# Optimization Methods: Assignments

Teaching Assistants

Spring 2019

## 1 A brief term paper (only one)

*Deadline Students: Before the final exams*
In this assignment you will explore methods in optimization that have not been covered in class, but are closely related. Read up on the topic, and write up a document summarizing your understanding. Additionally, use reference implementations or solvers available online using these methods to illustrate how the method works for a toy problem.

- A student with roll number $r$ is expected to turn in a term paper on topic $n$, where $n = (r \bmod 5) + 1$.
- Turn in the term paper (as a PDF file) and any source files in a `.tar.gz` archive on moodle. Keep the report **around 4 pages** not including references. (say in IEEE two column format or similar)

Some pointers on the bare minimum expected is given below.

### 1.1 Ellipsoid Method

`@Sudheer Achary`

- Briefly explain show the steps for ellipsoid method.
- Find feasible solution for any 2 examples using ellipsoid method and show steps for each iteration.
- Explain time complexity of ellipsoid method.

### 1.2 Interior Point Methods

`@Kumar Abhishek`
In this report you are expected to do the following:

- Explain about interior point methods.
- Choose atleast 2 interior point methods. State the types of optimization problem which can be solved using it.
- Describe the method explaining each step and mention the pros and cons of the methods.
- Take any appropriate example which you find interesting for each method and solve it using any solver. (or Take worked out examples and codes to explain the working of the methods)

### 1.3 Cutting Plane Methods

`@Yaghyavardhan Singh Khangarot`

In the report you have to do the following:

- Explain about Cutting Plane Methods.
- Choose any 2 Cutting Plane methods and Describe the method explaining each step and mention the pros and cons of the methods.
- Take any example (one for each method) and show the solution for them in the report (with codes).

### 1.4 Quadratic Programming: Conjugate Gradients

`@Avinash Vadlamudi`

- Explain the Conjugate gradient method for optimization of a second degree expression and also state its limitations. Write the conjugate gradient algorithm.
- Explain any method as to how to solve a second degree expression which doesn't satisfy the limitations of the conjugate gradient method, but using the approach similar to conjugate gradient method [Hint: You can ask for extra information]

Solve the Following Problems or you can solve any 2 problems of your choice related to conjugate gradient method:

1. Using Conjugate Descent method, find the solution for $\text{minimize} \quad f(x) = 2x^2 + 3y^2 - 6x + 5y$
2. If $d_0, d_1, ..., d_n$ be Q-conjugate directions, then show that $r_{k+1}^T d_i = 0 \quad \forall i = 0, 1, ..., k$ [ Hint: Use Induction Method ]

### 1.5 Cone Programming

`@Anuj Rathore`

- Explain about Conic Linear Program with few examples (with codes).
- Describe Conic Optimization and duality in Conic Programming.
- Explain about Quadratic cones with example (with codes).
- Explain about power cone and exponential cone.

Mention the references used in the report. You need to submit report along with the codes.

# 2 Pegasos

`@Jerin Philip <jerin.philip@research.iiit.ac.in>`
*Deadline: 1 May 2019, 23:59pm*

## 2.1 Statement

In this project, you'll implement your own version of the **Primal Estimated sub-GrAdient SOlver for SVM (Pegasos)**. Pegasos uses sub-gradient descent to arrive at the optima.

1. Implement a non-kernelized Pegasos, optimizing the primal objective.
2. Extend the above further enabling use of Mercer kernels.
3. Demonstrate successful results for a simple binary classification task.

You may refer to this paper (1) for implementation details. We suggest binary classification among two classes in *FashionMNIST*(2) as your optimization problem. (Focus on the optimization procedure, you're free to use other datasets or reduce the ones given to smaller complexity. But be sure to include clear description in your reports what you've used.)

## 2.2 Expected Deliverables

The maximum a student can get is 20 points.

1. Source-code of your implementation, with instructions to reproduce. Link to a screencast of the code running on your system. [10 points]
2. Report containing the following: [10 points]
   (a) Core contents in your source code embedded in the report, with text explaining these (gradient descent procedures, kernelization, loss functions used etc).
   (b) Basic performance metrics of your optimization procedure (Runs in $T$ time for data of complexity $D$).
3. Bonus Parts (requires source codes, report as above) [20 points]
   (a) Extend your binary SVM implementation to a multi-class setting. [7.5 points]
   (b) Comparisons across multiple objectives and kernels in your implementation. [7.5 points]
   (c) Comparisons with a few standard libraries on the same data as what you used above. [5 points]

## 2.3 Submission Instructions

- Submit the source codes of your implementation and report as a PDF file compressed into a `.tar.gz` archive, on moodle. If you use a reference implementation - link to it, cite it properly. Do not produce somebody else's work as your own work.
- You're allowed to collaborate in teams of size 2, provided you use `git` and include the git history (`.git` folder) in your submission. If the history indicates fair work-division, your individual marks will be total marks divided by 1.5 in this case.
- Warning: *Stanford Moss* will be run before any evaluations. Suspected plagiarism will get you zero.
- The submissions will ideally be evaluated offline provided your submissions are functional. You may be summoned for manual evaluation if there are errors or discrepencies in your submission.

# References


[1] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for SVM," *Mathematical programming*, vol. 127, no. 1, pp. 3–30, 2011.

[2] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.