Optimization Methods for Signal and Image Processing
(Lecture notes for EECS 598-006)

Jeff Fessler
University of Michigan

January 7, 2019

Chapter 0

EECS 598 Course introduction: w19

Contents (class version)

These notes were typeset using LaTeX. One way to learn LaTeX is to use http://overleaf.com.

0.1 Course logistics

EECS 598-006: Optimization Methods for Signal and Image Processing 3 credits

Lecture: Tue,Thu 9-10:30AM, 133 Chrysler

Instructor: Prof. Jeff Fessler fessler@umich.edu `https://web.eecs.umich.edu/~fessler/`
Office hours: Wed 10-11:50AM, 4431 EECS.
Include `[eecs598-w19]` in email subject for possibly less slow response.
Use Canvas/Piazza when possible.

GSI (office hours held in 3312 EECS):

- Cameron Blocker cblocker@umich.edu `https://cameronblocker.com/`
  Mon 1:30-2:30 Tue 10:30-11:30 Thu 2:00-3:00

Course materials: **Action**: bookmark these links.

- Primary site is Canvas: `https://umich.instructure.com/courses/276189`
  (homework, solutions, lecture notes, announcements, etc.)
- Annotated versions of class notes:
  `https://tinyurl.com/w19-598-opt-lecture`
- Secondary site (demos, back-ups):
  `http://web.eecs.umich.edu/~fessler/course/598`

Course goals

Modern signal and image processing methods for numerous applications are often formulated as optimization problems. This course is a sequel to EECS 551 that builds on the matrix methods therein to provide both the mathematical foundation and hands-on experience with optimization methods for modern signal and image processing applications.

EECS 551 considers several signal procession optimization problems:

- least-squares problems - solved with pseudo-inverse / SVD for small problems, or by (**preconditioned**) **gradient descent** (**GD**)/(**PGD**) iterations for large problems;
- classification using nearest subspace, subspace learned by SVD;
- orthogonal Procrustes problem - solved by SVD;
- low-rank matrix approximation in many variants - solved by SVD;
- $\arg\max_{\boldsymbol{x} \neq \boldsymbol{0}} \|\boldsymbol{A}\boldsymbol{x}\| / \|\boldsymbol{x}\|$ solved by SVD for small problems or by **power iteration**
- binary classifier design using logistic regression - using GD iteration;
- low-rank matrix completion - using **majorize-minimize** (**MM**) iteration.

Most of the above problems have SVD-based solutions, but SVD is impractical for large-scale problems.

And there are many other applications involving cost functions (such as logistic regression and matrix completion) where there is no closed-form solution and iterative algorithms are essential.

A better title for this course might be:

"signal and image processing problems that require iterative optimization algorithms to solve."

The goal of this course is to bridge the gap between EECS 551 and the modern signal processing literature that is replete with optimization approaches. The focus will be on large-scale problems where some off-the-shelf optimization algorithms may be too slow or use too much memory to be practical.

Related courses

- EECS 600 (Function Space Methods for Systems Theory) is much more theoretical than this course because it deals with infinite dimensional spaces, whereas 598 will focus completely on finite-dimensional problems. 600 is far more proof oriented than 598, but there will be some proofs presented and expected in 598 as well.
- IOE 511/Math562 (Continuous Optimization Methods) has quite a bit of overlap in terms of the optimization methods. IOE 511 uses Matlab. 598 will focus on signal and image processing applications.
- IOE 611/Math663 (Nonlinear Programming) covers very important Convex Optimization principles. It uses the CVX package in Matlab which does not scale well for large problems. 598 focuses more on large-scale signal and image processing applications.
- EECS 556 (Image Processing) introduces some of the applications (e.g., image deblurring) that will be considered as examples in 598. So there will be some overlap with past offerings of EECS 556, as well as the other courses listed above, but it is fine for students to take this course and also any or all of EECS 556, EECS 600 and IOE 611. Note that the IOE/Math courses can serve as a cognate for ECE students but 598 cannot.

Prerequisites

- EECS 551 (Matrix methods for...) or Prof. Nadakuditi's similar EECS 598 course
  An ordinary linear algebra course is insufficient!
  EECS 551 notes from F18 here:
  http://web.eecs.umich.edu/~fessler/course/551/f18-eecs551-notes-final.pdf
- EECS 501 helpful but not essential
- JULIA coding experience at level of EECS 551.

Exams (other ECE exams)

Midterm Exam: Wed. Feb. 27 6-8PM, Room 1303, 1500 EECS
Final Exam: ~~Wed. May 1, 10:30am – 12:30pm~~, Room TBA Thu. Apr. 25, 1:30-3:30 PM

Grades

Homework Written problems submitted to gradescope Autograder problems Jupyter notebooks Perusall annotations	30%
clicker	5%
Midterm	30%
Final exam	35%

Final grade cutoffs will be 90/80/70% or lower.
Exam scores may be standardized.

Honor code

The UM College of Engineering Honor Code applies. You should be familiar with it.
See https://ossa.engin.umich.edu/honor-council/ for details.
See collaboration policies below.

Homework

Typically due on Thursday at 4PM. Typically an automatic extension to Friday at 4PM. No further.
Submit scans of solutions to https://gradescope.com. (HW1 on Canvas!)
Hopefully will be graded and "returned" via gradescope within a week.
Written regrade requests via gradescope within 3 days of return date.

Actions:

- Check for your name on gradescope (should be there thanks to Canvas integration)
- Review gradescope scan/pdf submission process. There are also video instructions.

Collaboration policy: Homework assignments are to be completed on your own. You are allowed to consult with other students (and instructors) during the conceptualization of a solution, but all written work, whether in scrap or final form, is to be generated by you, working alone. Also, you are not allowed to use, or in any way derive advantage from, the existence of solutions prepared in prior years. Violation of this policy is an honor code violation. If you have questions about this policy, please contact me. While collaboration can sometimes be helpful to learning, if overused, it can inhibit the development of your problem solving skills.

Homework grading

Homework grading is constrained by GEO union policies. See:
http://web.eecs.umich.edu/~fessler/course/551/r/grading,geo.txt
http://web.eecs.umich.edu/~fessler/course/551/r/grader-duties.pdf

Manually graded problems will be on a scale of 0-3:

0 No solution was attempted
1 A solution was attempted but the approach used did not recognizably conform to any in the solution set
2 The approach used recognizably conformed to one in the solution set, but the answer was incorrect.
3 A solution approach recognizably conformed to one in the solution set, and the answer was correct.

JULIA-based autograder problems (details on HW1) typically will be 10 points each (10 or 0).

Ethics

Sharing any materials from this class with other individuals not in the class without written instructor permission will be treated as an Honor Code violation. Posting your own solutions (including code) on public sites like github.com is also prohibited. Keep your materials private! In particular, uploading any materials from this class to web sites akin to `coursehero.com` will be reported to the Honor Council.

Missing class

- Classes are captured/recorded and viewable on Canvas.
- Missed clicker questions and other in-class activities cannot be made up.
- Annotated notes are available online.

Mental health resources

As a student you may experience a range of issues that can cause barriers to learning, such as strained relationships, increased anxiety, alcohol/drug problems, feeling down, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may lead to diminished academic performance or reduce a student's ability to participate in daily activities. The University of Michigan is committed to advancing the mental health and well-being of its students. If you or someone you know is feeling overwhelmed, depressed, and/or in need of support, services are available. You can learn more about the broad range of confidential mental health services available on campus via `http://umich.edu/~mhealth`.

Books and other resources

Useful reference: Boyd & Vandenberghe, "Convex optimization" [1]
Free pdf available `http://www.stanford.edu/~boyd/cvxbook.html`

Books that may be useful: [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19]

Clickers

`http://caenfaq.engin.umich.edu/10909-clickers/` Bring batteries!
Action: Buy at `http://computershowcase.umich.edu/remotes/` ($29 used, buy back for $19)
Action: Register your clicker at Canvas.
Clicker question scoring: 2 points for answering, 3 points for correct answer. (Learning, not assessment.)

Why? Because **active learning** "has been definitively shown to be superior to lectures in promoting both comprehension and memory."

Annotations

Many weeks will involve reading assignments from the literature, posted on `perusall.com`, that you will access via Canvas for reading and for annotating the text with questions and comments.

Your annotations will be visible to the entire class and instructor. You can clarify confusing concepts, fill in missing steps in derivations, flag errors, provide better illustrations of the ideas, ask great questions, and answer (correctly) other's questions.

Annotations will be graded based on quantity, quality and timeliness, where the required quantity will depend on the assignment length, and quality will be measured by Perusall's algorithms. More on that on the first reading assignment. This is really about learning, not assessment, so genuine effort is what matters.

Why? Annotating the text helps you and us. First, you get practice reading technical material. Once you graduate, books (and papers) will be your primary vehicle for learning and learning does not stop when you graduate. Learning from the literature is an important lifelong skill. Second, by reading with attention and with an inquiring mind, you take ownership of your learning. Third, by annotating the text, you reverse the roles of student and teacher: for a change you are the one determining what's wrong or confusing. In a traditional class, the teacher tells you what is wrong or confusing about your work. When you annotate the text because you are confused, you have identified a problem in the text: you are right and the author is wrong! By communicating that confusion to others, you create an opportunity to address the confusion and learn. If many people in the class express confusion about a particular topic, we will know that we need to address that confusion in class or online.

Your goal in annotating is demonstrating substantive, thorough, timely, and thoughtful reading.

- ◦ Insufficient: "This confuses me"
- ◦ Better: "This equation appears to contradict (previous equation) or seems counter-intuitive because ..."
- ◦ Insufficient: Yes/No answers to questions without explanation.
- ◦ For more examples, see [this link]

Again, the purpose of these annotations is active learning.

Julia language

All code examples and homework code templates will use the **Julia programming language**.

To see why, and for pointers to tutorials and documentation and getting started instructions, see the Ch. 0 lecture notes from EECS 551 in F18 here:
`http://web.eecs.umich.edu/~fessler/course/551/l/n-00-intro.pdf`

Everything will use JULIA 1.0, including the autograder.
Beware of online Q/A for older versions of JULIA!

JULIA 0.7 is basically the same as JULIA 1.0 except that 0.7 gives deprecation warnings for obsolete usage.

For those who used JULIA 0.6 (or earlier versions) previously, see list of key changes between 0.6 and 1.0 at:
`http://web.eecs.umich.edu/~fessler/course/551/julia/changes.txt`

Action: install JULIA 1.0 (and 0.7 too if you have used only 0.6 previously)

For a YouTube tutorial see `https://www.youtube.com/watch?v=4igzy3bGVkQ`

`https://eecs556.autograder.eecs.umich.edu`
`mailto:eecs556@autograder.eecs.umich.edu`

(We use 556 number to avoid confusion with other 598 sections.)

A'

A.transpose()

PDF lecture note features

These notes highlight some important **terms** in red.
Many important terms have links in the pdf documents to **Wikipedia** in violet. Some links look like: [wiki]
Those links are clickable in the pdf and should cause your browser to open at the appropriate url.

Define. Key definitions are shaded like this.

Particularly important topics are shaded like this.

JULIA `code is shaded like this.`

Boxes with this color need completion during class.

A road hazard or **"dangerous bend" symbol** in the margin warns of tricky material.

A double diamond symbol is "experts only" material included for reference that is likely beyond the scope of the exams. ♦♦

These notes are not a textbook; they are designed for classroom use. There will be other reading assignments on Perusall to supplement these notes.

These notes are formatted with 16:10 aspect ratio to match the projector in the lecture room; that format is well-suited for printing two slides per paper side. If you print, please save paper by using that option.
Action: Print the next chapter (not this one) or bring pdf to class on a suitable device for annotating.

0.2 Course topics

Applications discussed will include: sparse coding, sparse linear regression, compressed sensing, image denoising, image deblurring, phase retrieval, matrix completion, matrix sensing, dictionary learning, sparsifying transform learning, biconvex problems like blind deconvolution and matrix factorization.

Optimization methods discussed and implemented will include: gradient methods (gradient descent, BB, fast gradient descent, preconditioned steepest descent and conjugate gradients), second-order methods briefly (Newton, quasi-Newton, LBFGS), stochastic gradient descent / incremental gradient methods, sketching methods, coordinate descent and block coordinate descent methods, projection onto convex sets (POCS), sketching methods, proximal methods (ISTA/PGM, FISTA/FPGM, POGM fast proximal gradient methods), splitting methods (Augmented Lagrangian methods and ADMM), majorize-minimize (MM) methods, Frank-Wolfe and primal-dual methods.

Every method introduced will have at least one signal / image processing application where it is applicable.

Optimization concepts will include: convexity, Lipschitz continuity, convex relaxation, and duality.

Bibliography

[1] S. Boyd and L. Vandenberghe. *Convex optimization*. UK: Cambridge, 2004.

[2] D. P. Bertsekas. *Convex analysis and optimization*. Athena, 2003.

[3] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. New York: Academic-Press, 1982.

[4] D. P. Bertsekas. *Nonlinear programming*. 2nd ed. Belmont: Athena Scientific, 1999.

[5] A. Beck. *First-order methods in optimization*. Soc. Indust. Appl. Math., 2017.

[6] T. F. Coleman and Y. Li. *Large-scale numerical optimization*. Philadelphia: Soc. Indust. Appl. Math., 1990.

[7] C. T. Kelley. *Iterative methods for optimization*. Soc. Indust. Appl. Math., 1999.

[8] B. T. Polyak. *Introduction to optimization*. New York: Optimization Software Inc, 1987.

[9] A. Ben-Tal and A. Nemirovski. *Lecture on modern convex optimization: Analysis, algorithms, and engineering applications*. Philadelphia: Soc. Indust. Appl. Math., 2001.

[10] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Kluwer, 2004.

[11] J. Nocedal and S. J. Wright. *Numerical optimization*. New York: Springer, 1999.

[12] A. Nemirovski and D. Yudin. *Problem complexity and method efficiency in optimization*. wiley, 1983.

[13] E. Polak. *Optimization. Algorithms and consistent approximations*. Philadelphia: Soc. Indust. Appl. Math., 1997.

[14] S. Sra, S. Nowozin, and S. J. Wright. *Optimization for machine learning*. Cambridge: MIT, 2011.

[15] J. M. Borwein and A. S. Lewis. *Convex analysis and nonlinear optimization*. New York: Springer, 2000.

[16] K. Lange. *Optimization*. Springer, 2013.

[17] K. Lange. *MM optimization algorithms*. Soc. Indust. Appl. Math., 2016.

[18] Y. Nesterov. *Introductory lectures on convex optimization*. Springer, 2008.

[19] D. P. Palomar and Y. C. Eldar. *Convex optimization in signal processing and communications*. Cambridge, 2011.

[20] M-Z. Poh, N. C. Swenson, and R. W. Picard. “A wearable sensor for unobtrusive, long-term assessment of electrodermal activity”. In: *IEEE Trans. Biomed. Engin.* 57.5 (May 2010), 1243–52.

[21] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. “Julia: A fresh approach to numerical computing”. In: *SIAM Review* 59.1 (2017), 65–98.

[22] M. Innes. “Flux: Elegant machine learning with Julia”. In: *J. of Open Source Software* 3.25 (2018), p. 602.

[23] T. Besard, C. Foket, and B. De Sutter. “Effective extensible programming: unleashing Julia on GPUs”. In: *IEEE Trans. Parallel. Dist. Sys.* (2018).

[24] A. J. Laub. *Matrix analysis for scientists and engineers*. Soc. Indust. Appl. Math., 2005.