SURVEY OF ADDITIONAL TOPICS

$\boxed{\text{Matrix Factorization}}$

$X \approx A \cdot B$, X is the (centered) data matrix

1) PCA

$$\min_{A,B} \|X - A \cdot B\|_F^2$$

s.t. $A \in \mathbb{R}^{d \times k}$
$B \in \mathbb{R}^{k \times n}$
$A^T A = I$

2) k-means

$$\min_{A,B} \|X - A \cdot B\|_F^2$$

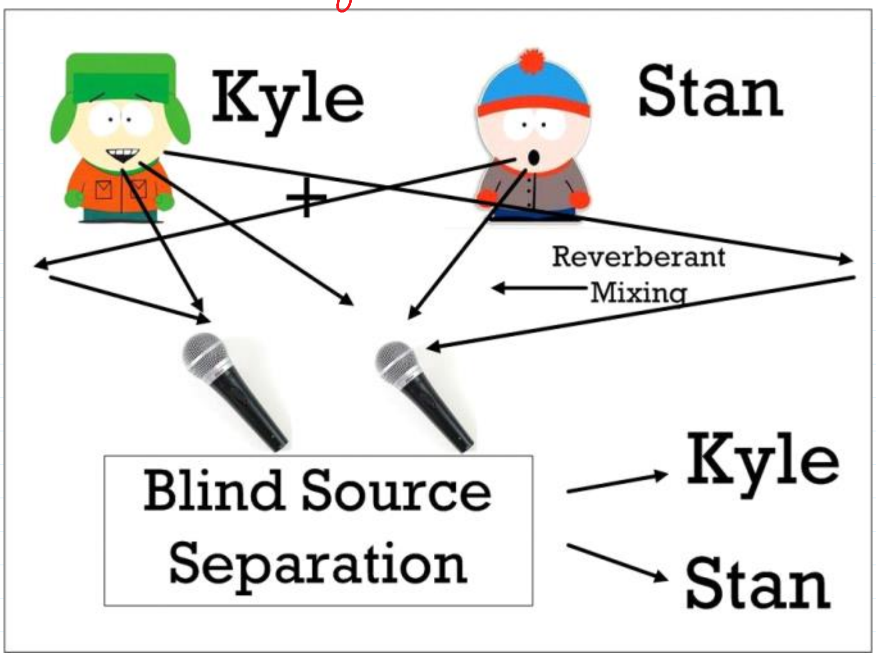
s.t. $A \in \mathbb{R}^{d \times k}$
$B \in \mathbb{R}^{k \times n}$
columns of B are indicator vectors

3) Independent component analysis (ICA)

$$X \approx \underset{d \times d}{A} \cdot \underset{d \times n}{S}$$

$S = [s_i]$ such that for each t, $s_{1t}, \dots, s_{dt}$ are realizations of independent RVs.

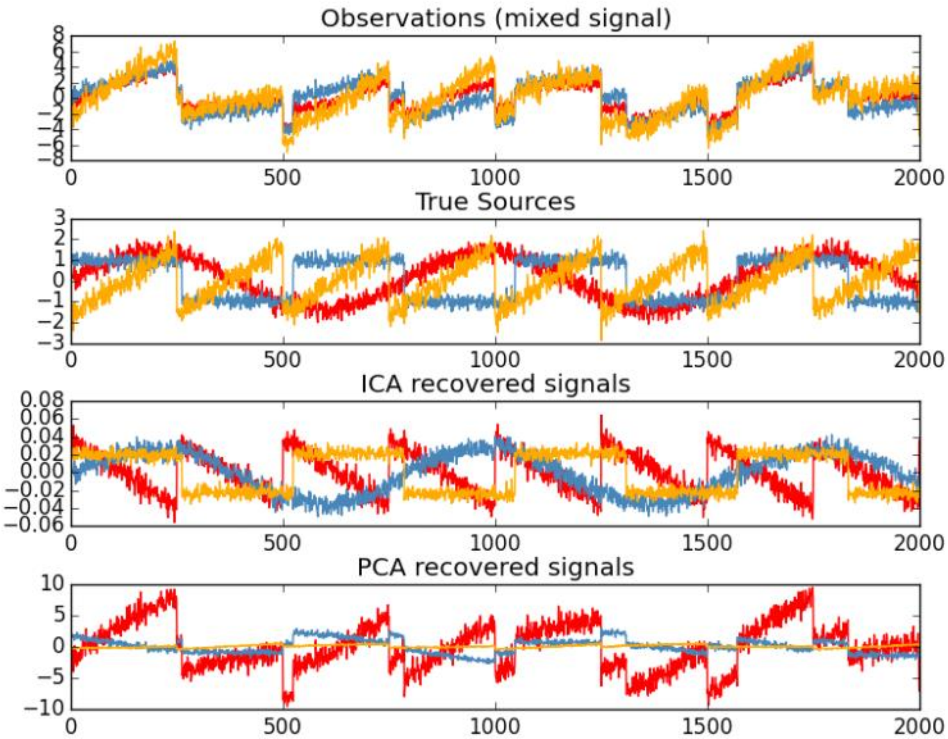
"Cocktail Party Problem"



$X = [x_{it}]$

x_{it} = mic i measurement at time t

s_{it} = speaker i speech signal at time t



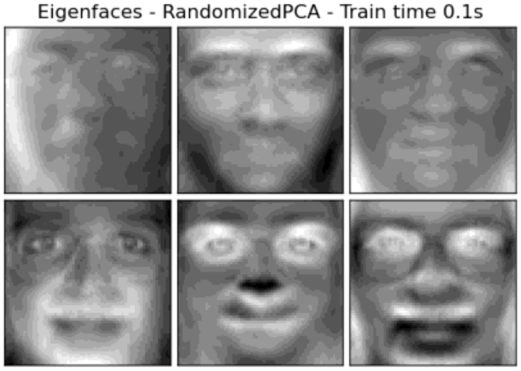
4) Nonnegative matrix factorization (NMF)

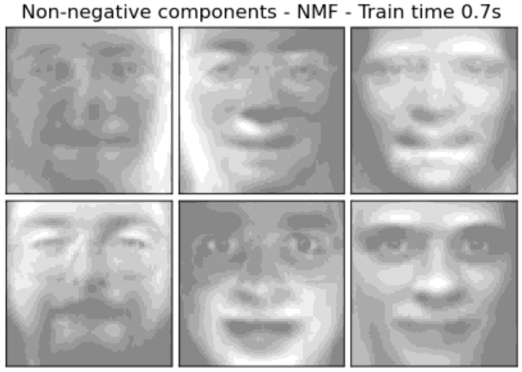
$$\min \| X - A \cdot B \|_F^2$$

s.t. $A \in \mathbb{R}^{d \times k}$

$B \in \mathbb{R}^{k \times n}$

elements of A, B are nonnegative.





5) Sparse coding / dictionary learning

$$\min_{D,A} \| X - D \cdot A \|_F^2$$

s.t. $D \in \mathbb{R}^{d \times s}$ $\quad (s > d)$

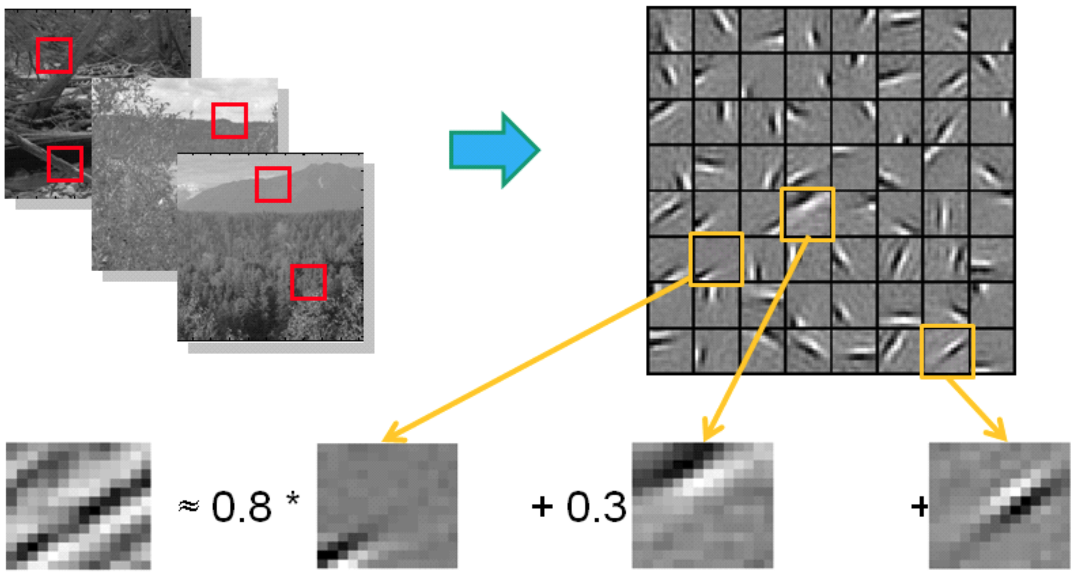
$A \in \mathbb{R}^{s \times n}$

columns of D have unit norm

columns of A sparse

Intuitively, find a set of components (dictionary columns) such that every column of X is explained as a superposition of a small number of components.

Sparse coding illustration



$[a_1, \ldots, a_{64}] = [0, 0, \ldots, 0, \mathbf{0.8}, 0, \ldots, 0, \mathbf{0.3}, 0, \ldots, 0, \mathbf{0.5}, 0]$
(feature representation)

Slide credit: Andrew Ng

Compact & easily interpretable

Algorithmic strategy: alternating minimization

7) Matrix completion

$X = [x_{ij}]$, $\Omega \subseteq \{1, \ldots, d\} \times \{1, \ldots, n\}$

$(d \times n)$

x_{ij} is only observed for $(i,j) \in \Omega$

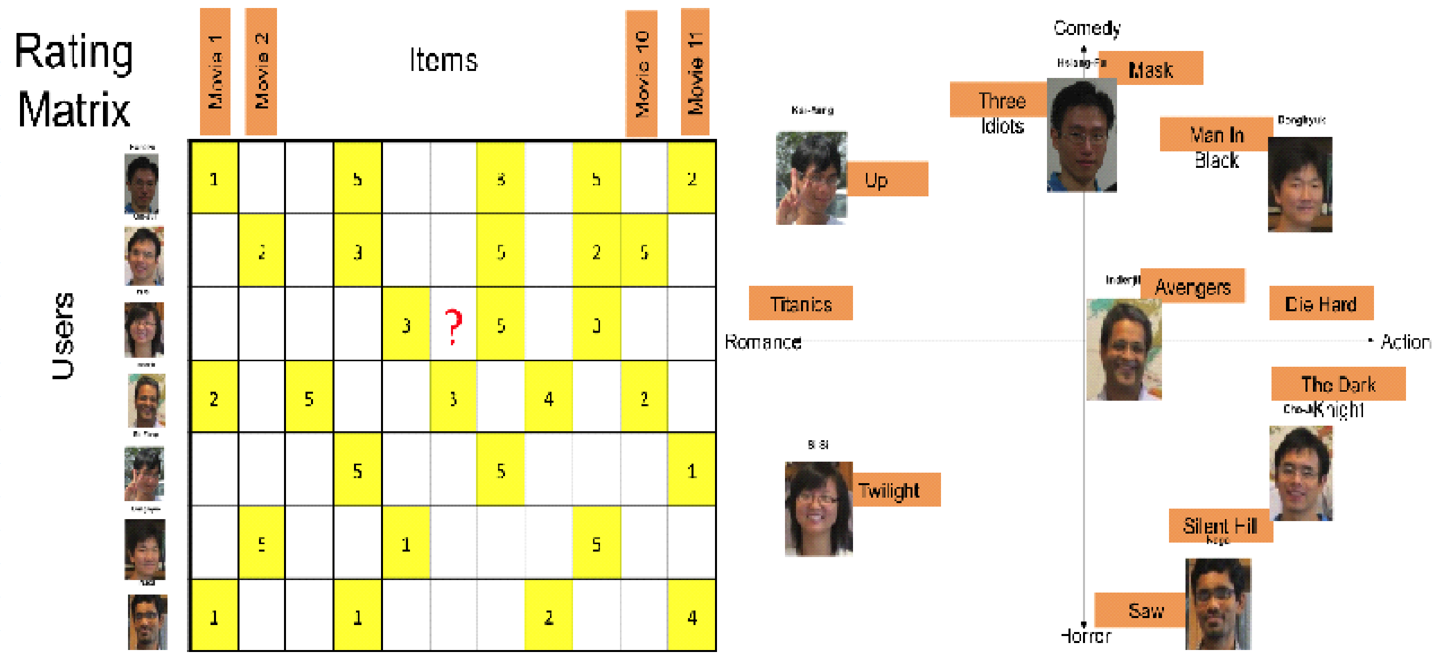
Basic approach: assume X has rank $r < \min(d,n)$.

$$\min_{A,B} \| X - A \cdot B \|_{F,\Omega}^2$$ ← sum of squares of entries indexed by Ω

$$\min_{A,B} \| X - AB \|_{F,\Omega}^2$$ entries indexed by Ω

s.t. $A \in \mathbb{R}^{d \times r}$

$B \in \mathbb{R}^{r \times n}$



8) Sparse PCA (θ_i's constrained to be sparse)

9) Probabilistic PCA : generative model whose maximum likelihood estimate coincides with PCA. Useful for extending PCA to

- missing data
- mixture models

10) Factor analysis : slightly more flexible generative model

relative to PPCA.

ii) Latent semantic indexing: Use PCA/SVD to get low rank approximation of X, where columns of X correspond to documents, rows to words in a vocabulary, and entries of X are word counts.

$\boxed{\text{Nuclear Norm Regularization}}$

Let $X \in \mathbb{R}^{d \times n}$ be a data matrix. Suppose we seek ~~a~~ the best rank r approximation to X. Then we know to just apply PCA/SVD. But what if the true r is unknown?

One option is to solve

$$\min_{W \in \mathbb{R}^{d \times n}} \|X - W\|_F^2 + \lambda \cdot \operatorname{rank}(W)$$

However, the rank function is nonconvex. Analogous to how the ℓ_1 norm is a convex proxy for the sparsity of a vector, the <u>nuclear norm</u>,

$$\|W\|_* := \sum \sigma_i \qquad \text{(sum of singular values)}$$

is the tightest convex relaxation of rank. This leads to

$$\min_{W \in \mathbb{R}^{d \times n}} \|X - W\|_F^2 + \lambda \cdot \|W\|_*$$

which is now a convex problem. It can be solved using ADMM where the prox operator for the nuclear norm is given by singular value thresholding = soft thresholding applied to the singular values of the argument.

For matrix completion, one solves

$$\min_{W} \|X - W\|_{F,\Omega}^2 + \lambda \|W\|_*$$

This approach yields a global minimum, unlike the alternating algorithm mentioned earlier.

As another application, consider robust PCA:

$$\min \|X - W\|_F^2 + \lambda \|L\|_* + \gamma \|S\|_{1,1}$$

$$\text{s.t.} \quad W = L + S$$

sum of

s.t. $W = L + S$

sum of absolute values of all entries

S corresponds to outliers, and
L gives the low dim. representation.
(just apply standard PCA to L).

Group Lasso

Recall that the ℓ_1 or "lasso" penalty promotes sparsity and is useful for feature selection. The "group lasso" penalty is useful for group feature selection.

Consider a prediction problem (classification or regression) where the features can be naturally grouped.

Example: In classification of brain images, groups of pixels correspond to anatomical units (e.g., hippocampus, visual cortex)

Let $G_1, \ldots, G_m$ be a partition of $\{1, \ldots, d\}$, so that

- $G_r \cap G_s = \emptyset$ if $r \neq s$
- $\bigcup_{r=1}^{m} G_r = \{1, \ldots, d\}$.

Let w_G denote the vector w restricted to features in G, e.g.,

$$w = \begin{bmatrix} 11 \\ -4 \\ -1 \\ 17 \\ 8 \end{bmatrix}, \quad G = \{2,5\} \implies w_G = \begin{bmatrix} -4 \\ 8 \end{bmatrix}.$$

The group lasso penalty is $\sum_{r=1}^{M} \|w_{G_r}\|_2$. Therefore, to perform linear regression with group feature selection, we would solve

$$\min_{w} \frac{1}{n}\sum_{i=1}^{n} (y_i - w^T x_i - b)^2 + \sum_r \|w_{G_r}\|_2 .$$

The intuition is that $\sum \|w_{G_r}\|$ can be viewed as the l_1 norm of $(\|w_{G_1}\|_2, \ldots, \|w_{G_M}\|_2)$, which encourages most values of $\|w_{G_r}\|_2$ to be zero, i.e., w_{G_r} = zero vector.

Multiclass SVM

One way to define a linear SVM in the multiclass case is

$$f(x) = \arg\max_{k=1,\dots,K} \langle w_k, x \rangle$$

where w_k is associated with class k, and solves

$$\min_{w_1,\dots,w_K} \frac{1}{2}\sum_{k=1}^{K} \|w_k\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i$$

$$\text{s.t.} \quad \langle w_{y_i} - w_k, x_i \rangle \geq 1 - \xi_i \qquad \forall i, \forall k \neq y_i$$

$$\xi_i \geq 0 \qquad \forall i$$

The above formulation can be kernelized using the dual optimization problem.

Q: How could we incorporate embedded feature selection into the linear multiclass SVM?

A: Group lasso penalty where groups correspond to features

$\boxed{\text{Multitask Learning}}$

Suppose there are N different (but possibly related) classification problems, referred to as tasks, and let

$$\{(x_j^{(i)}, y_j^{(i)}) \mid j = 1, \dots, n_i\}$$

be training data for the i^{th} task.

In multi-task learning, the goal is to learn the N classifiers simultaneously, in hopes that if some tasks are sufficiently similar, training data can be pooled, thus leading to a larger effective sample size for some or all tasks.

Let's consider the linear case. Let $w^{(i)} \in \mathbb{R}^d$ be the parameter associated with task i, and write

$$W = \begin{bmatrix} w^{(1)} & \cdots & w^{(N)} \end{bmatrix} = \begin{bmatrix} w_1^T \\ \vdots \\ w_d^T \end{bmatrix} \quad (d \times n)$$

A basic approach is to solve

$$\min_W \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(y_i, \langle w_j^{(i)}, x_i \rangle) + \lambda R(W)$$

$$\min_{W} \quad \frac{1}{N}\sum_{i=1} \frac{1}{n_i}\sum_{j=1} \ell(y_i, \langle w_j, x_i \rangle) + \lambda R(W)$$

where R is a regularizer that encourages $w^{(1)}, \dots, w^{(N)}$ to be similar. Can you suggest a good R?

Here are some possibilities:

- shared mean:

$$R(W) = \sum_{i=1}^{N} \left\| w^{(i)} - \frac{1}{N}\sum_{k=1}^{N} w^{(k)} \right\|_2^2$$

- nuclear norm :

$$R(W) = \|W\|_*$$

- group lasso :

$$R(W) = \sum_{\ell=1}^{d} \|W_\ell\|_2$$

For which of the above regularizers can the method be kernelized?