28.9.9), a concave function satisfies the following inequality:

$$\Psi_2(\boldsymbol{x}) \leq \Psi_2(\boldsymbol{x}^{(n)}) + (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \, \nabla \Psi_2(\boldsymbol{x}^{(n)}) \,.$$

Thus, the following function will be a valid surrogate for such cost functions:

$$\phi^{(n)}(\boldsymbol{x}) = \Psi_1(\boldsymbol{x}) + \Psi_2(\boldsymbol{x}^{(n)}) + (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \, \nabla \Psi_2(\boldsymbol{x}^{(n)}) \,.$$

If Ψ_1 is convex, then so will be this surrogate.

14.3.2.2 Surrogates for nested concave terms

Some cost functions have the form

$$\Psi(\boldsymbol{x}) = \sum_k \sigma_k(v_k(\boldsymbol{x})),$$

where $v_k : \mathbb{R}^{n_{\mathrm{p}}} \to \mathbb{R}$ is a convex function and $\sigma_k : \mathbb{R} \to \mathbb{R}$ is a differentiable *concave* function, *e.g.*, [48]. Again using concavity (28.9.9), we have the inequality

$$\sigma_k(v) \leq \sigma_k(u) + \dot{\sigma}_k(u)(v - u),$$

so the following function is a surrogate for Ψ:

$$\phi^{(n)}(\boldsymbol{x}) = \sum_k \left(\sigma_k(v_k(\boldsymbol{x}^{(n)})) + \dot{\sigma}_k(v_k(\boldsymbol{x}^{(n)})) \left[v_k(\boldsymbol{x}) - v_k(\boldsymbol{x}^{(n)})\right]\right) \equiv \sum_k \dot{\sigma}_k(v_k(\boldsymbol{x}^{(n)})) v_k(\boldsymbol{x}).$$

The surrogate $\phi^{(n)}$ is convex because the v_k functions are convex by assumption.

14.3.2.3 Gradient-based surrogates

When $\Psi(\boldsymbol{x})$ is differentiable, its **1st-order Taylor series** expansion (28.8.3) is given by

$$\Psi(\boldsymbol{x}) = \Psi(\boldsymbol{x}^{(n)}) + (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \int_0^1 \nabla \Psi(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{x}^{(n)}) \, \mathrm{d}\alpha \,.$$

This form suggests constructing a surrogate function as follows:

$$\phi^{(n)}(\boldsymbol{x}) \triangleq \Psi(\boldsymbol{x}^{(n)}) + (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \int_0^1 \nabla \phi^{(n)}(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{x}^{(n)}) \, \mathrm{d}\alpha \,.$$

This construction will satisfy the "matched value" surrogate condition in (14.1.4) and will also satisfy the majorization condition in (14.1.4) if

e,ox,graddiff

$$\boldsymbol{d}' \left[\nabla \phi^{(n)}(\boldsymbol{x}^{(n)} + \boldsymbol{d}) - \nabla \Psi(\boldsymbol{x}^{(n)} + \boldsymbol{d})\right] \geq 0, \qquad \forall \boldsymbol{d} \in \mathbb{R}^{n_{\mathrm{p}}}. \tag{14.3.8}$$

A simpler 1D version of this condition is described in Lemma 14.4.1. We use this condition to aid the design of quadratic surrogates described next.

14.3.2.4 Quadratic surrogates

Consider again the quadratic surrogates described by (14.3.5), for which

$$\nabla \phi^{(n)}(\boldsymbol{x}) = \nabla \Psi(\boldsymbol{x}^{(n)}) + \boldsymbol{J}_n \left[\boldsymbol{x} - \boldsymbol{x}^{(n)}\right],$$

where $\boldsymbol{J}_n$ is a Hermitian positive-semidefinite matrix that may depend on $\boldsymbol{x}^{(n)}$ but is independent of $\boldsymbol{x}$. For choosing $\boldsymbol{J}_n$, we would like to find alternatives to the sufficient "maximum curvature"condition (14.3.1) while ensuring $\phi^{(n)}$ is a surrogate. Substituting into (14.3.8) and rearranging leads to the following sufficient condition:

e,ox,design1,dJd

$$\boldsymbol{d}' \boldsymbol{J}_n \boldsymbol{d} \geq \boldsymbol{d}' \left[\nabla \Psi(\boldsymbol{x}^{(n)} + \boldsymbol{d}) - \nabla \Psi(\boldsymbol{x}^{(n)})\right], \qquad \forall \boldsymbol{d} \in \mathbb{R}^{n_{\mathrm{p}}}. \tag{14.3.9}$$

Alternatively, suppose that we can find $\boldsymbol{J}_n$ that satisfies

e,ox,quadhess,outer

$$\boldsymbol{J}_n \succeq \frac{1}{\|\boldsymbol{d}\|^2} \left[\nabla \Psi(\boldsymbol{x}^{(n)} + \boldsymbol{d}) - \nabla \Psi(\boldsymbol{x}^{(n)})\right] \boldsymbol{d}', \qquad \forall \boldsymbol{d} \neq \boldsymbol{0}. \tag{14.3.10}$$

Multiplying on the left by $\boldsymbol{d}'$ and on the right by $\boldsymbol{d}$ shows that (14.3.9) and hence (14.3.8) are satisfied.

Thus, a quadratic function of the form (14.3.5) with a Hessian that satisfies the conditions (14.3.9) or (14.3.10) is a valid surrogate for Ψ.

14.3.2.5 Relationship between quadratic surrogates and the Lipschitz condition for gradient projection

The gradient projection method considered in Theorem 12.2.1 assumes that the cost function gradient satisfies a Lipschitz condition of the form (28.8.5), *i.e.*, $\|\nabla \Psi(\boldsymbol{x} + \boldsymbol{d}) - \nabla \Psi(\boldsymbol{x})\| \leq L \|\boldsymbol{d}\|$. This is essentially a bound on the curvature of Ψ. Indeed, any cost function that satisfies that condition has a quadratic surrogate of the form (14.3.5) that satisfies the sufficient condition (14.3.8) for the choice $\boldsymbol{J}_n = L\boldsymbol{I}$ (in which case the surrogate is separable). To show this, simply apply the **Cauchy-Schwarz** inequality to the sufficient condition (14.3.9) and then apply the Lipschitz condition. Several recent papers have assumed Lipschitz conditions and developed algorithms based on what is essentially separable quadratic surrogates [43].

Further exploration of this interesting connection between the gradient projection algorithm and optimization transfer using quadratic surrogates is an *open problem*.

s,ox,design0

14.3.3 Separable surrogates for convex cost functions (s,ox,design0)

Although convex functions are easier to minimize than non-convex functions, the M-step (14.1.1) can still be challenging. One possible way to simplify the M-step is to form a **separable** surrogate function. (Often this technique is applied to form a separable surrogate function for an initial nonseparable surrogate function.) The next two subsections each present a method for forming a separable surrogate function, using generalizations of tricks developed by De Pierro [10, 11].

Interestingly, neither of the two methods described below require Ψ to be differentiable.

14.3.3.1 Separable surrogates for additive updates

To create a surrogate function that leads to an additive update, for the nth iteration, we write the cost function as follows:

$$\Psi(\boldsymbol{x}) = \Psi\left(\sum_{j=1}^{n_{\mathrm{p}}} x_j \boldsymbol{e}_j\right) = \Psi\left(\sum_{j=1}^{n_{\mathrm{p}}} \alpha_j^{(n)} \left[\frac{x_j - x_j^{(n)}}{\alpha_j^{(n)}} \boldsymbol{e}_j + \boldsymbol{x}^{(n)}\right]\right),$$

where $\boldsymbol{e}_j$ denotes the jth unit vector in $\mathbb{R}^{n_{\mathrm{p}}}$, and $\alpha_j^{(n)}$ denotes any positive coefficients that sum (over j) to unity. When Ψ is convex (*e.g.*, when we apply this approach to a convex surrogate), we have the following majorization:

$$\Psi(\boldsymbol{x}) \leq \phi^{(n)}(\boldsymbol{x}) \triangleq \sum_{j=1}^{n_{\mathrm{p}}} \alpha_j^{(n)} \, \Psi\left(\frac{x_j - x_j^{(n)}}{\alpha_j^{(n)}} \boldsymbol{e}_j + \boldsymbol{x}^{(n)}\right).$$

(This is a modification of a trick developed by De Pierro [11], *cf.* (14.5.9).) By this construction, this surrogate function satisfies the key surrogate conditions (14.1.4). Because $\phi^{(n)}$ is separable, the M-step simplifies into n_{p} 1D minimization problems:

$$\boldsymbol{x}^{(n+1)} = \arg\min_{\boldsymbol{x}} \phi^{(n)}(\boldsymbol{x}) \iff x_j^{(n+1)} = \arg\min_{x_j} \Psi\left(\frac{x_j - x_j^{(n)}}{\alpha_j^{(n)}} \boldsymbol{e}_j + \boldsymbol{x}^{(n)}\right).$$

There are a variety of 1D minimization algorithms available [49].

This approach is "additive" because it leads to an update of the following form

e,ox,sep,add

$$x_j^{(n+1)} = x_j^{(n)} + \alpha_j^{(n)} \arg\min_{\delta_j} \Psi(\delta_j \boldsymbol{e}_j + \boldsymbol{x}^{(n)}). \tag{14.3.11}$$

If Ψ is quadratic, then there is an explicit form for this minimization:

$$x_j^{(n+1)} = x_j^{(n)} - \frac{\alpha_j^{(n)}}{\frac{\partial^2}{\partial x_j^2} \Psi(\boldsymbol{x}^{(n)})} \frac{\partial}{\partial x_j} \Psi(\boldsymbol{x}^{(n)}).$$

In matrix-vector form, the update is

e,ox,design0,sep,quad

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \mathsf{diag}\left\{ \frac{\alpha_j^{(n)}}{\frac{\partial^2}{\partial x_j^2}\, \Psi(\boldsymbol{x}^{(n)})} \right\} \nabla \Psi(\boldsymbol{x}^{(n)}) \,. \tag{14.3.12}$$

Because the $\alpha_j^{(n)}$ values must sum to unity, the average $\alpha_j^{(n)}$ value is $1/n_{\mathrm{p}}$. In imaging problems, n_{p} can be very large, so the $\alpha_j^{(n)}$ values can be quite small, leading to slow convergence of this algorithm.

14.3.3.2 Separable surrogates for multiplicative updates

Alternatively, for problems with nonnegativity constraints we can form multiplicative updates by using the following approach to rewriting the cost function:

$$\begin{aligned}\Psi(\boldsymbol{x}) = \Psi\left(\sum_{j=1}^{n_{\mathrm{p}}} x_j \boldsymbol{e}_j\right) &= \Psi\left(\sum_{j=1}^{n_{\mathrm{p}}} \left(\frac{\alpha_j^{(n)} x_j^{(n)}}{\sum_{j'=1}^{n_{\mathrm{p}}} \alpha_{j'}^{(n)} x_{j'}^{(n)}}\right) \frac{x_j}{x_j^{(n)}} \left[\sum_{j'=1}^{n_{\mathrm{p}}} \alpha_{j'}^{(n)} x_{j'}^{(n)}\right] \boldsymbol{e}_j\right) \\ &\leq \sum_{j=1}^{n_{\mathrm{p}}} \left(\frac{\alpha_j^{(n)} x_j^{(n)}}{\sum_{j'=1}^{n_{\mathrm{p}}} \alpha_{j'}^{(n)} x_{j'}^{(n)}}\right) \Psi\left(\frac{x_j}{x_j^{(n)}} \left[\sum_{j'=1}^{n_{\mathrm{p}}} \alpha_{j'}^{(n)} x_{j'}^{(n)}\right] \boldsymbol{e}_j\right).\end{aligned}$$

This formulation of a surrogate function leads to a multiplicative update:

e,ox,design0,sep,mult

$$x_j^{(n+1)} = \frac{x_j^{(n)}}{\sum_{j'=1}^{n_{\mathrm{p}}} \alpha_{j'}^{(n)} x_{j'}^{(n)}} \cdot \mathop{\arg\min}_{\delta_j \geq 0} \Psi(\delta_j \boldsymbol{e}_j) \,. \tag{14.3.13}$$

Analyzing the convergence properties of this multiplicative iteration is an *open problem*.

s,ox,mixture

14.3.4 Example: mixture model estimation (s,ox,mixture)

One application of optimization transfer is estimation of **mixture distributions**. For simplicity, we illustrate the case of a scalar gaussian mixture where the only unknown parameters are the mean values of the gaussian components. In this setting, the negative log-likelihood is

$$Ł(\boldsymbol{x}) = \sum_{i=1}^{n_{\mathrm{d}}} -\log \mathsf{p}(y_i; \boldsymbol{x}) = \sum_{i=1}^{n_{\mathrm{d}}} -\log\left(\sum_{j=1}^{n_{\mathrm{p}}} q_j\, \mathsf{p}(y_i; x_j)\right),$$

where q_j are the (assumed known) mixture probabilities and

$$\mathsf{p}(y_i; x_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}}\, \mathrm{e}^{-(y_i - x_j)^2/(2\sigma_j^2)} \,,$$

where σ_j^2 are the (assumed known) component variances.

To design a surrogate function for this negative log-likelihood, we rewrite it as follows:

$$Ł(\boldsymbol{x}) = \sum_{i=1}^{n_{\mathrm{d}}} -\log\left(\sum_{j=1}^{n_{\mathrm{p}}} q_j\, \mathsf{p}(y_i; x_j)\right) = -\sum_{i=1}^{n_{\mathrm{d}}} \log\left(\sum_{j=1}^{n_{\mathrm{p}}} z_{ij}^{(n)} \frac{\mathsf{p}(y_i; x_j)}{\mathsf{p}\big(y_i; x_j^{(n)}\big)} \sum_k q_k\, \mathsf{p}\big(y_i; x_k^{(n)}\big)\right),$$

where

$$z_{ij}^{(n)} \triangleq \frac{q_j\, \mathsf{p}\big(y_i; x_j^{(n)}\big)}{\sum_k q_k\, \mathsf{p}\big(y_i; x_k^{(n)}\big)}.$$

Using the convexity of the negative logarithm function yields:

$$\begin{aligned}
\text{Ł}(\boldsymbol{x}) \leq \phi^{(n)}(\boldsymbol{x}) &\triangleq -\sum_{i=1}^{n_\mathrm{d}} \sum_{j=1}^{n_\mathrm{p}} z_{ij}^{(n)} \log\left(\frac{\mathsf{p}(y_i; x_j)}{\mathsf{p}\big(y_i; x_j^{(n)}\big)} \sum_k q_k \, \mathsf{p}\big(y_i; x_k^{(n)}\big) \right) \\
&\equiv \phi^{(n)}(\boldsymbol{x}) \triangleq -\sum_{j=1}^{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{d}} z_{ij}^{(n)} \log \mathsf{p}(y_i; x_j) \\
&\equiv \sum_{j=1}^{n_\mathrm{p}} \sum_{i=1}^{n_\mathrm{d}} z_{ij}^{(n)} \frac{(y_i - x_j)^2}{2\sigma_j^2}.
\end{aligned}$$

Minimizing this surrogate function with respect to $\boldsymbol{x}$ is a trivial WLS problem. This example contains the essence of the EM algorithm for mixture models [6, 50].

s,ox,recon

14.4 Surrogate design for image reconstruction cost functions (s,ox,recon)

In principle the preceding tools are applicable to general cost functions. Now let us focus on surrogate design methods that have been developed specifically for the types of cost functions that arise in image reconstruction problems. In particular, most of the image reconstruction problems considered in this *book* involve cost functions of the following form

e,ox,recon,kost

$$\Psi(\boldsymbol{x}) = \sum_{i=1}^{N_{\mathrm{i}}} \psi_i([\boldsymbol{B}\boldsymbol{x}]_i) \tag{14.4.1}$$

where $[\boldsymbol{B}\boldsymbol{x}]_i = \sum_{j=1}^{n_{\mathrm{p}}} b_{ij} x_j$, for some application-dependent choices for the **potential functions** ψ_i and the $N_{\mathrm{i}} \times n_{\mathrm{p}}$ matrix $\boldsymbol{B}$. For fast convergence, we must seek algorithms tailored to this form.

Some of the relevant properties of this form of Ψ include the following.

- $\Psi(\boldsymbol{x})$ is a sum of scalar functions $\psi_i(\cdot)$.
- The ψ_i functions often have bounded curvature, and often are convex.
- The arguments of the ψ_i functions involve inner products.
- Often the inner product coefficients are all nonnegative.

The cornucopia of algorithms that have been proposed in the image reconstruction literature exploit these properties (implicitly or explicitly) in different ways.

In the context of robust linear regression, estimators formed by minimizing such cost functions are called **M-estimators** [5].

The gradient of this type of cost function has the following form:

e,ox,recon,grad

$$\frac{\partial}{\partial x_j} \Psi(\boldsymbol{x}) = \sum_{i=1}^{N_{\mathrm{i}}} b_{ij}\, \dot{\psi}_i([\boldsymbol{B}\boldsymbol{x}]_i), \qquad \nabla\, \Psi(\boldsymbol{x}) = \boldsymbol{B}' \dot{\boldsymbol{\psi}}(\boldsymbol{B}\boldsymbol{x}), \tag{14.4.2}$$

where $\dot{\psi}_i(t) = \frac{\mathrm{d}}{\mathrm{d}t}\, \psi_i(t)$ and for $\boldsymbol{t} \in \mathbb{R}^{N_{\mathrm{i}}}$ we define

e,ox,recon,dPot

$$\dot{\boldsymbol{\psi}}(\boldsymbol{t}) \triangleq \begin{bmatrix} \dot{\psi}_1(t_1) \\ \vdots \\ \dot{\psi}_{N_{\mathrm{i}}}(t_{N_{\mathrm{i}}}) \end{bmatrix}. \tag{14.4.3}$$

For this type of **additively separable** cost function, designing surrogates is greatly simplified because we can consider one term in the sum at a time. We first consider each function $\psi_i(\cdot)$ and find a corresponding surrogate function $\mathsf{h}_i(t; s)$ that satisfies the following two conditions:

e,ox,hi,geq

$$\begin{aligned} \mathsf{h}_i(s; s) &= \psi_i(s), \qquad \forall s \\ \mathsf{h}_i(t; s) &\geq \psi_i(t), \qquad \forall t, s. \end{aligned} \tag{14.4.4}$$

If one can find such h_i functions, then a natural surrogate for $\Psi(\boldsymbol{x})$ is the following

e,ox,surn,recon

$$\phi^{(n)}(\boldsymbol{x}) = \sum_{i=1}^{N_{\mathrm{i}}} \mathsf{h}_i([\boldsymbol{B}\boldsymbol{x}]_i; [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i). \tag{14.4.5}$$

It follows from (14.4.4) that this $\phi^{(n)}$ satisfies the surrogate conditions (14.1.4), as well as the tangent condition (14.1.5) if the h_i functions are differentiable. So now the design problem becomes finding a suitable 1D surrogate function h_i for each ψ_i function.

14.4.1 Convex surrogates

When ψ_i is **nonconvex**, a natural first step is to find a surrogate function h_i that is convex. Specializing (14.3.4) to the 1D case, a simple choice is

e,ox,hi,convex

$$\mathsf{h}_i(t; s) = \psi_i(s) + \dot{\psi}_i(s)(t-s) + (t-s)^2 \int_0^1 (1-\alpha)\, \check{c}_i(\alpha t + (1-\alpha)\, s; s)\, \mathrm{d}\alpha, \tag{14.4.6}$$

where $\check{c}_i$, the curvature of $\mathsf{h}_i(\cdot; s)$, is chosen to satisfy

$$\begin{aligned} \check{c}_i(t; s) &\geq \ddot{\psi}_i(t), \qquad \forall t, s \\ \check{c}_i(t; s) &\geq 0, \qquad \forall t, s, \end{aligned}$$

in analogy with (14.3.1) and (14.3.2), assuming that ψ_i is twice differentiable. Any function constructed via (14.4.6) with curvatures that satisfy these conditions will satisfy (14.4.4) and hence is a valid surrogate function for ψ_i. And thus $\phi^{(n)}$ in (14.4.5) is a valid surrogate for Ψ. Because these h_i functions have nonnegative curvatures, they are convex, and thus it is readily verified that $\phi^{(n)}$ is a convex surrogate function. The natural choice for the curvature, *cf.* (14.3.3), is simply

$$\check{c}_i(t;s) \triangleq \max\left\{\ddot{\psi}_i(t), 0\right\},$$

although other choices may be used if it simplifies the integral in (14.4.6). This construction shows that convex surrogates exist for any cost function of the form (14.4.1) with twice differentiable ψ_i functions.

14.4.2 Derivative-based surrogate

The following Lemma describes a sufficient condition that is useful for designing surrogates. It provides the 1D analog of (14.3.8).

l,ox,deriv **Lemma 14.4.1** *If $\psi(t)$ and $h(t)$ are differentiable and if the following three conditions are satisfied:*

$$\begin{aligned} h(s) &= \psi(s), \quad \textit{for some } s \in \mathbb{R} \\ \dot{h}(t) &\geq \dot{\psi}(t), \quad \forall t > s \\ \dot{h}(t) &\leq \dot{\psi}(t), \quad \forall t < s, \end{aligned} \tag{14.4.7}$$

e,ox,sps,curv,suff

then $h(\cdot)$ is a valid surrogate for ψ, i.e., *$h(t) \geq \psi(t), \forall t$.*
Proof:
It follows from the assumptions that $\dot{h}(t)(t-s) \geq \dot{\psi}(t)(t-s)$. Applying the 1st-order Taylor expansion (28.8.3):

$$\begin{aligned} h(t) &= h(s) + \int_0^1 \dot{h}(\alpha s + (1-\alpha)t)\, \mathrm{d}\alpha (t-s) \\ &\geq \psi(s) + \int_0^1 \dot{\psi}(\alpha s + (1-\alpha)\, t)\, \mathrm{d}\alpha (t-s) = \psi(t)\,. \end{aligned}$$

□

The conditions (14.4.7) allow one to focus on the slopes when designing surrogate functions.

s,ox,recon,mult 14.4.3 Surrogates for multiplicative updates

Suppose each ψ_i is convex on $(0, \infty)$ and we want to minimize Ψ subject to the nonnegativity constraint $\boldsymbol{x} \succeq \mathbf{0}$. In the context of emission tomography, De Pierro noted that if $b_{ij} \geq 0, \ \forall i, j$, then

$$[\boldsymbol{B}\boldsymbol{x}]_i = \sum_{j=1}^{n_\mathrm{p}} b_{ij} x_j = \sum_{j=1}^{n_\mathrm{p}} \left(\frac{b_{ij} x_j^{(n)}}{[\boldsymbol{B}\boldsymbol{x}^{(n)}]_i} \right) \left(\frac{x_j}{x_j^{(n)}} [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i \right).$$

Using the convexity of each ψ_i:

$$\Psi(\boldsymbol{x}) = \sum_{i=1}^{N_\mathrm{i}} \psi_i([\boldsymbol{B}\boldsymbol{x}]_i) \leq \phi^{(n)}(\boldsymbol{x}) \triangleq \sum_{i=1}^{N_\mathrm{i}} \sum_{j=1}^{n_\mathrm{p}} \left(\frac{b_{ij} x_j^{(n)}}{[\boldsymbol{B}\boldsymbol{x}^{(n)}]_i} \right) \psi_i \left(\frac{x_j}{x_j^{(n)}} [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i \right).$$

This leads to the parallelizable multiplicative update (*cf.* (17.4.4)):

$$\begin{aligned} x_j^{(n+1)} &= x_j^{(n)} \alpha_j^{(n)} \\ \alpha_j^{(n)} &= \operatorname*{arg\,min}_{\alpha_j} \sum_{i=1}^{N_\mathrm{i}} \left(\frac{b_{ij}}{[\boldsymbol{B}\boldsymbol{x}^{(n)}]_i} \right) \psi_i(\alpha_j [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i). \end{aligned}$$

s,ox,recon,parab 14.4.4 Parabola surrogates (s,ox,recon,parab)

If the curvature $\check{c}_i(t;s)$ in (14.4.6) is independent of t (but possibly dependent on s), then $\mathsf{h}_i(\cdot;s)$ is a **parabola surrogate** for ψ_i and $\phi^{(n)}$ is a **quadratic surrogate** for Ψ. For parabola surrogates, we use the following notation:

e,ox,recon,qi

$$q_i(t;s) = \psi_i(s) + \dot{\psi}_i(s)(t-s) + \frac{1}{2}\, \check{c}(\psi_i, s)(t-s)^2. \tag{14.4.8}$$

There are a variety of methods for choosing the curvatures $\check{c}_i$, depending on the nature of ψ_i.

14.4.4.1 Optimal curvature

As described in §14.1.5, for fast convergence we would like the curvature of the surrogate function to be small. Yet for monotonic algorithms we must maintain the surrogate conditions (14.1.4). We therefore define the **optimal curvature** to be the smallest curvature that still ensures (14.4.4), as follows:

e,ox,recon,curv,opt

$$\breve{c}_{\text{opt}}(\psi, s) \triangleq \min\left\{ c \geq 0 : \, \psi(s) + \dot{\psi}(s)(t-s) + \frac{1}{2}c(t-s)^2 \geq \psi(t), \; \forall t \right\}. \qquad (14.4.9)$$

Generally speaking, this optimal curvature requires only that ψ between once differentiable. We use this choice whenever we can find the minimizer analytically.

In [73], we showed that if

- ψ is strictly convex on $[0, \infty)$, and
- $\dot{\psi}$ is strictly *concave* on $[0, \infty)$,

then the optimal curvature (for minimization subject to the nonnegative constraint $\boldsymbol{x} \succeq \mathbf{0}$ is as follows:

e,ox,recon,curv,opt,convex

$$\breve{c}_{\text{opt}}(\psi, s) = \begin{cases} \left[-2\dfrac{\psi(0) - \psi(s) + \dot{\psi}(s)\, s}{s^2} \right]_+, & s > 0 \\ \left[-\ddot{\psi}(s) \right]_+, & s = 0. \end{cases} \qquad (14.4.10)$$

These conditions apply to emission tomography, and similar conditions apply to (monoenergetic) transmission tomography with the usual Poisson noise models [73]. (See Chapter 19.)

In [74] de Leeuw expresses the optimal curvature (14.4.9) as follows:

e,ox,recon,curv,deleeuw

$$\breve{c}_{\text{opt}}(\psi, s) \triangleq \sup_{t \neq s} \frac{\psi(t) - \left[\psi(s) + \dot{\psi}(s)(t-s) \right]}{\frac{1}{2}(t-s)^2}, \qquad (14.4.11)$$

calling it "sharp" majorization.

14.4.4.2 Maximum curvature

If ψ is twice differentiable, then a simple choice for $\breve{c}$ is the **maximum curvature**

e,ox,recon,curv,max

$$\breve{c}_{\max}(\psi, s) = \max_t \ddot{\psi}(t). \qquad (14.4.12)$$

This choice is usually very easy to determine, but often leads to unnecessarily slow convergence.

s,ox,recon,deriv

14.4.4.3 Derivative-based curvatures (s,ox,recon,deriv)

Finding an analytical expression for the optimal curvatures (14.4.9) is something of an art. When one cannot determine the optimal curvatures, the following Lemma gives a curvature expression that is often easily evaluated [75], so it may be helpful in choosing the curvature of a parabola surrogate function.

l,ox,curv

Lemma 14.4.2 *If $\psi(t)$ is differentiable, and $\breve{c}(\psi, s)$ satisfies*

e,ox,recon,curv,ratio,geq

$$\breve{c}(\psi, s) \geq \frac{\dot{\psi}(t) - \dot{\psi}(s)}{t - s} \qquad \forall t \neq s, \qquad (14.4.13)$$

then the parabola (14.4.8) with this curvature is a surrogate for ψ.

Proof:

For $t \neq s$,

$$\begin{aligned} [\dot{q}(t) - \dot{\psi}(t)](t-s) &= \left[\dot{\psi}(s) + \breve{c}(\psi, s)(t-s) - \dot{\psi}(t) \right](t-s) \\ &= (t-s)^2 \left[\breve{c}(\psi, s) - \frac{\dot{\psi}(t) - \dot{\psi}(s)}{t-s} \right] \geq 0. \end{aligned}$$

The conditions of Lemma 14.4.1 are satisfied, so $q(t)$ is a surrogate for ψ. □

In particular, if

e,ox,recon,curv,ratio,max

$$\breve{c}(\psi, s) = \max_{t \neq s} \frac{\dot{\psi}(t) - \dot{\psi}(s)}{t - s} \tag{14.4.14}$$

is finite, then this maximum is a natural choice for the curvature.

For an example in holographic imaging, see [75]. In many cases we can consider intervals rather than the entire real line, such as in the following examples.

x,ox,curv,emis

Example 14.4.3 *Consider the emission tomography problem described in Chapter 8 in which $\psi(t) = (t + r) - y \log(t + r)$, where $r > 0$ and $y \geq 0$. Here we also include the constraint that $t, s \geq 0$. Then $\dot{\psi}(t) = 1 - y/(t + r)$ and $\ddot{\psi}(t) = y/(t + r)^2$. So the maximum curvature is $\max_{t \geq 0} \ddot{\psi}(t) = \ddot{\psi}(0) = y/r^2$. This can be undesirably large.*

To apply (14.4.14), we need to compute

$$\begin{aligned}\breve{c}(\psi, s) &= \max_{t \geq 0} \frac{\dot{\psi}(t) - \dot{\psi}(s)}{t - s} = \max_{t \geq 0} \frac{[1 - y/(t + r)] - [1 - y/(s + r)]}{t - s} \\ &= \max_{t \geq 0} \frac{y}{(t + r)(s + r)} = \frac{y}{r(s + r)}.\end{aligned}$$

When $s > 0$, this can be much smaller than the maximum curvature $\ddot{\psi}(0) = y/r^2$, which could accelerate convergence. However, as shown in , the optimal (smallest possible) curvature is (14.4.10).

x,ox,curv,trans

Example 14.4.4 *Consider the monoenergetic transmission tomography problem described in Chapter 9 in which the negative marginal log-likelihood is $\psi(t) = b\,\mathrm{e}^{-t} - y \log(b\,\mathrm{e}^{-t}) \equiv b\,\mathrm{e}^{-t} + yt$, where $b > 0$ and $y \geq 0$, and again $t, s \geq 0$. Then $\dot{\psi}(t) = y - b\,\mathrm{e}^{-t}$, so we need to compute*

$$\breve{c}(\psi, s) = \max_{t \geq 0} \frac{\dot{\psi}(t) - \dot{\psi}(s)}{t - s} = \max_{t \geq 0} \frac{y - b\,\mathrm{e}^{-t} - (y - b\,\mathrm{e}^{-s})}{t - s} = b\,\mathrm{e}^{-s} \max_{t \geq 0} \frac{1 - \mathrm{e}^{s-t}}{t - s}.$$

Substituting $x = s - t$ we have

$$\breve{c}(\psi, s) = b\,\mathrm{e}^{-s} \max_{x \leq s} \frac{\mathrm{e}^x - 1}{x} = b\,\mathrm{e}^{-s} \frac{\mathrm{e}^s - 1}{s} = b \frac{1 - \mathrm{e}^{-s}}{s}.$$

The optimal curvature for this case is again (14.4.10), as shown in §19.5, but the preceding derivation certainly is simpler.

s,ox,recon,huber

14.4.4.4 Huber's curvatures (s,ox,recon,huber)

Citing a 1975 report by Dutter, Huber [5, p. 184] described a parabola surrogate (which he called a **comparison function**) for a broad class of **potential functions** ψ_i. Specifically, Huber considered ψ_i functions of the form

$$\psi_i(t) = \psi(t - t_i)$$

for some $t_i \in \mathbb{R}$, where ψ is assumed to satisfy the conditions of Theorem 14.4.5 below. Fig. 14.4.1 shows an example of such a potential function and its quadratic surrogate for the case $\psi(t) = \sqrt{t^2 + 1}$.

Remarkably, for Huber's class of potential functions there is a very simple expression for the **optimal curvature** defined in (14.4.9), as shown in the following theorem [5, p. 185].

t,ox,parab

Theorem 14.4.5 *Suppose $\psi : \mathbb{R} \to \mathbb{R}$ satisfies the following conditions*[4].

$\psi(t)$ *is differentiable,*

$\psi(t) = \psi(-t)$, $\forall t$ *(symmetry),*

$\omega_\psi(t) \triangleq \dot{\psi}(t)/t$ *is bounded and monotone nonincreasing for* $t > 0$.

Then the parabola function defined by (cf. *(14.4.8)):*

$$q(t; s) \triangleq \psi(s) + \dot{\psi}(s)(t - s) + \frac{1}{2}\,\omega_\psi(s)(t - s)^2$$

[4] Huber [5, p. 184] makes the stronger assumption that ψ is convex, and uses that additional assumption to show that his iteration (14.3.6) *strictly* decreases the cost function each iteration until a minimizer is reached.

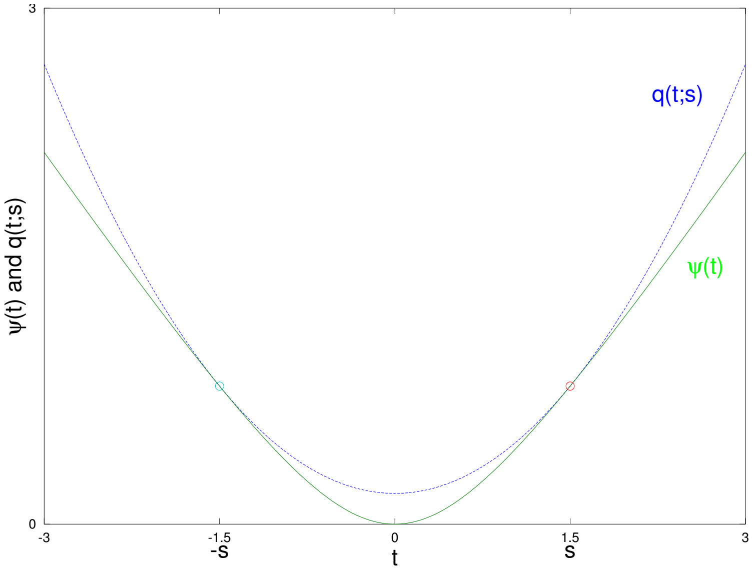


Figure 14.4.1: Illustration of a nonquadratic potential function $\psi(t) = \sqrt{t^2+1} - 1$ and a parabola surrogate function $q(t; s)$ for $s = 1.5$.

fig_parabl

e,ox,recon,huber,q

$$= \left[\psi(s) - \frac{\omega_\psi(s)}{2} s^2\right] + \frac{\omega_\psi(s)}{2} t^2, \tag{14.4.15}$$

is a surrogate function for ψ, i.e., *it satisfies the conditions (14.4.4). Furthermore, the curvature* ω_ψ *is optimal in the sense of being the smallest value that satisfies those conditions.*

Proof:
Because (14.4.15) implies $q(t; s) = q(t; |s|) = q(|t|; |s|)$, it suffices to consider $t, s \geq 0$. Note that $\dot{q}(t; s) \triangleq \frac{\partial}{\partial t} q(t; s) = \omega_\psi(s)\, t$. Because ω_ψ is nonincreasing for $t > 0$, it satisfies the following.

- If $t > s \geq 0$ then $\omega_\psi(s) \geq \omega_\psi(t) = \dot{\psi}(t)\,/t$, so $\dot{\psi}(t) \leq t\, \omega_\psi(s) = \dot{q}(t; s)$.
- If $s \geq t > 0$ then $\omega_\psi(s) \leq \omega_\psi(t) = \dot{\psi}(t)\,/t$, so $\dot{\psi}(t) \geq t\, \omega_\psi(s) = \dot{q}(t; s)$.

Thus the conditions of Lemma 14.4.1 are met, so q is a surrogate for ψ.

Because $q(t; s)$ touches $\psi(t)$ at both $t = s$ and $t = -s$, any smaller value of the curvature would cause $q(-s; s)$ to be less than $\psi(-s)$, so ω_ψ is optimal. □

A very wide variety of potential functions satisfy the conditions of this theorem. Table 2.1 in §2.7 summarizes several of these choices. Lange [76, p. 442] lists several more and describes a general procedure for deriving new choices. Table 2.1 also lists some choices with unbounded ω_ψ functions. Optimization with such ψ functions is more complicated[5].

The **potential weighting functions** $\omega_\psi(t)$ that determine the curvature of the surrogate parabola (14.4.15) significantly affect image properties such as robustness to noise outliers and edge-preservation, as discussed in §1.10.3. Fig. 2.7.1 illustrates several of the ω_ψ functions from Table 2.1.

The optimality of the curvature $\omega_\psi(\cdot)$ in Theorem 14.4.5 holds when all values of $t \in \mathbb{R}$ are considered in (14.4.9). In some situations, we know lower and upper bounds on t, in which case we can replace (14.4.9) by the weaker condition:

e,ox,recon,curv,opt,interval

$$\breve{c}_{\text{opt}}(\psi, s) \triangleq \min \left\{ c \geq 0 :\, \psi(s) + \dot{\psi}(s)(t - s) + \frac{1}{2} c (t - s)^2 \geq \psi(t),\ \forall t \in [t_{\min}, t_{\max}] \right\}. \tag{14.4.16}$$

A procedure for computing this $\breve{c}_{\text{opt}}(\psi, s)$ under the conditions of Theorem 14.4.5 is given in [78], with application to a coordinate descent algorithm. Adapting this approach to a simultaneous update algorithm is an *open problem*.

[5] For methods based on coordinate descent, one reasonable minimization approach uses a half-interval search over a suitably bracketed search range [77, p. 485].

s,ox,recon,halfquad

14.4.4.5 Half-quadratic curvatures (s,ox,recon,halfquad)

Huber's approach to finding quadratic surrogate functions has been rediscovered in different disguises. Several papers in the image restoration/reconstruction literature have described so-called **half-quadratic** methods for minimizing nonquadratic cost functions [48, 79–84], particularly in the context of **edge-preserving regularization**. As early as 1994, Kunsch noted that half-quadratic methods are a special case of Huber's algorithm [85], yet this relationship seems not to be well known. A typical cost function considered in the half-quadratic literature has the form

$$\Psi(\boldsymbol{x}) = \sum_{i=1}^{N_{\mathrm{i}}} \psi([\boldsymbol{B}\boldsymbol{x}]_i - t_i),$$

where ψ is some nonquadratic potential function designed to preserve edges. Because this cost function is difficult to minimize directly, a sequence of quadratic surrogate functions, or augmented cost functions, are minimized. Two such approaches have been proposed in the literature, one called the "multiplicative form" and the other called the "additive form." However, both of these forms are simply obfuscated versions of Huber's quadratic surrogates. In the half-quadratic literature, the potential functions ψ often are assumed to satisfy more restrictive conditions than those in Theorem 14.4.5.

14.4.4.5.1 Multiplicative form For the multiplicative form of the half-quadratic approach, *e.g.*, [48, 79, 83, 86], it is assumed that the potential function has the following form:

$$\psi(t) = \frac{1}{2}\sigma(t^2),$$

where $\sigma(t)$ must satisfy several conditions including [48]:

- σ is twice continuously differentiable,
- σ is strictly concave,
- $0 < \dot{\sigma} \leq M < \infty$.

Using such conditions, and sometimes stronger assumptions [83], papers in the half-quadratic literature show that there is some function ζ such that

e,ox,recon,hq,min,w

$$\psi(t) = \min_w \left[w \frac{1}{2} t^2 + \zeta(w) \right], \tag{14.4.17}$$

and this property is used to derive an augmentation algorithm of the form (14.1.6).

We now show that any potential function ψ that satisfies the above conditions will also satisfy the conditions in Theorem 14.4.5.

1. Clearly $\psi(-t) = \psi(t)$ by construction.
2. $\dot{\psi}(t) = \dot{\sigma}(t)t$ so $\omega_\psi(t) = \dot{\psi}(t)/t = \dot{\sigma}(t)$ which is bounded by assumption. Furthermore, because σ is twice differentiable and concave, for $t > 0$ we have $\frac{\mathrm{d}^2}{\mathrm{d}t^2}\, \omega_\psi = \ddot{\sigma} < 0$, so ω_ψ is decreasing on $[0, \infty)$.

Thus the conditions of Theorem 14.4.5 are satisfied. However, Theorem 14.4.5 is more general because twice differentiability is not assumed. Furthermore, constructing ω_ψ is more direct than considering the function ζ, simplifying the analysis.

It follows from the conditions (14.4.4) that $\psi(t) = \min_s q(t; s)$, so in light of (14.4.15), the minimizing "w" in (14.4.17) above is simply ω_ψ.

14.4.4.5.2 Additive form For the additive form, one first finds a *constant* curvature $\breve{c}$ that satisfies the condition

$$q(t; s) \triangleq \psi(s) + \dot{\psi}(s)(t - s) + \frac{\breve{c}}{2}(t - s)^2 \geq \psi(t), \qquad \forall t.$$

For example, if ψ is twice differentiable, then $\breve{c}_{\max} = \max_t \ddot{\psi}(t)$ is the logical choice. Rearranging $q(t; s)$, we have

$$\psi(t) = \min_s q(t; s) = \min_s \left[\psi(s) - \frac{1}{2\,\breve{c}}\, \dot{\psi}^2(s) + \frac{1}{2} \left(\sqrt{\breve{c}} t - \frac{\breve{c}\, s - \dot{\psi}(s)}{\sqrt{\breve{c}}} \right)^2 \right].$$

This is equivalent to the form considered in [81, 82, 87, 88]. In other words, the "additive form" of the half-quadratic method is nothing more than a surrogate parabola with a sufficiently large constant curvature. In the half-quadratic literature, results from convex analysis [89] (such as the **Fenchel-Legendre transform** [90]) are used to express the arguments in brackets in the **augmentation** form

$$\min_u \left[z(u) + \frac{1}{2} (\sqrt{\breve{c}} t - u/\sqrt{\breve{c}})^2 \right],$$

but in the final analysis it is equivalent to form above.

Nikolova and Ng compared the additive form and the multiplicative form of the half-quadratic methods [88], and found that the multiplicative form converged faster. This is the expected conclusion because for this family of potential functions, Huber's curvatures ω_ψ are optimal (Theorem 14.4.5), whereas the constant upper bound $\breve{c}_{\max}$ will be unnecessarily large, slowing convergence. However, using constant curvatures can simplify preconditioning [87].

As noted in [91], the half-quadratic method is equivalent to a simple **gradient linearization iteration**.

s,ox,recon,ps

14.5 Quadratic surrogate algorithms (s,ox,recon,ps)

Huber's conditions in Theorem 14.4.5 are *sufficient*, but by no means *necessary* for finding parabolic surrogate functions. For example, one can find parabolic surrogate functions even for the nonconvex log-likelihood that arises in transmission tomography [73], see Chapter 19. All that we need is to find a parabola q_i for each ψ_i such that (14.4.4) is satisfied. Such a parabola will always have the form (14.4.8), where one must choose the curvatures $\breve{c}$ to satisfy (14.4.4). For convergence rate reasons, we want the curvatures $\breve{c}$ to be as small as possible, (*cf.* §15.5.3).

Having found such parabola surrogates, we can form a quadratic surrogate for the cost function $\Psi(\boldsymbol{x})$ as follows:

e,ox,recon,ps

$$\Psi(\boldsymbol{x}) = \sum_{i=1}^{N_\mathrm{i}} \psi_i([\boldsymbol{B}\boldsymbol{x}]_i) \leq \phi^{(n)}(\boldsymbol{x}) \triangleq \sum_{i=1}^{N_\mathrm{i}} q_i^{(n)}([\boldsymbol{B}\boldsymbol{x}]_i), \tag{14.5.1}$$

where from (14.4.8)

e,ox,recon,qin

$$\begin{aligned} q_i^{(n)}(t) &\triangleq q_i(t; [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i) && (14.5.2) \\ &= \psi_i([\boldsymbol{B}\boldsymbol{x}^{(n)}]_i) + \dot{\psi}_i([\boldsymbol{B}\boldsymbol{x}^{(n)}]_i)(t - [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i) + \frac{\breve{c}_i^{(n)}}{2} (t - [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i)^2 && (14.5.3) \end{aligned}$$

for some curvature choice $\breve{c}_i^{(n)} \geq \breve{c}_{\mathrm{opt}}(\psi_i, [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i)$. Thus the quadratic surrogate is

$$\begin{aligned} \phi^{(n)}(\boldsymbol{x}) &= \sum_{i=1}^{N_\mathrm{i}} q_i^{(n)}([\boldsymbol{B}\boldsymbol{x}]_i) \\ &= \sum_{i=1}^{N_\mathrm{i}} \left(\psi_i([\boldsymbol{B}\boldsymbol{x}^{(n)}]_i) + \dot{\psi}_i([\boldsymbol{B}\boldsymbol{x}^{(n)}]_i)([\boldsymbol{B}\boldsymbol{x}]_i - [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i) + \frac{\breve{c}_i^{(n)}}{2} ([\boldsymbol{B}\boldsymbol{x}]_i - [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i)^2 \right). \end{aligned}$$

Equivalently, after some simplifications akin to (1.10.13), a quadratic surrogate has the form

e,ox,recon,sur,nonsep

$$\phi^{(n)}(\boldsymbol{x}) = \Psi(\boldsymbol{x}^{(n)}) + \nabla \Psi(\boldsymbol{x}^{(n)})(\boldsymbol{x} - \boldsymbol{x}^{(n)}) + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \boldsymbol{B}' \operatorname{diag}\{\breve{c}_i^{(n)}\} \boldsymbol{B} (\boldsymbol{x} - \boldsymbol{x}^{(n)}). \tag{14.5.4}$$

This surrogate is a special case of the quadratic form (14.3.5) with Hessian $\boldsymbol{J}_n = \boldsymbol{B}' \operatorname{diag}\{\breve{c}_i^{(n)}\} \boldsymbol{B}$.

Having chosen curvatures $\{\breve{c}_i^{(n)}\}$ and thereby designed a quadratic surrogate, the next step is to minimize it (M-step) per (14.1.1). In principle, any general purpose minimization method could be applied. The remainder of this section describes some of the more important choices.

14.5.1 Huber's algorithm

Minimizing the quadratic surrogate function (14.5.4) analytically yields the following iteration [5] (*cf.* (14.3.6)):

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \left[\boldsymbol{B}' \operatorname{diag}\{\breve{c}_i^{(n)}\} \boldsymbol{B} \right]^{-1} \nabla \Psi(\boldsymbol{x}^{(n)}).$$

This iteration is closely related to **iteratively reweighted least squares** methods [92].

Although $\boldsymbol{B}$ is sparse in many imaging problems, the product $\boldsymbol{B}'\boldsymbol{B}$ usually is not sparse, so a direct implementation of the above iteration is impractical in most imaging problems. This impracticality stems from the fact that $\phi^{(n)}$ is a

nonseparable surrogate. In practice, rather than computing a matrix inverse, one would implement the above iteration using the update $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \boldsymbol{\delta}^{(n)}$, where $\boldsymbol{\delta}^{(n)}$ is the solution to the linear system of equations

$$\left[\boldsymbol{B}' \,\mathsf{diag}\left\{\breve{c}_i^{(n)}\right\} \boldsymbol{B}\right] \boldsymbol{\delta}^{(n)} = \nabla \Psi(\boldsymbol{x}^{(n)}) .$$

Because this is a large system, typically an iterative method (such as the conjugate gradient method of §15.6.2) would be used, and implemented in a way that avoids computing $\boldsymbol{B}'\boldsymbol{B}$ explicitly (see (15.5.3)). However, because $\phi^{(n)}$ is only a surrogate for the actual cost function Ψ, it may be inefficient to try to precisely minimize the surrogate for the M-step. It may be more reasonable to simply descend $\phi^{(n)}$, using even as few as one iterations, and then to find a new surrogate.

s,ox,recon,qscd

14.5.2 Coordinate descent of quadratic surrogates

The coordinate descent method described in §11.10.5 is a particularly efficient approach to minimizing quadratic surrogates when the matrix $\boldsymbol{B}$ is precomputed and stored by columns. For each iteration, we first find the quadratic surrogate, *i.e.*, we determine the curvatures in (14.5.3). Then we apply $M \geq 1$ cycles of coordinate descent to descend the surrogate. Then we find new curvatures based on the updated estimate and iterate.

Equating the partial derivative of (14.5.1) to zero using (14.5.3) suggests the following update:

$$x_j^{\mathrm{new}} = x_j^{\mathrm{old}} - \frac{\sum_{i=1}^{N_{\mathrm{i}}} b_{ij}^* \, \dot{q}_i^{(n)}\left([\boldsymbol{B}\boldsymbol{x}^{\mathrm{old}}]_i\right)}{\sum_{i=1}^{N_{\mathrm{i}}} |b_{ij}|^2 \, \breve{c}_i^{(n)}},$$

where

e,ox,recon,ps,dqin

$$\dot{q}_i^{(n)}(t) = \dot{\psi}_i([\boldsymbol{B}\boldsymbol{x}^{(n)}]_i) + \breve{c}_i^{(n)}(t - [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i) . \qquad (14.5.5)$$

However, naive implementations of this update can be very inefficient. To implement this algorithm efficiently, one must maintain $\dot{q}_i^{(n)}([\boldsymbol{B}\boldsymbol{x}]_i)$ as a state vector, by noting that

$$\dot{q}_i^{(n)}([\boldsymbol{B}\boldsymbol{x}^{\mathrm{new}}]_i) = \dot{q}_i^{(n)}\left([\boldsymbol{B}\boldsymbol{x}^{\mathrm{old}}]_i\right) + \breve{c}_i^{(n)} \sum_{j=1}^{n_{\mathrm{p}}} b_{ij}(x_j^{\mathrm{new}} - x_j^{\mathrm{old}}).$$

This is shown in (14.5.6) in the algorithm below. It ensures that $\dot{q}_i$ is always the derivative of $q_i^{(n)}$ evaluated at $\boldsymbol{B}$ times the most recently updated $\boldsymbol{x}$.

Furthermore, by evaluating (14.5.5) at $t = [\boldsymbol{B}\boldsymbol{x}^{(n+1)}]_i$ and rearranging, we see that as long as $\breve{c}_i^{(n)} \neq 0$:

$$[\boldsymbol{B}\boldsymbol{x}^{(n+1)}]_i = [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i + \frac{\dot{q}_i^{(n)}\left([\boldsymbol{B}\boldsymbol{x}^{(n+1)}]_i\right) - \dot{\psi}_i([\boldsymbol{B}\boldsymbol{x}^{(n)}]_i)}{\breve{c}_i^{(n)}},$$

so we can determine $[\boldsymbol{B}\boldsymbol{x}^{(n+1)}]_i$ from the updated state vector $\dot{q}_i^{(n)}\left([\boldsymbol{B}\boldsymbol{x}^{(n+1)}]_i\right)$ *without* performing explicit matrix multiplication. Combining these tricks leads to the efficient **quadratic surrogate coordinate descent** (**QSCD**) algorithm shown in Fig. 14.5.1.

Specific cases of this algorithm are given in and . These algorithms include the nonnegativity constraint as shown in the algorithm above. More generally one can use box constraints (28.9.1). Like all optimization transfer methods, the QSCD algorithm monotonically decreases the cost function Ψ.

Mat *The QSCD approach is poorly suited to interpreted languages like* MATLAB*'s m-files, due to its triply-nested loops.* However, given $\boldsymbol{B}$, $\left\{d_j^{(n)}\right\}$, $\left\{\breve{c}_i^{(n)}\right\}$, and $\{q_i\}$, the inner two loops are independent of the specific cost function.

s,ox,recon,cdqs

14.5.3 Coordinate descent via quadratic surrogates

In the preceding QSCD algorithm, we first find a quadratic surrogate, and then apply coordinate descent to that surrogate. An alternative approach would be to apply the coordinate descent method directly to the original cost function, but to use 1D parabola surrogates as each x_j is updated in (11.10.4). Such an **coordinate-descent via quadratic surrogates** (**CDQS**) approach was described in [93, p. 39]. Because the curvatures are updated for each j, CDQS may converge somewhat faster per iteration than QSCD. However, the computation per iteration for CDQS is much higher than for QSCD, so generally QSCD is the preferable approach.

```
Quadratic surrogate coordinate descent (QSCD)
Initialization: $\tilde{\ell}_i = [\boldsymbol{B x}^{(0)}]_i, \quad i = 1,\ldots,N_{\mathrm{i}}$
Choose a lower bound $\breve{c}_{\min} > 0$
for $n = 0,1,\ldots,$ {
        $\dot{q}_i = \dot{\psi}_i^{(n)} = \dot{\psi}_i\left(\tilde{\ell}_i\right), \qquad i = 1,\ldots,N_{\mathrm{i}}$ % initialize/save derivatives
        $\breve{c}_i^{(n)} = \max\left\{\breve{c}_i\left(\psi_i, \tilde{\ell}_i\right), \breve{c}_{\min}\right\}, \qquad i = 1,\ldots,N_{\mathrm{i}}$ % curvatures
        $d_j^{(n)} = \sum_{i=1}^{N_{\mathrm{i}}} |b_{ij}|^2 \, \breve{c}_i^{(n)}, \qquad j = 1,\ldots,n_{\mathrm{p}}$ % stepsize denominators

    for $m = 1,\ldots,M$ { % optional subiterations
        for $j = 1,\ldots,n_{\mathrm{p}}$ {
            $x_j^{(n+1)} = \left[x_j^{(n)} - \frac{1}{d_j^{(n)}} \sum_{i=1}^{N_{\mathrm{i}}} b_{ij}^* \, \dot{q}_i\right]_+$
            $\dot{q}_i := \dot{q}_i + \breve{c}_i^{(n)} \, b_{ij} (x_j^{(n+1)} - x_j^{(n)}), \qquad \forall i : b_{ij} \neq 0$    (14.5.6)
        }
    }
        $\tilde{\ell}_i := \tilde{\ell}_i + \frac{\dot{q}_i - \dot{\psi}_i^{(n)}}{\breve{c}_i^{(n)}}, \qquad i = 1,\ldots,N_{\mathrm{i}}$
}
```

e,ox,recon,ps,state,q

f,ox,recon,ps,cd

Figure 14.5.1: QSCD algorithm.

s,ox,recon,ps,psd

14.5.4 Preconditioned steepest descent of surrogates

As described in §14.2.4, if constraints such as nonnegativity are unimportant, then a reasonable approach might be to use a preconditioned gradient as a search direction $\boldsymbol{d}^{(n)}$, and then minimize *the surrogate function* along that direction. When $\phi^{(n)}$ is quadratic, we can find analytically that the minimizer over α in (14.2.3) is

$$\alpha_n = \frac{-\nabla \Psi(\boldsymbol{x}^{(n)}) \, \boldsymbol{d}^{(n)}}{[\boldsymbol{B d}^{(n)}]' \operatorname{diag}\{\breve{c}_i^{(n)}\} \, \boldsymbol{B d}^{(n)}}.$$

Using (14.4.2) and (14.4.3), this leads to the algorithm shown in Fig. 14.5.2.

The dominant computation required is one "forward projection" $\boldsymbol{B d}^{(n)}$ and one "back projection" $\boldsymbol{B}' \dot{\boldsymbol{\psi}}$ each iteration. So this algorithm is very practical.

14.5.5 Surrogate semi-conjugate gradients

As discussed in §14.2.5, alternatively we can modify the preceding algorithm by choosing search directions using **conjugate gradient** principles, namely:

$$\boldsymbol{d}^{(n)} = \begin{cases} -\boldsymbol{P} \, \boldsymbol{g}^{(n)}, & n = 0 \\ -\boldsymbol{P} \, \boldsymbol{g}^{(n)} + \gamma_n \boldsymbol{d}^{(n-1)}, & n > 0, \end{cases}$$

where γ_n is chosen according to (11.8.7), (11.8.12), or (11.8.6).

This approach was applied to image denoising in [84].

s,ox,line

14.5.6 Monotonic line searches (s,ox,line)

For general cost functions $\Psi(\boldsymbol{x})$ there are several well-known, general-purpose methods for performing line searches such as (11.5.1) and (11.8.15). See [49]. For cost functions having the **partially separable** form (14.4.1), we can

PSD of surrogates

Initialize:

$$\boldsymbol{l}^{(0)} = \boldsymbol{B}\boldsymbol{x}^{(0)}$$

For $n = 0, 1, 2, \ldots$

$$\begin{aligned}
\boldsymbol{g}^{(n)} &= \nabla \Psi(\boldsymbol{x}^{(n)}) = \boldsymbol{B}' \dot{\boldsymbol{\psi}}(\boldsymbol{l}^{(n)}) \\
\boldsymbol{d}^{(n)} &= -\boldsymbol{P}\,\boldsymbol{g}^{(n)} \\
\boldsymbol{v}^{(n)} &= \boldsymbol{B}\boldsymbol{d}^{(n)} \\
\alpha_n &= \frac{-\left\langle \boldsymbol{g}^{(n)},\, \boldsymbol{d}^{(n)} \right\rangle}{\sum_{i=1}^{N_{\mathrm{i}}} \breve{c}_i^2 \left(v_i^{(n)}\right)} \geq 0 \\
\boldsymbol{x}^{(n+1)} &= \boldsymbol{x}^{(n)} + \alpha_n \boldsymbol{d}^{(n)} \\
\boldsymbol{l}^{(n+1)} &= \boldsymbol{l}^{(n)} + \alpha_n\, \boldsymbol{v}^{(n)}
\end{aligned} \tag{14.5.7}$$

e,ox,recon,ps,psd,ln

f,ox,recon,ps,psd

Figure 14.5.2: QS-PSD algorithm

use parabola surrogates to derive a line-search method that monotonically decreases the cost function, at least in the absence of constraints, as described in [69].

Let $\boldsymbol{d}^{(n)}$ denote the desired search direction and define the 1D cost function

$$f(\alpha) = \Psi(\boldsymbol{x}^{(n)} + \alpha \boldsymbol{d}^{(n)}) = \sum_{i=1}^{N_{\mathrm{i}}} \psi_i\left(\ell_i^{(n)} + \alpha p_i^{(n)}\right),$$

where $\ell_i^{(n)} = [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i$, $p_i^{(n)} = [\boldsymbol{B}\boldsymbol{d}^{(n)}]_i$. (An efficient implementation will update $\ell_i^{(n)}$ recursively as in (14.5.7).) We would like to find the minimizer of $f(\alpha)$. We will do this approximately by starting with an initial guess $\alpha_0 = 0$ and then performing a few iterations yielding a sequence $\{\alpha_k\}$.

Assume that each potential function ψ_i has a parabola surrogate of the form (14.4.8) for some curvature function $\breve{c}_i(\cdot) = \breve{c}(\psi_i, \cdot)$. Then similar to (14.5.1) we can define a surrogate function for f as follows:

$$\begin{aligned}
f(\alpha) \leq g_k^{(n)}(\alpha) &\triangleq \sum_{i=1}^{N_{\mathrm{i}}} q_i\left(\ell_i^{(n)} + \alpha p_i^{(n)}; \ell_i^{(n)} + \alpha_k p_i^{(n)}\right) \\
&= \sum_{i=1}^{N_{\mathrm{i}}} \psi_i\left(\ell_i^{(n)} + \alpha_k p_i^{(n)}\right) + \dot{\psi}_i\left(\ell_i^{(n)} + \alpha_k p_i^{(n)}\right)(\alpha - \alpha_k) p_i^{(n)} + \frac{1}{2} \breve{c}_i\left(\ell_i^{(n)} + \alpha_k p_i^{(n)}\right) \left[(\alpha - \alpha_k) p_i^{(n)}\right]^2 .
\end{aligned}$$

To update α we minimize the surrogate $g_k^{(n)}$ yielding

$$\alpha_{k+1} = \alpha_k - \frac{\sum_{i=1}^{N_{\mathrm{i}}} \dot{\psi}_i\left(\ell_i^{(n)} + \alpha_k p_i^{(n)}\right) p_i^{(n)}}{\sum_{i=1}^{N_{\mathrm{i}}} \breve{c}_i\left(\ell_i^{(n)} + \alpha_k p_i^{(n)}\right) \left|p_i^{(n)}\right|^2}. \tag{14.5.8}$$

e,ox,line,update

This algorithm monotonically decreases $f(\alpha)$ each iteration. It may require a bit more work per iteration than conventional line searches, but it may converge faster (or more robustly) due to its monotonicity. Most of the work is computing $\ell_i^{(n)}$ and $p_i^{(n)}$ and these are required by all descent methods.

MIRT *See* `pl_pcg_qs_ls.m`.

s,ox,sps

14.5.7 Separable quadratic surrogates (s,ox,sps)

The quadratic surrogate function (14.5.4) is **nonseparable**, so for imaging-sized problems, it would have to be minimized iteratively, resulting in iterations within iterations. This is particularly unappealing when constraints such as nonnegativity are required. (Without such constraints, the **PCG algorithm** would be a natural method to minimize a quadratic $\phi^{(n)}$ iteratively.)

To further simplify the M-step, we form a **separable quadratic surrogate** by using the following trick due to De Pierro [11]:

e,ox,dp,add

$$[\boldsymbol{B}\boldsymbol{x}]_i = \sum_{j=1}^{n_{\mathrm{p}}} b_{ij} x_j = \sum_{j=1}^{n_{\mathrm{p}}} \pi_{ij} \left(\frac{b_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i \right), \tag{14.5.9}$$

provided $\sum_{j=1}^{n_{\mathrm{p}}} \pi_{ij} = 1$ and π_{ij} is zero only if b_{ij} is zero. If the π_{ij} values are nonnegative, then we can apply the **convexity inequality** §28.9 to the parabola q_i to write

$$\begin{aligned} q_i^{(n)}([\boldsymbol{B}\boldsymbol{x}]_i) &= q_i^{(n)} \left(\sum_{j=1}^{n_{\mathrm{p}}} \pi_{ij} \left(\frac{b_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i \right) \right) \\ &\leq \sum_{j=1}^{n_{\mathrm{p}}} \pi_{ij}\, q_i^{(n)} \left(\frac{b_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i \right), \end{aligned}$$

where $q_i^{(n)}$ was defined in (14.5.3). Combining these yields the following separable quadratic surrogate for $\Psi(\boldsymbol{x})$:

$$\phi^{(n)}(\boldsymbol{x}) \leq \phi_{\mathrm{SQS}}^{(n)}(\boldsymbol{x}) \triangleq \sum_{\substack{i=1 \\ \pi_{ij} \neq 0}}^{N_{\mathrm{i}}} \sum_{j=1}^{n_{\mathrm{p}}} \pi_{ij}\, q_i^{(n)} \left(\frac{b_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i \right) = \sum_{j=1}^{n_{\mathrm{p}}} \phi_j^{(n)}(x_j)$$

where

$$\phi_j^{(n)}(x_j) = \sum_{\substack{i=1 \\ \pi_{ij} \neq 0}}^{N_{\mathrm{i}}} \pi_{ij}\, q_i^{(n)} \left(\frac{b_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i \right).$$

The derivatives of $\phi_j^{(n)}$ are

e,ox,sps,denjn

$$\begin{aligned} \frac{d}{dx_j} \phi_j^{(n)}(x_j) &= \sum_{\substack{i=1 \\ \pi_{ij} \neq 0}}^{N_{\mathrm{i}}} b_{ij}^*\, \dot{q}_i^{(n)} \left(\frac{b_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i \right) \\ d_j^{(n)} \triangleq \frac{d^2}{dx_j^2} \phi_j^{(n)}(\cdot) &= \sum_{\substack{i=1 \\ \pi_{ij} \neq 0}}^{N_{\mathrm{i}}} \frac{|b_{ij}|^2}{\pi_{ij}} \ddot{q}_i^{(n)} = \sum_{\substack{i=1 \\ \pi_{ij} \neq 0}}^{N_{\mathrm{i}}} \frac{|b_{ij}|^2}{\pi_{ij}} \check{c}_i^{(n)}, \end{aligned} \tag{14.5.10}$$

so the following "matched derivative" condition holds:

$$\left. \frac{d}{dx_j} \phi_j^{(n)}(x_j) \right|_{x_j = x_j^{(n)}} = \sum_{i=1}^{N_{\mathrm{i}}} b_{ij}^*\, \dot{q}_i^{(n)}([\boldsymbol{B}\boldsymbol{x}^{(n)}]_i) = \sum_{i=1}^{N_{\mathrm{i}}} b_{ij}^*\, \dot{\psi}_i([\boldsymbol{B}\boldsymbol{x}^{(n)}]_i) = \frac{\partial}{\partial x_j} \Psi(\boldsymbol{x}^{(n)}).$$

Combining, the separable quadratic surrogate satisfies the conditions (14.1.4) and has the form

e,ox,sps,sur

$$\phi_{\mathrm{SQS}}^{(n)}(\boldsymbol{x}) = \Psi(\boldsymbol{x}^{(n)}) + \nabla \Psi(\boldsymbol{x}^{(n)})(\boldsymbol{x} - \boldsymbol{x}^{(n)}) + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}^{(n)})' \operatorname{diag}\{d_j^{(n)}\} (\boldsymbol{x} - \boldsymbol{x}^{(n)}). \tag{14.5.11}$$

Because this quadratic surrogate is separable, it has a diagonal Hessian, so it is trivial to perform the M-step (14.1.1), yielding the following general **separable quadratic surrogate** algorithm.

Separable quadratic surrogate algorithm

Initialization: choose π_{ij} factors such that $\sum_{j=1}^{n_\mathrm{p}} \pi_{ij} = 1$ and $\pi_{ij} \geq 0$ and $\pi_{ij} = 0$ only if $b_{ij} = 0$, *e.g.*, $\pi_{ij} = |b_{ij}|/b_i$, $b_i = \sum_{j=1}^{n_\mathrm{p}} |b_{ij}|$.

```
For each iteration:
    Compute $[\boldsymbol{B}\boldsymbol{x}^{(n)}]_i, \forall i$
    Choose curvatures $\breve{c}_i^{(n)} = \breve{c}(\psi_i, [\boldsymbol{B}\boldsymbol{x}^{(n)}]_i)$ so (14.4.8) satisfies (14.4.4)
```

e,ox,denjn

$$d_j^{(n)} = \sum_{\substack{i=1\\ \pi_{ij}\neq 0}}^{N_\mathrm{i}} \frac{|b_{ij}|^2}{\pi_{ij}} \breve{c}_i^{(n)}, \qquad j = 1, \ldots, n_\mathrm{p} \tag{14.5.12}$$

e,ox,sps

$$x_j^{(n+1)} = \left[x_j^{(n)} - \frac{1}{d_j^{(n)}} \frac{\partial}{\partial x_j} \Psi(\boldsymbol{x}^{(n)}) \right]_+, \qquad j = 1, \ldots, n_\mathrm{p}. \tag{14.5.13}$$

In matrix-vector form, the update is:

$$\boldsymbol{x}^{(n+1)} = \left[\boldsymbol{x}^{(n)} - \mathsf{diag}\left\{ \frac{1}{d_j^{(n)}} \right\} \nabla \Psi(\boldsymbol{x}^{(n)}) \right]_+,$$

which is a kind of diagonally-preconditioned projected gradient descent. This algorithm is entirely parallelizable because one can update all pixels simultaneously. Provided one has chosen properly the curvatures $\left\{\breve{c}_i^{(n)}\right\}$ in (14.4.8), this algorithm will monotonically decrease the cost function $\Psi(\boldsymbol{x}^{(n)})$.

This algorithm is a prototype for several of the algorithms discussed in this *book*. It is basically a form of diagonally-scaled **gradient descent**, but the diagonal scaling is constructed using the form of (14.4.1) to guarantee that the cost function monotonicity decreases. Because $\phi_{\mathrm{SQS}}^{(n)}(\cdot)$ is separable, box constraints are also easily enforced.

The price we pay for the intrinsic monotonicity is the "stepsize calculation" (14.5.12). However, in some problems, we can choose $\breve{c}_i^{(n)}$ values that are independent of iteration, such as the precomputed **maximum curvature** (14.4.12). Because these $\breve{c}_i$ values are independent of $\boldsymbol{x}^{(n)}$, we can precompute the d_j values prior to iterating. However, using the maximum curvature may reduce the convergence rate, so one must examine the trade-off between number of iterations and work per iteration in any given problem.

Comparing (14.5.11) with the nonseparable surrogate (14.5.4), we find that

e,ox,sps,BCB<=D

$$\boldsymbol{B}' \,\mathsf{diag}\left\{\breve{c}_i^{(n)}\right\} \boldsymbol{B} \preceq \mathsf{diag}\left\{d_j^{(n)}\right\}. \tag{14.5.14}$$

(For an alternate proof based on the Geršgorin disk theorem and diagonal dominance, see Corollary 27.2.7 and Theorem 27.2.8.) An interesting special case is

$$\mathsf{diag}\{(\boldsymbol{1}'\boldsymbol{v})\,\boldsymbol{v}\} \succeq \boldsymbol{v}\,\boldsymbol{v}',$$

for any real vector $\boldsymbol{v}$ with nonnegative elements.

14.5.7.1 Comparison to general additive update

The additive update (14.3.12) for separable surrogates was derived for general quadratic cost functions. In contrast, the update (14.5.13) is applicable only to the family of cost functions given in (14.4.1). How do these iterative method compare? Specifically, suppose that $\Psi(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{x})' \,\mathsf{diag}\{w_i\}(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{x})$. Then the update (14.3.12) has a step size of

$$\frac{\alpha_j^{(n)}}{\frac{\partial^2}{\partial x_j^2} \Psi(\boldsymbol{x}^{(n)})} = \frac{\alpha_j^{(n)}}{\sum_{i=1}^{N_\mathrm{i}} |b_{ij}|^2 w_i},$$

whereas the update (14.5.13) has a step size of

$$\frac{1}{\displaystyle\sum_{\substack{i=1\\ \pi_{ij}\neq 0}}^{N_\mathrm{i}} \frac{|b_{ij}|^2}{\pi_{ij}} w_i}.$$

If we were to choose $\pi_{ij} = \alpha_j^{(n)}$, then the two algorithms would be identical. However, as discussed in §15.6.5.2, that choice for π_{ij} would be suboptimal. The earlier method (14.3.12) was generally applicable to convex cost functions, whereas the method (14.5.13) is tailored to the types of cost functions that arise in image reconstruction problems. Here we see that a derivation matched to the form of the cost function of interest provides greater flexibility in the algorithm design, potentially leading to faster convergence.

s,ox,ex2

14.5.7.2 Example: denoising SQS (s,ox,ex2)

x,ox,ex2

Example 14.5.1 *Consider the measurement model*

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon}$ is zero-mean uncorrelated noise vector. The ***signal denoising*** *problem is to estimate $\boldsymbol{x}$ from $\boldsymbol{y}$. Numerous algorithms exist for this problem; we consider estimating $\boldsymbol{x}$ by minimizing a regularized least-squares cost function of the following form:*

$$\begin{aligned}\Psi(\boldsymbol{x}) &= \sum_{i=1}^{n_\mathrm{d}} \frac{1}{2}(y_i - x_i)^2 + \sum_{k=1}^{n_\mathrm{p}-1} \psi_k(x_{k+1} - x_k) \\ &= \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + \sum_k \psi_k([\boldsymbol{C}\boldsymbol{x}]_k),\end{aligned}$$

where, for the 1D problem considered in this example, $\boldsymbol{C}$ is the $n_\mathrm{p} - 1 \times n_\mathrm{p}$ first finite-difference matrix[6] *defined in (1.8.4), so that $[\boldsymbol{C}\boldsymbol{x}]_k = x_{k+1} - x_k,\ k = 1, \ldots, n_\mathrm{p} - 1$. Each ψ_k is in the large family of (usually nonquadratic)* ***potential functions*** *considered in §14.4.4.4, with surrogate curvatures denoted $\omega_k(t) = \dot{\psi}_k(t)/t$. In particular, if ψ_k is a (corner-rounded) version of the absolute value function, such as the hyperbola potential or the Fair potential, then this problem provides a method for (anisotropic) TV denoising. Thus a suitable* ***quadratic surrogate function*** *is*

e,ox,ex2,surr

$$\phi^{(n)}(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + \sum_k q_k([\boldsymbol{C}\boldsymbol{x}]_k; [\boldsymbol{C}\boldsymbol{x}^{(n)}]_k), \qquad (14.5.15)$$

where q_k was defined by (14.4.15). To find the minimizer of $\phi^{(n)}(\cdot)$ using Huber's algorithm (14.3.6), note that

$$\frac{\partial}{\partial x_j}\phi^{(n)}(\boldsymbol{x}) = x_j - y_j + \sum_k c_{kj}\,\omega_k([\boldsymbol{C}\boldsymbol{x}^{(n)}]_k)[\boldsymbol{C}\boldsymbol{x}]_k,$$

so

$$\begin{aligned}\nabla\,\phi^{(n)}(\boldsymbol{x})\big|_{\boldsymbol{x}=\boldsymbol{x}^{(n)}} &= \boldsymbol{x}^{(n)} - \boldsymbol{y} + \boldsymbol{C}'\Omega(\boldsymbol{x}^{(n)})\boldsymbol{C}\boldsymbol{x}^{(n)} \\ &= \boldsymbol{x}^{(n)} - \boldsymbol{y} + \boldsymbol{C}'\dot{\boldsymbol{\psi}}(\boldsymbol{C}\boldsymbol{x}^{(n)}) = \nabla\,\Psi(\boldsymbol{x}^{(n)}),\end{aligned}$$

where $n_\mathrm{k} = n_\mathrm{p} - 1$, $\dot{\boldsymbol{\psi}}$ was defined in (14.4.3), $\Omega(\boldsymbol{x}) \triangleq \mathsf{diag}\{\omega_k([\boldsymbol{C}\boldsymbol{x}]_k)\}$, and

$$\nabla^2\,\phi^{(n)}(\boldsymbol{x}) = \boldsymbol{I} + \boldsymbol{C}'\Omega(\boldsymbol{x}^{(n)})\boldsymbol{C}.$$

So Huber's algorithm is simply:

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \left[\boldsymbol{I} + \boldsymbol{C}'\Omega(\boldsymbol{x}^{(n)})\boldsymbol{C}\right]^{-1}\left[\boldsymbol{y} - \boldsymbol{x}^{(n)} - \boldsymbol{C}'\Omega(\boldsymbol{x}^{(n)})\boldsymbol{C}\boldsymbol{x}^{(n)}\right].$$

This algorithm requires inversion of the banded Hessian matrix $\boldsymbol{I} + \boldsymbol{C}'\Omega(\boldsymbol{x}^{(n)})\boldsymbol{C}$ each iteration, and efficient algorithms for this problem exist, based on the ***Cholesky decomposition****, e.g., [94, 95]. Because each ω_k is bounded above by $\omega_k(0)$, to reduce computation, one could apply Böhning and Lindsay's algorithm (14.3.7) as follows:*

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \left[\boldsymbol{I} + \boldsymbol{C}'\Omega_0\boldsymbol{C}\right]^{-1}\left[\boldsymbol{y} - \boldsymbol{x}^{(n)} - \boldsymbol{C}'\Omega(\boldsymbol{x}^{(n)})\boldsymbol{C}\boldsymbol{x}^{(n)}\right],$$

where $\Omega_0 \triangleq \mathsf{diag}\{\omega_k(0)\}$. Alternatively, one could apply a coordinate descent method to minimize the quadratic surrogate (14.5.15).

[6] In this case, the matrix $\boldsymbol{B}$ in (14.4.1) is $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{I}_{n_\mathrm{d}} \\ \boldsymbol{C} \end{bmatrix}$.

To form an algorithm that is easily implemented in MATLAB*, we apply De Pierro's trick (14.5.9) [11]:*

$$x_{j+1} - x_j = \frac{1}{2}\left(2\left[x_{j+1} - x_{j+1}^{(n)}\right] + \left[x_{j+1}^{(n)} - x_j^{(n)}\right]\right) + \frac{1}{2}\left(-2\left[x_j - x_j^{(n)}\right] + \left[x_{j+1}^{(n)} - x_j^{(n)}\right]\right),$$

so for any convex function q (see §28.9):

$$q(x_{j+1} - x_j) \leq \frac{1}{2} q\left(2\left[x_{j+1} - x_{j+1}^{(n)}\right] + \left[x_{j+1}^{(n)} - x_j^{(n)}\right]\right) + \frac{1}{2} q\left(-2\left[x_j - x_j^{(n)}\right] + \left[x_{j+1}^{(n)} - x_j^{(n)}\right]\right).$$

Thus, considering (14.5.15) and the preceding inequality, a ***separable quadratic surrogate*** *for this problem is*

$$\phi_2^{(n)}(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + \sum_k \left[\frac{1}{2} q_k\left(2\left[x_{j+1} - x_{j+1}^{(n)}\right] + \left[x_{j+1}^{(n)} - x_j^{(n)}\right]\right) + \frac{1}{2} q_k\left(-2\left[x_j - x_j^{(n)}\right] + \left[x_{j+1}^{(n)} - x_j^{(n)}\right]\right)\right],$$

where

$$\nabla\, \phi_2^{(n)}(\boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{x}^{(n)}} = \boldsymbol{x}^{(n)} - \boldsymbol{y} + \boldsymbol{C}'\Omega(\boldsymbol{x}^{(n)})\boldsymbol{C}\boldsymbol{x}^{(n)} = \nabla\, \Psi(\boldsymbol{x}^{(n)})$$

and

$$\nabla^2\, \phi_2^{(n)}(\boldsymbol{x}) = \boldsymbol{I} + \mathsf{diag}\left\{d_j^{(n)}\right\}$$

where

$$d_j^{(n)} = \begin{cases} 2\,\omega_1([\boldsymbol{C}\boldsymbol{x}^{(n)}]_1), & j = 1 \\ 2\,\omega_{j-1}([\boldsymbol{C}\boldsymbol{x}^{(n)}]_{j-1}) + 2\,\omega_j([\boldsymbol{C}\boldsymbol{x}^{(n)}]_j), & j = 2, \ldots, n_{\mathrm{p}} - 1 \\ 2\,\omega_{n_{\mathrm{p}}-1}\left([\boldsymbol{C}\boldsymbol{x}^{(n)}]_{n_{\mathrm{p}}-1}\right), & j = n_{\mathrm{p}}. \end{cases}$$

Thus, Huber's algorithm for this alternative surrogate is:

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \left[\boldsymbol{I} + \mathsf{diag}\left\{d_j^{(n)}\right\}\right]^{-1}\left[\boldsymbol{y} - \boldsymbol{x}^{(n)} - \boldsymbol{C}'\Omega(\boldsymbol{x}^{(n)})\boldsymbol{C}\boldsymbol{x}^{(n)}\right],$$

which is trivial to implement because the required inverse only involves a diagonal matrix. To simplify even further, let $\omega_{\max} = \max_k \omega_k(0)$ *. Then a version of Böhning and Lindsay's algorithm for this problem is simply:*

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \frac{1}{1 + 4\omega_{\max}}\left[\boldsymbol{y} - \boldsymbol{x}^{(n)} - \boldsymbol{C}'\Omega(\boldsymbol{x}^{(n)})\boldsymbol{C}\boldsymbol{x}^{(n)}\right].$$

s,ox,it

14.5.7.3 Example: iterative thresholding (s,ox,it)

x,ox,it

Example 14.5.2 *Consider the regularized least-squares cost function of the following form:*

e,ox,it,Kx

$$\Psi_1(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_{\boldsymbol{W}^{1/2}}^2 + \beta\,\|\boldsymbol{U}'\boldsymbol{x}\|_1, \qquad (14.5.16)$$

where here $\boldsymbol{U}$ *is a unitary matrix such as an orthonormal wavelet basis. This type of cost function arises in* ***compressed sensing*** *formulations where* $\boldsymbol{U}'\boldsymbol{x}$ *is assumed to be* ***sparse****. In this example we use separable quadratic surrogates to derive an* ***iterative soft thresholding*** *(****IST****) algorithm [96, 97].*

When $\boldsymbol{U}$ *is unitary, this problem is equivalent to finding* $\hat{\boldsymbol{x}} = \boldsymbol{U}\hat{\boldsymbol{z}}$*, where*

$$\hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} \Psi(\boldsymbol{z}), \qquad \Psi(\boldsymbol{z}) = \Psi(\boldsymbol{U}\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{U}\boldsymbol{z}\|_{\boldsymbol{W}^{1/2}}^2 + \beta\,\|\boldsymbol{z}\|_1.$$

Rewrite the data-fitting term as follows:

$$\begin{aligned} \text{Ł}(\boldsymbol{z}) &\triangleq \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{U}\boldsymbol{z}\|_{\boldsymbol{W}^{1/2}}^2 \\ &= \text{Ł}(\boldsymbol{z}^{(n)}) + (\boldsymbol{z} - \boldsymbol{z}^{(n)})'\nabla\,\text{Ł}(\boldsymbol{z}^{(n)}) + \frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}^{(n)})'\boldsymbol{U}'\boldsymbol{A}'\boldsymbol{W}\boldsymbol{A}\boldsymbol{U}(\boldsymbol{z} - \boldsymbol{z}^{(n)}). \end{aligned}$$

Let $\check{c}$ be a constant such that $\boldsymbol{A}'\boldsymbol{W}\boldsymbol{A} \preceq \check{c}\,\boldsymbol{I}$. See for example Theorem 14.3.1 or let $\check{c} = \max_j d_j$ defined in (14.5.10). Then a suitable ***quadratic surrogate function*** *for the data-fit term is*

$$\begin{aligned}\phi_1^{(n)}(\boldsymbol{x}) &= \text{Ł}(\boldsymbol{z}^{(n)}) + (\boldsymbol{z} - \boldsymbol{z}^{(n)})' \nabla\, \text{Ł}(\boldsymbol{z}^{(n)}) + \frac{\check{c}}{2} (\boldsymbol{z} - \boldsymbol{z}^{(n)})'(\boldsymbol{z} - \boldsymbol{z}^{(n)}) \\ &\equiv \frac{\check{c}}{2} \left\| \boldsymbol{z} - \boldsymbol{z}^{(n)} + \frac{1}{\check{c}} \nabla\, \text{Ł}(\boldsymbol{z}^{(n)}) \right\|^2,\end{aligned}$$

because $\boldsymbol{U}'\boldsymbol{U} = \boldsymbol{I}$. Thus a surrogate for the overall cost function is

e,ox,it,surr

$$\phi^{(n)}(\boldsymbol{z}) \triangleq \frac{\check{c}}{2} \left\| \boldsymbol{z} - \boldsymbol{z}^{(n)} + \frac{1}{\check{c}} \nabla\, \text{Ł}(\boldsymbol{z}^{(n)}) \right\|^2 + \beta \left\| \boldsymbol{z} \right\|_1. \tag{14.5.17}$$

Following Problem 1.12, define the soft-thresholding function

$$\operatorname{soft}\{t; \alpha\} = \arg\min_s \frac{1}{2} |t - s|^2 + \alpha\, |s| = t\,(1 - \alpha / |t|)\, \mathbb{I}_{\{|t| > \alpha\}}.$$

Because $\phi^{(n)}(\cdot)$ in (14.5.17) is ***separable****, its minimizer is expressed easily in terms of the soft-thresholding function:*

e,ox,it,znn,cgrad

e,ox,it,znn

$$\begin{aligned}\boldsymbol{z}^{(n+1)} = \arg\min_{\boldsymbol{z}} \phi^{(n)}(\boldsymbol{z}) &= \operatorname{soft}\left\{ \boldsymbol{z}^{(n)} - \frac{1}{\check{c}} \nabla\, \text{Ł}(\boldsymbol{z}^{(n)});\, \frac{\beta}{\check{c}} \right\} &(14.5.18) \\ &= \operatorname{soft}\left\{ \boldsymbol{z}^{(n)} + \frac{1}{\check{c}} \boldsymbol{U}'\boldsymbol{A}'\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{U}\boldsymbol{z}^{(n)});\, \frac{\beta}{\check{c}} \right\}. &(14.5.19)\end{aligned}$$

We can rewrite this algorithm in terms of $\boldsymbol{x}^{(n)}$ as follows:

e,ox,it,xnn

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{U} \operatorname{soft}\left\{ \boldsymbol{U}' \left(\boldsymbol{x}^{(n)} + \frac{1}{\check{c}} \boldsymbol{A}'\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{(n)}) \right);\, \frac{\beta}{\check{c}} \right\}. \tag{14.5.20}$$

This method is appealingly simple, but has the practical drawback of slow convergence because the curvature $\check{c}$ can be large.

MIRT *See* `mri_cs_ist_example.m`.

See Problem 14.10 for generalizations to non-quadratic data-fit terms. Generalizing this example to use **ordered subsets** is an interesting *open problem*. Although we focused on $\|\cdot\|_1$ in this example, the principles generalize to any sparsity prior, including $\|\cdot\|_1$, by using the other thresholding functions considered in Problem 1.12.

In this derivation we used the scaled identity $\check{c}\,\boldsymbol{I}$ to upper bound the Hessian. For certain types of transforms $\boldsymbol{U}$, such as wavelet transforms, one can use specific diagonal upper bounds that may provide faster convergence [98].

s,ox,group

14.5.8 Grouped coordinate algorithms (s,ox,group)

All of the algorithms described above either update all pixels simultaneously, or update pixels one-by-one sequentially (*i.e.*, in the coordinate descent approach). In some cases it can be beneficial to update a *group* of parameters simultaneously, holding all other parameters at their most recent values. Such a method is called **grouped coordinate-descent** (**GCD**) or **block coordinate-descent**. This type of approach is applicable to both the cost function Ψ as well as to the surrogate functions in the optimization transfer framework.

We first establish some notation. An **index set** $\mathcal{S}$ is a nonempty subset of the set $\{1, \ldots, n_{\mathrm{p}}\}$. The set $\tilde{\mathcal{S}}$ denotes the complement of $\mathcal{S}$ intersected with $\{1, \ldots, n_{\mathrm{p}}\}$. Let $|\mathcal{S}|$ denote the cardinality of $\mathcal{S}$. We use $\boldsymbol{x}_{\mathcal{S}}$ to denote the $|\mathcal{S}|$-dimensional vector consisting of the $|\mathcal{S}|$ elements of $\boldsymbol{x}$ indexed by the members of $\mathcal{S}$. We similarly define $\boldsymbol{x}_{\tilde{\mathcal{S}}}$ as the $n_{\mathrm{p}} - |\mathcal{S}|$ dimensional vector consisting of the remaining elements of $\boldsymbol{x}$. For example, if $n_{\mathrm{p}} = 5$ and $\mathcal{S} = \{1, 3, 4\}$, then $\tilde{\mathcal{S}} = \{2, 5\}$, $\boldsymbol{x}_{\mathcal{S}} = (x_1, x_3, x_4)$, and $\boldsymbol{x}_{\tilde{\mathcal{S}}} = (x_2, x_5)$. Occasionally we use $\mathcal{S}$ as a superscript of a function or matrix, to serve as a reminder that the function or matrix depends on the choice of $\mathcal{S}$.

One more notational convention will be used hereafter. Functions like $\Psi(\boldsymbol{x})$ expect a n_{p}-dimensional vector argument, but it is often convenient to split the argument $\boldsymbol{x}$ into two vectors: $\boldsymbol{x}_{\mathcal{S}}$ and $\boldsymbol{x}_{\tilde{\mathcal{S}}}$, as defined above. Therefore, we define expressions such as the following to be equivalent: $\Psi(\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}_{\tilde{\mathcal{S}}}) = \Psi(\boldsymbol{x})$.

14.5.8.1 Direct minimization

In a direct **grouped coordinate-descent** (**GCD**) method, one sequences through different index sets $\mathcal{S}_n$ and updates only the elements $\boldsymbol{x}_{\mathcal{S}_n}$ of $\boldsymbol{x}$ while holding the other parameters $\boldsymbol{x}_{\tilde{\mathcal{S}}_n}$ fixed [49]. At the nth iteration one would like to assign $\boldsymbol{x}_{\mathcal{S}_n}^{(n+1)}$ to the argument that minimizes $\Psi\left(\boldsymbol{x}_{\mathcal{S}_n}, \boldsymbol{x}_{\tilde{\mathcal{S}}_n}^{(n)}\right)$ over $\boldsymbol{x}_{\mathcal{S}_n}$, as summarized in the following algorithm.

"Direct" GCD "Algorithm"

$$\begin{aligned} &\text{Choose } \mathcal{S}_n \\ \boldsymbol{x}_{\tilde{\mathcal{S}}_n}^{(n+1)} &= \boldsymbol{x}_{\tilde{\mathcal{S}}_n}^{(n)} \\ \boldsymbol{x}_{\mathcal{S}_n}^{(n+1)} &= \operatorname*{arg\,min}_{\boldsymbol{x}_{\mathcal{S}_n}} \Psi\left(\boldsymbol{x}_{\mathcal{S}_n}, \boldsymbol{x}_{\tilde{\mathcal{S}}_n}^{(n)}\right). \end{aligned} \qquad (14.5.21)$$

e,ox,gcd,direct

As always, the minimization will be subject to any applicable constraints. This type of approach has been applied in a variety of fields [99–102]. This approach is globally convergent under remarkably broad conditions [103].

In many imaging problems, the minimization (14.5.21) is difficult, even if $\mathcal{S}_n$ only consists of a few pixels. One could apply any of the previously described iterative methods, such as the Newton-Raphson algorithm, to perform the minimization (14.5.21), which would require subiterations within the iterations. A more elegant and general approach is to combine the grouped coordinate concept with the optimization transfer approach.

14.5.8.2 Surrogate minimization

Combining the GCD approach with a surrogate function leads to "indirect" GCD algorithms. For a fully simultaneous iteration where all pixels are updated simultaneously, the surrogate function condition (14.1.2) guarantees monotonic decreases in Ψ. However, by working with smaller groups of parameters at a time, one can relax (14.1.2) considerably. Essentially, the condition (14.1.2) only needs to hold *on the restriction of* Ψ *to the parameters being updated.* In mathematical terms, when updating the group $\boldsymbol{x}_{\mathcal{S}}$, we choose a surrogate function $\phi_{\mathcal{S}}$ that satisfies the following condition:

$$\Psi(\boldsymbol{x}^{(n)}) - \Psi\left(\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}_{\tilde{\mathcal{S}}}^{(n)}\right) \geq \phi_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}^{(n)}; \boldsymbol{x}^{(n)}) - \phi_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}; \boldsymbol{x}^{(n)}), \qquad \forall \boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}^{(n)}, \qquad (14.5.22)$$

e,ox,gcd,mono

or the following equivalent pair of conditions:

$$\begin{aligned} \phi_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}^{(n)}; \boldsymbol{x}^{(n)}) &= \Psi(\boldsymbol{x}^{(n)}), \qquad \forall \boldsymbol{x}^{(n)} \\ \phi_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}; \boldsymbol{x}^{(n)}) &\geq \Psi\left(\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}_{\tilde{\mathcal{S}}}^{(n)}\right), \qquad \forall \boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}^{(n)}. \end{aligned}$$

Incorporating such surrogate functions into the grouped coordinate descent algorithm (14.5.21) leads to the following very general family of methods.

"Indirect" GCD "Algorithm"

$$\begin{aligned} &\text{Choose } \mathcal{S}_n \\ &\text{Choose } \phi_{\mathcal{S}_n}(\boldsymbol{x}_{\mathcal{S}_n}; \boldsymbol{x}^{(n)}) \text{ satisfying (14.5.22)} \\ \boldsymbol{x}_{\tilde{\mathcal{S}}_n}^{(n+1)} &= \boldsymbol{x}_{\tilde{\mathcal{S}}_n}^{(n)} \\ \boldsymbol{x}_{\mathcal{S}_n}^{(n+1)} &= \operatorname*{arg\,min}_{\boldsymbol{x}_{\mathcal{S}_n}} \phi_{\mathcal{S}_n}(\boldsymbol{x}_{\mathcal{S}_n}, \boldsymbol{x}_{\tilde{\mathcal{S}}_n}^{(n)}). \end{aligned} \qquad (14.5.23)$$

e,ox,gcd

Many of the algorithms discussed in this *book* belong to this class of methods. Specific examples of such methods in the imaging literature include [104–106].

14.5.9 Examples (s,ox,ex)

s,ox,ex

This section gives some examples of the use of optimization transfer with quadratic surrogate for minimizing non-quadratic cost functions.

s,ox,ex1

14.5.9.1 Simple 1D minimization (s,ox,ex1)

x,ox,surrogate

Example 14.5.3 *Consider the problem of minimizing the cost function*

$$\Psi(x) = \psi(x) + 2\,\psi(x-1),$$

where the strictly convex ***potential function*** ψ *is the following hyperbola:*

$$\psi(x) = \sqrt{x^2+1}.$$

(This potential function arises in edge-preserving image recovery; see [76] and Table 2.1.) By Theorem 14.4.5 the following quadratic function is a suitable surrogate function for ψ*:*

$$q(x; x^{(n)}) = \psi(x^{(n)}) + \dot{\psi}(x^{(n)})(x - x^{(n)}) + \frac{1}{2}\,\omega_\psi(x^{(n)})(x - x^{(n)})^2,$$

thus a suitable surrogate function for Ψ *is*

$$\phi^{(n)}(x) \triangleq q(x; x^{(n)}) + 2q(x-1; x^{(n)} - 1).$$

Applying Huber's algorithm (14.3.6) using ω_ψ *from Table 2.1 yields the following simple iteration:*

$$x^{(n+1)} = x^{(n)} - \frac{\dot{\psi}(x^{(n)}) + 2\,\dot{\psi}(x^{(n)} - 1)}{\omega_\psi(x^{(n)}) + 2\,\omega_\psi(x^{(n)} - 1)} = x^{(n)} - \frac{\frac{x^{(n)}}{\sqrt{(x^{(n)})^2+1}} + 2\frac{x^{(n)}-1}{\sqrt{(x^{(n)}-1)^2+1}}}{\frac{1}{\sqrt{(x^{(n)})^2+1}} + 2\frac{1}{\sqrt{(x^{(n)}-1)^2+1}}}.$$

Fig. 14.5.3 shows the cost function Ψ*, the surrogate function* $\phi^{(n)}$*, and illustrates that a few iterations of this algorithm leads rapidly to the minimizer of* Ψ*.*

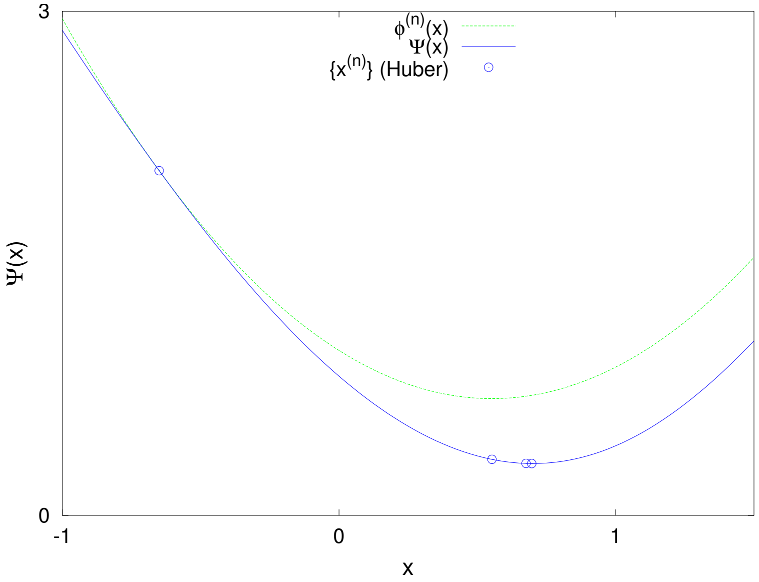


Figure 14.5.3: Finding the minimizer of a 1D function $\Psi(x)$ using optimization transfer (Huber's iteration) with a quadratic surrogate function $\phi^{(n)}(x)$. fig_ox_recon_ex1a

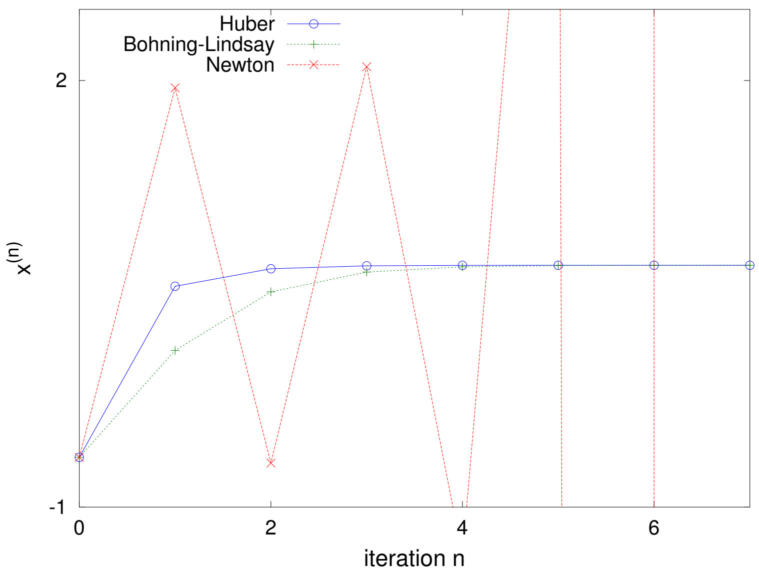
Because the curvature of ψ *is bounded by unity, Böhning and Lindsay's lower-bound algorithm (14.3.7) has the following even simpler form:*

$$x^{(n+1)} = x^{(n)} - \frac{\dot{\psi}(x^{(n)}) + 2\,\dot{\psi}(x^{(n)} - 1)}{\omega_\psi(0) + 2\,\omega_\psi(0)} = x^{(n)} - \frac{1}{3}\left[\frac{x^{(n)}}{\sqrt{(x^{(n)})^2+1}} + 2\frac{x^{(n)}-1}{\sqrt{(x^{(n)}-1)^2+1}}\right].$$

For comparison, the Newton-Raphson algorithm (11.4.3) for this problem is given by

$$x^{(n+1)} = x^{(n)} - \frac{\frac{x^{(n)}}{\sqrt{(x^{(n)})^2+1}} + 2\frac{x^{(n)}-1}{\sqrt{(x^{(n)}-1)^2+1}}}{\left[(x^{(n)})^2+1\right]^{-3/2} + 2\left[(x^{(n)}-1)^2+1\right]^{-3/2}}.$$

Fig. 14.5.4 compares $\{x^{(n)}\}$ for the three algorithms given above: Huber's, Böhning and Lindsay's, and Newton Raphson. Divergence of NR is caused by the small curvature of the hyperbola ψ for large arguments, leading to a poor quadratic approximation and inappropriately large steps. Huber's method converges faster than Böhning and Lindsay's, because the latter uses unnecessarily large curvatures for the surrogate functions.



fig_ox_recon_ex1b

Figure 14.5.4: Iterates $x^{(n)}$ for 1D example for three algorithms.

s,ox,l1

14.5.9.2 l_1 regression (s,ox,l1)

x,ox,l1,reg

Example 14.5.4 *Consider the **ℓ_1 regression** problem of minimizing the following cost function*

$$\check{\Psi}(\boldsymbol{x}) = \sum_{i=1}^{n_\mathrm{d}} |[\boldsymbol{A}\boldsymbol{x}]_i - b_i| = \sum_{i=1}^{n_\mathrm{d}} \check{\psi}_i([\boldsymbol{A}\boldsymbol{x}]_i), \qquad \check{\psi}_i(t) = |t - b_i|\,,$$

*for $\boldsymbol{x} \in \mathbb{R}^{n_\mathrm{p}}$ and $\boldsymbol{A} \in \mathbb{R}^{n_\mathrm{d} \times n_\mathrm{p}}$. This notorious cost function is convex but need not be strictly convex, so it may not have a unique minimizer. It is also not everywhere differentiable. We can remedy these problems by replacing the absolute value function with a strictly convex **potential function** that is a close approximation, such as the following hyperbola:*

e,ox,l1,hyper

$$\psi(t) = \sqrt{t^2 + \varepsilon} \approx |t|, \tag{14.5.24}$$

*for some small $\varepsilon > 0$. This approximation is called **corner rounding**. So we replace $\check{\Psi}$ with*

$$\Psi(\boldsymbol{x}) = \sum_{i=1}^{n_\mathrm{d}} \psi_i([\boldsymbol{A}\boldsymbol{x}]_i), \qquad \psi_i(t) = \psi(t - b_i)\,.$$

We now have a convex minimization problem; if n_p is small, then Huber's algorithm (14.3.6) is a reasonable choice.

By Theorem 14.4.5, a parabola of the form (14.4.15) is a suitable surrogate function for ψ, when $\omega_\psi(s) = \frac{1}{\sqrt{s^2+\varepsilon}}$. Thus, following (14.4.5), a suitable surrogate function for Ψ is

$$\phi^{(n)}(x) \triangleq \sum_{i=1}^{n_\mathrm{d}} q([\boldsymbol{A}\boldsymbol{x}]_i - b_i\,;\, [\boldsymbol{A}\boldsymbol{x}^{(n)}]_i - b_i).$$

The gradient of this surrogate at $\boldsymbol{x}^{(n)}$ is

$$\nabla\, \phi^{(n)}(x)\big|_{\boldsymbol{x}=\boldsymbol{x}^{(n)}} = \boldsymbol{A}'\boldsymbol{D}(\omega_\psi([\boldsymbol{A}\boldsymbol{x}^{(n)}]_i - b_i))\,([\boldsymbol{A}\boldsymbol{x}^{(n)}]_i - b_i) = \nabla\Psi(\boldsymbol{x}^{(n)})\,.$$

The Hessian of this surrogate is

$$\nabla^2\nabla\, \phi^{(n)}(\boldsymbol{x}) = \boldsymbol{A}'\boldsymbol{D}(\omega_\psi([\boldsymbol{A}\boldsymbol{x}^{(n)}]_i - b_i))\boldsymbol{A}.$$

Thus, Huber's algorithm (14.3.6) is the following simple iteration:

e,ox,l1,huber

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \left[\boldsymbol{A}'\boldsymbol{D}(\omega_\psi([\boldsymbol{A}\boldsymbol{x}^{(n)}]_i - b_i))\boldsymbol{A}\right]^{-1} \boldsymbol{A}'\boldsymbol{D}(\omega_\psi([\boldsymbol{A}\boldsymbol{x}^{(n)}]_i - b_i))\,([\boldsymbol{A}\boldsymbol{x}^{(n)}]_i - b_i). \tag{14.5.25}$$

Again, because the curvature of ψ is bounded above by $1/\sqrt{\varepsilon}$, one could also apply Böhning and Lindsay's lower-bound algorithm (14.3.7). However, if ε is small, then the large value of the upper bound, $1/\sqrt{\varepsilon}$, may cause very slow convergence.

As $\varepsilon \to 0$, the weighting function $\omega_\psi(s)$ approaches $1/|s|$, which was used in [107] with a heuristic modification when the argument becomes "very small." It is unclear whether that approach would converge, whereas Huber's algorithm (14.5.25) is at least guaranteed to decrease Ψ every iteration.

Some authors [108] have used Huber's algorithm for ***total variation (TV)*** *regularization* without ***corner rounding****, in which case $\omega_\psi(t) = \frac{\operatorname{sgn}(t)}{t} = \frac{1}{|t|}$, arguing that the* ***singularity issue*** *at $t = 0$ "is not in fact an issue" if the image is initialized properly.*

An alternative to corner rounding is to use **Young's inequality** [wiki]: $|st| \leq \frac{1}{2}|s|^2 + \frac{1}{2}|t|^2 \implies |t| \leq \frac{1}{2}\phi(t;s) \triangleq |s| + \frac{1}{2|s|}|t|^2$ if $s \neq 0$ as discussed in [109]. However, the $|s|$ term in the denominator diverges as $s \to 0$, leading to numerical problems [109]. This approach should be avoided; instead use an explicit approximation like (14.5.24), or an alternative algorithm designed for nonsmooth regularizers.

s,ox,fista

14.6 Acceleration via momentum, e.g., FISTA (s,ox,fista)

One way to accelerate optimization transfer algorithms is to modify the update to include a **momentum term**, as described below.

Optimization transfer with momentum (*e.g.*, FISTA)

Initialize: $\boldsymbol{x}^{(0)}$, $\boldsymbol{z}^{(1)} = \boldsymbol{x}^{(0)}$, and choose momentum factors $\{\beta_n\}$

for $n = 1, 2, \ldots$

$$\boldsymbol{x}^{(n)} = \underset{\boldsymbol{x} \in \mathcal{X}}{\arg\min}\, \phi(\boldsymbol{x}; \boldsymbol{z}^{(n)})$$

$$\boldsymbol{z}^{(n+1)} = \boldsymbol{x}^{(n)} + \beta_n (\boldsymbol{x}^{(n)} - \boldsymbol{x}^{(n-1)}). \tag{14.6.1}$$

e,ox,momentum

The update (14.6.1) is called a **momentum term**. If we choose $\beta_n = 0$ then the method simplifies to a conventional optimization transfer algorithm.

One example of such an approach is the **fast iterative soft thresholding algorithm** (**FISTA**) [43], for which

e,ox,momentum,fista,bet_n

$$\beta_n = \frac{t_n - 1}{t_{n+1}}, \qquad t_1 = 1, \qquad t_{n+1} = \frac{1 + \sqrt{1 + 4t_n^2}}{2}, \tag{14.6.2}$$

where one can verify that the momentum factor $\{\beta_n\}$ increases monotonically towards 1 as iteration n increases.

x,os,fista,quad

Example 14.6.1 *For a quadratic surrogate of the form (14.3.5), when $\mathcal{X} = \mathbb{R}^{n_{\mathrm{p}}}$, the first step simplifies to*

$$\boldsymbol{x}^{(n)} = \boldsymbol{z}^{(n)} - \boldsymbol{J}_n^{-1} \nabla \Psi(\boldsymbol{z}^{(n)}),$$

leading to the recursion

$$\boldsymbol{x}^{(n+1)} = \left((1+\beta_n)\boldsymbol{x}^{(n)} - \beta_n \boldsymbol{x}^{(n-1)}\right) - \boldsymbol{J}_{n+1}^{-1} \nabla \Psi\left((1+\beta_n)\boldsymbol{x}^{(n)} - \beta_n \boldsymbol{x}^{(n-1)}\right).$$

In particular, for a quadratic cost function where $\nabla \Psi(\boldsymbol{x}) = \boldsymbol{H}\boldsymbol{x} - \boldsymbol{b}$, and where $\boldsymbol{P} = \boldsymbol{J}_n^{-1}$, we have the recursion

e,ox,momentum,quad

$$\begin{bmatrix} \boldsymbol{x}^{(n+1)} \\ \boldsymbol{x}^{(n)} \end{bmatrix} = \begin{bmatrix} (1+\beta_n)\boldsymbol{S} & -\beta_n \boldsymbol{S} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{(n)} \\ \boldsymbol{x}^{(n-1)} \end{bmatrix} - \begin{bmatrix} \boldsymbol{P}\boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix}, \tag{14.6.3}$$

where $\boldsymbol{S} \triangleq \boldsymbol{I} - \boldsymbol{P}\boldsymbol{H}$. One can verify that the fixed point of this iteration is $\hat{\boldsymbol{x}} = \boldsymbol{H}^{-1}\boldsymbol{b}$ if $\boldsymbol{P}$ and $\boldsymbol{H}$ are nonsingular. If the sequence $\{\beta_n\}$ approaches a limit β, then the asymptotic convergence rate is governed by the eigenvalues of the matrix

$$\begin{bmatrix} (1+\beta)\boldsymbol{S} & -\beta \boldsymbol{S} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix}$$

so noting that $\boldsymbol{u} = \lambda\, \boldsymbol{v}$ and simplifying yields

$$(\lambda(1+\beta) - \beta)\, \boldsymbol{S}\, \boldsymbol{v} = \lambda^2\, \boldsymbol{v}\,.$$

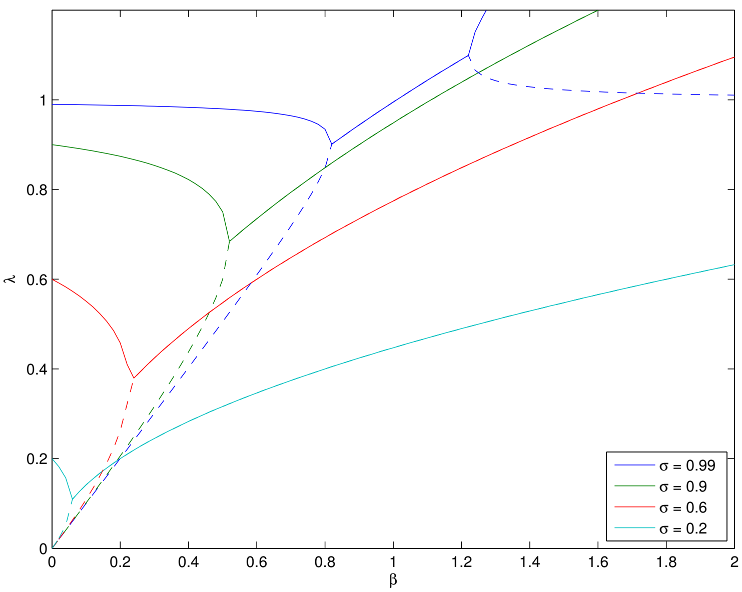
If σ is an eigenvalue of $\boldsymbol{S}$ with corresponding eigenvector $\boldsymbol{v}$, i.e., *$\boldsymbol{S}\,\boldsymbol{v} = \sigma\,\boldsymbol{v}$, then*

$$\lambda^2 - (1+\beta)\sigma\lambda + \beta\sigma = 0 \Longrightarrow \lambda = \frac{(1+\beta)\sigma \pm \sqrt{(1+\beta)^2\sigma^2 - 4\beta\sigma}}{2}.$$

If $\sigma \in [0,1)$, which occurs when $\boldsymbol{P}^{-1} \succeq \boldsymbol{H}$, as discussed in (14.1.10), then it appears that the ***spectral radius*** *that describes the* ***root convergence factor*** *of (14.6.1) is given by $\rho = \frac{(1+\beta)\sigma + \sqrt{|(1+\beta)^2\sigma^2 - 4\beta\sigma|}}{2}$ and it appears that in the* scalar *case, as a function of β, convergence is fastest when $0 = (1+\beta)^2\sigma^2 - 4\beta\sigma$,* i.e., *$\beta = 2/\sigma - 1 - \sqrt{(2/\sigma - 1)^2 - 1}$, for which $\rho = \frac{(1+\beta)\sigma}{2} = 1 - \sqrt{1-\sigma} \leq \sigma$. So (14.6.1) can converge faster than ordinary optimization transfer if we choose β appropriately.*

Fig. 14.6.1 shows λ for various σ and β values.

Fig. 14.6.2 shows the root convergence factor of (14.6.1) when β is chosen optimally as a function of σ for a 1D problem.



fig_ox_fista_rate1

Figure 14.6.1: Plots of λ for various σ and β.

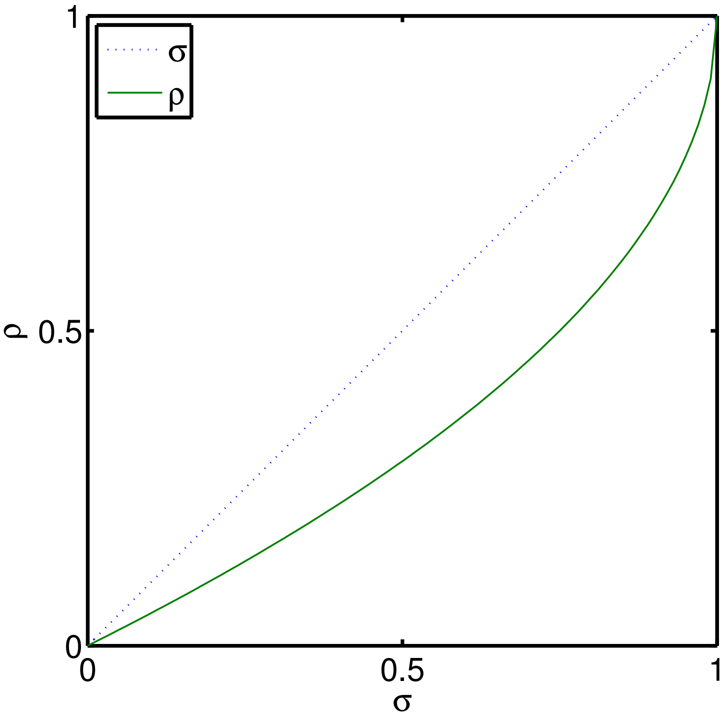


Figure 14.6.2: Plot of the root convergence factor ρ of the optimization transfer method accelerated via momentum versus σ, the root convergence factor for the optimization transfer algorithm for a 1D quadratic problem.

fig_ox_fista_rate1b

s,ox,amp

14.7 Alternating-minimization procedure (s,ox,amp)

Csiszár and Tusnády [144] described an intriguing framework for developing iterative algorithms, called the **alternating minimization** procedure. Let $\mathcal{P}$ and $\mathcal{Q}$ denote two convex sets, and let $\mathsf{D}(\boldsymbol{p} \,\|\, \boldsymbol{q})$ denote some measure of "distance" between $\boldsymbol{p} \in \mathcal{P}$ and $\boldsymbol{q} \in \mathcal{Q}$, *i.e.*, $D : \mathcal{P} \times \mathcal{Q} \to \mathbb{R}$. Then given an initial $\boldsymbol{q}^{(0)} \in \mathcal{Q}$, define the following iteration, which alternates between updating $\boldsymbol{p}$ and $\boldsymbol{q}$:

$$\boxed{\begin{aligned} \text{Step 1: } \boldsymbol{p}^{(n+1)} &= \arg\min_{\boldsymbol{p}\in\mathcal{P}} \mathsf{D}(\boldsymbol{p} \,\|\, \boldsymbol{q}^{(n)}) && (14.7.1) \\ \text{Step 2: } \boldsymbol{q}^{(n+1)} &= \arg\min_{\boldsymbol{q}\in\mathcal{Q}} \mathsf{D}\big(\boldsymbol{p}^{(n+1)} \,\|\, \boldsymbol{q}\big). && (14.7.2) \end{aligned}}$$

e,ox,amp,p
e,ox,amp

Conditions that ensure convergence of the sequences to the minimizer of $\mathsf{D}(\boldsymbol{p} \,\|\, \boldsymbol{q})$ over $\mathcal{P} \times \mathcal{Q}$ are discussed in [144].

To make such an algorithm useful for statistical image reconstruction, one must choose appropriate spaces $\mathcal{P}$ and $\mathcal{Q}$ and an appropriate function D such that the minimizer of D corresponds to the minimizer of the cost function Ψ of interest. Examples of this approach are given in §15.6.4, §18.13, and [145, 146].

In the context of image reconstruction problems, typical cost functions have the "**partially separable**" form (14.4.1). When minimizing such cost functions over a parameter space $\mathcal{X} \subset \mathbb{R}^{n_{\mathrm{p}}}$, reasonable choices for $\mathcal{P}$ and $\mathcal{Q}$ are as follows:

$$\begin{aligned} \mathcal{P} &\triangleq \left\{ \boldsymbol{p} = \{p_{ij}\} \in \mathbb{R}^{N_{\mathrm{i}} \times n_{\mathrm{p}}} : \sum_{j=1}^{n_{\mathrm{p}}} p_{ij} = t_i,\ i = 1,\ldots,N_{\mathrm{i}} \right\} \\ \mathcal{Q} &\triangleq \left\{ \boldsymbol{q} = \boldsymbol{q}(\boldsymbol{x}) \in \mathbb{R}^{N_{\mathrm{i}} \times n_{\mathrm{p}}} : q_{ij} = b_{ij} x_j,\ i = 1,\ldots,N_{\mathrm{i}}, \quad \text{for some } \boldsymbol{x} \in \mathcal{X} \right\}, \end{aligned}$$

where $t_i \triangleq \arg\min_t \psi_i(t)$. Natural choices for D have the form

e,ox,amp,D,recon

$$\mathsf{D}(\boldsymbol{p} \,\|\, \boldsymbol{q}) = \sum_{i=1}^{N_{\mathrm{i}}} \sum_{j=1}^{n_{\mathrm{p}}} \psi_{ij}(p_{ij}, q_{ij}), \qquad (14.7.3)$$

for some functions $\psi_{ij}(\cdot)$ chosen by the algorithm designer subject to the requirement that the minimizer of D should correspond to the minimizer of the cost function Ψ:

e,ox,amp,necc

$$\arg\min_{\boldsymbol{x}} \Psi(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}:\boldsymbol{q}(\boldsymbol{x})\in\mathcal{Q}} \min_{\boldsymbol{p}\in\mathcal{P}} \mathsf{D}(\boldsymbol{p} \,\|\, \boldsymbol{q}(\boldsymbol{x})). \qquad (14.7.4)$$

For the above choices of $\mathcal{P}$, $\mathcal{Q}$, and D, the alternating minimization procedure reduces to the following steps.

Given a previous guess $\boldsymbol{x}^{(n)}$, define $q_{ij}^{(n)} = b_{ij} x_j^{(n)}$. Then for Step 1 of the procedure, we find

$$\boldsymbol{p}^{(n+1)} = \arg\min_{\boldsymbol{p}\in\mathcal{P}} \mathsf{D}(\boldsymbol{p} \,\|\, \boldsymbol{q}^{(n)})$$

where

$$\mathsf{D}(\boldsymbol{p} \,\|\, \boldsymbol{q}^{(n)}) = \sum_{i=1}^{N_{\mathrm{i}}} \sum_{j=1}^{n_{\mathrm{p}}} \psi_{ij}\big(p_{ij}, q_{ij}^{(n)}\big).$$

Because $\mathcal{P}$ has a product form and D is **additively separable**, Step 1 separates into the following N_{i} minimization problems:

e,ox,ampl,s1

$$\left\{ p_{ij}^{(n+1)} \right\}_{j=1}^{n_{\mathrm{p}}} = \arg\min_{\left\{ \{p_{ij}\}_{j=1}^{n_{\mathrm{p}}} \,:\, \sum_{j=1}^{n_{\mathrm{p}}} p_{ij} = t_i \right\}} \sum_{j=1}^{n_{\mathrm{p}}} \psi_{ij}\big(p_{ij}, q_{ij}^{(n)}\big), \qquad i = 1,\ldots,N_{\mathrm{i}}. \qquad (14.7.5)$$

One can solve each of these minimization problems using the method of Lagrange multipliers.

For Step 2, we find

e,ox,ampl,s2

$$\boldsymbol{x}^{(n+1)} = \arg\min_{\boldsymbol{x}} \mathsf{D}\big(\boldsymbol{p}^{(n+1)} \,\|\, \boldsymbol{q}(\boldsymbol{x})\big) = \arg\min_{\boldsymbol{x}} \sum_{i=1}^{N_{\mathrm{i}}} \sum_{j=1}^{n_{\mathrm{p}}} \psi_{ij}\Big(p_{ij}^{(n+1)}, q_{ij}(\boldsymbol{x})\Big). \qquad (14.7.6)$$

Due to the form of $\mathcal{Q}$ and the additive separable form of D, Step 2 simplifies to

$$x_j^{(n+1)} = \arg\min_{x_j} \sum_{i=1}^{N_\mathrm{i}} \psi_{ij}\left(p_{ij}^{(n+1)}, b_{ij} x_j\right), \qquad j = 1, \ldots, n_\mathrm{p}.$$

Clearly the utility of the alternating minimization procedure hinges on whether one can find choices for ψ_{ij} for which the minimizer of D corresponds to the minimizer of Ψ, *and* for which Steps 1 and 2 are easily solved. We will see that such choices are readily available for the weighted least-squares cost function (§15.6.4) and for the emission tomography log-likelihood, (§18.13). Finding a suitable choice for the transmission tomography log-likelihood is an *open problem*.

14.7.1 Relationship to optimization transfer

It has perhaps not been widely appreciated that, at least in the context of image reconstruction, alternating minimization methods are simply a special case of **optimization transfer**. This section describes the relationship.

We can write the two steps of an alternating minimization method as follows.

e,ox,amp,new

$$\boxed{\begin{aligned} &\text{Step 1: Find } \boldsymbol{p}^{(\star)}(\boldsymbol{x}^{(n)}), \text{ where } \boldsymbol{p}^{(\star)}(\boldsymbol{x}) \triangleq \arg\min_{\boldsymbol{p} \in \mathcal{P}} \mathsf{D}(\boldsymbol{p} \,\|\, \boldsymbol{q}(\boldsymbol{x})) \\ &\text{Step 2: } \boldsymbol{x}^{(n+1)} = \arg\min_{\boldsymbol{x} \,:\, \boldsymbol{q}(\boldsymbol{x}) \in \mathcal{Q}} \mathsf{D}(\boldsymbol{p}^{(\star)}(\boldsymbol{x}^{(n)}) \,\|\, \boldsymbol{q}(\boldsymbol{x})) \qquad (14.7.7) \end{aligned}}$$

Clearly, by this construction, the following equalities are satisfied

$$\begin{aligned} \mathsf{D}(\boldsymbol{p}^{(\star)}(\boldsymbol{x}) \,\|\, \boldsymbol{q}(\boldsymbol{x})) &\le \mathsf{D}(\boldsymbol{p} \,\|\, \boldsymbol{q}(\boldsymbol{x})), \qquad \forall \boldsymbol{x} \in \boldsymbol{X}, \qquad \forall \boldsymbol{p} \in \mathcal{P} \\ \mathsf{D}\big(\boldsymbol{p}^{(\star)}(\boldsymbol{x}^{(n)}) \,\|\, \boldsymbol{q}(\boldsymbol{x}^{(n+1)})\big) &\le \mathsf{D}(\boldsymbol{p}^{(\star)}(\boldsymbol{x}^{(n)}) \,\|\, \boldsymbol{q}), \qquad \forall \boldsymbol{q} \in \mathcal{Q}, \end{aligned}$$

so in particular

$$\mathsf{D}\big(\boldsymbol{p}^{(\star)}(\boldsymbol{x}^{(n+1)}) \,\|\, \boldsymbol{q}(\boldsymbol{x}^{(n+1)})\big) \le \mathsf{D}\big(\boldsymbol{p}^{(\star)}(\boldsymbol{x}^{(n)}) \,\|\, \boldsymbol{q}(\boldsymbol{x}^{(n+1)})\big) \le \mathsf{D}(\boldsymbol{p}^{(\star)}(\boldsymbol{x}^{(n)}) \,\|\, \boldsymbol{q}(\boldsymbol{x}^{(n)}))\,.$$

In all examples of alternating minimization methods for image reconstruction that I have seen, the function D is chosen such that

e,ox,amp,suff

$$\Psi(\boldsymbol{x}) = \mathsf{D}(\boldsymbol{p}^{(\star)}(\boldsymbol{x}) \,\|\, \boldsymbol{q}(\boldsymbol{x})) = \min_{\boldsymbol{p} \in \mathcal{P}} \mathsf{D}(\boldsymbol{p} \,\|\, \boldsymbol{q}(\boldsymbol{x})), \qquad (14.7.8)$$

which is sufficient to ensure (14.7.4). Combining the two previous equations, we see immediately that $\Psi\big(\boldsymbol{x}^{(n+1)}\big) \le \Psi(\boldsymbol{x}^{(n)})$, *i.e.*, alternating minimization methods of the above form are monotonic descent methods.

In terms of an optimization transfer viewpoint, the equivalent surrogate function is

$$\phi^{(n)}(\boldsymbol{x}) = \mathsf{D}(\boldsymbol{p}^{(\star)}(\boldsymbol{x}^{(n)}) \,\|\, \boldsymbol{q}(\boldsymbol{x}))$$

for which

$$\begin{aligned} \phi^{(n)}(\boldsymbol{x}^{(n)}) &= \mathsf{D}(\boldsymbol{p}^{(\star)}(\boldsymbol{x}^{(n)}) \,\|\, \boldsymbol{q}(\boldsymbol{x}^{(n)})) = \Psi(\boldsymbol{x}^{(n)}) \\ \phi^{(n)}(\boldsymbol{x}) &= \mathsf{D}(\boldsymbol{p}^{(\star)}(\boldsymbol{x}^{(n)}) \,\|\, \boldsymbol{q}(\boldsymbol{x})) \ge \mathsf{D}(\boldsymbol{p}^{(\star)}(\boldsymbol{x}) \,\|\, \boldsymbol{q}(\boldsymbol{x})) = \Psi(\boldsymbol{x})\,. \end{aligned}$$

This surrogate function satisfies the required conditions (14.1.4) when (14.7.8) holds.

Some authors apparently find the "geometric" perspective offered by Csiszár and Tusnády's paper [144] to be insightful. Although the optimization transfer approach may lack this geometric structure, only basic algebra is required to derive an algorithm using optimization transfer.

s,ox,amp,inc

14.7.2 Incremental methods

Neal and Hinton [147] developed an **incremental** extension of EM algorithm, and the incremental concept is also applicable to alternating minimization when D has the form (14.7.3). Instead of updating *every* element of $\boldsymbol{p}$ in (14.7.1), it may be easier to update only a *subset* of the p_{ij} terms, and hold the remaining p_{ij} terms to their previous values. Then one updates $\boldsymbol{q}$ using (14.7.2) using all of the current p_{ij} estimates, some of which were more recently updated than others.

With such an incremental update of $\boldsymbol{p}$, the algorithm will monotonically decrease D, but no longer will the algorithm monotonically decrease the original cost function. However, convergence can still be established under certain assumptions [148–151]. See §14.11 for a general formulation in terms of optimization transfer.

s,ox,dc

14.8 Difference of convex functions procedure (s,ox,dc)

Suppose we have a cost function of the form

$$\Psi(\boldsymbol{x}) = f(\boldsymbol{x}) - g(\boldsymbol{x})$$

where both f and g are convex (and closed and proper). Then by (28.9.15)

$$g(\boldsymbol{x}) = \inf_{\boldsymbol{z}} \left(g^\star(\boldsymbol{z}) - \langle \boldsymbol{z},\, \boldsymbol{x} \rangle \right),$$

minimizing $\Psi(\boldsymbol{x})$ is equivalent to

$$\min_{\boldsymbol{x},\boldsymbol{z}} f(\boldsymbol{x}) - \langle \boldsymbol{z},\, \boldsymbol{x} \rangle + g^\star(\boldsymbol{z}).$$

This approach is called the **difference of convex functions** (**DC**) method [152–156], also known as the concave-convex procedure (**CCCP**) [157].

s,ox,em

14.9 Expectation-maximization (EM) methods (s,ox,em)

The **expectation-maximization** (**EM**) family of algorithms provides methods for computing the maximum-likelihood (**ML**) estimate $\hat{\boldsymbol{\theta}}$ of an unknown parameter vector $\boldsymbol{\theta}^{\text{true}}$ residing in subset Θ of $\mathbb{R}^{n_{\text{p}}}$, from a measured realization $\boldsymbol{y} \in \mathbb{R}^{n_{\text{d}}}$ of an observable random vector $\boldsymbol{Y}$ having the probability mass function[7] $\mathsf{p}(\boldsymbol{y}; \boldsymbol{\theta}^{\text{true}})$, where $\boldsymbol{\theta}^{\text{true}}$ is the true value of the unknown parameter. Computationally, for ML estimation we must find the maximizer of a log-likelihood merit function[8]:

e,ox,em,problem

$$\hat{\boldsymbol{\theta}} \triangleq \arg\max_{\boldsymbol{\theta} \in \Theta} \mathsf{L}(\boldsymbol{\theta}), \tag{14.9.1}$$

where the log-likelihood is given by

e,ox,em,L

$$\mathsf{L}(\boldsymbol{\theta}) \triangleq \log \mathsf{p}(\boldsymbol{y}; \boldsymbol{\theta}). \tag{14.9.2}$$

For a wide variety of important statistical estimation problems, direct maximization of $\mathsf{L}(\boldsymbol{\theta})$ is intractable, so one must resort to iterative methods.

As noted by Lange et al. [3], "Many specific EM algorithms can even be derived more easily by invoking optimization transfer rather than missing data." Indeed, this conclusion applies to *all* of the EM algorithms considered in this *book*. In short, my view is that the EM approach is an obsolete method for deriving optimization algorithms for image reconstruction algorithms[9]. Our presentation is thus primarily for historical completeness.

As illustrated in Example 14.9.3 below, in many estimation problems one can imagine a hypothetical experiment that would yield a more extensive measurement vector that, if observed, would greatly simplify the parameter estimation problem. This measurement vector is called the "complete data" in EM terminology, and is the basic ingredient in deriving a specific EM algorithm[10]. The EM method is a special case of the **optimization transfer** approach described in §14.1, in which we replace the maximization of $\mathsf{L}(\boldsymbol{\theta})$ with the maximization of another functional $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(n)})$ that is related to the condition log-likelihood of the complete-data given the observed data. Of course if we choose the complete-data vector unwisely, then the problem of maximizing $Q(\cdot; \boldsymbol{\theta}^{(n)})$ may end up being as difficult as the original maximization problem, requiring inconvenient numerical methods like line searches. But with a wise choice for the complete-data vector, often one can maximize $Q(\cdot; \boldsymbol{\theta}^{(n)})$ *analytically*, a tremendous simplification. Even if one cannot maximize Q analytically, one can often choose complete-data vectors such that it is easier to evaluate $Q(\cdot; \boldsymbol{\theta}^{(n)}) - Q(\boldsymbol{\theta}^{(n)}; \boldsymbol{\theta}^{(n)})$ than $\mathsf{L}(\cdot) - \mathsf{L}(\boldsymbol{\theta}^{(n)})$, so line searches for maximizing $Q(\cdot; \boldsymbol{\theta}^{(n)})$ would be cheaper than line searches for maximizing $\mathsf{L}(\cdot)$. The statistical formulation of the functionals Q inherently ensures that increases in Q yield increases in $\mathsf{L}(\boldsymbol{\theta})$.

To construct the surrogate functions Q, we must identify an admissible complete-data vector defined in the following sense.

[7] For simplicity, we restrict our description to discrete-valued random variables. The method is easily extended to general distributions [160].

[8] For consistency with the majority of the EM literature, we use $\boldsymbol{\theta}$ to denote the unknown parameter vector and $\mathsf{L}(\boldsymbol{\theta})$ to denote the merit function to be maximized, rather than considering minimization of a cost function $\Psi(\boldsymbol{x})$ as described in previous sections.

[9] However, it must be said that distinguished statisticians have been unable to prove that optimization transfer provides a strictly broader family of methods than EM. (See the discussion by Meng in [3].)

[10] The discussants of the seminal EM paper [6] complained about the term "algorithm" because the EM approach in fact yields a whole family of algorithms depending on the designer's choice of a complete data vector.

d,complete-data

Definition 14.9.1 *A random vector $\boldsymbol{Z}$ with probability mass function $\mathsf{p}(\boldsymbol{z};\boldsymbol{\theta})$ is an* admissible complete-data vector *for $\mathsf{p}(\boldsymbol{y};\boldsymbol{\theta})$ if the joint distribution of $\boldsymbol{Z}$ and $\boldsymbol{Y}$ satisfies*

e,ox,em,admit

$$\mathsf{p}(\boldsymbol{y},\boldsymbol{z};\boldsymbol{\theta}) = \mathsf{p}(\boldsymbol{y} \,|\, \boldsymbol{z})\,\mathsf{p}(\boldsymbol{z};\boldsymbol{\theta}), \tag{14.9.3}$$

i.e., *the conditional distribution $\mathsf{p}(\boldsymbol{y} \,|\, \boldsymbol{z})$ must be independent of $\boldsymbol{\theta}$.*

The definition used for the classical EM algorithm [6] is contained as a special case of Definition 14.9.1 by requiring $\boldsymbol{Y}$ to be a deterministic function of $\boldsymbol{Z}$ [160]. In non-imaging statistical problems, often $\boldsymbol{Z} = (\boldsymbol{Y}, \boldsymbol{Y}_{\text{missing}})$ where $\boldsymbol{Y}_{\text{missing}}$ denotes unobserved or "missing" data; in imaging problems this decomposition is rarely natural.

14.9.1 Algorithm

An essential ingredient of any EM algorithm is the following conditional expectation[11] of the log-likelihood of $\boldsymbol{Z}$:

e,ox,em,Q

e,ox,em,estep

$$\begin{aligned} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) &\triangleq \mathsf{E}[\log \mathsf{p}(\boldsymbol{Z};\boldsymbol{\theta}) \,|\, \boldsymbol{Y} = \boldsymbol{y}; \boldsymbol{\theta}^{(n)}] && (14.9.4)\\ &= \sum_{\boldsymbol{z}} [\log \mathsf{p}(\boldsymbol{z};\boldsymbol{\theta})]\, \mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y} = \boldsymbol{y}; \boldsymbol{\theta}^{(n)})\,. && (14.9.5) \end{aligned}$$

Let $\boldsymbol{\theta}^{(0)} \in \Theta$ be an initial parameter estimate. A generic EM algorithm produces a sequence of estimates $\{\boldsymbol{\theta}^{(n)}\}_{n=0}^{\infty}$ via the following recursion.

EM "Algorithm"

```
for $n = 0, 1, \ldots$ {
    1. Choose an admissible complete-data vector $\boldsymbol{Z}$
    2. E-step: "Compute" $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)})$ using (14.9.4)
    3. M-step:
        $\boldsymbol{\theta}^{(n+1)} = \arg\max_{\boldsymbol{\theta}\in\Theta} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)})$    (14.9.6)
}.
```

e,ox,em,mstep

If one chooses the complete-data vectors appropriately, then typically one can combine the E-step and M-step via an analytical maximization into a recursion of the form $\boldsymbol{\theta}^{(n+1)} = \mathcal{M}(\boldsymbol{\theta}^{(n)})$. The examples in later sections illustrate this important aspect of the EM method.

Note that if one chooses $\boldsymbol{Z} = \boldsymbol{Y}$, then one sees from (14.9.4) that $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) = \mathsf{L}(\boldsymbol{\theta})$, and the maximization problem (14.9.6) reverts to the original problem (14.9.1).

Rather than requiring a strict maximization in (14.9.6), one could settle simply for local maxima [160], or for mere increases in Q, in analogy with **generalized EM** (**GEM**) algorithms [6], as described in §14.9.5. These generalizations provide the opportunity to further refine the trade-off between convergence rate and computation per-iteration.

s,ox,em,mono

14.9.2 Monotonicity (s,ox,em,mono)

Many standard optimization methods require line searches with evaluation of $\mathsf{L}\big(\boldsymbol{\theta}^{(n+1)}\big) - \mathsf{L}(\boldsymbol{\theta}^{(n)})$ to ensure monotonicity. An appealing property of the EM method is that it provides intrinsically monotonic iterative algorithms, as established in following theorem.

t,ox,em,mono

Theorem 14.9.2 *Let $\{\boldsymbol{\theta}^{(n)}\}$ denote the sequence of estimates generated by an EM algorithm (14.9.6) with surrogate function Q satisfying (14.9.4) Then 1) $\mathsf{L}(\boldsymbol{\theta}^{(n)})$ is monotonically nondecreasing,* i.e., $\mathsf{L}\big(\boldsymbol{\theta}^{(n+1)}\big) \geq \mathsf{L}(\boldsymbol{\theta}^{(n)})$*, 2) if $\hat{\boldsymbol{\theta}}$ maximizes $\mathsf{L}(\cdot)$, then $\hat{\boldsymbol{\theta}}$ is a fixed point of the EM algorithm, and 3)*

$$\mathsf{L}\big(\boldsymbol{\theta}^{(n+1)}\big) - \mathsf{L}(\boldsymbol{\theta}^{(n)}) \geq Q(\boldsymbol{\theta}^{(n+1)};\boldsymbol{\theta}^{(n)}) - Q(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)}).$$

[11] The summation $\sum_{\boldsymbol{z}}$ is over the range of values of $\boldsymbol{Z}$, and $0 \log 0$ is interpreted as zero.

Proof (see *e.g.*, [16, p. 118] or [6]):
Applying Bayes' rule to (14.9.4) using (14.9.3) we see that

e,ox,em,qhl

$$\begin{aligned} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) &= \sum_{\boldsymbol{z}} \mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta}^{(n)}) \log \mathsf{p}(\boldsymbol{z};\boldsymbol{\theta}) \\ &= \sum_{\boldsymbol{z}} \mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta}^{(n)}) \log \frac{\mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta})\, \mathsf{p}(\boldsymbol{y};\boldsymbol{\theta})}{\mathsf{p}(\boldsymbol{y} \,|\, \boldsymbol{z})} \\ &= \mathsf{L}(\boldsymbol{\theta}) + H(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) - W(\boldsymbol{\theta}^{(n)}), \end{aligned} \tag{14.9.7}$$

where L is defined in (14.9.2),

e,ox,em,hs

$$H(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) \triangleq \mathsf{E}[\log \mathsf{p}(\boldsymbol{Z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta}) \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta}^{(n)}], \tag{14.9.8}$$

and

$$W(\boldsymbol{\theta}^{(n)}) \triangleq \sum_{\boldsymbol{z}} \mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta}^{(n)}) \log \mathsf{p}(\boldsymbol{y} \,|\, \boldsymbol{z}) \,.$$

Because W is independent of $\boldsymbol{\theta}$, it does not affect the maximization (14.9.6). Using **Jensen's inequality** (28.9.8):

$$\begin{aligned} H(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) - H(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)}) &= \sum_{\boldsymbol{z}} \mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta}^{(n)}) \log\left(\frac{\mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta})}{\mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta}^{(n)})} \right) \\ &\leq \log\left(\sum_{\boldsymbol{z}} \mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta}^{(n)}) \frac{\mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta})}{\mathsf{p}(\boldsymbol{z} \,|\, \boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta}^{(n)})} \right) = 0, \end{aligned}$$

so

e,ox,em,H,jensen

$$H(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) \leq H(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)}), \qquad \forall \boldsymbol{\theta},\ \forall \boldsymbol{\theta}^{(n)}. \tag{14.9.9}$$

From (14.9.7) it follows that

$$\begin{aligned} \mathsf{L}(\boldsymbol{\theta}) - \mathsf{L}(\boldsymbol{\theta}^{(n)}) &= [Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) - H(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)})] - [Q(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)}) - H(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)})] \\ &= [Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) - Q(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)})] + [H(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)}) - H(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)})] \\ &\geq Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) - Q(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)}), \end{aligned}$$

by (14.9.9). Thus, if we find *any* value $\boldsymbol{\theta}^{(n+1)}$ satisfying $Q(\boldsymbol{\theta}^{(n+1)};\boldsymbol{\theta}^{(n)}) \geq Q(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)})$, then we are assured that $\mathsf{L}\big(\boldsymbol{\theta}^{(n+1)}\big) \geq \mathsf{L}(\boldsymbol{\theta}^{(n)})$. The results then follow from the definition of the EM algorithm. □

This proof can be generalized for mixed discrete/continuous random variables. See [16, p. 119] for a rigorous treatment.

It follows from (14.9.7) and (14.9.9) that

e,ox,em,dQ=dL

$$\nabla^{10} Q(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)}) \triangleq \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(n)}} = \nabla \mathsf{L}(\boldsymbol{\theta}^{(n)}) \,. \tag{14.9.10}$$

Thus, the EM surrogate function Q satisfies the important condition (14.1.5) of a surrogate function for optimization transfer.

s,ox,em,rate

14.9.3 Asymptotic convergence rate (s,ox,em,rate)

Because EM algorithms are special cases of **optimization transfer** methods, analysis of the **asymptotic convergence rate** of EM algorithms follows directly from §14.1.5 (which was in fact inspired by similar analysis in [6]).

For a statistical interpretation of (14.1.9), we can apply (14.9.7) to note that

$$\boldsymbol{I} - \left[-\nabla^{20} Q\right]^{-1} \left[-\nabla^2 \mathsf{L}\right] = \boldsymbol{I} - \left[-\nabla^2 L - \nabla^{20} H\right]^{-1} \left[-\nabla^2 \mathsf{L}\right],$$

The Hessian matrix $-\nabla^2 \mathsf{L}$ is called the **observed Fisher information matrix** of the incomplete-data, whereas the matrix $-\nabla^2 H$ quantifies how much more informative is the hypothesized complete data relative to the measured incomplete data. From the above expression, in light of the convergence rate analysis in §14.1.5, we see that if one chooses an overly informative complete data, then the resulting EM algorithm will converge *very* slowly. This relationship is examined quantitatively in [34, 161] (*cf.* §15.5.3).

s,ox,em,ex

14.9.4 Example application with missing data (s,ox,em,ex)

x,ox,em

Example 14.9.3 *This example illustrates an EM approach in the context of a traditional missing-data problem. Suppose we attempt to collect three independent and identically distributed samples of a random vector* $\boldsymbol{Z} \in \mathbb{R}^2$ *having a normal distribution with unknown mean* $\boldsymbol{\theta} = [\theta_1 \; \theta_2]'$ *but* known[12] *covariance* $\boldsymbol{K} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}'$ *with* $\rho = 1/2$*. Due to instrumentation failure*[13]*, suppose that rather than recording* $[\boldsymbol{Z}_1 \; \boldsymbol{Z}_2 \; \boldsymbol{Z}_3] = \begin{bmatrix} Z_{11} & Z_{21} & Z_{31} \\ Z_{12} & Z_{22} & Z_{32} \end{bmatrix}$ *we only record (observe):*

$$[\boldsymbol{Y}_1 \; \boldsymbol{Y}_2 \; \boldsymbol{Y}_3] = \begin{bmatrix} y_{11} & y_{21} & \\ y_{12} & & y_{32} \end{bmatrix} = \begin{bmatrix} 60 & 0 & \\ 30 & & 0 \end{bmatrix}.$$

From the observed values $(\boldsymbol{Y}_1, \boldsymbol{Y}_2, \boldsymbol{Y}_3)$ *we wish to determine the ML estimate of* $\boldsymbol{\theta}$*. The naive estimate of* $\boldsymbol{\theta}$ *would just take the average of the available measurements in each row:*

e,ox,em,naive

$$\hat{\boldsymbol{\theta}}_{\text{naive}} = \begin{bmatrix} \frac{Y_{11}+Y_{21}}{2} \\ \frac{Y_{12}+Y_{32}}{2} \end{bmatrix} = \begin{bmatrix} \frac{60+0}{2} \\ \frac{30+0}{2} \end{bmatrix} = \begin{bmatrix} 30 \\ 15 \end{bmatrix}. \tag{14.9.11}$$

Another option would be to discard all the vectors with missing elements, which in this case would leave just $\boldsymbol{Y}_1$*, with corresponding estimate* $\hat{\boldsymbol{\theta}}_{\text{discard}} = (60, 30)$*. However, these are* not *the same as the ML estimate for this problem. Fig. 14.9.1 displays contours of the log-likelihood:*

$$\mathsf{L}(\boldsymbol{\theta}) \equiv \log \prod_{i=1}^{3} \frac{1}{\sqrt{|\det\{\boldsymbol{S}_i' \boldsymbol{K} \boldsymbol{S}_i\}|}} \exp\left(-\frac{1}{2} (\boldsymbol{y}_i - \boldsymbol{S}_i \boldsymbol{\theta})' [\boldsymbol{S}_i' \boldsymbol{K} \boldsymbol{S}_i]^{-1} (\boldsymbol{y}_i - \boldsymbol{S}_i \boldsymbol{\theta}) \right),$$

where

$$\boldsymbol{S}_1 \triangleq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{S}_2 \triangleq \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \boldsymbol{S}_3 \triangleq \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad \boldsymbol{y}_1 = \begin{bmatrix} 60 \\ 30 \end{bmatrix}, \quad \boldsymbol{y}_2 = 0, \quad \boldsymbol{y}_3 = 0.$$

From Fig. 14.9.1 it is clear that the ML estimate is $\hat{\boldsymbol{\theta}} = \begin{bmatrix} 28 & 8 \end{bmatrix}'$*. However, for larger problems we must apply numerical procedures to find* $\hat{\boldsymbol{\theta}}$*, and EM algorithms can be convenient choices.*

For this problem the natural complete-data vector is $\boldsymbol{Z} = \begin{bmatrix} Z_{11} & Z_{21} & Z_{31} & Z_{12} & Z_{22} & Z_{32} \end{bmatrix}'$*, where* $\boldsymbol{Z}_i$ *variables are independent. In this example, the corresponding conditional pdf mentioned in (14.9.3) is*

$$\mathsf{p}(\boldsymbol{y} \,|\, \boldsymbol{z}) = \delta(y_{11} - z_{11}) \; \delta(y_{21} - z_{21}) \; \delta(y_{12} - z_{12}) \; \delta(y_{32} - z_{32}),$$

where $\delta(\cdot)$ *denotes the Dirac unit impulse. To derive the corresponding EM algorithm, we first compute the complete-data log-likelihood, and then the Q function (14.9.4). The log pdf of* $\boldsymbol{Z}$ *is given by*

$$\begin{aligned} \log \mathsf{p}(\boldsymbol{Z}; \boldsymbol{\theta}) &= \log \prod_{i=1}^{3} \mathsf{p}(\boldsymbol{Z}_i; \boldsymbol{\theta}) \\ &\equiv -\sum_{i=1}^{3} \frac{1}{2} (\boldsymbol{Z}_i - \boldsymbol{\theta})' \boldsymbol{K}^{-1} (\boldsymbol{Z}_i - \boldsymbol{\theta}) \\ &\equiv -\frac{3}{2} \boldsymbol{\theta}' \boldsymbol{K}^{-1} \boldsymbol{\theta} + \boldsymbol{\theta}' \boldsymbol{K}^{-1} \sum_{i=1}^{3} \boldsymbol{Z}_i. \end{aligned}$$

Thus the Q function is:

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(n)}) = \mathsf{E}[\log \mathsf{p}(\boldsymbol{Z}; \boldsymbol{\theta}) \,|\, \boldsymbol{Y} = \boldsymbol{y}; \boldsymbol{\theta}^{(n)}] \equiv \boldsymbol{\theta}' \boldsymbol{K}^{-1} \sum_{i=1}^{3} \mathsf{E}[\boldsymbol{Z}_i \,|\, \boldsymbol{Y}_i = \boldsymbol{y}_i; \boldsymbol{\theta}^{(n)}] - \frac{3}{2} \boldsymbol{\theta}' \boldsymbol{K}^{-1} \boldsymbol{\theta}.$$

To find the maximizer over $\boldsymbol{\theta}$ *we zero the gradient of Q:*

$$\mathbf{0} = \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(n)}) = \boldsymbol{K}^{-1} \sum_{i=1}^{3} \mathsf{E}[\boldsymbol{Z}_i \,|\, \boldsymbol{Y}_i = \boldsymbol{y}_i; \boldsymbol{\theta}^{(n)}] - 3 \boldsymbol{K}^{-1} \boldsymbol{\theta}.$$

[12] The assumption of known covariance may be artificial, but greatly simplifies this didactic example.

[13] We assume that the cause of this failure is statistically independent from the measurement noise.

Solving for $\boldsymbol{\theta}$ yields the iteration:

$$\boldsymbol{\theta}^{(n+1)} = \frac{1}{3}\sum_{i=1}^{3} \mathsf{E}[\boldsymbol{Z}_i \,|\, \boldsymbol{Y}_i = \boldsymbol{y}_i; \boldsymbol{\theta}^{(n)}] \,. \tag{14.9.12}$$

e,ox,em,ex,1

This iteration has the following natural interpretation: using the most recent parameter estimate $\boldsymbol{\theta}^{(n)}$, we compute the conditional expectation of the complete-data vectors $\boldsymbol{Z}_i$, and then we compute the average *of those estimates of $\boldsymbol{Z}_i$ (which would have been the ML estimator for $\boldsymbol{\theta}$ if we had observed all of the $\boldsymbol{Z}_i$ values.)*

To finalize the iteration, we must evaluate the conditional expectations in (14.9.12). Because $\boldsymbol{Z}_i$ is a normal random vector and $\boldsymbol{Y}_i = \boldsymbol{S}_i\boldsymbol{Z}_i$, the conditional expectation of $\boldsymbol{Z}_i$ given $\boldsymbol{Y}_i$ has the following form [162, p. 302] [163, p. 325]:

$$\mathsf{E}[\boldsymbol{Z}_i \,|\, \boldsymbol{Y}_i; \boldsymbol{\theta}^{(n)}] = \mathsf{E}[\boldsymbol{Z}_i; \boldsymbol{\theta}^{(n)}] + \boldsymbol{K}_{\boldsymbol{Z}_i,\boldsymbol{Y}_i} \boldsymbol{K}_{\boldsymbol{Y}_i}^{-1} (\boldsymbol{Y}_i - \mathsf{E}[\boldsymbol{Y}_i; \boldsymbol{\theta}^{(n)}]),$$

where $\boldsymbol{K}_{\boldsymbol{Z}_i,\boldsymbol{Y}_i} = \mathsf{Cov}\{\boldsymbol{Z}_i, \boldsymbol{Y}_i\} = \mathsf{Cov}\{\boldsymbol{Z}_i, \boldsymbol{S}_i\boldsymbol{Z}_i\} = \mathsf{Cov}\{\boldsymbol{Z}_i\}\, \boldsymbol{S}_i'$. Thus in this example:

$$\mathsf{E}[\boldsymbol{Z}_i \,|\, \boldsymbol{Y}_i; \boldsymbol{\theta}^{(n)}] = \boldsymbol{\theta}^{(n)} + \boldsymbol{K}\boldsymbol{S}_i'[\boldsymbol{S}_i\boldsymbol{K}\boldsymbol{S}_i']^{-1}(\boldsymbol{Y}_i - \boldsymbol{S}_i\boldsymbol{\theta}^{(n)}).$$

Combining the preceding expressions yields the following final form for the EM algorithm for this example:

$$\begin{aligned}
\boldsymbol{\theta}^{(n+1)} &= \boldsymbol{\theta}^{(n)} + \boldsymbol{K}\frac{1}{3}\sum_{i=1}^{3} \boldsymbol{S}_i'[\boldsymbol{S}_i\boldsymbol{K}\boldsymbol{S}_i']^{-1}(\boldsymbol{y}_i - \boldsymbol{S}_i\boldsymbol{\theta}^{(n)}) \\
&= \boldsymbol{\theta}^{(n)} + \frac{1}{3}\left[\left(\begin{bmatrix} 60 \\ 30 \end{bmatrix} - \boldsymbol{\theta}^{(n)}\right) + \left(\begin{bmatrix} 1 \\ 1/2 \end{bmatrix}(0 - \theta_1^{(n)})\right) + \left(\begin{bmatrix} 1/2 \\ 1 \end{bmatrix}(0 - \theta_2^{(n)})\right)\right].
\end{aligned} \tag{14.9.13}$$

e,ox,em,ex

Fig. 14.9.1 illustrates the iterates produced by this EM algorithm when initialized with the naive estimator (14.9.11). In this simple example, the EM algorithm approaches the ML estimate fairly rapidly. Unfortunately the convergence is much slower in the tomographic applications discussed in subsequent chapters.

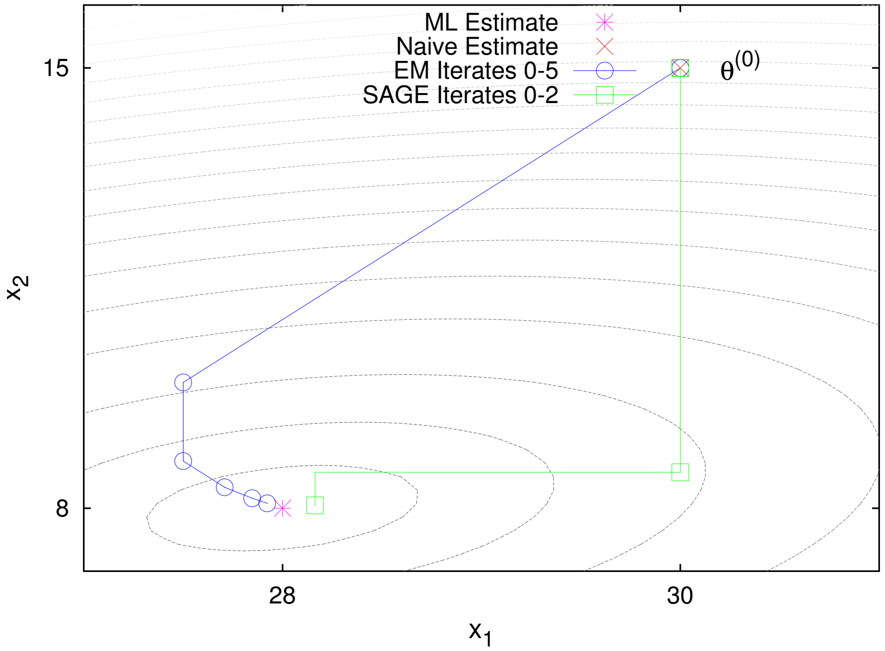


Figure 14.9.1: Contours of the log-likelihood $\mathsf{L}(\boldsymbol{\theta})$ in Example 14.9.3, along with a trajectory of EM iterates $\{\boldsymbol{\theta}^{(n)}\}$ (see (14.9.13)) and the SAGE iterates (see (14.12.7)) converging from the initial naive estimate $\boldsymbol{\theta}^{(0)}$ from (14.9.11) towards the ML estimate $\hat{\boldsymbol{\theta}}$.

fig_em1

s,ox,em,gem

14.9.5 Generalized EM (GEM) methods (s,ox,em,gem)

The final lines of proof of monotonicity in Theorem 14.9.2 show that it is unnecessary to perform an *exact* maximization in the M-step (14.9.6) of an EM algorithm. As long as we find a value $\boldsymbol{\theta}^{(n+1)}$ that *increases* the surrogate function Q, we are assured of a monotone increase in the log-likelihood L. Such algorithms were termed **generalized EM** (**GEM**) methods in [6], and have been applied in image reconstruction *e.g.*, [164].

14.9.6 Relationship to optimization transfer methods

The EM approach is a special case of an optimization transfer method. Let $\Psi(\boldsymbol{x}) = -\mathsf{L}(\boldsymbol{x})$ denote the cost function that is minimized for ML estimation. Define the following surrogate function

e,ox,em,surr

$$\phi^{(n)}(\boldsymbol{x}) = \Psi(\boldsymbol{x}) + \mathsf{D}(\mathsf{p}(\boldsymbol{z}|\boldsymbol{y};\boldsymbol{\theta}^{(n)}) \parallel \mathsf{p}(\boldsymbol{z}|\boldsymbol{y};\boldsymbol{\theta})), \tag{14.9.14}$$

where $\mathsf{D}(p \parallel q)$ is the **Kullback-Leibler divergence** for probability distributions, defined by

$$\mathsf{D}(p \parallel q) = \sum_{\boldsymbol{z}} p(\boldsymbol{z}) \log \frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}.$$

Because $\mathsf{D}(p \parallel q) \geq \mathsf{D}(p \parallel p) = 0$, (see §17.4.3), it follows that (14.9.14) satisfies the surrogate conditions (14.1.4).

By arguments similar to those in Theorem 14.9.2, the above surrogate $\phi^{(n)}$ is related to the EM surrogate $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)})$ as follows:

e,ox,em,sur,vs,Q

$$\phi^{(n)}(\boldsymbol{x}) = -Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(n)}) - W(\boldsymbol{\theta}^{(n)}) + H(\boldsymbol{\theta}^{(n)};\boldsymbol{\theta}^{(n)}), \tag{14.9.15}$$

i.e., they are the same except for a sign and an additive constant that is independent of $\boldsymbol{\theta}$.

s,ox,em,more

14.9.7 Generalizations (s,ox,em,more)

EM algorithms have been applied to a tremendous variety of applications, *e.g.*, [50, 99, 165–173]. Several papers have analyzed convergence more carefully, correcting an error in the proof of the original paper by Dempster et al. [144, 174, 175]. Many extensions of the basic EM algorithm have been proposed in the literature, *e.g.*, [161, 176–180], including consideration of problems with constraints [177, 181, 182]. For example, the cascade EM algorithm [183] is a generalization based on a hierarchy of nested complete-data spaces. Recursive EM algorithms for time-varying applications have also been proposed [184, 185], as have versions with asynchronous updates [186] for image segmentation with **Markov random fields** (**MRF**) s.

Some papers focus on computing the (approximate) covariance matrix of $\hat{\boldsymbol{\theta}}$ [187–191].

Because the EM algorithm converges so slowly, even at sublinear rates [192] in some cases, a particularly large segment of the literature has been devoted to methods for accelerating the EM algorithm [14, 36, 42, 193–196] Often the goal of such methods is to provide EM-like monotone increases in $\mathsf{L}(\boldsymbol{\theta})$ in the initial iterations far from the maximizer, but faster Newton-like convergence rates as the solution is approached. Because EM methods are special cases of optimization transfer methods many acceleration methods for OT also apply to EM, including momentum-type approaches [140].

One acceleration method that was originally motivated by imaging problems is the family of space-alternating generalized EM (**SAGE**) algorithms described in §14.12.

s,ox,em,inc

14.9.8 Incremental EM algorithms (s,ox,em,inc)

An interesting generalization is the **incremental EM algorithm** proposed by Neal and Hinton [147]. The method has been shown to converge [148], albeit non-monotonically in $\mathsf{L}(\cdot)$ [150]. It has been applied to PET (see §17.7.2) [151, 197]. A closely related method has been applied to mixture models [198, 199].

The incremental EM approach is based on defining

$$\bar{F}(\tilde{\mathsf{p}}, \boldsymbol{\theta}) \triangleq -\mathsf{L}(\boldsymbol{\theta}) + \mathsf{D}(\tilde{\mathsf{p}} \parallel \mathsf{p}(\boldsymbol{z}|\boldsymbol{y};\boldsymbol{\theta}))$$

for any probability distribution $\tilde{\mathsf{p}}$, and using (14.9.14) to write the EM algorithm as follows:

$$\begin{array}{lrcl} \text{E-step:} & \tilde{\mathsf{p}}^{(n)} & = & \arg\min_{\tilde{\mathsf{p}}} \bar{F}(\tilde{\mathsf{p}}, \boldsymbol{\theta}^{(n)}) \\ \text{M-step:} & \boldsymbol{\theta}^{(n+1)} & = & \arg\min_{\boldsymbol{\theta}} \bar{F}\left(\tilde{\mathsf{p}}^{(n)}, \boldsymbol{\theta}\right) \end{array}.$$

(The minimization in the E-step is restricted to valid probability distributions.)

In many applications including imaging problems, the conditional distribution of the complete data $\boldsymbol{z}$ often factors as follows:

$$\mathsf{p}(\boldsymbol{z}|\boldsymbol{y};\boldsymbol{\theta}^{(n)}) = \prod_{m=1}^{M} \mathsf{p}_m(\boldsymbol{z}_m|\boldsymbol{y}_m;\boldsymbol{\theta}^{(n)}),$$

where $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M)$ is some decomposition of the complete data. Rather than updating the (estimated) distribution of the entire vector $\boldsymbol{z}$ each iteration, one could update just the (conditional) distribution of $\boldsymbol{z}_m$ for the E-step, thereby possibly reducing computation. Let

$$\tilde{\mathsf{p}} = \prod_m \tilde{\mathsf{p}}_m$$

and, assuming that

$$\mathsf{p}(\boldsymbol{y}; \boldsymbol{\theta}) = \prod_m \mathsf{p}(\boldsymbol{y}_m; \boldsymbol{\theta})$$

define

$$\bar{F}(\tilde{\mathsf{p}}, \boldsymbol{\theta}) = \sum_m \bar{F}_m(\tilde{\mathsf{p}}_m, \boldsymbol{\theta})$$

where

$$\bar{F}_m(\tilde{\mathsf{p}}_m, \boldsymbol{\theta}) = -\log \mathsf{p}(\boldsymbol{y}_m; \boldsymbol{\theta}) + \mathsf{D}(\tilde{\mathsf{p}}_m \,\|\, \mathsf{p}_m(\boldsymbol{z}_m | \boldsymbol{y}_m; \boldsymbol{\theta})) \,. \qquad (14.9.16)$$

e,ox,em,inc,Finc_m

This suggests the following iterative algorithm.

Incremental EM method

for $m = 1, \ldots, M$

$$\tilde{\mathsf{p}}_l^{(n+m/M)} = \begin{cases} \arg\min_{\tilde{\mathsf{p}}_l} \bar{F}_l\left(\tilde{\mathsf{p}}_l, \boldsymbol{\theta}^{(n+(m-1)/M)}\right), & l = m \\ \tilde{\mathsf{p}}_l^{(n+(m-1)/M)}, & l \neq m \end{cases}$$

$$\boldsymbol{\theta}^{(n+m/M)} = \arg\min_{\boldsymbol{\theta}} \bar{F}\left(\tilde{\mathsf{p}}^{(n+m/M)}, \boldsymbol{\theta}\right), \qquad (14.9.17)$$

e,ox,em,inc

where $\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n+M/M)}$.

This algorithm monotonically decreases the function $\bar{F}(\tilde{\mathsf{p}}, \boldsymbol{\theta})$ each iteration, but is not guaranteed to monotonically increase the likelihood. §17.7.2 describes its application to emission tomography.

Notice that the entire parameter vector $\boldsymbol{\theta}$ is updated simultaneously during the M-step of each iteration of the incremental EM approach. In contrast, the SAGE algorithm of §14.12 only updates a portion of $\boldsymbol{\theta}$ at a time.

14.10 Incremental gradient or ordered subset methods (s,ox,os)

s,ox,os

Essentially all of the cost functions of interest in image reconstruction are sums of functions:

$$\Psi(\boldsymbol{x}) = \sum_{m=1}^{M} \Psi_m(\boldsymbol{x}), \qquad (14.10.1)$$

e,ox,Kx,part,sep

a form that is sometimes called **partially separable**, *e.g.*, [200] or **additive-separable** [201]. Most algorithms involve the gradient of Ψ, which is also partially separable:

$$\nabla\Psi(\boldsymbol{x}) = \sum_{m=1}^{M} \nabla\Psi_m(\boldsymbol{x}) \,.$$

Experience in both tomographic image reconstruction and other optimization problems has shown that in the early iterations of many algorithms, one can replace $\nabla\Psi(\boldsymbol{x}^{(n)})$ with $\nabla\Psi_m(\boldsymbol{x}^{(n)})$ yet still compute an $\boldsymbol{x}^{(n+1)}$ that is "improved" over $\boldsymbol{x}^{(n)}$ but with less computation. In the optimization literature, such approaches are called **incremental gradient** methods [147, 202–208], and these methods date back to the 1960s in that field *e.g.*, [201]. In tomography, these methods are called **ordered subsets** methods, because only a subset of the projection views are used each "subiteration," and the ordering of the views can be important [209–218]. The term **block iterative** has also been used [219, 220]. These methods are also related closely to **stochastic gradient descent** methods [wiki] [221–244].

In mathematical terms, suppose we have an iterative algorithm of the form

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - D_0(\boldsymbol{x}^{(n)}) \, \nabla\Psi(\boldsymbol{x}^{(n)}),$$

for some matrix $D_0(\cdot)$, that is typically diagonal. We can consider instead an algorithm with subiterations such as the following[14].

[14] In some cases it is desirable to change the matrix D_0 to a somewhat different matrix $D(\cdot)$.

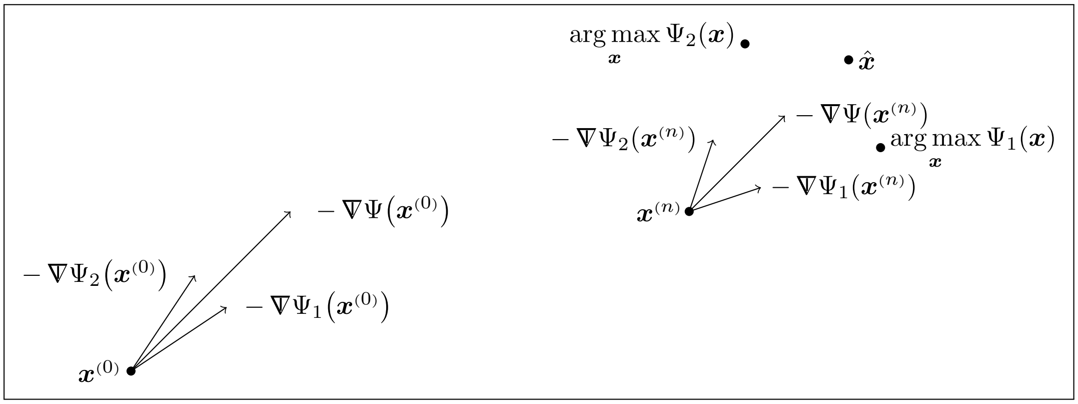

$\arg\max_{\boldsymbol{x}} \Psi_2(\boldsymbol{x})$
$\hat{\boldsymbol{x}}$
$-\nabla\Psi(\boldsymbol{x}^{(n)})$
$-\nabla\Psi_2(\boldsymbol{x}^{(n)})$
$\arg\max_{\boldsymbol{x}} \Psi_1(\boldsymbol{x})$
$-\nabla\Psi_1(\boldsymbol{x}^{(n)})$
$\boldsymbol{x}^{(n)}$
$-\nabla\Psi\left(\boldsymbol{x}^{(0)}\right)$
$-\nabla\Psi_2\left(\boldsymbol{x}^{(0)}\right)$
$-\nabla\Psi_1\left(\boldsymbol{x}^{(0)}\right)$
$\boldsymbol{x}^{(0)}$

Figure 14.10.1: Geometrical view of ordered subsets principle.

fig,os,geom

Incremental gradient or ordered-subset method

```
for $m = 1, \ldots, M$
```

$$\boldsymbol{x}^{(n+m/M)} = \boldsymbol{x}^{(n+(m-1)/M)} - M D(\boldsymbol{x}^{(n+(m-1)/M)})\, \nabla\Psi_m\left(\boldsymbol{x}^{(n+(m-1)/M)}\right), \qquad (14.10.2)$$

e,ox,os

where $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n+M/M)}$. A complete iteration involves cycling through all the subsets indexed by m so that all data is used. Several examples of such algorithms are given throughout the book. They are important in part due to their commercial availability for both PET and SPECT systems.

Convergence results for *relaxed* ordered-subsets algorithms are given in §14.6. Without relaxation however, such algorithms usually do not converge (except when $M = 1$) [245]. Nevertheless, in the early iterations, using ordered subsets usually accelerates "convergence" by a factor of roughly M, due in part to "inflating" the step size by "M" in (14.10.2).

Fig. 14.10.1 illustrates geometrically why the OS principle can work. For $\boldsymbol{x}^{(n)}$ far from the minimizer $\hat{\boldsymbol{x}}$, if we choose the subsets to be "balanced" in some appropriate sense, then the gradient vectors $\nabla\Psi_m(\boldsymbol{x}^{(n)})$ for $m = 1, \ldots, M$ all point roughly in the same direction as the overall gradient $\nabla\Psi(\boldsymbol{x}^{(n)})$, *i.e.*,

,ox,os,grad,approx

$$\nabla\Psi(\boldsymbol{x}^{(n)}) \approx M\, \nabla\Psi_m(\boldsymbol{x}^{(n)}), \quad m = 1, \ldots, M. \qquad (14.10.3)$$

Because $\nabla\Psi(\boldsymbol{x}^{(n)})$ need not point exactly towards $\hat{\boldsymbol{x}}$ anyway, it seems reasonable to use $M\, \nabla\Psi_m(\boldsymbol{x}^{(n)})$ in place of $\nabla\Psi(\boldsymbol{x}^{(n)})$ for the update because less computation is needed to evaluate $\nabla\Psi_m$. Unfortunately, as the sequence $\{\boldsymbol{x}^{(n)}\}$ approach the minimizer $\hat{\boldsymbol{x}}$, the subset gradient vectors point in increasingly different directions, so an unrelaxed OS algorithm will oscillate and possibly reach a limit cycle. Indeed, in the absence of constraints, in the limit the subset gradient vectors must satisfy

$$\mathbf{0} = \nabla\Psi(\hat{\boldsymbol{x}}) = \sum_{m=1}^{M} \nabla\Psi_m(\hat{\boldsymbol{x}}),$$

so the vectors are necessarily pointing in opposite directions!

Fig. 14.10.1 also illustrates the desirability of choosing the subsets that satisfy the following **subset gradient balance** condition:

$$\nabla\Psi_1(\boldsymbol{x}) \approx \nabla\Psi_2(\boldsymbol{x}) \approx \cdots \approx \nabla\Psi_M(\boldsymbol{x})$$

for $\boldsymbol{x}$ far from $\hat{\boldsymbol{x}}$.

Fig. 14.10.2 and Fig. 14.10.3 show subset choices that do and do not satisfy the subset gradient balance condition. The cost function is a **weighted least-squares** form $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$, where the rows of $\boldsymbol{A}$ correspond to a discretization of the 2D Radon transform for tomography. For Fig. 14.10.2, we decomposed the cost function Ψ into even ($m = 1$) and odd ($m = 2$) projection views. For a uniform initial image $\boldsymbol{x}^{(0)}$ (that is far from the minimizer $\hat{\boldsymbol{x}}$), Fig. 14.10.2 shows that the exact gradient $\nabla\Psi\left(\boldsymbol{x}^{(0)}\right)$ is closely matched by the subset gradients $2\,\nabla\Psi_1\left(\boldsymbol{x}^{(0)}\right)$ and $2\,\nabla\Psi_2\left(\boldsymbol{x}^{(0)}\right)$.

In contrast, in Fig. 14.10.3 we chose the first subset to correspond to the views with angles in $[0°, 90°)$ and the second subset to corresponded to $[90°, 180°)$. In this case, even for the same initial image $\boldsymbol{x}^{(0)}$ that is far from $\hat{\boldsymbol{x}}$, the subset gradients poorly match the exact cost function gradient. So this choice of subsets does not satisfy the subset gradient balance condition.

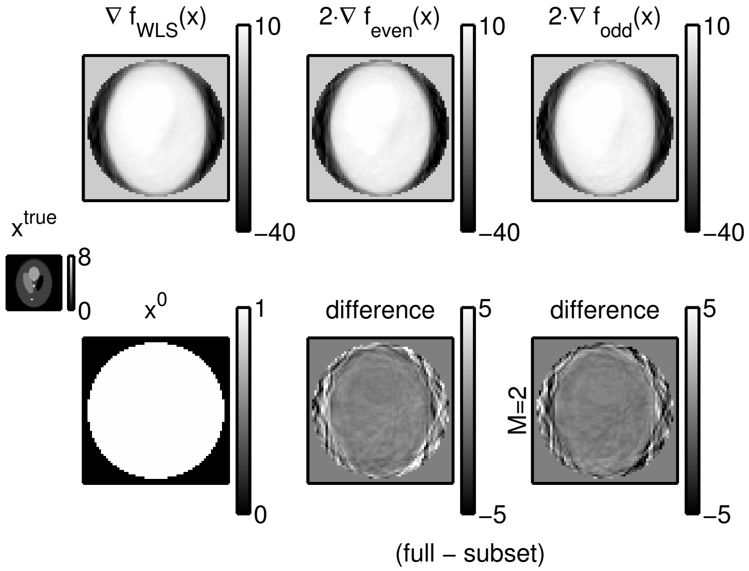


Figure 14.10.2: Here the initial image $\boldsymbol{x}^{(0)}$ is far from the solution so the incremental gradients, *i.e.*, the gradients computed from just the even or odd angles, agree well with the full gradient.

fig_os_gradient

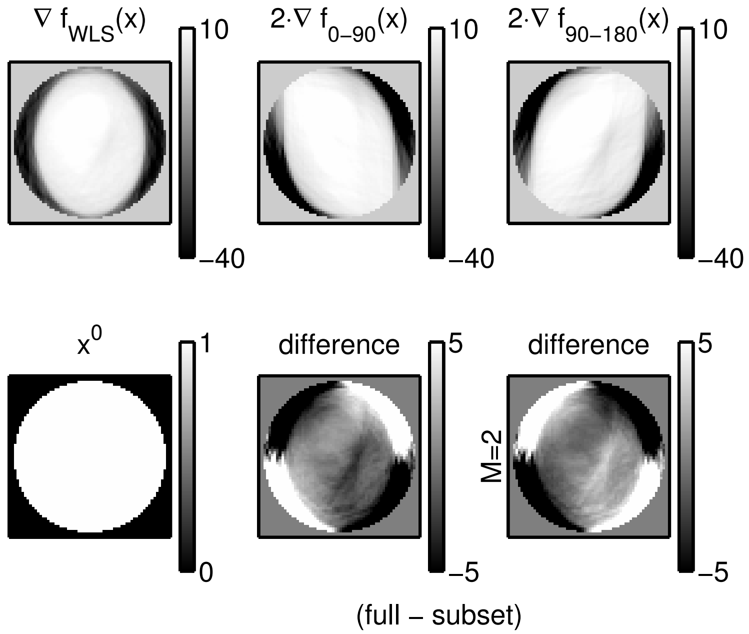


Figure 14.10.3: Here the first subset was angles 0-90°, and the second subset was angles 90-180°, roughly speaking. Now the incremental gradients do not agree as well with the full gradient. (Of course the *sum* of the two incremental gradients would still equal the full gradient.) This imbalance is expected to slow "convergence."

fig_os_imbalance

Instead of using a "gradient descent" type of iteration like (14.10.2), one could use the the gradient approximation (14.10.3) within a more sophisticated optimization method such as the **precondition conjugate gradients** (**PCG**) method [246].

For *regularized* cost functions, it is reasonable to try to initialize the iterations with as good of a starting point $\boldsymbol{x}^{(0)}$ as possible, meaning $\boldsymbol{x}^{(0)} \approx \hat{\boldsymbol{x}}$. In these cases, it is less clear whether OS approaches are effective.

When more than two subsets are used, the ordering of the subsets may affect the initial convergence rate. To make the fastest initial progress, it seem desirable to choose subsets whose corresponding gradient vectors are "as orthogonal as possible." In tomography, "balance" usually means using equally-spaced subsets of the projection angles, where these subsets are used in an ordering that skips around between angles that are spaced far apart.

One approach uses an access order that is equivalent to the FFT bit reversal ordering [247]. For example, if we have 48 projection angles numbered 0 through 47, then for $M = 8$ we could use the following subsets

$$\begin{array}{lll@{\qquad}lll}
\mathcal{S}_1 &=& \{0, 8, 16, 24, 32, 40\}, & \mathcal{S}_5 &=& \{1, \ldots\}, \\
\mathcal{S}_2 &=& \{4, 12, \ldots, 44\}, & \mathcal{S}_6 &=& \{5, \ldots\}, \\
\mathcal{S}_3 &=& \{2, 10, \ldots\}, & \mathcal{S}_7 &=& \{3, \ldots\}, \\
\mathcal{S}_4 &=& \{6, \ldots\}, & \mathcal{S}_8 &=& \{7, 15, \ldots, 47\}.
\end{array}$$

x,ox,os,sps

Example 14.10.1 *Consider the general* ***separable quadratic surrogates*** *(**SQS**) algorithm given by (14.5.13). Let* $\{\mathcal{S}_m\}$ *be a partition of the set* $\{1, \ldots, N_{\mathrm{i}}\}$. *Then the natural ordered-subsets (**OS-SQS**) version is*

e,ox,os,sps

$$x_j^{(n+m/M)} = \left[x_j^{(n+(m-1)/M)} - \frac{M}{d_j^{(n)}} \sum_{i \in \mathcal{S}_m} b_{ij}\, \dot{\psi}_i\left([\boldsymbol{B}\boldsymbol{x}^{(n+(m-1)/M)}]_i \right) \right]_+, \qquad j = 1, \ldots, n_{\mathrm{p}}, \tag{14.10.4}$$

where $d_j^{(n)}$ *was defined in (14.5.10). Because OS algorithms are not guaranteed to monotonically decrease the cost function, it is particularly reasonable to use the precomputed maximum curvatures (14.4.12), or the following "**Newton curvatures**"*

e,ox,recon,curv,newton

$$\check{c}_i = \ddot{\psi}_i\left(\arg\min_t \psi_i(t) \right). \tag{14.10.5}$$

This saves computation because we can precompute the d_j *values prior to iterating.*

If the subsets have different sizes $|\mathcal{S}_m|$, *then we would replace the multiplier* M *in (14.10.4) with* $N_{\mathrm{i}} / |\mathcal{S}_m|$.

Incremental gradient methods without locks (to reduce synchronization overhead in parallel systems) have also been proposed [233]. One can form convergent OS methods by increasing the number of subsets as iterations proceed [224, 248, 249]. See also [237].

s,ox,inc

14.11 Incremental optimization transfer (IOT) (s,ox,inc)

This section generalizes the **incremental EM** approach of §14.9.8 by allowing broad family of surrogate functions within the optimization transfer framework, rather than being limited to EM-type surrogates.

As in §14.10, assume that the cost function Ψ has the partially separable form (14.10.1). Assume furthermore that, for each sub-cost function $\Psi_m(\boldsymbol{x})$, we can find a surrogate function ϕ_m that satisfies the usual majorization conditions:

e,ox,inc,major

$$\begin{aligned}
\phi_m(\boldsymbol{x}; \boldsymbol{x}) &= \Psi_m(\boldsymbol{x}), & \forall \boldsymbol{x} \\
\phi_m(\boldsymbol{x}; \bar{\boldsymbol{x}}) &\geq \Psi_m(\boldsymbol{x}), & \forall \boldsymbol{x}, \bar{\boldsymbol{x}}.
\end{aligned} \tag{14.11.1}$$

Inspired by (14.9.16), define the following "divergence" functions:

e,ox,inc,Dam

$$\mathsf{D}_m(\boldsymbol{x} \,\|\, \bar{\boldsymbol{x}}) \triangleq \phi_m(\boldsymbol{x}; \bar{\boldsymbol{x}}) - \Psi_m(\boldsymbol{x}), \quad m = 1, \ldots, M. \tag{14.11.2}$$

By construction, $\mathsf{D}_m(\boldsymbol{x} \,\|\, \bar{\boldsymbol{x}}) \geq 0$ and $\mathsf{D}_m(\boldsymbol{x} \,\|\, \boldsymbol{x}) = 0$, and in particular:

$$\arg\min_{\bar{\boldsymbol{x}}} \mathsf{D}(\boldsymbol{x} \,\|\, \bar{\boldsymbol{x}}) = \boldsymbol{x}.$$

Now define the following augmented cost function:

e,ox,inc,F

$$\bar{F}(\boldsymbol{x}; \bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_M) = \Psi(\boldsymbol{x}) + \sum_{m=1}^M \mathsf{D}_m(\boldsymbol{x} \,\|\, \bar{\boldsymbol{x}}_m) = \sum_{m=1}^M \phi_m(\boldsymbol{x}; \bar{\boldsymbol{x}}_m). \tag{14.11.3}$$

Because of the properties of $\mathsf{D}_m(\cdot \,\|\, \cdot)$, the estimator $\hat{\boldsymbol{x}}$ can be expressed as follows:

e,ox,inc,xh,F

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \Psi(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}} \min_{\bar{\boldsymbol{x}}_1,\ldots,\bar{\boldsymbol{x}}_M} \bar{F}(\boldsymbol{x}; \bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_M)\,. \tag{14.11.4}$$

So the problem of minimizing Ψ is equivalent to minimizing $\bar{F}$. Given an initial set of estimates $\bar{\boldsymbol{x}}_m^{(0)}$ for $m = 1, \ldots, M$, we minimize $\bar{F}$ by applying a cyclic coordinate descent approach in which we alternate between updating $\boldsymbol{x}$ and one of the $\bar{\boldsymbol{x}}_m$ vectors, as follows.

Generic incremental optimization transfer (**IOT**) method

```
Initialize $\bar{\boldsymbol{x}}_1^{(0)}, \ldots, \bar{\boldsymbol{x}}_M^{(0)}$
for $n = 0, 1, \ldots$
    for $m = 1, \ldots, M$
        $\boldsymbol{x}^{\text{new}} := \arg\min_{\boldsymbol{x}} \bar{F}\left(\boldsymbol{x}; \bar{\boldsymbol{x}}_1^{(n+1)}, \ldots, \bar{\boldsymbol{x}}_{m-1}^{(n+1)}, \bar{\boldsymbol{x}}_m^{(n)}, \bar{\boldsymbol{x}}_{m+1}^{(n)}, \ldots, \bar{\boldsymbol{x}}_M^{(n)}\right)$   (14.11.5)
        $\bar{\boldsymbol{x}}_m^{(n+1)} = \arg\min_{\bar{\boldsymbol{x}}_m} \bar{F}\left(\boldsymbol{x}^{\text{new}}; \bar{\boldsymbol{x}}_1^{(n+1)}, \ldots, \bar{\boldsymbol{x}}_{m-1}^{(n+1)}, \bar{\boldsymbol{x}}_m, \bar{\boldsymbol{x}}_{m+1}^{(n)}, \ldots, \bar{\boldsymbol{x}}_M^{(n)}\right) = \boldsymbol{x}^{\text{new}}$.   (14.11.6)
```

e,ox,inc,xnew

e,ox,inc,xbm,new

The implementation details depend on the structure of the surrogates, but regardless the method will require storing at least M vectors of length n_{p}. This is the primary drawback of incremental methods when M and n_{p} are large.

The operation (14.11.6) is trivial; all of the "work" occurs in (14.11.5). This is an "incremental" form of optimization transfer because each subiteration updates only one of the $\bar{\boldsymbol{x}}_m$ vectors in (14.11.6), and thus only one of the surrogates ϕ_m. If one were to update all of the $\bar{\boldsymbol{x}}_m$ vectors in (14.11.6), then the algorithm would revert to an ordinary optimization transfer method, and, in many applications, the update (14.11.5) would require about M times more work when all of the $\bar{\boldsymbol{x}}_m$ values have been updated than when only one of the $\bar{\boldsymbol{x}}_m$ values have changed.

Each update will decrease $\bar{F}$ monotonically. And by (14.11.4), if the iterates converge to a minimizer of $\bar{F}$, then that estimate also minimizes Ψ. However, Ψ need not decrease monotonically with this algorithm.

An alternative way to write the update is as follows:

$$\boldsymbol{x}^{(n+1+(m-1)/M)} = \arg\min_{\boldsymbol{x}} \bar{F}\left(\boldsymbol{x}; \boldsymbol{x}^{(n+1+0/M)}, \ldots, \boldsymbol{x}^{(n+1+(m-2)/M)}, \boldsymbol{x}^{(n+(m-1)/M)}, \ldots, \boldsymbol{x}^{(n+(M-1)/M)}\right).$$

14.11.1 Quadratic surrogates

The **IOT** approach is particularly simple if the surrogates ϕ_m are quadratic, *i.e.*, if

$$\phi_m(\boldsymbol{x}; \bar{\boldsymbol{x}}) = \Psi_m(\bar{\boldsymbol{x}}) + (\boldsymbol{x} - \bar{\boldsymbol{x}})' \, \nabla \Psi_m(\bar{\boldsymbol{x}}) + \frac{1}{2} (\boldsymbol{x} - \bar{\boldsymbol{x}})' \, \breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}) (\boldsymbol{x} - \bar{\boldsymbol{x}}),$$

where each $\breve{\boldsymbol{C}}_m$ is a $n_{\mathrm{p}} \times n_{\mathrm{p}}$ Hermitian positive-semidefinite matrix that one must choose to ensure (14.11.1). Furthermore, for simplicity consider the case of unconstrained estimation. Given $\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_M$, the minimizer over $\boldsymbol{x}$ in (14.11.5) above satisfies

$$\boldsymbol{0} = \sum_{m=1}^{M} \left[\nabla \Psi_m(\bar{\boldsymbol{x}}_m) + \breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}_m)(\boldsymbol{x}^{\text{new}} - \bar{\boldsymbol{x}}_m) \right].$$

In other words,

$$\left[\sum_{m=1}^{M} \breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}_m) \right] \boldsymbol{x}^{\text{new}} = \sum_{m=1}^{M} \left[\breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}_m) \, \bar{\boldsymbol{x}}_m - \nabla \Psi_m(\bar{\boldsymbol{x}}_m) \right].$$

This relationship leads to the following general form for (unconstrained) **incremental quadratic surrogate algorithms.**

```
Incremental quadratic surrogate algorithm (unconstrained) - efficient form
Initialize $\{\bar{\boldsymbol{x}}_m : m = 1, \ldots, M\}$
 $\boldsymbol{C}_m := \breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}_m), \quad m = 1, \ldots, M$
 $\boldsymbol{v}_m := \boldsymbol{C}_m \bar{\boldsymbol{x}}_m - \nabla \Psi_m(\bar{\boldsymbol{x}}_m), \quad m = 1, \ldots, M$
 $\hat{\boldsymbol{C}} := \sum_{m=1}^M \boldsymbol{C}_m$
 $\hat{\boldsymbol{v}} := \sum_{m=1}^M \boldsymbol{v}_m$
for $n = 1, 2, \ldots$
    for $m = 1, \ldots, M$
        $\hat{\boldsymbol{x}} := \hat{\boldsymbol{C}}^{-1} \hat{\boldsymbol{v}}$        (14.11.8)
        $\hat{\boldsymbol{C}} := \hat{\boldsymbol{C}} - \boldsymbol{C}_m$
        $\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}} - \boldsymbol{v}_m$
        $\boldsymbol{C}_m := \breve{\boldsymbol{C}}_m(\hat{\boldsymbol{x}})$        (14.11.9)
        $\boldsymbol{v}_m := \boldsymbol{C}_m \hat{\boldsymbol{x}} - \nabla \Psi_m(\hat{\boldsymbol{x}})$        (14.11.10)
        $\hat{\boldsymbol{C}} := \hat{\boldsymbol{C}} + \boldsymbol{C}_m$
        $\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}} + \boldsymbol{v}_m .$
    end
end
```

e,ox,inc,xnew,quad,eff

Figure 14.11.1: Efficient “state vector” implementation of a generic incremental optimization transfer (**IOT**) algorithm based on quadratic surrogate functions with curvatures $\breve{\boldsymbol{C}}_m$.

fig,ox,inc,qs,eff

```
Incremental quadratic surrogate algorithm (unconstrained)
Initialize $\{\bar{\boldsymbol{x}}_m : m = 1, \ldots, M\}$
for $n = 1, 2, \ldots$
    for $m = 1, \ldots, M$
        $\bar{\boldsymbol{x}}_m := \left[ \sum_{l=1}^M \breve{\boldsymbol{C}}_l(\bar{\boldsymbol{x}}_l) \right]^{-1} \sum_{l=1}^M \left[ \breve{\boldsymbol{C}}_l(\bar{\boldsymbol{x}}_l) \, \bar{\boldsymbol{x}}_l - \nabla \Psi_l(\bar{\boldsymbol{x}}_l) \right] .$        (14.11.7)
    end
end
```

e,ox,inc,xnew,quad

If each $\breve{\boldsymbol{C}}_m$ is diagonal, this form requires storing $3M + 2$ vectors of length n_{p}.

One can avoid performing the summations at every subiteration by maintaining those sums as a **state vector** and state matrix that are updated incrementally, as shown in Fig. 14.11.1. If each $\breve{\boldsymbol{C}}_m$ is diagonal, then this implementation requires storing $2M + 3$ vectors of length n_{p}. If the curvature matrices $\breve{\boldsymbol{C}}_m$ are precomputed (*i.e.*, are independent of $\bar{\boldsymbol{x}}_m$), then further efficiencies are realized by not updating $\hat{\boldsymbol{C}}$. If $\hat{\boldsymbol{C}}$ is diagonal, then the main “work” in this algorithm is in (14.11.9) and (14.11.10).

If the cost function Ψ is strictly convex, then at least in the case of quadratic surrogates the augmented function $\bar{F}$ will be convex with respect to $\boldsymbol{x}$, so the minimizer in (14.11.5) exists. The above algorithm will decrease $\bar{F}$ monotonically, and one can establish general conditions that ensure convergence of all of the $\bar{\boldsymbol{x}}_m$ vectors to $\hat{\boldsymbol{x}}$ [250]. Furthermore, if each $\breve{\boldsymbol{C}}_m$ is diagonal, then one can readily include **box constraints**.

To streamline the implementation, it may be helpful in some cases to find a single matrix $\boldsymbol{D}$, often diagonal, that dominates each $\breve{\boldsymbol{C}}_m$, *i.e.*, for which $\boldsymbol{D} \succeq \breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}})$ for all m and all $\bar{\boldsymbol{x}}$. When such a matrix $\boldsymbol{D}$ is found and used in place of each $\breve{\boldsymbol{C}}_m$, the inner update simplifies to:

$$\bar{\boldsymbol{x}}_m := \frac{1}{M} \sum_{l=1}^M \left[\bar{\boldsymbol{x}}_l - \boldsymbol{D}^{-1} \nabla \Psi_l(\bar{\boldsymbol{x}}_l) \right]$$

e,ox,inc,alg,D

$$= \frac{1}{M}\sum_{l=1}^{M} \bar{\boldsymbol{x}}_l - \frac{1}{M}\boldsymbol{D}^{-1}\sum_{l=1}^{M} \nabla\Psi_l(\bar{\boldsymbol{x}}_l)\,. \tag{14.11.11}$$

If $\boldsymbol{D}$ is diagonal, then an efficient “state vector” implementation of this method requires storing $M+3$ vectors of length n_{p}.

MIRT *See* `pl_iot.m`.

14.11.2 Incremental aggregated gradient (IAG) methods

In cases where the algorithm (14.11.11) converges, the $\bar{\boldsymbol{x}}_m$ vectors will approach a common limit as the iterations proceed, so it might seem natural to replace the first “$\bar{\boldsymbol{x}}_l$” term in (14.11.11) with the most recent estimate of $\hat{\boldsymbol{x}}$, *i.e.*, $\bar{\boldsymbol{x}}_{m-1}$. (The “$m-1$” must be taken modulo M.) We can write such a modified update as follows:

e,ox,inc,blatt

$$\bar{\boldsymbol{x}}_m := \bar{\boldsymbol{x}}_{m-1} - \boldsymbol{D}^{-1}\frac{1}{M}\sum_{l=1}^{M} \nabla\Psi_l(\bar{\boldsymbol{x}}_l)\,. \tag{14.11.12}$$

Furthermore, the simplest choice for $\boldsymbol{D}$ is $\boldsymbol{D} = \frac{1}{\alpha}\boldsymbol{I}$ for some sufficiently small α. In this case the update (14.11.12) simplifies precisely to the **incremental aggregated gradient** (**IAG**) algorithm described by Blatt et al. [251]. If $\boldsymbol{D}$ is diagonal, this method also requires storing $M+3$ vectors of length n_{p}.

Modifying (14.11.11) to become (14.11.12) seems to lose all assurances of monotonicity and would seem to risk compromising convergence. Interestingly, Blatt et al. nevertheless have shown convergence of (14.11.12) under reasonable assumptions on Ψ [251]. It is likely that IAG and IOT have identical asymptotic convergence rates. It is an *open problem* to compare the convergence rates of these two methods in the early iterations. Another *open problem* is to see if one can derive (14.11.12) by some other majorization approach.

Another interesting *open problem* is to generalize (14.11.12) to the case of nonquadratic penalized likelihood, as described in the next section, where the likelihood surrogates are updated less frequently than the penalty surrogates.

s,ox,inc,pl

14.11.3 Nested quadratic surrogates for penalized-likelihood problems (s,ox,inc,pl)

For most penalized-likelihood image reconstruction problems, the cost function has the form

$$\Psi(\boldsymbol{x}) = \sum_{m=1}^{M} \text{Ł}_m(\boldsymbol{x}) + \mathsf{R}(\boldsymbol{x}),$$

where $\text{Ł}_m(\boldsymbol{x})$ denotes the negative log-likelihood associated with the mth data block, and $\mathsf{R}(\boldsymbol{x})$ is a roughness penalty function. Usually it is much more expensive to evaluate gradients of Ł_m than of $\mathsf{R}(\boldsymbol{x})$. In such cases the general algorithm implementation (14.11.7) can be suboptimal. It can be more efficient to find individual quadratic surrogates for the log-likelihood terms and for the roughness penalty, and to update the penalty surrogates more frequently. For the log-likelihood terms, we first find quadratic surrogates of the form

e,os,inc,qs,sur_m

$$\phi_m(\boldsymbol{x};\bar{\boldsymbol{x}}) = \text{Ł}_m(\bar{\boldsymbol{x}}) + (\boldsymbol{x}-\bar{\boldsymbol{x}})'\,\boldsymbol{g}_m(\bar{\boldsymbol{x}}) + \frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{x}})'\,\breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}})(\boldsymbol{x}-\bar{\boldsymbol{x}}), \tag{14.11.13}$$

where the gradient vector is:

$$\boldsymbol{g}_m(\boldsymbol{x}) \triangleq \nabla\,\text{Ł}_m(\boldsymbol{x}),$$

and where each $\breve{\boldsymbol{C}}_m$ is a $n_{\mathrm{p}} \times n_{\mathrm{p}}$ Hermitian positive-semidefinite matrix that majorizes Ł_m as follows:

e,ox,inc,major,L

$$\begin{aligned} \phi_m(\boldsymbol{x};\boldsymbol{x}) &= \text{Ł}_m(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \\ \phi_m(\boldsymbol{x};\bar{\boldsymbol{x}}) &\geq \text{Ł}_m(\boldsymbol{x}), \qquad \forall \boldsymbol{x}, \bar{\boldsymbol{x}}. \end{aligned} \tag{14.11.14}$$

Typically the form of each partial negative log-likelihood function is

$$\text{Ł}_m(\boldsymbol{x}) = \sum_{i\in\mathcal{S}_m} \mathsf{h}_i([\boldsymbol{A}\boldsymbol{x}]_i),$$

where often we have quadratic surrogates available for each h_i function:

$$\mathsf{h}_i(t) \leq q_i(t;s) \triangleq \mathsf{h}_i(s) + \dot{\mathsf{h}}_i(s)(t-s) + \frac{1}{2}\,\breve{c}_i(s)(t-s)^2.$$

In these cases, $[\nabla \text{Ł}_m(\boldsymbol{x})]_j = \sum_{i \in \mathcal{S}_m} a_{ij}\, \dot{\mathsf{h}}_i([\boldsymbol{A}\boldsymbol{x}]_i)$ and the natural (nonseparable) option for $\breve{\boldsymbol{C}}_m$ in (14.11.13) is

$$\breve{\boldsymbol{C}}_m(\boldsymbol{x}) = \boldsymbol{A}_m' \,\mathsf{diag}\{\breve{c}_i([\boldsymbol{A}\boldsymbol{x}]_i)\}\, \boldsymbol{A}_m = \boldsymbol{A}' \,\mathsf{diag}\Big\{\breve{c}_i([\boldsymbol{A}\boldsymbol{x}]_i)\, \mathbb{I}_{\{i \in \mathcal{S}_m\}}\Big\}\, \boldsymbol{A},$$

where $\boldsymbol{A}_m$ denotes the $|\mathcal{S}_m| \times n_{\mathrm{p}}$ matrix consisting of the rows of $\boldsymbol{A}$ for which $i \in \mathcal{S}_m$. Another (separable) option is to choose

$$\breve{\boldsymbol{C}}_m(\boldsymbol{x}) = \mathsf{diag}\{d_{mj}(\boldsymbol{x})\} \text{ where } d_{mj}(\boldsymbol{x}) \triangleq \sum_{i \in \mathcal{S}_m} |a_{ij}|\, |a|_i\, \breve{c}_i([\boldsymbol{A}\boldsymbol{x}]_i),$$

where $|a|_i \triangleq \sum_{j=1}^{n_{\mathrm{p}}} |a_{ij}|$.

Instead of minimizing $\Psi(\boldsymbol{x})$ directly, we apply alternating minimization to the following augmented cost function:

$$\begin{aligned}
\bar{F}(\boldsymbol{x}; \bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_M) &\triangleq \sum_{m=1}^{M} \phi_m(\boldsymbol{x}; \bar{\boldsymbol{x}}_m) + \mathsf{R}(\boldsymbol{x}) \\
&\equiv \sum_{m=1}^{M} \left(\boldsymbol{x}'\, \boldsymbol{g}_m(\bar{\boldsymbol{x}}_m) + \frac{1}{2} (\boldsymbol{x} - \bar{\boldsymbol{x}}_m)'\, \breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}_m) (\boldsymbol{x} - \bar{\boldsymbol{x}}_m) \right) + \mathsf{R}(\boldsymbol{x}),
\end{aligned}$$

and at each subiteration we need to minimize this over $\boldsymbol{x}$, per (14.11.5).

If $\mathsf{R}(\boldsymbol{x})$ is quadratic, then so is $\bar{F}$ and minimizing over $\boldsymbol{x}$ is relatively easy. However, if $\mathsf{R}(\boldsymbol{x})$ is non-quadratic, then minimizing $\bar{F}$ remains challenging. In fact, minimizing $\bar{F}$ with respect to $\boldsymbol{x}$ is essentially a penalized weighted least-squares (**PWLS**) problem. We suggest to apply optimization transfer yet again so as to minimize $\bar{F}(\cdot)$ over $\boldsymbol{x}$ using subiterations. Suppose that we have a (possibly separable) quadratic surrogate for the penalty function of the form

$$\mathsf{R}(\boldsymbol{x}; \tilde{\boldsymbol{x}}) = \mathsf{R}(\tilde{\boldsymbol{x}}) + (\boldsymbol{x} - \tilde{\boldsymbol{x}})' \nabla \mathsf{R}(\tilde{\boldsymbol{x}}) + \frac{1}{2} (\boldsymbol{x} - \tilde{\boldsymbol{x}})' \boldsymbol{K}(\tilde{\boldsymbol{x}}) (\boldsymbol{x} - \tilde{\boldsymbol{x}}),$$

for some curvature matrix $\boldsymbol{K}$. Then we define a surrogate function for $\bar{F}(\cdot)$ as follows:

$$\Phi(\boldsymbol{x}; \tilde{\boldsymbol{x}}; \bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_M) \triangleq \bar{F}(\boldsymbol{x}; \bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_M) - \mathsf{R}(\boldsymbol{x}) + \mathsf{R}(\boldsymbol{x}; \tilde{\boldsymbol{x}}).$$

In particular, note that

$$\arg\min_{\boldsymbol{x}} \bar{F}(\boldsymbol{x}; \ldots) = \arg\min_{\boldsymbol{x}} \min_{\tilde{\boldsymbol{x}}} \Phi(\boldsymbol{x}; \tilde{\boldsymbol{x}}; \ldots).$$

Iteratively descending $\Phi(\cdot)$ with respect to its first two arguments will monotonically descend $\bar{F}(\cdot)$ with respect to $\boldsymbol{x}$. The gradient of Φ is

$$\begin{aligned}
\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}; \tilde{\boldsymbol{x}}; \bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_M) &= \sum_{m=1}^{M} \boldsymbol{g}_m(\bar{\boldsymbol{x}}_m) + \sum_{m=1}^{M} \breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}_m)(\boldsymbol{x} - \bar{\boldsymbol{x}}_m) + \nabla \mathsf{R}(\tilde{\boldsymbol{x}}) + \boldsymbol{K}(\tilde{\boldsymbol{x}})(\boldsymbol{x} - \tilde{\boldsymbol{x}}) \\
&= \left[\hat{\boldsymbol{C}} + \boldsymbol{K}(\tilde{\boldsymbol{x}})\right] \boldsymbol{x} - \sum_{m=1}^{M} \left[\breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}_m)\, \bar{\boldsymbol{x}}_m - \boldsymbol{g}_m(\bar{\boldsymbol{x}}_m)\right] + \nabla \mathsf{R}(\tilde{\boldsymbol{x}}) - \boldsymbol{K}(\tilde{\boldsymbol{x}}) \tilde{\boldsymbol{x}},
\end{aligned}$$

where $\hat{\boldsymbol{C}} \triangleq \sum_{m=1}^{M} \breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}_m)$. Equating to zero and solving yields the following inner iteration:

$$\boldsymbol{x}^{\mathrm{new}} := \left[\hat{\boldsymbol{C}} + \boldsymbol{K}(\boldsymbol{x}^{\mathrm{old}})\right]^{-1} \left[\sum_{l=1}^{M} \left(\breve{\boldsymbol{C}}_l(\bar{\boldsymbol{x}}_l)\, \bar{\boldsymbol{x}}_l - \boldsymbol{g}_l(\bar{\boldsymbol{x}}_l) \right) - \nabla \mathsf{R}(\boldsymbol{x}^{\mathrm{old}}) + \boldsymbol{K}(\boldsymbol{x}^{\mathrm{old}}) \boldsymbol{x}^{\mathrm{old}} \right].$$

Precomputing summations when possible leads to the efficient implementation shown in Fig. 14.11.2. If each $\breve{\boldsymbol{C}}_m$ is diagonal, then the storage requirements are $2M + 3$ vectors of size equal to that of $\boldsymbol{x}$.

In the algorithm description above, we update the penalty term $\mathsf{R}(\boldsymbol{x})$ every time a data block is updated, in fact, even more often. An alternative is to treat the penalty term as one of the "blocks" in (14.10.1), *e.g.*, [245, 252]. Such a strategy would not have "subset gradient balance" but might still be effective in some situations.

```
Penalized-likelihood incremental optimization transfer (PLIOT)
Initialize $\{\bar{\boldsymbol{x}}_m : m = 1, \ldots, M\}$, e.g., by an aggressive non-convergent OS method
Intialize states:
        $\boldsymbol{C}_m := \breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}_m), \; m = 1, \ldots, M$
        $\boldsymbol{v}_m := \boldsymbol{C}_m \bar{\boldsymbol{x}}_m - \nabla \text{Ł}_m(\bar{\boldsymbol{x}}_m), \; m = 1, \ldots, M$
        $\hat{\boldsymbol{C}} := \sum_{m=1}^M \boldsymbol{C}_m$
        $\hat{\boldsymbol{v}} := \sum_{m=1}^M \boldsymbol{v}_m$
Initialize $\hat{\boldsymbol{x}}$, e.g., $\hat{\boldsymbol{x}} = \bar{\boldsymbol{x}}_M$
for $n = 1, 2, \ldots$
    for $m = 1, \ldots, M$
        for one or more subiterations
            $\hat{\boldsymbol{x}} := \left[\hat{\boldsymbol{C}} + \boldsymbol{K}(\hat{\boldsymbol{x}})\right]^{-1} \left[\hat{\boldsymbol{v}} - \nabla \mathsf{R}(\hat{\boldsymbol{x}}) + \boldsymbol{K}(\hat{\boldsymbol{x}}) \, \hat{\boldsymbol{x}}\right].$   (14.11.15)   e,ox,inc,xnew,pliot
        end
        $\hat{\boldsymbol{C}} := \hat{\boldsymbol{C}} - \boldsymbol{C}_m$
        $\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}} - \boldsymbol{v}_m$
        $\boldsymbol{C}_m := \breve{\boldsymbol{C}}_m(\hat{\boldsymbol{x}})$   (14.11.16)
        $\boldsymbol{v}_m := \boldsymbol{C}_m \, \hat{\boldsymbol{x}} - \nabla \text{Ł}_m(\hat{\boldsymbol{x}})$   (14.11.17)
        $\hat{\boldsymbol{C}} := \hat{\boldsymbol{C}} + \boldsymbol{C}_m$
        $\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}} + \boldsymbol{v}_m$
    end
end
```

fig,ox,inc,pliot

Figure 14.11.2: PLIOT algorithm for quadratic surrogates.

```
Penalized-likelihood incremental optimization transfer (PLIOT)-v2
Initialize $\{\bar{\boldsymbol{x}}_m : m = 1, \ldots, M\}$, e.g., by an aggressive non-convergent OS method
Intialize states:
        $\boldsymbol{C}_m := \breve{\boldsymbol{C}}_m(\bar{\boldsymbol{x}}_m),\ m = 1, \ldots, M$
        $\boldsymbol{v}_m := \boldsymbol{C}_m \bar{\boldsymbol{x}}_m - \nabla \text{Ł}_m(\bar{\boldsymbol{x}}_m),\ m = 1, \ldots, M$
        $\hat{\boldsymbol{C}} := \sum_{m=1}^{M} \boldsymbol{C}_m$
        $\hat{\boldsymbol{v}} := \sum_{m=1}^{M} \boldsymbol{v}_m$
Initialize $\hat{\boldsymbol{x}}$ as follows:
    for one or more subiterations
        $\hat{\boldsymbol{x}} := \left[\hat{\boldsymbol{C}} + \boldsymbol{K}(\hat{\boldsymbol{x}})\right]^{-1} \left[\hat{\boldsymbol{v}} - \nabla \mathsf{R}(\hat{\boldsymbol{x}}) + \boldsymbol{K}(\hat{\boldsymbol{x}})\, \hat{\boldsymbol{x}}\right].$    (14.11.18)
    end
for $n = 1, 2, \ldots$
    for $m = 1, \ldots, M$
        $\hat{\boldsymbol{C}} := \hat{\boldsymbol{C}} - \boldsymbol{C}_m$
        $\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}} - \boldsymbol{v}_m$
        $\boldsymbol{C}_m := \breve{\boldsymbol{C}}_m(\hat{\boldsymbol{x}})$    (14.11.19)
        $\boldsymbol{v}_m := \boldsymbol{C}_m\, \hat{\boldsymbol{x}} - \nabla \text{Ł}_m(\hat{\boldsymbol{x}})$    (14.11.20)
        $\hat{\boldsymbol{C}} := \hat{\boldsymbol{C}} + \boldsymbol{C}_m$
        $\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}} + \boldsymbol{v}_m$
        Update $\hat{\boldsymbol{x}}$ using subiterations of (14.11.18)
    end
end
```

e,ox,inc,xnew,pliot,2

fig,ox,inc,pliot,2

Figure 14.11.3: PLIOT-v2 algorithm for quadratic surrogates.

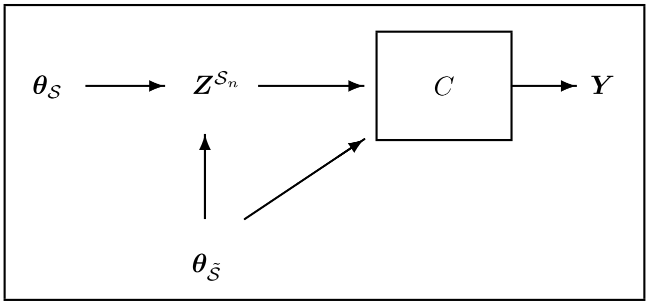

$\boldsymbol{\theta}_{\mathcal{S}}$
$\boldsymbol{Z}^{\mathcal{S}_n}$
C
$\boldsymbol{Y}$
$\boldsymbol{\theta}_{\tilde{\mathcal{S}}}$

Figure 14.12.1: Representing the observed data $\boldsymbol{Y}$ as the output of a possibly noisy "channel" C whose input is the hidden-data $\boldsymbol{Z}^{\mathcal{S}_n}$. fig,sage,channel

s,ox,sage

14.12 Space-alternating generalized EM (SAGE) methods (s,ox,sage)

In imaging applications, EM algorithms typically converge very slowly. Significant acceleration of convergence is possible by appropriately applying the grouped coordinate ascent concept described in §14.5.8. The resulting family of algorithms is called **space alternating generalized EM** (SAGE) algorithms [17, 18]. This section describes the SAGE approach, primarily for historical completeness. The optimization transfer approach with quadratic surrogates seems to offer comparable convergence rates with more flexible yet simpler derivations, so it seems preferable to SAGE algorithms for the imaging problems considered in this *book*. However, SAGE has appeared as a natural approach in other applications, *e.g.*, [253, 254].

14.12.1 Problem

As in §14.9, our goal is to compute the ML estimate $\hat{\boldsymbol{\theta}}$ of a parameter vector $\boldsymbol{\theta}$ given a realization $\boldsymbol{y}$ of a random vector $\boldsymbol{Y}$ with distribution $\mathsf{p}(\boldsymbol{y}; \boldsymbol{\theta}^{\text{true}})$.

The basic idea behind the SAGE method combines ideas from the EM method and the grouped coordinate ascent approach. By introducing a "hidden-data" space for $\boldsymbol{\theta}_{\mathcal{S}}$ based on the statistical structure of the likelihood, we replace the maximization of $\mathsf{L}\big(\boldsymbol{\theta}_{\mathcal{S}}, \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}\big)$ over Θ with the maximization of another functional $Q^{\mathcal{S}_n}(\boldsymbol{\theta}_{\mathcal{S}}; \boldsymbol{\theta}^{(n)})$. If the hidden-data space is chosen wisely, then one can maximize the function $Q^{\mathcal{S}_n}(\cdot; \boldsymbol{\theta}^{(n)})$ *analytically*, obviating the need for line searches. Just as for an EM algorithm, the functionals $Q^{\mathcal{S}_n}$ are constructed to ensure that increases in $Q^{\mathcal{S}_n}$ yield increases in L. Furthermore, we have found empirically for tomography that by using a hidden-data space whose Fisher information is small, the analytical maximum of $Q^{\mathcal{S}_n}(\cdot; \boldsymbol{\theta}^{(n)})$ increases $\mathsf{L}\big(\cdot, \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}\big)$ nearly as much as maximizing $\mathsf{L}\big(\cdot, \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}\big)$ itself. This was formalized in an Appendix of [17], where we proved that less informative hidden-data spaces lead to faster asymptotic convergence rates. In summary, the SAGE method uses the underlying statistical structure of the problem to replace cumbersome or expensive numerical maximizations with analytical or simpler maximizations.

14.12.2 Hidden-data space

To generate the functions $Q^{\mathcal{S}_n}$ for each index set $\mathcal{S}_n$ of interest, we must identify an admissible hidden-data space defined in the following sense:

d,hidden-data

Definition 14.12.1 *A random vector* $\boldsymbol{Z}^{\mathcal{S}_n}$ *with distribution* $\mathsf{p}(\boldsymbol{z}; \boldsymbol{\theta})$ *is an* admissible hidden-data space *with respect to* $\boldsymbol{\theta}_{\mathcal{S}_n}$ *for* $\mathsf{p}(\boldsymbol{y}; \boldsymbol{\theta})$ *if the joint distribution of* $\boldsymbol{Z}^{\mathcal{S}_n}$ *and* $\boldsymbol{Y}$ *satisfies*

e,admit

$$\mathsf{p}(\boldsymbol{y}, \boldsymbol{z}; \boldsymbol{\theta}) = \mathsf{p}\big(\boldsymbol{y} | \boldsymbol{z}; \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}\big)\, \mathsf{p}(\boldsymbol{z}; \boldsymbol{\theta}), \tag{14.12.1}$$

i.e., *the conditional distribution* $\mathsf{p}\big(\boldsymbol{y} | \boldsymbol{z}; \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}\big)$ *must be functionally independent of* $\boldsymbol{\theta}_{\mathcal{S}_n}$. *In other words,* $\boldsymbol{Z}^{\mathcal{S}_n}$ *must be a complete-data space (in the sense of (14.9.3)) for* $\boldsymbol{\theta}_{\mathcal{S}_n}$ *given that* $\boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}$ *is known.*

A few remarks may clarify this definition's relationship to related methods.

- The complete-data space (14.9.3) for the classical EM algorithm [6] is contained as a special case of Definition 14.12.1 by choosing $\mathcal{S} = \{1, \ldots, n_{\mathrm{p}}\}$ and requiring $\boldsymbol{Y}$ to be a deterministic function of $\boldsymbol{Z}^{\mathcal{S}_n}$ [160].
- Under the decomposition (14.12.1), one can think of $\boldsymbol{Y}$ as the output of a noisy channel that may depend on $\boldsymbol{\theta}_{\tilde{\mathcal{S}}}$ but not on $\boldsymbol{\theta}_{\mathcal{S}}$, as illustrated in Fig. 14.12.1.

```
for $n = 0, 1, \ldots$ {
    1. Choose an index set $\mathcal{S}_n$
    2. Choose an admissible hidden-data space $\boldsymbol{Z}^{\mathcal{S}_n}$ for $\boldsymbol{\theta}_{\mathcal{S}_n}$.
    3. E-step: Compute $Q^{\mathcal{S}_n}(\boldsymbol{\theta}_{\mathcal{S}_n}; \boldsymbol{\theta}^{(n)})$ using (14.12.2)
    4. M-step:
        $\boldsymbol{\theta}_{\mathcal{S}_n}^{(n+1)} = \arg\max_{\boldsymbol{\theta}_{\mathcal{S}_n}} Q^{\mathcal{S}_n}(\boldsymbol{\theta}_{\mathcal{S}_n}; \boldsymbol{\theta}^{(n)}),$    (14.12.3)    e,sage
        $\boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}^{(n+1)} = \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}^{(n)}.$    (14.12.4)    e,copy
    5. Optional[a]: Repeat steps 3 and 4.
}
```

[a] Including the optional subiterations of the E- and M-steps yields a "greedier" algorithm. In the few examples we have tried in image reconstruction, the additional greediness was not beneficial. (This is consistent with the benefits of *under*-relaxation for coordinate-ascent analyzed in [255].) In other applications however, such subiterations may improve the convergence rate, and may be computationally advantageous over line-search methods that require analogous subiterations applied directly to $\mathsf{L}(\boldsymbol{\theta})$.

t,sage

Table 14.1: SAGE "Algorithm"

- We use the term "hidden" rather than "complete" to describe $\boldsymbol{Z}^{\mathcal{S}_n}$, because in general $\boldsymbol{Z}^{\mathcal{S}_n}$ will not be complete for $\boldsymbol{\theta}$ in the original sense of Dempster et al. [6]. Even the aggregate of $\boldsymbol{Z}^{\mathcal{S}_n}$ over all of $\mathcal{S}_n$ will not in general be an admissible complete-data space for $\boldsymbol{\theta}$.
- The most significant generalization over the EM complete-data that is embodied by (14.12.1) is that the conditional distribution of $\boldsymbol{Y}$ on $\boldsymbol{Z}^{\mathcal{S}_n}$ *is* allowed to depend on all of the other parameters $\boldsymbol{\theta}_{\tilde{\mathcal{S}}}$ (see Fig. 14.12.1). In the superimposed signal application described in [17], it was precisely this dependency that led to improved convergence rates. It also allows significantly more flexibility in the design of the distribution of $\boldsymbol{Z}^{\mathcal{S}_n}$.
- In principle one could further generalize the SAGE method by combining it with other generalizations such as the cascade EM algorithm [183].

14.12.3 Algorithm

An essential ingredient of any SAGE algorithm is the following conditional expectation of the log-likelihood of $\boldsymbol{Z}^{\mathcal{S}_n}$:

$$\begin{aligned} Q^{\mathcal{S}_n}(\boldsymbol{\theta}_{\mathcal{S}_n}; \boldsymbol{\theta}^{(n)}) &= Q^{\mathcal{S}_n}(\boldsymbol{\theta}_{\mathcal{S}_n}; \boldsymbol{\theta}_{\mathcal{S}_n}^{(n)}, \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}^{(n)}) \\ &\triangleq \mathsf{E}\left[\log \mathsf{p}\left(\boldsymbol{Z}^{\mathcal{S}_n}; \boldsymbol{\theta}_{\mathcal{S}_n}, \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}^{(n)}\right) | \boldsymbol{Y} = \boldsymbol{y}; \boldsymbol{\theta}^{(n)}\right] \qquad (14.12.2) \\ &= \sum_{\boldsymbol{z}} \left[\log \mathsf{p}\left(\boldsymbol{z}; \boldsymbol{\theta}_{\mathcal{S}_n}, \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}^{(n)}\right)\right] \mathsf{p}(\boldsymbol{z} | \boldsymbol{Y} = \boldsymbol{y}; \boldsymbol{\theta}^{(n)}) \,. \end{aligned}$$

Let $\boldsymbol{\theta}^0 \in \Theta$ be an initial parameter estimate. A generic SAGE algorithm produces a sequence of estimates $\{\boldsymbol{\theta}^{(n)}\}_{n=0}^{\infty}$ via the recursion shown in Table 14.1.

The maximization in (14.12.3) is over the set

e,Theta

$$\Theta^{\mathcal{S}_n}(\boldsymbol{\theta}^{(n)}) \triangleq \left\{ \boldsymbol{\theta}_{\mathcal{S}_n} : (\boldsymbol{\theta}_{\mathcal{S}_n}, \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}^{(n)}) \in \Theta \right\}. \qquad (14.12.5)$$

If one chooses the index sets and hidden data spaces appropriately, then typically one can combine the E-step and M-step via an analytical maximization into a recursion of the form $\boldsymbol{\theta}_{\mathcal{S}_n}^{(n+1)} = T_{\mathcal{S}_n}(\boldsymbol{\theta}^{(n)})$. The examples in later sections illustrate this important aspect of the SAGE method.

If one chooses $\boldsymbol{Z}^{\mathcal{S}_n} = \boldsymbol{Y}$, then for that $\mathcal{S}_n$ we see from (14.12.2) that $Q^{\mathcal{S}_n}(\boldsymbol{\theta}_{\mathcal{S}}; \boldsymbol{\theta}^{(n)}) = \mathsf{L}\left(\boldsymbol{\theta}_{\mathcal{S}}, \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}\right)$. Thus, grouped coordinate-ascent (§14.5.8) is a special case of the SAGE method, and one can apply GCA to index sets $\mathcal{S}_n$ for which $\mathsf{L}\left(\boldsymbol{\theta}_{\mathcal{S}}, \boldsymbol{\theta}_{\tilde{\mathcal{S}}_n}\right)$ is easily maximized.

Rather than requiring a strict maximization in (14.12.3), one could settle simply for local maxima [160], or for mere increases in $Q^{\mathcal{S}_n}$, in analogy with GEM algorithms [6]. These generalizations provide the opportunity to further refine the trade-off between convergence rate and computation per-iteration.

14.12.4 Choosing index sets

To implement a SAGE algorithm, one must choose a sequence of index sets $\mathcal{S}_n,\ n = 0, 1, \ldots$. This choice is as much art as science, and will depend on the structure and relative complexities of the E- and M-steps for the problem. To illustrate the trade-offs, we focus on imaging problems, for which there are at least four natural choices for the index sets: 1) the entire image, 2) individual elements of $\boldsymbol{\theta}$ (typically pixels), *i.e.*,

e,ox,sage,scalar

$$\mathcal{S}_n = \{1 + (n \bmod n_{\mathrm{p}})\}\,, \tag{14.12.6}$$

(this was used in the ICM-EM algorithm of [256]), 3) grouping by rows or by columns, and 4) "red-black" type orderings. These four choices lead to different trade-offs between convergence rate and ability to parallelize. A "red-black" grouping was used in a modified EM algorithm in [257] to address the M-step coupling introduced by the smoothness penalties. However, those authors subsequently concluded [258] that the simultaneous-update algorithm by De Pierro [11] (see §18.5.3) is preferable. Those methods use the same complete-data space as in the conventional EM algorithm for image reconstruction [259], so the convergence rate is still slow. Because the E-step for image reconstruction naturally decomposes into n_{p} separate calculations (one for each element of $\boldsymbol{\theta}$), it is natural to update individual elements of $\boldsymbol{\theta}$ as in (14.12.6). By using less informative hidden-data spaces, we showed in [260, 261] that the SAGE algorithm converges faster than the GEM algorithm of Hebert and Leahy [262], which in turn is faster than the method of De Pierro [11]. Thus, for image reconstruction it appears that (14.12.6) is particularly well-suited to 2D reconstruction on serial computers.

Other **domain partitions** have been considered [263, 264].

For image *restoration* problems with spatially-invariant systems, one can compute the E-step of the usual EM algorithm for such problems (see (18.14.4) and (15.6.12) for example) using fast Fourier transforms (FFTs). A SAGE algorithm with single-element index sets (14.12.6) would require direct convolutions. Depending on the width and spectrum of the point-spread function, the improved convergence rate of SAGE using (14.12.6) may be offset by the use of direct convolution. A compromise would be to group the pixels alternately by rows and by columns. This would allow the use of 1D FFTs for the E-step, yet could still retain some of the improved convergence rate. Nevertheless, the SAGE method may be most beneficial in applications with spatially-variant system responses.

Regardless of how one chooses the index sets, we have constructed $Q^{\mathcal{S}_n}$ to ensure that increases in $Q^{\mathcal{S}_n}$ lead to monotone increases in L, by arguments very similar to those of Theorem 14.9.2.

14.12.5 Convergence rate

The **asymptotic convergence rate** of SAGE algorithms is analyzed in [17] following the methods in [35].

s,ox,sage,ex

14.12.6 SAGE example with missing data (s,ox,sage,ex)

x,ox,sage

Example 14.12.2 *Here we continue Example 14.9.3 and develop a SAGE algorithm for that problem.*

Let $\boldsymbol{S}_{ij}$ denote the jth column of $\boldsymbol{S}_i$. When updating θ_j, consider the following hidden data

$$\boldsymbol{Z}_i^j \sim \mathsf{N}(\boldsymbol{S}_{ij}\theta_j, \boldsymbol{K}_i), \qquad i = 1, \ldots, 3$$

where $\boldsymbol{K}_i = \boldsymbol{S}_i' \boldsymbol{K} \boldsymbol{S}_i$, so the observed data is related to the hidden data as follows:

$$\boldsymbol{Y}_i = \boldsymbol{Z}_i^j + \sum_{k \neq j} \boldsymbol{S}_{ij}[ik]\,\theta_k, \qquad j = 1, \ldots, 2.$$

The log pdf of $\boldsymbol{Z}^j$ is given by

$$\begin{aligned}
\log f(\boldsymbol{Z}^j; \theta_j) &= \log \prod_{i=1}^3 f(\boldsymbol{Z}_i^j; \theta_j) \\
&\equiv -\sum_{i=1}^3 \frac{1}{2} (\boldsymbol{Z}_i^j - \boldsymbol{S}_{ij}\theta_j)' \boldsymbol{K}_i^{-1} (\boldsymbol{Z}_i^j - \boldsymbol{S}_{ij}\theta_j) \\
&\equiv \theta_j' \sum_{i=1}^3 \boldsymbol{S}_{ij}' \boldsymbol{K}_i^{-1} \boldsymbol{Z}_i^j - \frac{1}{2} \theta_j' \left[\sum_{i=1}^3 \boldsymbol{S}_{ij}' \boldsymbol{K}_i^{-1} \boldsymbol{S}_{ij} \right] \theta_j.
\end{aligned}$$

Thus the Q^j function is:

$$\begin{aligned} Q^j(\theta_j;\boldsymbol{\theta}^{(n)}) &= \mathsf{E}\big[\log f(\boldsymbol{Z}^j;\theta_j)|\boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\theta}^{(n)}\big] \\ &\equiv \theta_j' \sum_{i=1}^{3} \boldsymbol{S}_{ij}'\boldsymbol{K}_i^{-1}\,\mathsf{E}\Big[\boldsymbol{Z}_i^j|\boldsymbol{Y}_i=\boldsymbol{y}_i;\boldsymbol{\theta}^{(n)}\Big] - \frac{1}{2}\theta_j'\left[\sum_{i=1}^{3}\boldsymbol{S}_{ij}'\boldsymbol{K}_i^{-1}\boldsymbol{S}_{ij}\right]\theta_j. \end{aligned}$$

Using properties of normal random vectors, one can show:

$$\mathsf{E}\Big[\boldsymbol{Z}_i^j|\boldsymbol{Y}_i=\boldsymbol{y}_i;\boldsymbol{\theta}^{(n)}\Big] = \boldsymbol{S}_{ij}\theta_j^{(n)} + (\boldsymbol{y}_i - \boldsymbol{S}_i\boldsymbol{\theta}^{(n)}).$$

Substituting into the Q^j function, minimizing and simplifying yields the following update for (14.12.3):

$$\theta_j^{(n+1)} = \theta_j^{(n)} + \frac{\sum_{i=1}^{3}\boldsymbol{S}_{ij}'\boldsymbol{K}_i^{-1}(\boldsymbol{y}_i - \boldsymbol{S}_i\boldsymbol{\theta}^{(n)})}{\sum_{i=1}^{3}\boldsymbol{S}_{ij}'\boldsymbol{K}_i^{-1}\boldsymbol{S}_{ij}}. \tag{14.12.7}$$

e,ox,sage,ex

Fig. 14.9.1 compares the SAGE iterates to those of the EM algorithm derived in Example 14.9.3. The SAGE algorithm appears to converge faster.

14.13 Nonlinear least squares: Gauss-Newton and Levenberg-Marquardt

s,ox,nls (s,ox,nls)

An important family of cost functions arise in **nonlinear least-squares** problems:

$$\Psi(\boldsymbol{x}) = \frac{1}{2} \left\|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x})\right\|^2,$$

where $\boldsymbol{f} : \mathbb{R}^{n_{\mathrm{p}}} \to \mathbb{R}^{n_{\mathrm{d}}}$ is a nonlinear vector-valued function. The classical **Gauss-Newton** algorithm or **Levenberg-Marquardt** algorithm [284, 285] are often used for this type of minimization problem, but they do not readily accommodate constraints such as nonnegativity on the parameters. Nor are they intrinsically monotonic. An optimization transfer approach could be useful in some applications.

The cost function gradient is

$$\nabla \Psi(\boldsymbol{x}) = (\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y}) \nabla \boldsymbol{f}(\boldsymbol{x})$$

and the Hessian is

$$\nabla^2 \Psi(\boldsymbol{x}) = \sum_{i=1}^{n_{\mathrm{d}}} (f_i(\boldsymbol{x}) - y_i) \boldsymbol{H}_i + \nabla \boldsymbol{f}(\boldsymbol{x}) \nabla \boldsymbol{f}(\boldsymbol{x}),$$

where $\boldsymbol{H}_i \triangleq \nabla^2 f_i(\boldsymbol{x})$.

The **Gauss-Newton** method ignores the $\boldsymbol{H}_i$ terms in the Hessian of Ψ, making the approximation:

$$\Psi(\boldsymbol{x}) \approx \Psi(\boldsymbol{x}^{(n)}) + \nabla \Psi(\boldsymbol{x}^{(n)})(\boldsymbol{x} - \boldsymbol{x}^{(n)}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{(n)})' \nabla \boldsymbol{f}(\boldsymbol{x}) \nabla \boldsymbol{f}(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{x}^{(n)}).$$

This leads to the update:

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - [\nabla \boldsymbol{f}(\boldsymbol{x}) \nabla \boldsymbol{f}(\boldsymbol{x})]^{-1} \nabla \Psi(\boldsymbol{x}^{(n)}). \qquad (14.13.1)$$

e,ox,nls,gn

This update is valid if $[\nabla \boldsymbol{f}(\boldsymbol{x}) \nabla \boldsymbol{f}(\boldsymbol{x})]$ is invertible. If not, then the **Levenberg-Marquardt** alternative is:

e,ox,nls,lm

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \left[\nabla \boldsymbol{f}(\boldsymbol{x}) \nabla \boldsymbol{f}(\boldsymbol{x}) + \lambda^{(n)} \boldsymbol{I}\right]^{-1} \nabla \Psi(\boldsymbol{x}^{(n)}), \qquad (14.13.2)$$

where the $\lambda^{(n)}$ parameters are chosen to ensure descent.

If we can find matrices $\boldsymbol{U}_i$ and $\boldsymbol{U}$ such that

$$(f_i(\boldsymbol{x}) - y_i) \boldsymbol{H}_i \preceq \boldsymbol{U}_i, \qquad \nabla \boldsymbol{f}(\boldsymbol{x}) \nabla \boldsymbol{f}(\boldsymbol{x}) \preceq \boldsymbol{U}$$

for all $\boldsymbol{x}$, then we can use a quadratic surrogate of the form (14.3.5) with

$$\boldsymbol{J}_0 = \sum_i \boldsymbol{U}_i + \boldsymbol{U}.$$

Developing general methods for finding such upper bounds is an *open problem*.

The Gauss-Newton approximation can be derived by linearizing the nonlinear function $\boldsymbol{f}(\boldsymbol{x})$, leading to the approximate cost function:

$$\begin{aligned}
\tilde{\Psi}(\boldsymbol{x}) &\triangleq \frac{1}{2} \left\|\boldsymbol{y} - f(\boldsymbol{x}^{(n)}) - \nabla \boldsymbol{f}(\boldsymbol{x}^{(n)})(\boldsymbol{x} - \boldsymbol{x}^{(n)})\right\|^2 \\
&\equiv \Psi(\boldsymbol{x}^{(n)}) + \nabla \Psi(\boldsymbol{x}^{(n)}) (\boldsymbol{x} - \boldsymbol{x}^{(n)}) + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}^{(n)})' (\nabla \boldsymbol{f}(\boldsymbol{x}^{(n)}) \nabla \boldsymbol{f}(\boldsymbol{x}^{(n)})) (\boldsymbol{x} - \boldsymbol{x}^{(n)}).
\end{aligned}$$

It seems plausible that a reasonable quadratic surrogate can be developed using this approximation as a starting point.

14.14 Summary (s,ox,summ)

s,ox,summ

In summary, this chapter has presented general tools for developing **optimization transfer** or **majorize-minimize** (**MM**) algorithms, particularly for image reconstruction problems. A few examples illustrated the ideas; many more appear elsewhere in the book. Other examples exist as well, *e.g.*, for **multidimensional scaling** [286, p. 120] [287].

s,ox,prob

14.15 Problems (s,ox,prob)

p,ox,mono

Problem 14.1 *From the perspective of algorithm development, is condition (14.1.4) more general, less general, or equivalent to condition (14.1.2)?*

p,ox,tangent

Problem 14.2 *Prove the* ***matched tangent*** *condition (14.1.5), under appropriate conditions on the parameter set* $\mathcal{X}$.

p,ox,min,psd

Problem 14.3 *Analyze the asymptotic convergence rate of the surrogate minimization method described in §14.2.4. As a simpler alternative, consider the case where* α *is a constant.* *(Solve?)*

p,ox,line1

Problem 14.4 *Analyze the asymptotic convergence rate of a version of the steepest descent algorithm that uses just one subiteration of the line-search method described in §14.5.6.* *(Solve?)*

p,ox,sps,gen

Problem 14.5 *Generalize the derivation of the SQS in §14.5.7 to the case of cost functions of the form*

$$\Psi(\boldsymbol{x}) = \sum_{i=1}^{n_{\mathrm{d}}} \mathsf{h}_i(f_i(\boldsymbol{x})), \qquad f_i(\boldsymbol{x}) = \sum_{j=1}^{n_{\mathrm{p}}} f_{ij}(x_j),$$

where the h_i *functions are convex but the functions* $f_{ij}(\cdot)$ *need not be linear. (You should derive a surrogate that is* separable *but not necessarily quadratic. You need not try to minimize that surrogate.)*

p,ox,rot,inv

Problem 14.6 *As described in §2.4.2, generalize the algorithm derived in Example 14.5.1 to a 2D problem with a rotationally invariant (isotropic) regularizer of the form (2.4.4). If the potential function is the absolute value, with corner rounding, then this problem is a version of (isotropic) TV denoising. Hint: use (14.4.15). (Using optimization transfer, this generalization is much simpler than in the half-quadratic approach [82].)*

p,ox,asymm

Problem 14.7 *The symmetry condition in Huber's curvature theorem Theorem 14.4.5 seems unnecessarily restrictive. This problem explores relaxing that condition.*

1. *Consider the "asymmetric parabola" function*

$$\psi(t) = \begin{cases} \alpha t^2/2, & t \geq 0 \\ \beta t^2/2, & t \leq 0, \end{cases}$$

   where $\alpha, \beta \geq 0$. *Show that the parabola* $q(t; s)$ *in (14.4.8) is a valid surrogate for* ψ *if it has curvature* $\breve{c} = \max\{\alpha, \beta\}$.

2. *Prove or disprove: the curvature* $\breve{c}$ *above is optimal.*

3. *Now consider the following more general conditions (same as Huber's except for symmetry).*
   - ψ *is differentiable*
   - *For* $t > 0$, $\dot{\psi}(t)/t$ *is bounded and nonincreasing as* $t \to \infty$.
   - *For* $t < 0$, $\dot{\psi}(t)/t$ *is bounded and nonincreasing as* $t \to -\infty$.

   Prove or disprove. Under the above conditions, the following curvature ensures that $q(t; s)$ *in (14.4.15) is a valid surrogate for* ψ*:*

$$\breve{c}_\psi(t) = \max\left\{\frac{\dot{\psi}(t)}{t}, \frac{\dot{\psi}(-t)}{-t}\right\}.$$

   If this choice does not work, then find an appropriate (preferably optimal) choice.

(Solve?)

p,ox,sqs,complex

Problem 14.8 *Generalize the SQS derivation to the case where* $\boldsymbol{y}$, $\boldsymbol{x}$, *and* $\boldsymbol{B}$ *are all complex.* *(Need typed.)*

p,ox,sqs,np=2

Problem 14.9 *For the case* $n_{\mathrm{p}} = 2$, *consider a convex cost function* $\Psi(\boldsymbol{x})$ *with matrix upper bound on its curvature:* $\begin{bmatrix} a & b \\ b & c \end{bmatrix} \succeq \nabla^2 \Psi(\boldsymbol{x})$. *Find the optimal separable quadratic surrogate having curvature* $\boldsymbol{C} = \alpha \boldsymbol{I}_2$ *that minimizes the root convergence factor* $\rho\left(\boldsymbol{I} - \boldsymbol{C}^{-1} \begin{bmatrix} a & b \\ b & c \end{bmatrix}\right)$. *(Solve?)*

p,ox,it,gen

Problem 14.10 *Generalize the iterative soft-thresholding algorithm of Example 14.5.2 to the case of non-quadratic data-fit terms*

$$\Psi(\boldsymbol{x}) = \sum_{i=1}^{n_{\mathrm{d}}} \mathsf{h}_i([\boldsymbol{A}\boldsymbol{x}]_i) + \beta \left\| \boldsymbol{U}' \boldsymbol{x} \right\|_1 .$$

Make appropriate reasonable assumptions about the h_i *functions.*

p,ox,it,res

Problem 14.11 *Apply the IST algorithm of Example 14.5.2 to a 2D image restoration problem with shift-invariant blur. Hint. One can modify* `mri_cs_ist_example.m` *to use* `Gblur` *instead of* `Gdft`*.*

p,ox,sep,mult

Problem 14.12 *Analyze the convergence properties of the separable multiplicative update (14.3.13).* *(Solve?)*

p,ox,em,sur,vs,Q

Problem 14.13 *Prove the equality (14.9.15) relating majorizers to EM surrogates.*

p,ox,inc,wls

Problem 14.14 *Develop an incremental algorithm for the WLS cost function by studying §14.7.2, §15.6.4, and §17.7.2. (Need typed.)*

p,ox,recon,mult

Problem 14.15 *Generalize the algorithm described in §14.4.3 to the case where each* ψ_i *is convex on* $(-r_i, \infty)$*, where* $r_i \geq 0$*. Hint. Consider §17.5.2.* *(Need typed.)*

p,ox,hq,mag

Problem 14.16 *For the case of complex* $\boldsymbol{x}$ *with highly oscillatory phase, a regularizer of the form* $\sum_k \psi_k([\boldsymbol{C}\,|\boldsymbol{x}|]_k)$ *where* $|\boldsymbol{x}|$ *denotes the vector whose elements are the magnitude of the elements of* $\boldsymbol{x}$ *was proposed in [288, 289], and a half-quadratic method was presented for minimization. Develop an optimization transfer approach and compare the surrogate functions.* *(Solve?)*

p,ox,kinetic

Problem 14.17 *Develop an optimization transfer method for performing* ***nonlinear least-squares*** *estimation of* ***kinetic model parameters****. Specifically, consider the cost function*

$$\underset{\boldsymbol{a},\boldsymbol{\alpha}}{\arg\min} \sum_{i=1}^{n_{\mathrm{d}}} \left| y_i - \sum_j a_j \, \mathrm{e}^{-t_i \alpha_j} \right|^2 .$$

Compare to classical NLS estimation methods in terms of convergence speed and reliability. *(Solve?)*

p,ox,pca

Problem 14.18 *In the problem known as "sparse nonnegative matrix factorization" or "nonnegative sparse coding," one wishes to minimize a cost function of the following form [290, 291]*

$$\underset{\boldsymbol{A},\boldsymbol{B}}{\arg\min} \frac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{A}\boldsymbol{B} \right\|^2 + \beta \sum_{i,j} \psi(b_{ij}),$$

with nonnegativity constraints on the elements of $\boldsymbol{A}$ *and* $\boldsymbol{B}$*. This is a generalization of* ***principle components analysis*** *(****PCA****) [292] that requires nonnegativity and sparseness. Develop an optimization transfer solution to this problem and compare to the algorithm in [291].* *(Solve?)*

p,ox,gg

Problem 14.19 *Show that if* $0 < p \leq 1$ *and* $s \neq 0$ *then*

$$|t|^p \leq |t| \, p \, |s|^{p-1} + (1-p) \, |s|^p .$$

This majorizer is useful for wavelet-based image restoration methods with heavy-tailed priors on the wavelet coefficients [293, 294] that encourage ***sparse*** *wavelet representations. (See Problem 1.12.)* *(Need typed.)*

p,ox,zoom

Problem 14.20 *Use optimization transfer to develop an algorithm for edge-preserving image expansion or zooming. This will be a generalization of Problem 1.23 [295, 296].*

p,ox,ist,tv

Problem 14.21 *The standard* ***iterative soft thresholding*** *(****IST****) algorithm described in §14.5.7.3 is not applicable to a standard* ***total variation*** *type of regularizer* $\|\boldsymbol{C}\boldsymbol{x}\|_1$ *because the usual choices for* $\boldsymbol{C}$ *like* $\boldsymbol{D}_N$ *in (1.8.4) are not invertible. However if we add one more row to* $\boldsymbol{D}_N$ *as follows:* $\boldsymbol{C} = \begin{bmatrix} 1\,0\,\ldots\,0 \\ \boldsymbol{D}_N \end{bmatrix}$ *then this matrix is invertible and its inverse is lower-triangular. Develop a IST algorithm for 1D problems with this regularizer. Does the principle generalize to 2D?*

p,ox,qn

Problem 14.22 *Compare a general-purpose optimization method such as a limited-memory* ***quasi-Newton*** *(****QN****) method to one of the special purpose optimization transfer methods described in this chapter for an image reconstruction algorithm. Open-source QN software is available online for example at*
`https://software.sandia.gov/trac/poblano` *(Solve?)*

14.16 Bibliography

jacobson:07:aet [1] M. W. Jacobson and J. A. Fessler. "An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms." In: *IEEE Trans. Im. Proc.* 16.10 (Oct. 2007), 2411–22. DOI: 10.1109/TIP.2007.904387 (cit. on p. 14.4).

robini:15:ghq [2] M. C. Robini and Y. Zhu. "Generic half-quadratic optimization for image reconstruction." In: *SIAM J. Imaging Sci.* 8.3 (2015), 1752–97. DOI: 10.1137/140987845 (cit. on p. 14.4).

lange:00:otu [3] K. Lange, D. R. Hunter, and I. Yang. "Optimization transfer using surrogate objective functions." In: *J. Computational and Graphical Stat.* 9.1 (Mar. 2000), 1–20. URL: http://www.jstor.org/stable/1390605 (cit. on pp. 14.4, 14.6, 14.10, 14.13, 14.40).

ortega:70 [4] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. New York: Academic, 1970. DOI: 10.1137/1.9780898719468 (cit. on p. 14.4).

huber:81 [5] P. J. Huber. *Robust statistics*. New York: Wiley, 1981 (cit. on pp. 14.4, 14.11, 14.17, 14.20, 14.23).

dempster:77:mlf [6] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." In: *J. Royal Stat. Soc. Ser. B* 39.1 (1977), 1–38. URL: http://www.jstor.org/stable/2984875 (cit. on pp. 14.4, 14.16, 14.40, 14.41, 14.42, 14.44, 14.56, 14.57).

heiser:95:ccb [7] W. J. Heiser. "Convergent computing by iterative majorization: theory and applications in multidimensional data analysis." In: *Recent Advances in Descriptive Multivariate Analysis*. Ed. by W J Krzasnowski. Oxford: Clarendon, 1995, pp. 157–89 (cit. on p. 14.4).

kiers:02:sua [8] H. A. L. Kiers. "Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems." In: *Comp. Stat. Data Anal.* 41.1 (Nov. 2002), 157–70. DOI: 10.1016/S0167-9473(02)00142-1 (cit. on p. 14.4).

depierro:87:otc [9] A. R. De Pierro. "On the convergence of the iterative image space reconstruction algorithm for volume ECT." In: *IEEE Trans. Med. Imag.* 6.2 (June 1987), 174–5. DOI: 10.1109/TMI.1987.4307819 (cit. on p. 14.4).

depierro:93:otr [10] A. R. De Pierro. "On the relation between the ISRA and the EM algorithm for positron emission tomography." In: *IEEE Trans. Med. Imag.* 12.2 (June 1993), 328–33. DOI: 10.1109/42.232263 (cit. on pp. 14.4, 14.14).

depierro:95:ame [11] A. R. De Pierro. "A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography." In: *IEEE Trans. Med. Imag.* 14.1 (Mar. 1995), 132–7. DOI: 10.1109/42.370409 (cit. on pp. 14.4, 14.14, 14.27, 14.30, 14.58).

depierro:95:otc [12] A. R. De Pierro. "On the convergence of an EM-type algorithm for penalized likelihood estimation in emission tomography." In: *IEEE Trans. Med. Imag.* 14.4 (Dec. 1995), 762–5. DOI: 10.1109/42.476119 (cit. on p. 14.4).

lange:94:aab [13] K. Lange. "An adaptive barrier method for convex programming." In: *Methods and Appl. of Analysis* 1.4 (1994), 392–402. URL: http://www.intlpress.com/MAA/p/1994/1_4/MAA-1-4-392-402.pdf (cit. on p. 14.4).

lange:95:aga [14] K. Lange. "A gradient algorithm locally equivalent to the EM Algorithm." In: *J. Royal Stat. Soc. Ser. B* 57.2 (1995), 425–37. URL: http://www.jstor.org/stable/2345971 (cit. on pp. 14.4, 14.45).

lange:95:gca [15] K. Lange and J. A. Fessler. "Globally convergent algorithms for maximum a posteriori transmission tomography." In: *IEEE Trans. Im. Proc.* 4.10 (Oct. 1995), 1430–8. DOI: 10.1109/83.465107 (cit. on p. 14.4).

lange:99 [16] K. Lange. *Numerical analysis for statisticians*. New York: Springer-Verlag, 1999 (cit. on pp. 14.4, 14.8, 14.9, 14.42).

fessler:94:sag [17] J. A. Fessler and A. O. Hero. "Space-alternating generalized expectation-maximization algorithm." In: *IEEE Trans. Sig. Proc.* 42.10 (Oct. 1994), 2664–77. DOI: 10.1109/78.324732 (cit. on pp. 14.4, 14.56, 14.57, 14.58).

fessler:95:pml [18] J. A. Fessler and A. O. Hero. "Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms." In: *IEEE Trans. Im. Proc.* 4.10 (Oct. 1995), 1417–29. DOI: 10.1109/83.465106 (cit. on pp. 14.4, 14.56).

becker:97:eaw
[19] M. P. Becker, I. Yang, and K. Lange. "EM algorithms without missing data." In: *Stat. Meth. Med. Res.* 6.1 (Jan. 1997), 38–54. DOI: 10.1177/096228029700600104 (cit. on p. 14.4).

hunter:99:phd
[20] D. R. Hunter. "Optimization transfer algorithms in statistics." PhD thesis. Ann Arbor, MI: Univ. of Michigan, Ann Arbor, MI, 48109-2122, Apr. 1999. URL: https://hdl.handle.net/2027.42/131907 (cit. on p. 14.4).

hunter:04:ato
[21] D. R. Hunter and K. Lange. "A tutorial on MM algorithms." In: *American Statistician* 58.1 (Feb. 2004), 30–7. DOI: 10.1198/0003130042836 (cit. on p. 14.4).

lange:16
[22] K. Lange. *MM optimization algorithms*. Soc. Indust. Appl. Math., 2016. DOI: 10.1137/1.9781611974409 (cit. on p. 14.4).

cohen:96:ava
[23] L. D. Cohen. "Auxiliary variables and two-step iterative algorithms in computer vision problems." In: *J. Math. Im. Vision* 6.1 (Jan. 1996), 59–83. DOI: 10.1007/BF00127375(abstract) (cit. on p. 14.4).

allain:06:oga
[24] M. Allain, J. Idier, and Y. Goussard. "On global and local convergence of half-quadratic algorithms." In: *IEEE Trans. Im. Proc.* 15.5 (May 2006), 1130–42. DOI: 10.1109/TIP.2005.864173 (cit. on p. 14.4).

weiszfeld:1937:slp
[25] E. Weiszfeld. "Sur le point pour lequel la somme des distances de n points donnés est minimum." In: *Tôhoku Mathematical Journal* 43 (1937), 355–86 (cit. on p. 14.4).

voss:80:lco
[26] H. Voss and U. Eckhardt. "Linear convergence of generalized Weiszfeld's method." In: *Computing* 25.3 (Sept. 1980), 243–51. DOI: 10.1007/BF02242002 (cit. on p. 14.4).

censor:92:pma
[27] Y. Censor and S. A. Zenios. "Proximal minimization algorithm with D-functions." In: *J. Optim. Theory Appl.* 73.3 (June 1992), 451–64. DOI: 10.1007/BF00940051 (cit. on p. 14.4).

bertsekas:99
[28] D. P. Bertsekas. *Nonlinear programming*. 2nd ed. Belmont: Athena Scientific, 1999. URL: http://www.athenasc.com/nonlinbook.html (cit. on p. 14.4).

parikh:13:pa
[29] N. Parikh and S. Boyd. "Proximal algorithms." In: *Found. Trends in Optimization* 1.3 (2013), 123–231. DOI: 10.1561/2400000003 (cit. on p. 14.4).

booker:99:arf
[30] A. J. Booker et al. "A rigorous framework for optimization of expensive functions by surrogates." In: *Structural Optimization* 17.1 (Feb. 1999), 1–13. DOI: 10.1007/BF01197708 (cit. on p. 14.6).

robini:16:ihq
[31] M. Robini et al. "Inexact half-quadratic optimization for image reconstruction." In: *Proc. IEEE Intl. Conf. on Image Processing*. 2016, 3513–7. DOI: 10.1109/ICIP.2016.7533013 (cit. on p. 14.6).

deleeuw:94:bra
[32] J. de Leeuw. "Block relaxation algorithms in statistics." In: *Information Systems and Data Analysis*. Ed. by M M Richter H H Bock W Lenski. Berlin: Springer-Verlag, 1994, pp. 308–25 (cit. on p. 14.6).

deleeuw:03:bra
[33] J. de Leeuw and G. Michailidis. *Block relaxation algorithms in statistics*. Preprint by email from the author. see deleeuw:94:bra. Supposedly at http://www.stat.uc la.edu/ deleeuw/block.pdf, according to discussion by these authors of lange:00:otu. 2003 (cit. on p. 14.6).

fessler:93:ocd
[34] J. A. Fessler, N. H. Clinthorne, and W. L. Rogers. "On complete data spaces for PET reconstruction algorithms." In: *IEEE Trans. Nuc. Sci.* 40.4 (Aug. 1993), 1055–61. DOI: 10.1109/23.256712 (cit. on pp. 14.7, 14.8, 14.42).

hero:95:cin
[35] A. O. Hero and J. A. Fessler. "Convergence in norm for alternating expectation-maximization (EM) type algorithms." In: *Statistica Sinica* 5.1 (Jan. 1995), 41–54. URL: http://www3.stat.sinica.edu.tw/statistica/oldpdf/A5n13.pdf (cit. on pp. 14.8, 14.58).

lange:95:aqn
[36] K. Lange. "A Quasi-Newton acceleration of the EM Algorithm." In: *Statistica Sinica* 5.1 (Jan. 1995), 1–18. URL: http://www3.stat.sinica.edu.tw/statistica/j5n1/j5n11/j5n11.htm (cit. on pp. 14.9, 14.45).

jamshidian:97:aot
[37] M. Jamshidian and R. I. Jennrich. "Acceleration of the EM algorithm by using quasi-Newton methods." In: *J. Royal Stat. Soc. Ser. B* 59.3 (1997), 569–87. DOI: 10.1111/1467-9868.00083 (cit. on pp. 14.9, 14.10).

gu:04:msd-1
[38] T. Gu et al. "Multiple search direction conjugate gradient method I: methods and their propositions." In: *Int. J. of Computer Mathematics* 81.9 (Sept. 2004), 1133–43. DOI: 10.1080/00207160410001712305 (cit. on p. 14.9).

gu:04:msd-2
[39] T. Gu et al. "Multiple search direction conjugate gradient method II: theory and numerical experiments." In: *Int. J. of Computer Mathematics* 81.10 (Oct. 2004), 1289–307. DOI: 10.1080/00207160412331289065 (cit. on p. 14.9).

bridson:06:amp [40] R. Bridson and C. Greif. "A multi-preconditioned conjugate gradient algorithm." In: *SIAM J. Matrix. Anal. Appl.* 27.4 (2006), 1056–68. DOI: 10.1137/040620047 (cit. on p. 14.9).

florescu:14:amm [41] A. Florescu et al. "A majorize-minimize memory gradient method for complex-valued inverse problems." In: *Signal Processing* 103 (Oct. 2014), 285–95. DOI: 10.1016/j.sigpro.2013.09.026 (cit. on pp. 14.9, 14.10).

jamshidian:93:cga [42] M. Jamshidian and R. I. Jennrich. "Conjugate gradient acceleration of the EM algorithm." In: *J. Am. Stat. Assoc.* 88.421 (Mar. 1993), 221–8. URL: http://www.jstor.org/stable/2290716 (cit. on pp. 14.10, 14.45).

beck:09:afi [43] A. Beck and M. Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." In: *SIAM J. Imaging Sci.* 2.1 (2009), 183–202. DOI: 10.1137/080716542 (cit. on pp. 14.11, 14.14, 14.36).

bohning:88:moq [44] D. Böhning and B. G. Lindsay. "Monotonicity of quadratic approximation algorithms." In: *Ann. Inst. Stat. Math.* 40.4 (Dec. 1988), 641–63. DOI: 10.1007/BF00049423 (cit. on p. 14.11).

wright:07:srb [45] S. J. Wright, R. D. Nowak, and Mário A T Figueiredo. "Sparse reconstruction by separable approximation." In: *Proc. IEEE Conf. Acoust. Speech Sig. Proc.* 2007, 3373–6. DOI: 10.1109/ICASSP.2008.4518374 (cit. on p. 14.11).

luenberger:69 [46] D. G. Luenberger. *Optimization by vector space methods*. New York: Wiley, 1969. URL: http://books.google.com/books?id=lZU0CAH4RccC (cit. on p. 14.12).

konno:97:ool [47] H. Konno, P. T. Thach, and H. Tuy. *Optimization on low rank nonconvex structures*. Dordecth: Kluwer, 1997 (cit. on p. 14.13).

delaney:98:gce [48] A. H. Delaney and Y. Bresler. "Globally convergent edge-preserving regularized reconstruction: an application to limited-angle tomography." In: *IEEE Trans. Im. Proc.* 7.2 (Feb. 1998), 204–21. DOI: 10.1109/83.660997 (cit. on pp. 14.13, 14.22).

press:88 [49] W. H. Press et al. *Numerical recipes in C*. New York: Cambridge Univ. Press, 1988 (cit. on pp. 14.14, 14.25, 14.32).

redner:84:mdm [50] R. Redner and H. Walker. "Mixture densities, maximum likelihood, and the EM algorithm." In: *SIAM Review* 26.2 (Apr. 1984), 195–239. DOI: 10.1137/1026034 (cit. on pp. 14.16, 14.45).

li:13:ssr [51] X. Li and V. Voroninski. "Sparse signal recovery from quadratic measurements via convex programming." In: *SIAM J. Math. Anal.* 45.5 (2013), 3019–33. DOI: 10.1137/120893707.

gerchberg:72:apa [52] R. W. Gerchberg and W. O. Saxton. "A practical algorithm for the determination of phase from image and diffraction plane pictures." In: *Optik* 35.2 (Apr. 1972), 237–46.

fienup:78:roa [53] J. R. Fienup. "Reconstruction of an object from the modulus of its Fourier transform." In: *Optics Letters* 3.1 (July 1978), 27–9. DOI: 10.1364/OL.3.000027.

fienup:82:pra [54] J. R. Fienup. "Phase retrieval algorithms: a comparison." In: *Appl. Optics* 21.15 (Aug. 1982), 2758–69. DOI: 10.1364/AO.21.002758.

yang:94:gsa [55] G. Yang et al. "Gerchberg-Saxton and Yang-Gu algorithms for phase retrieval in an nonunitary transform system: a comparison." In: *Appl. Optics-IP* 33.2 (Jan. 1994), 209–18. DOI: 10.1364/AO.33.000209.

osullivan:08:ama [56] J. A. O'Sullivan and C. Preza. "Alternating minimization algorithm for quantitative differential-interference contrast (DIC) microscopy." In: *Proc. SPIE 6814 Computational Imaging VI*. 2008, 68140Y. DOI: 10.1117/12.785471.

moravec:07:cpr [57] M. L. Moravec, J. K. Romberg, and R. G. Baraniuk. "Compressive phase retrieval." In: *Proc. SPIE 6701 Wavelets XII*. 2007, p. 670120. DOI: 10.1117/12.736360.

chan:08:tiw [58] W. L. Chan et al. "Terahertz imaging with compressed sensing and phase retrieval." In: *Optics Letters* 33.9 (May 2008), 974–6. DOI: 10.1364/OL.33.000974.

candes:13:pea [59] E. J. Candes, T. Strohmer, and V. Voroninski. "PhaseLift: exact and stable signal recovery from magnitude measurements via convex programming." In: *Comm. Pure Appl. Math.* 66.8 (Aug. 2013), 1241–74. DOI: 10.1002/cpa.21432.

candes:13:prv [60] E. J. Candes et al. "Phase retrieval via matrix completion." In: *SIAM J. Imaging Sci.* 6.1 (2013), 199–225. DOI: 10.1137/110848074.

chouzenoux:13:abc [61] E. Chouzenoux, J. C. Pesquet, and A. Repetti. *A block coordinate variable metric forward-backward algorithm*. 2013. URL: http://www.optimization-online.org/DB_HTML/2013/12/4178.html.

qin:14:csp [62] S. Qin, X. Hu, and Q. Qin. "Compressed sensing phase retrieval with phase diversity." In: *Optics Communications* 310.1 (Jan. 2014), 193–8. DOI: 10.1016/j.optcom.2013.08.001.

shechtman:14:gep [63] Y. Shechtman, A. Beck, and Y. C. Eldar. "GESPAR: efficient phase retrieval of sparse signals." In: *IEEE Trans. Sig. Proc.* 62.4 (Feb. 2014), 928–38. DOI: 10.1109/TSP.2013.2297687.

waldspurger:15:prm [64] I. Waldspurger, A. d'Aspremont, and Stephane Mallat. "Phase recovery, MaxCut and complex semidefinite programming." In: *Mathematical Programming* 149.1 (Feb. 2015), 47–81. DOI: 10.1007/s10107-013-0738-9.

setsompop:08:mls [65] K. Setsompop et al. "Magnitude least squares optimization for parallel radio frequency excitation design demonstrated at 7 Tesla with eight channels." In: *Mag. Res. Med.* 59.4 (Apr. 2008), 908–15. DOI: 10.1002/mrm.21513.

kassakian:05:mls [66] P. Kassakian. "Magnitude least-squares fitting via semidefinite programming with applications to beamforming and multidimensional filter design." In: *Proc. IEEE Conf. Acoust. Speech Sig. Proc.* Vol. 3. 2005, 53–6. DOI: 10.1109/ICASSP.2005.1415644.

kassakian:06:caa [67] P. W. Kassakian. "Convex approximation and optimization with applications in magnitude filter design and radiation pattern synthesis." UCB/EECS-2006-64. PhD thesis. EECS Department, UC Berkeley, May 2006. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-64.html.

hayes:80:srf [68] M. Hayes, J. Lim, and A. Oppenheim. "Signal reconstruction from phase or magnitude." In: *IEEE Trans. Acoust. Sp. Sig. Proc.* 28.6 (Dec. 1980), 672–80. DOI: 10.1109/TASSP.1980.1163463.

fessler:99:cgp [69] J. A. Fessler and S. D. Booth. "Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction." In: *IEEE Trans. Im. Proc.* 8.5 (May 1999), 688–99. DOI: 10.1109/83.760336 (cit. on p. 14.26).

chouzenoux:11:amm [70] E. Chouzenoux, J. Idier, and Said Moussaoui. "A majorize-minimize strategy for subspace optimization applied to image restoration." In: *IEEE Trans. Im. Proc.* 20.6 (June 2011), 1517–28. DOI: 10.1109/TIP.2010.2103083.

candes:14:prv [71] E. Candes, X. Li, and M. Soltanolkotabi. *Phase retrieval via Wirtinger flow: Theory and algorithms*. 2014. URL: http://arxiv.org/abs/1407.1065.

netrapalli:13:pru [72] P. Netrapalli, P. Jain, and S. Sanghavi. *Phase retrieval using alternating minimization*. 2013. URL: http://arxiv.org/abs/1306.0160.

erdogan:99:maf [73] H. Erdogan and J. A. Fessler. "Monotonic algorithms for transmission tomography." In: *IEEE Trans. Med. Imag.* 18.9 (Sept. 1999), 801–14. DOI: 10.1109/42.802758 (cit. on pp. 14.19, 14.23).

deleeuw:09:sqm [74] J. de Leeuw and K. Lange. "Sharp quadratic majorization in one dimension." In: *Comp. Stat. Data Anal.* 53.7 (May 2009), 2471–84. DOI: 10.1016/j.csda.2009.01.002 (cit. on p. 14.19).

sotthivirat:04:pli [75] S. Sotthivirat and J. A. Fessler. "Penalized-likelihood image reconstruction for digital holography." In: *J. Opt. Soc. Am. A* 21.5 (May 2004), 737–50. DOI: 10.1364/JOSAA.21.000737 (cit. on pp. 14.19, 14.20).

lange:90:coe [76] K. Lange. "Convergence of EM image reconstruction algorithms with Gibbs smoothing." In: *IEEE Trans. Med. Imag.* 9.4 (Dec. 1990). Corrections, T-MI, 10:2(288), June 1991., 439–46. DOI: 10.1109/42.61759 (cit. on pp. 14.21, 14.33).

bouman:96:aua [77] C. A. Bouman and K. Sauer. "A unified approach to statistical tomography using coordinate descent optimization." In: *IEEE Trans. Im. Proc.* 5.3 (Mar. 1996), 480–92. DOI: 10.1109/83.491321 (cit. on p. 14.21).

yu:11:fmb [78] Z. Yu et al. "Fast model-based X-ray CT reconstruction using spatially non-homogeneous ICD optimization." In: *IEEE Trans. Im. Proc.* 20.1 (Jan. 2011), 161–75. DOI: 10.1109/TIP.2010.2058811 (cit. on p. 14.21).

geman:92:cra [79] D. Geman and G. Reynolds. "Constrained restoration and the recovery of discontinuities." In: *IEEE Trans. Patt. Anal. Mach. Int.* 14.3 (Mar. 1992), 367–83. DOI: 10.1109/34.120331 (cit. on p. 14.22).

delaney:95:afi [80] A. H. Delaney and Y. Bresler. "A fast iterative tomographic reconstruction algorithm." In: *Proc. IEEE Conf. Acoust. Speech Sig. Proc.* Vol. 4. 1995, 2295–8. DOI: 10.1109/ICASSP.1995.479950 (cit. on p. 14.22).

geman:95:nir
[81] D. Geman and C. Yang. “Nonlinear image recovery with half-quadratic regularization.” In: *IEEE Trans. Im. Proc.* 4.7 (July 1995), 932–46. DOI: 10.1109/83.392335 (cit. on p. 14.22).

aubert:97:avm
[82] G. Aubert and L. Vese. “A variational method in image recovery.” In: *SIAM J. Numer. Anal.* 34.5 (Oct. 1997), 1948–97. DOI: 10.1137/S003614299529230X (cit. on pp. 14.22, 14.61).

charbonnier:97:dep
[83] P. Charbonnier et al. “Deterministic edge-preserving regularization in computed imaging.” In: *IEEE Trans. Im. Proc.* 6.2 (Feb. 1997), 298–311. DOI: 10.1109/83.551699 (cit. on p. 14.22).

rivera:03:ehq
[84] M. Rivera and J. L. Marroquin. “Efficient half-quadratic regularization with granularity control.” In: *Im. and Vision Computing* 21.4 (Apr. 2003), 345–57. DOI: 10.1016/S0262-8856(03)00005-2 (cit. on pp. 14.22, 14.25).

kunsch:94:rpf
[85] H. Kunsch. “Robust priors for smoothing and image restoration.” In: *Ann. Inst. Stat. Math.* 46.1 (Mar. 1994), 1–19. DOI: 10.1007/BF00773588 (cit. on p. 14.22).

black:96:otu
[86] M. J. Black and A. Rangarajan. “On the unification of line processes, outlier rejection, and robust statistics with applications in early vision.” In: *Intl. J. Comp. Vision* 19.1 (July 1996), 57–91. DOI: 10.1007/BF00131148 (cit. on p. 14.22).

nikolova:01:fir
[87] M. Nikolova and M. Ng. “Fast image reconstruction algorithms combining half-quadratic regularization and preconditioning.” In: *Proc. IEEE Intl. Conf. on Image Processing*. Vol. 1. 2001, 277–80. DOI: 10.1109/ICIP.2001.959007 (cit. on pp. 14.22, 14.23).

nikolova:05:aoh
[88] M. Nikolova and M. K. Ng. “Analysis of half-quadratic minimization methods for signal and image recovery.” In: *SIAM J. Sci. Comp.* 27.3 (2005), 937–66. DOI: 10.1137/030600862 (cit. on pp. 14.22, 14.23).

idier:01:chq
[89] J. Idier. “Convex half-quadratic criteria and interacting auxiliary variables for image restoration.” In: *IEEE Trans. Im. Proc.* 10.7 (July 2001), 1001–9. DOI: 10.1109/83.931094 (cit. on p. 14.23).

rockafellar:70
[90] R. T. Rockafellar. *Convex analysis*. Princeton: Princeton University Press, 1970 (cit. on p. 14.23).

nikolova:07:teo
[91] M. Nikolova and R. H. Chan. “The equivalence of half-quadratic minimization and the gradient linearization iteration.” In: *IEEE Trans. Im. Proc.* 16.6 (June 2007), 1623–7. DOI: 10.1109/TIP.2007.896622 (cit. on p. 14.23).

green:84:irl
[92] P. J. Green. “Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives.” In: *J. Royal Stat. Soc. Ser. B* 46.2 (1984), 149–92. URL: http://www.jstor.org/stable/2345503 (cit. on p. 14.23).

fessler:00:sir
[93] J. A. Fessler. “Statistical image reconstruction methods for transmission tomography.” In: *Handbook of Medical Imaging, Volume 2. Medical Image Processing and Analysis*. Ed. by M. Sonka and J. Michael Fitzpatrick. Bellingham: SPIE, 2000, pp. 1–70. DOI: 10.1117/3.831079.ch1 (cit. on p. 14.24).

golub:89
[94] G. H. Golub and C. F. Van Loan. *Matrix computations*. 2nd ed. Johns Hopkins Univ. Press, 1989 (cit. on p. 14.29).

chen:08:a8c
[95] Y. Chen et al. “Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate.” In: *ACM Trans. Math. Software* 35.3 (Oct. 2008), 22:1–22:14. DOI: 10.1145/1391989.1391995 (cit. on p. 14.29).

figueiredo:03:aea
[96] M. A. T. Figueiredo and R. D. Nowak. “An EM algorithm for wavelet-based image restoration.” In: *IEEE Trans. Im. Proc.* 12.8 (Aug. 2003), 906–16. DOI: 10.1109/TIP.2003.814255 (cit. on p. 14.30).

daubechies:04:ait
[97] I. Daubechies, M. Defrise, and C. De Mol. “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint.” In: *Comm. Pure Appl. Math.* 57.11 (Nov. 2004), 1413–57. DOI: 10.1002/cpa.20042 (cit. on p. 14.30).

vonesch:09:afm
[98] C. Vonesch and M. Unser. “A fast multilevel algorithm for wavelet-regularized image restoration.” In: *IEEE Trans. Im. Proc.* 18.3 (Mar. 2009), 509–23. DOI: 10.1109/TIP.2008.2008073 (cit. on p. 14.31).

ziskind:88:mll
[99] I. Ziskind and M. Wax. “Maximum likelihood localization of multiple sources by alternating projection.” In: *IEEE Trans. Acoust. Sp. Sig. Proc.* 36.10 (Oct. 1988), 1553–60 (cit. on pp. 14.32, 14.45).

hathaway:91:gcm
[100] R. J. Hathaway and J. C. Bezdek. “Grouped coordinate minimization using Newton’s method for inexact minimization in one vector coordinate.” In: *J. Optim. Theory Appl.* 71.3 (Dec. 1991), 503–16. DOI: 10.1007/BF00941400 (cit. on p. 14.32).

jensen:91:gca
[101] S. T. Jensen, S. Johansen, and S. L. Lauritzen. “Globally convergent algorithms for maximizing a likelihood function.” In: *Biometrika* 78.4 (Dec. 1991), 867–77. DOI: 10.1093/biomet/78.4.867 (cit. on p. 14.32).

clinthorne:94:acd
[102] N. H. Clinthorne. “A constrained dual-energy reconstruction method for material-selective transmission tomography.” In: *Nucl. Instr. Meth. Phys. Res. A.* 353.1 (Dec. 1994), 347–8. DOI: 10.1016/0168-9002(94)91673-X (cit. on p. 14.32).

grippo:99:gcb
[103] L. Grippo and M. Sciandrone. “Globally convergent block-coordinate techniques for unconstrained optimization.” In: *Optim. Meth. Software* 10.4 (Apr. 1999), 587–637. DOI: 10.1080/10556789908805730 (cit. on p. 14.32).

sauer:95:pco
[104] K. D. Sauer, S. Borman, and C. A. Bouman. “Parallel computation of sequential pixel updates in statistical tomographic reconstruction.” In: *Proc. IEEE Intl. Conf. on Image Processing*. Vol. 3. 1995, 93–6. DOI: 10.1109/ICIP.1995.537422 (cit. on p. 14.32).

fessler:97:gca
[105] J. A. Fessler et al. “Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction.” In: *IEEE Trans. Med. Imag.* 16.2 (Apr. 1997), 166–75. DOI: 10.1109/42.563662 (cit. on p. 14.32).

fessler:97:gcd
[106] J. A. Fessler. “Grouped coordinate descent algorithms for robust edge-preserving image restoration.” In: *Proc. SPIE 3170 Im. Recon. and Restor. II*. 1997, 184–94. DOI: 10.1117/12.279713 (cit. on p. 14.32).

schlossmacher:73:ait
[107] E. J. Schlossmacher. “An iterative technique for absolute deviations curve fitting.” In: *J. Am. Stat. Assoc.* 68.344 (Dec. 1973), 857–9. URL: http://www.jstor.org/stable/2284512 (cit. on p. 14.35).

oliveira:09:atv
[108] J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo. “Adaptive total variation image deblurring: A majorization-minimization approach.” In: *Signal Processing* 89.9 (Sept. 2009), 1683–93. DOI: 10.1016/j.sigpro.2009.03.018 (cit. on p. 14.35).

chen:14:tis
[109] P-Y. Chen and I. W. Selesnick. “Translation-invariant shrinkage/thresholding of group sparse signals.” In: *Signal Processing* 94 (Jan. 2014), 476–89. DOI: 10.1016/j.sigpro.2013.06.011 (cit. on p. 14.35).

lu:11:mcn
[110] Z. Lu and T. K. Pong. “Minimizing condition number via convex programming.” In: *SIAM J. Matrix. Anal. Appl.* 32.4 (2011), 1193–211. DOI: 10.1137/100795097.

fletcher:85:sdm
[111] R. Fletcher. “Semi-definite matrix constraints in optimization.” In: *SIAM J. Cont. Opt.* 23.4 (July 1985), 493–. DOI: 10.1137/0323032.

boyd:04
[112] S. Boyd and L. Vandenberghe. *Convex optimization*. UK: Cambridge, 2004. URL: http://web.stanford.edu/~boyd/cvxbook.html.

drori:17:tei
[113] Y. Drori. “The exact information-based complexity of smooth convex minimization.” In: *J. Complexity* 39 (Apr. 2017), 1–16. DOI: 10.1016/j.jco.2016.11.001.

kim:16:ofo
[114] D. Kim and J. A. Fessler. “Optimized first-order methods for smooth convex minimization.” In: *Mathematical Programming* 159.1 (Sept. 2016), 81–107. DOI: 10.1007/s10107-015-0949-3.

mcgaffin:15:ado-arxiv
[115] M. G. McGaffin and J. A. Fessler. *Algorithmic design of majorizers for large-scale inverse problems*. 2015. URL: http://arxiv.org/abs/1508.02958.

vandenberghe:98:dmw
[116] L. Vandenberghe, S. Boyd, and S-P. Wu. “Determinant maximization with linear matrix inequality constraints.” In: *SIAM J. Matrix. Anal. Appl.* 19.2 (1998), 499–533. DOI: 10.1137/S0895479896303430.

bonettini:16:vmi
[117] S. Bonettini et al. “Variable metric inexact line-search-based methods for nonsmooth optimization.” In: *SIAM J. Optim.* 26.2 (2016), 891–921. DOI: 10.1137/15M1019325.

aylor:17:ewc-composite
[118] A. B. Taylor, J. M. Hendrickx, and Francois Glineur. “Exact worst-case performance of first-order methods for composite convex optimization.” In: *SIAM J. Optim.* 27.3 (Jan. 2017), 1283–313. DOI: 10.1137/16m108104x.

osborne:60:opc
[119] E. Osborne. “On pre-conditioning of matrices.” In: *J. Assoc. Comput. Mach.* 7.4 (Oct. 1960), 338–45. DOI: 10.1145/321043.321048.

bauer:63:osm
[120] F. L. Bauer. “Optimally scaled matrices.” In: *Numerische Mathematik* 5.1 (Dec. 1963), 73–87. DOI: 10.1007/BF01385880.

gray:06:tac
[121] R. M. Gray. *Toeplitz and circulant matrices: a review*. 2006. URL: http://www-ee.stanford.edu/~gray/toeplitz.html.

chan:07
[122] R. H-F. Chan and X-Q. Jin. *An introduction to iterative Toeplitz solvers*. Philadelphia: Soc. Indust. Appl. Math., 2007. URL: http://www.ec-securehost.com/SIAM/FA05.html.

strang:86:apf
[123] G. Strang. “A proposal for Toeplitz matrix calculations.” In: *Stud. Appl. Math.* 74 (1986), 171–6.

chan:89:teb
[124] R. Chan and G. Strang. “Toeplitz equations by conjugate gradients with circulant preconditioner.” In: *SIAM J. Sci. Stat. Comp.* 10.1 (1989), 104–19. DOI: 10.1137/0910009.

chan:94:cpf
[125] T. F. Chan and J. A. Olkin. “Circulant preconditioners for Toeplitz-block matrices.” In: *Numer. Algorithms* 6.1 (Mar. 1994), 89–101. DOI: 10.1007/BF02149764.

chan:96:cgm
[126] R. H. Chan and M. K. Ng. “Conjugate gradient methods for Toeplitz systems.” In: *SIAM Review* 38.3 (Sept. 1996), 427–82. DOI: 10.1137/S0036144594276474.

potts:01:pfi
[127] D. Potts and G. Steidl. “Preconditioners for ill-conditioned Toeplitz matrices constructed from positive kernels.” In: *SIAM J. Sci. Comp.* 22.5 (2001), 1741–61. DOI: 10.1137/S1064827599351428.

yagle:02:nfp
[128] A. E. Yagle. “New fast preconditioners for Toeplitz-like linear systems.” In: *Proc. IEEE Conf. Acoust. Speech Sig. Proc.* Vol. 2. 2002, 1365–8. DOI: 10.1109/ICASSP.2002.1006005.

chan:88:aoc
[129] T. F. Chan. “An optimal circulant preconditioner for Toeplitz systems.” In: *SIAM J. Sci. Stat. Comp.* 9.4 (July 1988), 766–71. DOI: 10.1137/0909051.

reeves:02:fro
[130] S. J. Reeves. “Fast restoration of PMMW imagery without boundary artifacts.” In: *Proc. SPIE 4719 Infrared and passive millimeter-wave imaging systems: Design, analysis, modeling, and testing*. 2002, 289–95. DOI: 10.1117/12.477469.

pan:05:fra
[131] R. Pan and S. J. Reeves. “Fast restoration and superresolution with edge-preserving regularization.” In: *Proc. SPIE 5077 Passive Millimeter-Wave Imaging Tech. VI and Radar Sensor Tech. VII*. 2005, 93–9. DOI: 10.1117/12.487300.

pan:05:fhm
[132] R. Pan and S. J. Reeves. “Fast Huber-Markov edge-preserving image restoration.” In: *Proc. SPIE 5674 Computational Imaging III*. 2005, 138–46. DOI: 10.1117/12.587823.

pan:06:ehm
[133] R. Pan and S. J. Reeves. “Efficient Huber-Markov edge-preserving image restoration.” In: *IEEE Trans. Im. Proc.* 15.12 (Dec. 2006), 3728–35. DOI: 10.1109/TIP.2006.881971.

goldfarb:12:fms
[134] D. Goldfarb and S. Ma. “Fast multiple-splitting algorithms for convex optimization.” In: *SIAM J. Optim.* 22.2 (2012), 533–56. DOI: 10.1137/090780705.

johnstone:17:lag
[135] P. R. Johnstone and P. Moulin. “Local and global convergence of a general inertial proximal splitting scheme for minimizing composite functions.” In: *Computational Optimization and Applications* 67.2 (June 2017), 259–92. DOI: 10.1007/s10589-017-9896-7.

elad:07:cas
[136] M. Elad, B. Matalon, and M. Zibulevsky. “Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization.” In: *Applied and Computational Harmonic Analysis* 23.3 (Nov. 2007), 346–67. DOI: 10.1016/j.acha.2007.02.002.

gutknecht:86:ksi
[137] M. H. Gutknecht, W. Niethammer, and R. S. Varga. “K-step iterative methods for solving nonlinear systems of equations.” In: *nummath* 48.6 (June 1986), 699–712. DOI: 10.1007/BF01399689.

chambolle:15:otc
[138] A. Chambolle and C. Dossal. “On the convergence of the iterates of the "Fast iterative shrinkage/Thresholding algorithm".” In: *J. Optim. Theory Appl.* 166.3 (Sept. 2015), 968–82. DOI: 10.1007/s10957-015-0746-4.

kim:18:ala
[139] D. Kim and J. A. Fessler. “Another look at the Fast Iterative Shrinkage/Thresholding Algorithm (FISTA).” In: *SIAM J. Optim.* 28.1 (2018), 223–50. DOI: 10.1137/16M108940X.

varadhan:08:sag
[140] R. Varadhan and C. Roland. “Simple and globally convergent methods for accelerating the convergence of any EM algorithm.” In: *Scandinavian Journal of Statistics* 35 (2008), 335–53. DOI: 10.1111/j.1467-9469.2007.00585.x (cit. on p. 14.45).

tseng:08:oap
[141] P. Tseng. *On accelerated proximal gradient methods for convex-concave optimization*. 2008. URL: http://pages.cs.wisc.edu/~brecht/cs726docs/Tseng.APG.pdf.

jiang:12:aia
[142] K. Jiang, D. Sun, and K-C. Toh. “An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP.” In: *SIAM J. Optim.* 22.3 (2012), 1042–64. DOI: 10.1137/110847081.

scheinberg:16:pip
[143] K. Scheinberg and X. Tang. “Practical inexact proximal quasi-Newton method with global complexity analysis.” In: *Mathematical Programming* 160.1 (Nov. 2016), 495–529. DOI: 10.1007/s10107-016-0997-3.

csiszar:84:iga

[144] I. Csiszar and G. Tusnady. “Information geometry and alternating minimization procedures.” In: *Statistics and Decisions, Supplement Issue* 1 (1984), 205–37 (cit. on pp. 14.38, 14.39, 14.45).

byrne:93:iir

[145] C. L. Byrne. “Iterative image reconstruction algorithms based on cross-entropy minimization.” In: *IEEE Trans. Im. Proc.* 2.1 (Jan. 1993). Erratum and addendum: 4(2):226-7, Feb. 1995., 96–103. DOI: 10.1109/83.210869 (cit. on p. 14.38).

osullivan:98:ama

[146] J. A. O’Sullivan. “Alternating minimization algorithms: From Blahut-Arimoto to expectation-maximization.” In: *Codes, curves, and signals*. Ed. by A Vardy. Hingham, MA: Kluwer, 1998, pp. 173–92. DOI: 10.1007/978-1-4615-5121-8_13 (cit. on p. 14.38).

neal:98:avo

[147] R. Neal and G. E. Hinton. “A view of the EM algorithm that justifies incremental, sparse and other variants.” In: *Learning in Graphical Models*. Ed. by M. I. Jordan. Dordrencht: Kluwer, 1998, pp. 355–68. URL: http://www.cs.toronto.edu/~radford/em.abstract.html (cit. on pp. 14.39, 14.45, 14.46).

gunawardana:99:coe

[148] A. Gunawardana and W. Byrne. *Convergence of EM variants*. Tech. rep. Johns Hopkins University: ECE Dept., Feb. 1999. URL: http://www.clsp.jhu.edu/cgi-bin/zilla/showpage.prl?top (cit. on pp. 14.39, 14.45).

byrne:99:coe

[149] W. Byrne and A. Gunawardana. “Convergence of EM variants.” In: *Proc. 1999 IEEE Info. Theory Wkshp. on Detection, Estimation, Classification and Imaging (DECI)*. 1999, p. 64 (cit. on p. 14.39).

byrne:00:coe

[150] W. Byrne and A. Gunawardana. “Comments on “Efficient training algorithms for HMMs using incremental estimation”.” In: *IEEE Trans. Speech & Audio Proc.* 8.6 (Nov. 2000), 751–4. DOI: 10.1109/89.876315 (cit. on pp. 14.39, 14.45).

gunawardana:01:tig

[151] A. Gunawardana. “The information geometry of EM variants for speech and image processing.” PhD thesis. Baltimore: Johns Hopkins University, 2001. URL: https://dl.acm.org/citation.cfm?id=933549 (cit. on pp. 14.39, 14.45).

gasso:08:snc

[152] G. Gasso, A. Rakotomamonjy, and S. Canu. “Solving non-convex lasso type problems with DC programming.” In: *Proc. IEEE Wkshp. Machine Learning for Signal Proc.* 2008, 450–5. DOI: 10.1109/MLSP.2008.4685522 (cit. on p. 14.40).

tao:98:adc

[153] P. D. Tao and L. T. H. An. “A D.C. optimization algorithm for solving the trust-region subproblem.” In: *siam-joo* 8.2 (May 1998), 476–505. DOI: 10.1137/S1052623494274313 (cit. on p. 14.40).

schule:05:dtb

[154] T. Schule et al. “Discrete tomography by convex-concave regularization and D.C. programming.” In: *Discrete Applied Mathematics* 151.1-3 (Oct. 2005), 229–43. DOI: 10.1016/j.dam.2005.02.028 (cit. on p. 14.40).

an:10:dtb

[155] L. T. H. An, N. T. Phuc, and P. D. Tao. “Discrete tomography based on DC programming and DCA.” In: *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010 IEEE RIVF International Conference on*. 2010, 1–6. DOI: 10.1109/RIVF.2010.5633367 (cit. on p. 14.40).

kim:13:gc3

[156] K-S. Kim et al. “Globally convergent 3D dynamic PET reconstruction with patch-based non-convex low rank regularization.” In: *Proc. IEEE Intl. Symp. Biomed. Imag.* 2013, 1158–61. DOI: 10.1109/ISBI.2013.6556685 (cit. on p. 14.40).

yuille:03:tcc

[157] A. L. Yuille and A. Rangarajan. “The concave-convex procedure.” In: *Neural Computation* 15.4 (Apr. 2003), 915–36. DOI: 10.1162/08997660360581958 (cit. on p. 14.40).

kim:15:dpr

[158] K. Kim et al. “Dynamic PET reconstruction using temporal patch-based low rank penalty for ROI- based brain kinetic analysis.” In: *Phys. Med. Biol.* 60.5 (Mar. 2015), 2019–46. DOI: 10.1088/0031-9155/60/5/2019.

zhang:18:mot

[159] S. Zhang and J. Xin. “Minimization of transformed L_1 penalty: theory, difference of convex function algorithm, and robust application in compressed sensing.” In: *Mathematical Programming* 169.1 (May 2018), 307–36. DOI: 10.1007/s10107-018-1236-x.

hero:93:acp

[160] A. O. Hero and J. A. Fessler. *Asymptotic convergence properties of EM-type algorithms*. Tech. rep. 282. Univ. of Michigan, Ann Arbor, MI, 48109-2122: Comm. and Sign. Proc. Lab., Dept. of EECS, Apr. 1993. URL: http://web.eecs.umich.edu/~fessler/papers/files/tr/93,282,hero-em.pdf (cit. on pp. 14.40, 14.41, 14.56, 14.57).

meng:97:tea

[161] X. L. Meng and D. van Dyk. “The EM algorithm - An old folk song sung to a fast new tune.” In: *J. Royal Stat. Soc. Ser. B* 59.3 (1997), 511–67. DOI: 10.1111/1467-9868.00082 (cit. on pp. 14.42, 14.45).

stark:86
[162] H. Stark and J. W. Woods. *Probability, random processes, and estimation theory for engineers*. Englewood Cliffs, NJ: Prentice-Hall, 1986 (cit. on p. 14.44).

kay:93
[163] S. M. Kay. *Fundamentals of statistical signal processing: Estimation theory*. New York: Prentice-Hall, 1993 (cit. on p. 14.44).

hebert:89:age
[164] T. Hebert and R. Leahy. "A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors." In: *IEEE Trans. Med. Imag.* 8.2 (June 1989), 194–202. DOI: 10.1109/42.24868 (cit. on p. 14.44).

rubin:82:fml
[165] D. B. Rubin and T. H. Szatrowski. "Finding maximum likelihood estimates of patterned covariance matrices by the EM algorithm." In: *Biometrika* 69.3 (Dec. 1982), 657–60. DOI: 10.1093/biomet/69.3.657 (cit. on p. 14.45).

little:83:oje
[166] R. J. Little and D. B. Rubin. "On jointly estimating parameters and missing data by maximizing the complete-data likelihood." In: *American Statistician* 37.3 (Aug. 1983), 218–20. URL: http://www.jstor.org/stable/2683374 (cit. on p. 14.45).

cover:84:aaf
[167] T. M. Cover. "An algorithm for maximizing expected log investment return." In: *IEEE Trans. Info. Theory* 30.2 (Mar. 1984), 369–73. DOI: 10.1109/TIT.1984.1056869 (cit. on p. 14.45).

laird:87:mlc
[168] N. Laird, N. Lange, and D. Stram. "Maximum likelihood computations with repeated measures: Application of the EM algorithm." In: *J. Am. Stat. Assoc.* 82.387 (Mar. 1987), 97–105. URL: http://www.jstor.org/stable/2289129 (cit. on p. 14.45).

feder:88:peo
[169] M. Feder and E. Weinstein. "Parameter estimation of superimposed signals using the EM algorithm." In: *IEEE Trans. Acoust. Sp. Sig. Proc.* 36.4 (Apr. 1988), 477–89. DOI: 10.1109/29.1552 (cit. on p. 14.45).

weiss:88:mla
[170] A. J. Weiss, A. S. Willsky, and B. C. Levy. "Maximum likelihood array processing for the estimation of superimposed signals." In: *Proc. IEEE* 76.2 (Feb. 1988). Correction in IEEE Proc., Vol. 76, No. 6, p. 734, June 1988., 203–5. DOI: 10.1109/5.4396 (cit. on p. 14.45).

feder:89:mln
[171] M. Feder, A. Oppenheim, and E. Weinstein. "Maximum likelihood noise cancellation using the EM algorithm." In: *IEEE Trans. Acoust. Sp. Sig. Proc.* 37.2 (Feb. 1989), 204–16. DOI: 10.1109/29.21683 (cit. on p. 14.45).

segal:91:ema
[172] M. Segal, E. Weinstein, and B. R. Musicus. "Estimate-maximize algorithms for multichannel time and signal estimation." In: *IEEE Trans. Sig. Proc.* 39.1 (Jan. 1991), 1–16. DOI: 10.1109/78.80760 (cit. on p. 14.45).

snyder:92:dst
[173] D. L. Snyder, T. J. Schulz, and J. A. O'Sullivan. "Deblurring subject to nonnegativity constraints." In: *IEEE Trans. Sig. Proc.* 40.5 (May 1992), 1143–50. DOI: 10.1109/78.134477 (cit. on p. 14.45).

boyles:83:otc
[174] R. Boyles. "On the convergence of the EM algorithm." In: *J. Royal Stat. Soc. Ser. B* 45.1 (1983), 47–50. URL: http://www.jstor.org/stable/2345622 (cit. on p. 14.45).

wu:83:otc
[175] C. F. J. Wu. "On the convergence properties of the EM algorithm." In: *Ann. Stat.* 11.1 (Mar. 1983), 95–103. DOI: 10.1214/aos/1176346060 (cit. on p. 14.45).

meng:92:ret
[176] X. L. Meng and D. B. Rubin. "Recent extensions to the EM algorithm." In: *Bayesian Statistics 4*. Ed. by J M Bernardo et al. Oxford: Clarendon Press, 1992, pp. 307–20 (cit. on p. 14.45).

kim:95:tre
[177] D. K. Kim and J. M. G. Taylor. "The restricted EM algorithm for maximum likelihood estimation under linear restrictions on the parameters." In: *J. Am. Stat. Assoc.* 90.430 (June 1995), 708–16. URL: http://www.jstor.org/stable/2291083 (cit. on p. 14.45).

liu:98:pef
[178] C. H. Liu, D. B. Rubin, and Y. N. Wu. "Parameter expansion for EM acceleration - The PX-EM algorithm." In: *Biometrika* 85.4 (Dec. 1998), 755–70. DOI: 10.1093/biomet/85.4.755 (cit. on p. 14.45).

liu:99:pes
[179] C. H. Liu and Y. N. Wu. "Parameter expansion scheme for data augmentation." In: *J. Am. Stat. Assoc.* 94.448 (Dec. 1999), 1264–74. URL: http://www.jstor.org/stable/2669940 (cit. on p. 14.45).

matsuyama:03:tae
[180] Y. Matsuyama. "The alpha-EM algorithm: surrogate likelihood maximization using alpha-logarithmic information measures." In: *IEEE Trans. Info. Theory* 49.3 (Mar. 2003), 692–706. DOI: 10.1109/TIT.2002.808105 (cit. on p. 14.45).

hathaway:85:acf
[181] R. J. Hathaway. "A constrained formulation of maximum-likelihood estimation for normal mixture distributions." In: *Ann. Stat.* 13.2 (June 1985), 795–800. DOI: 10.1214/aos/1176349557 (cit. on p. 14.45).

nettleton:99:cpo
[182] D. Nettleton. “Convergence properties of the EM algorithm in constrained parameter spaces.” In: *The Canadian Journal of Statistics* 27.3 (1999), 639–48. URL: http://www.jstor.org/stable/3316118 (cit. on p. 14.45).

segal:88:tce
[183] M. Segal and E. Weinstein. “The cascade EM algorithm.” In: *Proc. IEEE* 76.10 (Oct. 1988), 1388–90. DOI: 10.1109/5.16341 (cit. on pp. 14.45, 14.57).

titterington:84:rpe
[184] D. M. Titterington. “Recursive parameter estimation using incomplete data.” In: *J. Royal Stat. Soc. Ser. B* 46.2 (1984), 257–67. URL: http://www.jstor.org/stable/2345509 (cit. on p. 14.45).

frenkel:99:rem
[185] L. Frenkel and M. Feder. “Recursive expectation-maximization (EM) algorithms for time-varying parameters with applications to multiple target tracking.” In: *IEEE Trans. Sig. Proc.* 47.2 (Feb. 1999), 306–20. DOI: 10.1109/78.740104 (cit. on p. 14.45).

roche:11:otc
[186] A. Roche et al. “On the convergence of EM-like algorithms for image segmentation using Markov random fields.” In: *Med. Im. Anal.* 15.6 (2011), 830–9. DOI: 10.1016/j.media.2011.05.002 (cit. on p. 14.45).

louis:82:fto
[187] T. A. Louis. “Finding the observed information matrix when using the EM algorithm.” In: *J. Royal Stat. Soc. Ser. B* 44.2 (1982), 226–33. URL: http://www.jstor.org/stable/2345828 (cit. on p. 14.45).

meng:91:uet
[188] X. L. Meng and D. B. Rubin. “Using EM to obtain asymptotic variance-covariance matrices: the SEM algorithm.” In: *J. Am. Stat. Assoc.* 86.416 (Dec. 1991), 899–909. URL: http://www.jstor.org/stable/2290503 (cit. on p. 14.45).

meng:93:mle
[189] X. L. Meng and D. B. Rubin. “Maximum likelihood estimation via the ECM algorithm: A general framework.” In: *Biometrika* 80.2 (June 1993), 267–78. DOI: 10.1093/biomet/80.2.267 (cit. on p. 14.45).

segal:94:vfm
[190] M. R. Segal, P. Bacchetti, and N. P. Jewell. “Variances for maximum penalized likelihood estimates obtained via the EM algorithm.” In: *J. Royal Stat. Soc. Ser. B* 56.2 (1994), 345–52. URL: http://www.jstor.org/stable/2345905 (cit. on p. 14.45).

vandyk:95:mle
[191] D. A. van Dyk, X. L. Meng, and D. B. Rubin. “Maximum likelihood estimation via the ECM algorithm: computing the asymptotic variance.” In: *Statistica Sinica* 5.1 (Jan. 1995), 55–76. URL: http://www3.stat.sinica.edu.tw/statistica/j5n1/j5n14/j5n14.htm (cit. on p. 14.45).

horng:87:eos
[192] S. C. Horng. “Examples of sublinear convergence of the EM algorithm.” In: *Proc. of Stat. Comp. Sect. of Amer. Stat. Assoc.* 1987, 266–71 (cit. on p. 14.45).

meilijson:89:afi
[193] I. Meilijson. “A fast improvement to the EM algorithm on its own terms.” In: *J. Royal Stat. Soc. Ser. B* 51.1 (1989), 127–38. URL: http://www.jstor.org/stable/2345847 (cit. on p. 14.45).

liu:94:tea
[194] C. H. Liu and D. B. Rubin. “The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence.” In: *Biometrika* 81.4 (Dec. 1994), 633–48. DOI: 10.1093/biomet/81.4.633 (cit. on p. 14.45).

meng:94:otg
[195] X. L. Meng and D. B. Rubin. “On the global and componentwise rates of convergence of the EM algorithm.” In: *Linear Algebra and its Applications* 199.1 (Mar. 1994), 413–25. DOI: 10.1016/0024-3795(94)90363-8 (cit. on p. 14.45).

meng:94:otr
[196] X. L. Meng. “On the rate of convergence of the ECM algorithm.” In: *Ann. Stat.* 22.1 (Mar. 1994), 326–39. DOI: 10.1214/aos/1176325371 (cit. on p. 14.45).

hsiao:02:apc
[197] I. T. Hsiao, A. Rangarajan, and G. Gindi. “A provably convergent OS-EM like reconstruction algorithm for emission tomography.” In: *Proc. SPIE 4684 Medical Imaging: Image Proc.* 2002, 10–19. DOI: 10.1117/12.467144 (cit. on p. 14.45).

hathaway:86:aio
[198] R. J. Hathaway. “Another interpretation of EM algorithm for mixture distributions.” In: *Statist. Probab. Letters* 4.2 (Mar. 1986), 53–6. DOI: 10.1016/0167-7152(86)90016-7 (cit. on p. 14.45).

celeux:01:acw
[199] G. Celeux et al. “A component-wise EM algorithm for mixtures.” In: *J. Computational and Graphical Stat.* 10.4 (Dec. 2001), 697–712. URL: http://www.jstor.org/stable/1390967 (cit. on p. 14.45).

nocedal:96:lsu
[200] J. Nocedal. “Large scale unconstrained optimization.” In: *The State of the Art in Numerical Analysis*. Ed. by A. Watson and I. Duff. Oxford: Oxford University Press, 1996. URL: http://www.ece.nwu.edu/~nocedal/recent.html (cit. on p. 14.46).

kibardin:79:dif
[201] V. M. Kibardin. “Decomposition into functions in the minimization problem.” In: *Avtomatika i Telemekhanika* 9 (Sept. 1979). Translation: p. 1311-23 in Plenum Publishing Co. "Adaptive Systems", 66–79. URL: http://mi.mathnet.ru/eng/at/y1979/i9/p66 (cit. on p. 14.46).

bertsekas:97:anc
[202] D. P. Bertsekas. “A new class of incremental gradient methods for least squares problems.” In: *SIAM J. Optim.* 7.4 (Nov. 1997), 913–26. DOI: 10.1137/S1052623495287022 (cit. on p. 14.46).

nedic:00:cro
[203] A. Nedic and D. Bertsekas. “Convergence rate of incremental subgradient algorithms.” In: *Stochastic Optimization: Algorithms and Applications*. Ed. by S. Uryasev and P. M. Pardalos. New York: Kluwer, 2000, pp. 263–304. URL: http://web.mit.edu/6.291/www/ (cit. on p. 14.46).

nedic:01:dai
[204] A. Nedic, D. Bertsekas, and V. Borkar. “Distributed asynchronous incremental subgradient methods.” In: *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*. Ed. by D. Butnariu, Y. Censor, and S. Reich. Amsterdam: Elsevier, 2001, pp. 381–407. DOI: '10.1016/S1570-579X(01)80023-9' (cit. on p. 14.46).

nedic:01:ism
[205] A. Nedic and D. P. Bertsekas. “Incremental subgradient methods for nondifferentiable optimization.” In: *SIAM J. Optim.* 12.1 (2001), 109–38. DOI: 10.1137/S1052623499362111 (cit. on p. 14.46).

censor:04:ssp
[206] Y. Censor, A. R. De Pierro, and M. Zaknoon. “Steered sequential projections for the inconsistent convex feasibility problem.” In: *Nonlinear Analysis: Theory, Methods & Applications (Series A: Theory and Methods)* 59.3 (Nov. 2004), 385–405. DOI: 10.1016/j.na.2004.07.018 (cit. on p. 14.46).

mairal:15:imm
[207] J. Mairal. “Incremental majorization-minimization optimization with application to large-scale machine learning.” In: *SIAM J. Optim.* 25.2 (2015), 829–55. DOI: 10.1137/140957639 (cit. on p. 14.46).

lan:18:aor
[208] G. Lan and Y. Zhou. “An optimal randomized incremental gradient method.” In: *Mathematical Programming* 171.1-2 (Sept. 2018), 167–215. DOI: 10.1007/s10107-017-1173-0 (cit. on p. 14.46).

hebert:90:tdm
[209] T. Hebert, R. Leahy, and M. Singh. “Three-dimensional maximum-likelihood reconstruction for an electronically collimated single-photon-emission imaging system.” In: *J. Opt. Soc. Am. A* 7.7 (July 1990), 1305–13. DOI: 10.1364/JOSAA.7.001305 (cit. on p. 14.46).

holte:90:iir
[210] S. Holte et al. “Iterative image reconstruction for emission tomography: A study of convergence and quantitation problems.” In: *IEEE Trans. Nuc. Sci.* 37.2 (Apr. 1990), 629–35. DOI: 10.1109/23.106689 (cit. on p. 14.46).

herman:93:art
[211] G. T. Herman and L. B. Meyer. “Algebraic reconstruction techniques can be made computationally efficient.” In: *IEEE Trans. Med. Imag.* 12.3 (Sept. 1993), 600–9. DOI: 10.1109/42.241889 (cit. on p. 14.46).

hudson:94:air
[212] H. M. Hudson and R. S. Larkin. “Accelerated image reconstruction using ordered subsets of projection data.” In: *IEEE Trans. Med. Imag.* 13.4 (Dec. 1994), 601–9. DOI: 10.1109/42.363108 (cit. on p. 14.46).

manglos:95:tml
[213] S. H. Manglos et al. “Transmission maximum-likelihood reconstruction with ordered subsets for cone beam CT.” In: *Phys. Med. Biol.* 40.7 (July 1995), 1225–41. DOI: 10.1088/0031-9155/40/7/006 (cit. on p. 14.46).

kamphuis:98:ait
[214] C. Kamphuis and F. J. Beekman. “Accelerated iterative transmission CT reconstruction using an ordered subsets convex algorithm.” In: *IEEE Trans. Med. Imag.* 17.6 (Dec. 1998), 1001–5. DOI: 10.1109/42.746730 (cit. on p. 14.46).

kudo:99:cbi
[215] H. Kudo, H. Nakazawa, and T. Saito. “Convergent block-iterative method for general convex cost functions.” In: *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med.* 1999, 247–250 (cit. on p. 14.46).

erdogan:99:osa
[216] H. Erdogan and J. A. Fessler. “Ordered subsets algorithms for transmission tomography.” In: *Phys. Med. Biol.* 44.11 (Nov. 1999), 2835–51. DOI: 10.1088/0031-9155/44/11/311 (cit. on p. 14.46).

ahn:01:gco
[217] S. Ahn and J. A. Fessler. “Globally convergent ordered subsets algorithms: Application to tomography.” In: *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.* Vol. 2. 2001, 1064–8. DOI: 10.1109/NSSMIC.2001.1009736 (cit. on p. 14.46).

bental:01:tos
[218] A. Ben-Tal, T. Margalit, and A. Nemirovski. “The ordered subsets mirror descent optimization method with applications to tomography.” In: *SIAM J. Optim.* 12.1 (2001), 79–108. DOI: 10.1137/S1052623499354564 (cit. on p. 14.46).

byrne:96:bim
[219] C. L. Byrne. “Block-iterative methods for image reconstruction from projections.” In: *IEEE Trans. Im. Proc.* 5.5 (May 1996), 792–3. DOI: 10.1109/83.499919 (cit. on p. 14.46).

byrne:98:ate
[220] C. L. Byrne. “Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative methods.” In: *IEEE Trans. Im. Proc.* 7.1 (Jan. 1998), 100–9. DOI: 10.1109/83.650854 (cit. on p. 14.46).

robbins:51:asa
[221] H. Robbins and S. Monro. “A stochastic approximation method.” In: *Ann. Math. Stat.* 22.3 (Sept. 1951), 400–7. URL: http://www.jstor.org/stable/2236626 (cit. on p. 14.46).

kesten:58:asa
[222] H. Kesten. “Accelerated stochastic approximation.” In: *Ann. Math. Stat.* 29.1 (Mar. 1958), 41–59. URL: http://www.jstor.org/stable/2237294 (cit. on p. 14.46).

tsitsiklis:86:dad
[223] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. “Distributed asynchronous deterministic and stochastic gradient optimization algorithms.” In: *IEEE Trans. Auto. Control* 31.9 (Sept. 1986), 803–12. DOI: 10.1109/TAC.1986.1104412 (cit. on p. 14.46).

dupuis:91:osc
[224] P. Dupuis and R. Simha. “On sampling controlled stochastic approximation.” In: *IEEE Trans. Auto. Control* 36.8 (Aug. 1991), 915–24. DOI: 10.1109/9.133185 (cit. on pp. 14.46, 14.49).

spall:92:msa
[225] J. C. Spall. “Multivariate stochastic approximation using a simultaneous perturbation gradient approximation.” In: *IEEE Trans. Auto. Control* 37.3 (Mar. 1992), 332–41. DOI: 10.1109/9.119632 (cit. on p. 14.46).

spall:00:asa
[226] J. C. Spall. “Adaptive stochastic approximation by the simultaneous perturbation method.” In: *IEEE Trans. Auto. Control* 45.10 (Oct. 2000), 1839–53. DOI: 10.1109/TAC.2000.880982 (cit. on p. 14.46).

hutchison:04:sai
[227] D. W. Hutchison and J. C. Spall. “Stochastic approximation in finite samples using surrogate processes.” In: *Proc. Conf. Decision and Control*. Vol. 4. 2004, 4157–62. DOI: 10.1109/CDC.2004.1429404 (cit. on p. 14.46).

plakhov:04:asa
[228] A. Plakhov and P. Cruz. “A stochastic approximation algorithm with step size adaptation.” In: *J. of Mathematics and Sciences* 120.1 (Mar. 2004), 964–73. DOI: 10.1023/B:JOTH.0000013559.37579.b2 (cit. on p. 14.46).

bousquet:08:tto
[229] O. Bousquet and Leon Bottou. “The tradeoffs of large scale learning.” In: *Neural Info. Proc. Sys.* Vol. 20. 2008, 161–8. URL: http://papers.nips.cc/paper/3323-the-tradeoffs-of-large-scale-learning (cit. on p. 14.46).

nemirovski:09:rsa
[230] A. Nemirovski et al. “Robust stochastic approximation approach to stochastic programming.” In: *SIAM J. Optim.* 19.4 (2009), 1574–609. DOI: 10.1137/070704277 (cit. on p. 14.46).

byrd:11:otu
[231] R. Byrd et al. “On the use of stochastic Hessian information in optimization methods for machine learning.” In: *SIAM J. Optim.* 21.3 (2011), 977–95. DOI: 10.1137/10079923X (cit. on p. 14.46).

devolder:11:sfo
[232] O. Devolder. *Stochastic first order methods in smooth convex optimization*. 2011. URL: http://www.optimization-online.org/DB_HTML/2012/01/3321.html (cit. on p. 14.46).

niu:11:hal
[233] F. Niu et al. “HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent.” In: *Neural Info. Proc. Sys.* 2011, 693–701. URL: http://pages.cs.wisc.edu/~brecht/publications.html (cit. on pp. 14.46, 14.49).

duchi:12:rsf
[234] J. Duchi, P. Bartlett, and M. Wainwright. “Randomized smoothing for stochastic optimization.” In: *SIAM J. Optim.* 22.2 (2012), 674–701. DOI: 10.1137/110831659 (cit. on p. 14.46).

lan:12:aom
[235] G. Lan. “An optimal method for stochastic composite optimization.” In: *Mathematical Programming* 133.1 (June 2012), 365–97. DOI: 10.1007/s10107-010-0434-y (cit. on p. 14.46).

leroux:13:asg
[236] N. Le Roux, M. Schmidt, and F. Bach. *A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets*. 2013. URL: http://arxiv.org/abs/1202.6258 (cit. on p. 14.46).

johnson:13:asg
[237] R. Johnson and T. Zhang. “Accelerating stochastic gradient descent using predictive variance reduction.” In: *Neural Info. Proc. Sys.* 2013, 315–23. URL: http://papers.nips.cc/paper/4937-accelerating-stochastic-gradient-descent-using-predictive-variance-reduction.pdf (cit. on pp. 14.46, 14.49).

wang:13:vrf
[238] C. Wang et al. “Variance reduction for stochastic gradient optimization.” In: *Neural Info. Proc. Sys.* 2013. URL: http://papers.nips.cc/paper/5034-variance-reduction-for-stochastic-gradient-optimization (cit. on p. 14.46).

friedlander:14:hds
[239] M. P. Friedlander and M. Schmidt. *Hybrid deterministic-stochastic methods for data fitting*. 2014. URL: http://arxiv.org/abs/1104.2373 (cit. on p. 14.46).

nedic:14:oss
[240] A. Nedic and S. Lee. “On stochastic subgradient mirror-descent algorithm with weighted averaging.” In: *SIAM J. Optim.* 24.1 (2014), 84–107. DOI: 10.1137/120894464 (cit. on p. 14.46).

needell:14:sgd
[241] D. Needell, N. Srebro, and R. Ward. *Stochastic gradient descent and the randomized Kaczmarz algorithm*. 2014. URL: http://arxiv.org/abs/1310.5715 (cit. on p. 14.46).

nocedal:14:doa
[242] J. Nocedal. "Devising optimization algorithms for machine learning." In: *SIAM News* 47.7 (Sept. 2014), 1–2. URL: https://sinews.siam.org/Details-Page/Devising-Optimization-Algorithms-for-Machine-Learning (cit. on p. 14.46).

ghadimi:16:agm
[243] S. Ghadimi and G. Lan. "Accelerated gradient methods for nonconvex nonlinear and stochastic programming." In: *Mathematical Programming* 156.1 (Mar. 2016), 59–99. DOI: 10.1007/s10107-015-0871-8 (cit. on p. 14.46).

schmidt:17:mfs
[244] M. Schmidt, N. Le Roux, and F. Bach. "Minimizing finite sums with the stochastic average gradient." In: *Mathematical Programming* 162.1 (Mar. 2017), 83–112. DOI: 10.1007/s10107-016-1030-6 (cit. on p. 14.46).

depierro:01:fel
[245] A. R. De Pierro and M. E. B. Yamagishi. "Fast EM-like methods for maximum 'a posteriori' estimates in emission tomography." In: *IEEE Trans. Med. Imag.* 20.4 (Apr. 2001), 280–8. DOI: 10.1109/42.921477 (cit. on pp. 14.47, 14.53).

hong:10:upc
[246] I. K. Hong et al. "Ultrafast preconditioned conjugate gradient OSEM algorithm for fully 3D PET reconstruction." In: *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.* 2010, 2418–9. DOI: 10.1109/NSSMIC.2010.5874221 (cit. on p. 14.49).

guan:94:apa
[247] H. Guan and R. Gordon. "A projection access order for speedy convergence of ART (algebraic reconstruction technique): a multilevel scheme for computed tomography." In: *Phys. Med. Biol.* 39.11 (Nov. 1994), 2005–22. DOI: 10.1088/0031-9155/39/11/013 (cit. on p. 14.49).

byrd:12:sss
[248] R. H. Byrd et al. "Sample size selection in optimization methods for machine learning." In: *Mathematical Programming* 134.1 (Aug. 2012), 127–55. DOI: 10.1007/s10107-012-0572-5 (cit. on p. 14.49).

pasupathy:14:hmt
[249] R. Pasupathy et al. *How much to sample in simulation-based stochastic recursions?* 2014. URL: http://bibbase.org/network/publication/pasupathy-glynn-ghosh-hahemi-howmuchtosampleinsimulationbasedstochasticrecursions-2014 (cit. on p. 14.49).

ahn:06:cio
[250] S. Ahn et al. "Convergent incremental optimization transfer algorithms: Application to tomography." In: *IEEE Trans. Med. Imag.* 25.3 (Mar. 2006), 283–96. DOI: 10.1109/TMI.2005.862740 (cit. on p. 14.51).

blatt:07:aci
[251] D. Blatt, A. O. Hero, and H. Gauchman. "A convergent incremental gradient method with a constant step size." In: *SIAM J. Optim.* 18.1 (2007), 29–51. DOI: 10.1137/040615961 (cit. on p. 14.52).

depierro:98:fbe
[252] A. R. De Pierro. "Fast Bayesian estimation methods in emission tomography." In: *Proc. SPIE 3459 Bayesian inference for inverse problems*. 1998, 100–5. DOI: 10.1117/12.323789 (cit. on p. 14.53).

nelson:96:imr
[253] L. B. Nelson and H. V. Poor. "Iterative multiuser receivers for CDMA channels: an EM-based approach." In: *IEEE Trans. Comm.* 44.12 (Dec. 1996), 1700–10. DOI: 10.1109/26.545900 (cit. on p. 14.56).

johnston:99:fds
[254] L. A. Johnston and V. Krishnamurthy. "Finite dimensional smoothers for MAP state estimation of bilinear systems." In: *IEEE Trans. Sig. Proc.* 47.9 (Sept. 1999), 2444–59. DOI: 10.1109/78.782188 (cit. on p. 14.56).

fessler:94:pwl
[255] J. A. Fessler. "Penalized weighted least-squares image reconstruction for positron emission tomography." In: *IEEE Trans. Med. Imag.* 13.2 (June 1994), 290–300. DOI: 10.1109/42.293921 (cit. on p. 14.57).

abdalla:92:epi
[256] M. Abdalla and J. W. Kay. "Edge-preserving image restoration." In: *Stochastic Models, Statistical Methods, and Algorithms in Im. Analysis*. Ed. by P Barone, A Frigessi, and M Piccioni. Vol. 74. Lecture Notes in Statistics. New York: Springer, 1992, pp. 1–13. DOI: 10.1007/978-1-4612-2920-9_1 (cit. on p. 14.58).

depierro:87:ago
[257] A. R. De Pierro. *A generalization of the EM algorithm for maximum likelihood estimates from incomplete data*. Tech. rep. MIPG119. Univ. of Pennsylvania: Med. Im. Proc. Group, Dept. of Radiol., Feb. 1987 (cit. on p. 14.58).

herman:92:omf
[258] G. T. Herman, A. R. De Pierro, and N. Gai. "On methods for maximum a posteriori image reconstruction with a normal prior." In: *J. Visual Comm. Im. Rep.* 3.4 (Dec. 1992), 316–24. DOI: 10.1016/1047-3203(92)90035-R (cit. on p. 14.58).

lange:84:era
[259] K. Lange and R. Carson. "EM reconstruction algorithms for emission and transmission tomography." In: *J. Comp. Assisted Tomo.* 8.2 (Apr. 1984), 306–16 (cit. on p. 14.58).

fessler:93:ncd
[260] J. A. Fessler and A. O. Hero. “New complete-data spaces and faster algorithms for penalized-likelihood emission tomography.” In: *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.* Vol. 3. 1993, 1897–901. DOI: 10.1109/NSSMIC.1993.373624 (cit. on p. 14.58).

fessler:94:sag-tr
[261] J. A. Fessler and A. O. Hero. *Space-alternating generalized EM algorithms for penalized maximum-likelihood image reconstruction*. Tech. rep. 286. Univ. of Michigan, Ann Arbor, MI, 48109-2122: Comm. and Sign. Proc. Lab., Dept. of EECS, Feb. 1994. URL: http://web.eecs.umich.edu/~fessler/papers/files/tr/94,286,sage.pdf (cit. on p. 14.58).

hebert:89:abr
[262] T. Hebert and R. Leahy. “A Bayesian reconstruction algorithm for emission tomography using a Markov random field prior.” In: *Proc. SPIE 1092 Med. Im. III: Im. Proc.* 1989, 458–66. DOI: 10.1117/12.953287 (cit. on p. 14.58).

xie:06:anb
[263] D. Xie. “A new block parallel SOR method and its analysis.” In: *SIAM J. Sci. Comp.* 27.5 (2006), 1513–33. DOI: 10.1137/040604777 (cit. on p. 14.58).

xie:99:nps
[264] D. Xie and L. Adams. “New parallel SOR method by domain partitioning.” In: *SIAM J. Sci. Comp.* 20.6 (1999), 2261–81. DOI: 10.1137/S1064827597303370 (cit. on p. 14.58).

geiping:18:cob
[265] J. Geiping and M. Moeller. *Composite optimization by nonconvex majorization-minimization*. 2018. URL: http://arxiv.org/abs/1802.07072.

barber:16:mmc
[266] R. F. Barber and E. Y. Sidky. “MOCCA: mirrored convex/concave optimization for nonconvex composite functions.” In: *J. Mach. Learning Res.* 17.44 (2016), 1–51. URL: http://www.jmlr.org/papers/v17/15-583.html.

dauphin:14:iaa
[267] Y. Dauphin et al. *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization*. 2014. URL: http://arxiv.org/abs/1406.2572.

byrne:15:amo
[268] C. L. Byrne. *Alternating minimization, optimization transfer and proximal minimization are equivalent*. 2015.

byrne:15:amp
[269] C. L. Byrne and J. S. Lee. *Alternating minimization, proximal minimization and optimization transfer are equivalent*. 2015. URL: http://arxiv.org/abs/1512.03034.

parizi:15:gmm
[270] S. N. Parizi et al. *Generalized majorization-minimization*. 2015. URL: http://arxiv.org/abs/1506.07613.

selesnick:09:srw
[271] I. W. Selesnick and Mário A T Figueiredo. “Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors.” In: *Proc. SPIE 7446 Wavelets XIII*. Wavelets XIII. 2009, p. 74460D. DOI: 10.1117/12.826663.

bauschke:08:tpa
[272] H. Bauschke et al. “The proximal average: basic theory.” In: *SIAM J. Optim.* 19.2 (2008), 766–85. DOI: 10.1137/070687542.

bauschke:08:htt
[273] H. H. Bauschke, Y. Lucet, and M. Trienis. “How to transform one convex function continuously into another.” In: *SIAM Review* 50.1 (2008), 115–32. DOI: 10.1137/060664513.

yu:13:baa
[274] Y-L. Yu. “Better approximation and faster algorithm using the proximal average.” In: *Neural Info. Proc. Sys.* 2013. URL: http://papers.nips.cc/paper/4934-better-approximation-and-faster-algorithm-using-the-proximal-average.

mehranian:13:aos
[275] A. Mehranian et al. “An ordered-subsets proximal preconditioned gradient algorithm for edge-preserving PET image reconstruction.” In: *Med. Phys.* 40.5 (2013), p. 052503. DOI: 10.1118/1.4801898.

wang:12:plp
[276] G. Wang and J. Qi. “Penalized likelihood PET image reconstruction using patch-based edge-preserving regularization.” In: *IEEE Trans. Med. Imag.* 31.12 (Dec. 2012), 2194–204. DOI: 10.1109/TMI.2012.2211378.

yang:13:nro
[277] Z. Yang and M. Jacob. “Nonlocal regularization of inverse problems: A unified variational framework.” In: *IEEE Trans. Im. Proc.* 22.8 (Aug. 2013), 3192–203. DOI: 10.1109/TIP.2012.2216278.

perona:90:ssa
[278] P. Perona and J. Malik. “Scale-space and edge detection using anisotropic diffusion.” In: *IEEE Trans. Patt. Anal. Mach. Int.* 12.7 (July 1990), 629–39.

catte:92:iss
[279] F. Catté et al. “Image selective smoothing and edge detection by nonlinear diffusion.” In: *SIAM J. Numer. Anal.* 29.1 (1992), 182–93. DOI: 10.1137/0729012.

alvarez:92:iss
[280] L. Alvarez, P-L. Lions, and J-M. Morel. “Image selective smoothing and edge detection by nonlinear diffusion. II.” In: *SIAM J. Numer. Anal.* 29.3 (1992), 845–66. DOI: 10.1137/0729052.

weickert:98:ear

[281] J. Weickert, B. MT. H. Romeny, and M. A. Viergever. “Efficient and reliable schemes for nonlinear diffusion filtering.” In: *IEEE Trans. Im. Proc.* 7.3 (Mar. 1998), 398–410. DOI: 10.1109/83.661190.

mjolsness:90:ato

[282] E. Mjolsness and C. Garrett. “Algebraic transformations of objective functions.” In: *Neural Networks* 3.6 (1990), 651–69. DOI: 10.1016/0893-6080(90)90055-P.

rangarajan:99:rpf

[283] A. Rangarajan, H. Chui, and J. S. Duncan. “Rigid point feature registration using mutual information.” In: *Med. Im. Anal.* 3.4 (Dec. 1999), 425–40. DOI: 10.1016/S1361-8415(99)80034-6.

levenberg:1944:amf

[284] K. Levenberg. “A method for the solution of certain non-linear problems in least squares.” In: *Quart. Appl. Math.* 2.2 (July 1944), 164–8. URL: http://www.jstor.org/stable/43633451 (cit. on p. 14.60).

marquardt:63:aaf

[285] D. W. Marquardt. “An algorithm for least-squares estimation of nonlinear parameters.” In: *J. Soc. Indust. Appl. Math.* 11.2 (June 1963), 431–41. URL: http://www.jstor.org/stable/2098941 (cit. on p. 14.60).

costa:05:rgf

[286] J. Costa. “Random graphs for structure discovery in high-dimensional data.” PhD thesis. Ann Arbor, MI: Univ. of Michigan, Ann Arbor, MI, 48109-2122, 2005. URL: https://hdl.handle.net/2027.42/125332 (cit. on p. 14.60).

groenen:93

[287] P. J. F. Groenen. *The majorization approach to multidimensional scaling: some problems and extensions*. Leiden, The Netherlands: DSWO Press, 1993 (cit. on p. 14.60).

cetin:01:fes

[288] M. Cetin and W. C. Karl. “Feature-enhanced synthetic aperture radar image formation based on nonquadratic regularization.” In: *IEEE Trans. Im. Proc.* 10.4 (Apr. 2001), 623–31. DOI: 10.1109/83.913596 (cit. on p. 14.62).

cetin:02:epi

[289] M. Cetin, W. C. Karl, and A. S. Willsky. “Edge-preserving image reconstruction for coherent imaging applications.” In: *Proc. IEEE Intl. Conf. on Image Processing*. Vol. 2. 2002, 481–4. DOI: 10.1109/ICIP.2002.1039992 (cit. on p. 14.62).

sajda:04:nmf

[290] P. Sajda et al. “Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain.” In: *IEEE Trans. Med. Imag.* 23.12 (Dec. 2004), 1453–65. DOI: 10.1109/TMI.2004.834626 (cit. on p. 14.62).

hoyer:04:nnm

[291] P. O. Hoyer. “Non-negative matrix factorization with sparseness constraints.” In: *J. Mach. Learning Res.* 5 (Dec. 2004), 1457–69. URL: http://jmlr.org/papers/v5/hoyer04a.html (cit. on p. 14.62).

jolliffe:02

[292] I. T. Jolliffe. *Principal component analysis*. Springer, 2002. DOI: 10.1007/b98835 (cit. on p. 14.62).

figueiredo:07:mma

[293] M. A. T. Figueiredo, Jose M Bioucas-Dias, and R. D. Nowak. “Majorization-minimization algorithms for wavelet-based image restoration.” In: *IEEE Trans. Im. Proc.* 16.12 (Dec. 2007), 2980–91. DOI: 10.1109/TIP.2007.909318 (cit. on p. 14.62).

bioucasdias:06:bwb

[294] J. M. Bioucas-Dias. “Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors.” In: *IEEE Trans. Im. Proc.* 15.4 (Apr. 2006), 937–51. DOI: 10.1109/TIP.2005.863972 (cit. on p. 14.62).

schultz:94:aba

[295] R. R. Schultz and R. L. Stevenson. “A Bayesian approach to image expansion for improved definition.” In: *IEEE Trans. Im. Proc.* 3.3 (May 1994), 233–42. DOI: 10.1109/83.287017 (cit. on p. 14.62).

raj:07:fas

[296] A. Raj and K. Thakur. “Fast and stable Bayesian image expansion using sparse edge priors.” In: *IEEE Trans. Im. Proc.* 16.4 (Apr. 2007), 1073–84. DOI: 10.1109/TIP.2006.891339 (cit. on p. 14.62).