

EE364b Projects

John Duchi

April 18, 2018

The EE364b project is meant to involve some combination of (but not necessarily all of) independent research, implementation and testing, simulation and verification, and documentation. You can propose anything you like as a project; the descriptions below are meant only as broad categories and generic examples. We've also made some suggestions for projects on the course webpage.

All of your project submissions should be submitted to Gradescope.

1 Project requirements

The formal requirements for the project are: initial proposal, midterm peer review, midterm progress report, poster session, and final report. The idea is that you *apply* to do a project; the initial proposal and midterm progress report needs to be approved. We may not approve all initial projects; if your project is not approved, this will not effect your grade in any way in the course, as it simply means that you will take the final exam instead of doing a project. Your grade on the project will be primarily based on the poster and the final report.

To help you with the process, we ask you go undergo *peer review*, in which you exchange your drafts and provide feedback, before the midterm submission. The peer review is done electronically.

1.1 Detailed logistics

Here we outline the detailed logistics in (roughly) chronological order. Please read carefully to avoid confusion.

- You will submit your initial proposal, which is a maximum of 2 pages (with an additional page, if needed, for references) by Thursday, 4/26/18 at 5pm. Submit your project on Gradescope.
- The initial proposal should include the topic, team members, and some ideas about the approach. Include some background, basic problem, and references.
- By Sunday, 4/29/18 we will return our feedback and tell you whether the project is accepted.

- The midterm progress report will require somewhat more care and planning. We will provide a Google form a few days before the peer review for the midterm progress report. Fill out the Google form, which we will provide a few days before the peer review, to let us know that your group is planning to submit. Based on this, we will randomly assign 3 groups into a *peer review team*.
- By Monday 5/14/18, 8am, write a draft of your midterm progress report and email it to every member of your peer review team. You should cc ee364b.submission@gmail.com.
- By Wednesday 5/16/18, 8am, email back the peer reviews. This email should cc ee364b.submission@gmail.com. The peer reviews are written individually and not as a group.
- As a group, rate the peer reviews on a scale of 0–2.
 - 0/2. The review is not useful and does not demonstrate that the reviewer read your draft.
 - 1/2. The review is not useful but does demonstrate that the reviewer read your draft.
 - 2/2. The review is either useful *or* demonstrates that the reviewer made a sincere effort.

Remark. Keep in mind that a review that demonstrates misunderstanding can be a useful review; it can show why your communication was unclear.

- By Friday 5/18/18, 5pm, submit the report, all received peer reviews, and the ratings of these reviews on Gradescope.
- By the time you write the midterm progress report, you should have a good idea of your approach, with a clear problem statement and some preliminary results. Length: 4 pages maximum.
- By Tuesday 5/22/18, we will return our feedback and tell you whether the project is making acceptable progress. We reserve the right to reject a project based on the midterm progress report, even if it passed the initial proposal.
- *This date subject to change, but probably accurate.* On Friday, June 1, 5–7pm, you will present a poster in the Packard Atrium. If you are presenting, please come to Packard to collect the board and easel by 4:30pm. You are responsible for printing the poster. Boards and easels for the posters will be provided. The boards measure roughly 24" by 36".
- By Wednesday, June 6, 5pm, submit your final report on Gradescope, and turn in a physical copy into the submission box in Packard in the second-floor hall behind the kitchen. Length: 6 pages maximum.

1.2 Remarks

- Do not be shy to utilize Prof. Duchi or Yair; as soon as you have a project idea, talk to one of us for at least 5 minutes.
- All reports must be written using L^AT_EX in a style that follows our template and submitted as a physical copy. *No proprietary formats (word, powerpoint) will be accepted.*
- Your grade will be primarily based on the poster and the final report. We may deduct points for providing poor peer reviews, although to do so is not our intent.
- All page limits exclude references but include everything else, such as acknowledgements or figures.
- At any point, you may decide to drop the project and take the final or go Cr/NCr. You do not have to notify us of this. Just don't submit. However, if you were assigned to a peer review team, you are still responsible for providing feedback to the other groups. In other words, you will give but not ask for feedback.
- We ask you to cc your peer review transactions for the purpose of keeping a timestamp. Don't bother to email the ratings of the received reviews.

1.3 Peer review Instruction

Your job as a reviewer is to help your peers' report get accepted. Be encouraging and constructive.

When reviewing the midterm progress report, start off by summarizing the entire report in a single paragraph. This helps the authors understand how the message is conveyed to the reader. This isn't necessary for reviewing the initial proposal.

The following are some possible points you can comment on in the review.

- *Writing style.* You can find more information on this in the L^AT_EX template.
- *Clarity.* You, a student who has taken EE364a, should be able to understand the report. For example, if the report is filled with another field's jargon that you don't understand, the writer is at fault.
- *Relevance.* The project must have relevance to convex (or non-convex) optimization.

2 Types of projects

Your project should fall into one of the following categories. Once you have a idea for a potential project, you should briefly discuss it with Prof. Duchi or one of the TAs to make sure you are on the right track.

We are very interested in projects that contribute to the public good. These projects have a higher chance of being accepted. However, do not feel discouraged from doing a project that has a more research flavor.

2.1 A new modeling approach

A common project type will involve an innovation in *modeling* (in the sense used in optimization), *i.e.*, the way a practical problem is formulated as an optimization problem. In such a project, you develop a new approach to some engineering (or other practical) problem, possibly in simplified form. Such a project will likely have the following components:

- *Background and problem in general terms.* You must clearly describe the engineering problem, giving the background, describing how the problem is solved now, and what's lacking or inadequate in how it's done now. For example, current design practice might be ad hoc, ignore a number of constraints, etc.

This description can be vague, as in “The goal is to estimate the original uncorrupted signal, without excessive sensitivity to noise.” Don't mix in anything here that is related to how you're going to solve the problem.

- *Your new formulation.* Explain your formulation of the problem (or some part of it, or some simplified variation) as an optimization problem. Be clear about what the variables and constraints are, and whether the problem is convex, nonconvex, stochastic, infinite dimensional, etc. How accurate is your modeling (formulation)?
- *Solution.* To solve the problem you formulate you can use standard codes or modeling languages such as CVX, or (if appropriate) develop your own algorithm and solver. If the problem you formulate has structure that can be exploited, and very large problems (or very short solve times) are of interest, you can develop an efficient implementation. If appropriate and interesting, you can develop a distributed algorithm for solving your problem.

If your formulated problem is nonconvex, give an algorithm for solving it, or approximately solve it, using convex optimization. Can you give any performance guarantees on your algorithm? If you cannot provide a guarantees, why (and for what kinds of problem instances) do you expect your method to work well?

- *Verification.* You should apply your solution method to one or more problem instances to verify any claims you make in previous sections. If you claim your algorithm is faster than some others, or can solve larger problem instances, convince us of this. If you made any approximation or simplification in formulating or solving your problem, show that the end result is still useful (and hopefully good). For example, if you have a new method for design of some circuit, you might provide some SPICE simulations of the designs produced by your method. If your method involves a heuristic for solving a hard (say, combinatorial) problem, then you can consider some cases that are small

enough for you to compute the global optimum (or a lower bound), which can then be compared to the approximate solutions produced by your method.

We emphasize that your formulation does not have to be a convex optimization problem. If it is nonconvex, however, you must use some convex optimization based methods for solving it, perhaps approximately or heuristically. You could form a relaxation, use a randomized method, use branch and bound (with convex optimization based lower bounds), or repeated linearization/convexification to (approximately) solve your problem. However, it is *unacceptable* to simply form a nonconvex problem and then approximately solve it using some standard nonlinear programming method or code.

Generic titles for projects of this style are “Optimal XXX via convex optimization”, or “Optimal design of XXX via convex optimization”. Here are some some samples:

- “Color correction via convex optimization”
- “Coordinated vehicle trajectory design via convex relaxations”
- “Transmit filter design via convex optimization”
- “DRAM cell design via geometric programming”
- “Optimal power assignment in wideband FDMA system with nonlinearities”

The principal final artifact for a project of this type is a research paper, along with all supporting material such as (well documented) codes, scripts, and data.

Your model for such a project is an EE364a final exam question, perhaps expanded by a factor of three. These exam problems take a general reader up to speed on some practical problem, defining all the relevant variables, objectives, constraints, problem data, relations between critical quantities, and so on; it leads to a numerical example that shows that the convex optimization method does, in fact, do something.

2.2 Pedagogical project

In a *pedagogical* project, the goal is not (directly) to do original research, but to create materials that explain, illustrate, or elucidate some aspect or application of convex optimization. For example, you might work out or collect from the literature the details of some application (say, digital filter design), develop simple codes to carry out the designs, and then work out a number of interesting examples.

For example, you might develop a simple digital filter design toolbox, with an accompanying IPython Notebook. As another (more generic) example, you might develop implementations of some algorithms for convex optimization, that are optimized for clarity and simplicity (not necessarily performance), that students in future years can use. Alternatively, you could commit to making a number of Wikipedia entries in some specific areas related to convex optimization.

Here are some sample titles of this kind of project:

- “Digital filter design”
- “Trajectory optimization”
- “Antenna array weight design”
- “Experiment design”
- “Extremal volume ellipsoids”
- “Hyperplane classification”

The final artifacts for such a project are: A well written lecture or set of lecture notes, with code and plots for several examples. Alternatively, you can create a small software toolbox, along with an IPython Notebook (or similar) illustrating an aspect or application of convex optimization. Good writing and clear documentation is extremely important in such a project.

2.3 Open-source software project

You will create a new open-source software implementing one or more algorithms, or you will contribute (substantially) to an existing open-source software project.

If you contribute to an existing project, the existing software project must either be related to convex optimization (*e.g.*, CVXPY), or your contribution must introduce convex optimization to it (*e.g.*, implementing MPC in a Python module for dynamics and control). Your contribution need not be algorithmic—for example, you can write documentation for or improve accesibility of an existing software project, in which case your written report would consist of a brief description of your contributions, accompanied by (say) the documentation you wrote. If you choose to do this, please make sure that we think your contribution is sufficient in scope.

If you create a new software project related to convex optimization, you must decide what to implement, and in addition to developing the code, you must develop documentation, test suites, etc. The goal is to make the software available publicly. The software can be in Matlab/Octave, C/C++, Python, Julia, or even R. You can develop low level code (calling LAPACK, BLAS, or some sparse libraries), or build on top of, or contribute to, higher level software such as CVX (Matlab), CVXPY (Python), or R. It should be as simple as possible and totally portable. You should avoid using anything but open source code.

One variation on this type of project is to develop new algorithms for solving large-scale instances of some family of problems, *e.g.*, from machine learning, signal processing, or finance. You will exploit problem structure and use some tricks to solve really big problems, beyond the reach of standard methods and software.

Final artifact: A summary of your contributions to an existing project, including any code/documentation/websites you created, or a directory, to be made publicly available, that contains your software project, including the source code (makefiles, etc.), and possibly

executables for several architectures, very clear documentation, and a number of examples and test routines. All materials for these projects should be on github.

Here are some sample titles of this kind of project:

- “C/LAPACK implementation of Newton and barrier methods for dense problems”
- “Algorithms for large-scale total variation reconstruction”
- “Python package for distributed convex optimization”
- “CUDA implementation of first-order dense cone programming solver”

2.4 Theory

Of course, this is not the main focus of the course. But we are open to a well-conceived project focusing on some theory related to the course material, *e.g.*, self-concordance and complexity analysis, relaxations, randomized algorithms.