CS 169/268 Introduction to Optimization

Syllabus, Fall 2015

Instructor: Eric Mjolsness

emj@uci.edu

Rough outline of topics:

1. **Introduction to (introduction to) optimization**
    a. **1D unconstrained opt methods**
    b. **problem definitions**
        i. **zoo of opt problems**
            1. **discrete vs. continuous *(& interior point methods)***
            2. **local vs. global opt; local vs. global convergence**
            3. **unconstrained vs. convex vs. constrained**
        ii. **structure & topology of spaces (domain, range, function spaces)**
        iii. **special cases in opt:**
            1. **linearity, quadraticality, convexity, sampled objectives, …**
            2. **integer- or discrete-valued solutions**
            3. **combinatorial optimization**
2. **Unconstrained optimization**
    a. **optimality conditions**
    b. **nonderivative methods**
        i. **Nelder-Mead, Simulated Annealing, Genetic Alg.s, Diff. Evol., …**
    c. **convergence rates & condition number**
    d. **gradient methods (conjugate gradients; block coordinate descent; …)**
    e. **Newton & quasi-Newton methods**
    f. **multiscale/multigrid methods**
3. **Equality-constrained optimization**
    a. **optimality conditions: Lagrange multipliers**

    b. gradient projection
    c. augmented Lagrangian method & lasso
4. Discrete optimization
    a. combinatorial optimization
        i. Linear programming: Simplex and Interior Point methods
        ii. branch and bound
        iii. linear & quadratic assignment
        iv. computational complexity & NP-completeness
    b. interior-point methods
5. Inequality-constrained optimization
    a. optimality conditions: Kuhn-Tucker
    b. barrier & penalty methods
    c. duality
    d. gradient methods eg. active sets
    e. convex optimization
6. Application areas
    a. logistics & operations research
    b. mechanical & electrical engineering
    c. machine learning
    d. computer vision
    e. robotic planning, at multiple levels
7. Advanced topics (may or may not get here)
    a. Nondifferentiable problems:
        i. subgradient methods
        ii. cutting planes
    b. Trust region methods
    c. application class: Finite Element Method

d. **Algebraic multigrid**

e. **scaling up**

Assignments and Grading:

For undergraduates : There will be a several homework sets worth 25%, quizzes worth 15%, a midterm exam worth 30% and a group project worth 30%.

For graduate students: There will be a several homework sets worth 30%, quizzes worth 10%, a midterm exam worth 30% and a group project worth 30%. The graduate student work will be more extensive by about x2, and more advanced.

The Group Projects will be described in a separate document.

Midterm exam: Tuesday, November 3, in class. Note that this is the first class after the end of Daylight Savings time.

Final Projects due: Roughly, at the regularly scheduled Final Exam time for this class.

References

Everyone should get access to at least one good optimization book, somehow. Here (below) are some possibilities.

Strongly Recommended:

A. Belegundu & T. Chandrupatla, Optimization Concepts and Applications in Engineering, 2nd ed. Cambridge U. Press.

Optional:

D. Bertsekas, Nonlinear Programming, 2nd ed. Athena Scientific. (For more serious students of numerical optimization.)

R. Baldick, Applied Optimization, Cambridge U. Press. (More elementary and less complete, but contains many good examples.)

Alternatives and background reading on special topics:

S. Boyd, Convex Optimization. (Somewhat specialized to … convex optimization.)

K. Lange, Optimization, Springer 2nd ed. *(A statistics-oriented viewpoint on optimization. Useful treatments of analysis (Ch 2), EM methods (Ch 8-9), Lasso (Ch 13), and calculus of variations (ch 17).)*

C. Papadimitriou, Combinatorial Optimization. (Somewhat specialized to … combinatorial optimization; also a bit dated.)

D. Luenberger, Linear and Nonlinear Programming. (A bit dated, but classic.)

W. Press et al., Numerical Recipes, Cambridge U. Press. (***Warning***: Contains lots of actual, useful working code - and therefore must *not* be consulted on some but not all of our homeworks or homework problems!! But generally you get a chance later on to compare with good external code. All of it is available online.)

Deeper theory background:

D. Luenberger, Optimization by Vector Space Methods, Wiley Interscience Press. (What we are *really* doing in optimization is working in various function spaces.)

Software resources

Mathematica, Matlab, R, … mathematical Problem-Solving Environments (PSEs) have built-in optimization capabilities. Many open source codes also exist. But sometimes you will be asked to make your own implementation, from zero starting code; only afterwards is it fair to compare it with other implementations. Generally you may use small amounts of non-executable pseudocode if you cite them. Whether you can also use executable code (eg in PSEs) and/or read other people's source code depends on the individual assignment, and it must always be with full attribution.

("Code" is at least: Any expression in any formal language L which can be compiled into or interactively interpreted as a computer program by software specific to language L. Pseudocode looks like code and/or mathematical notation, but is either not formalized, or cannot be automatically compiled or interpreted by any language-specific software you have access to.)