CS 769: Optimization in Machine Learning

COURSE INTRO

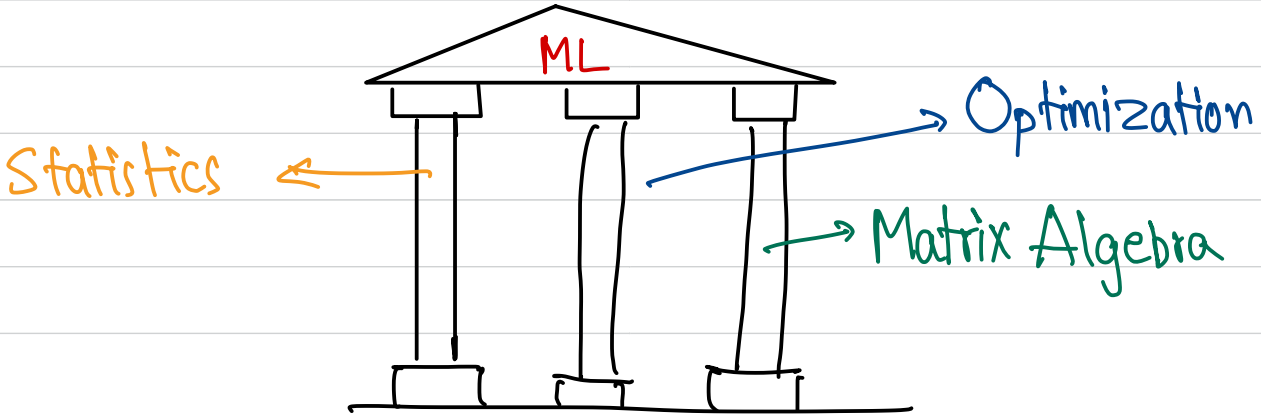
Why take this course?

Optimization is everywhere: Big Data, ML, Controls, Simulations, etc

Generally, there are two fundamental steps in approaching and solving the problems from the above areas:

1. **MATHEMATICAL MODELING** : Defining and modeling the problem
2. **COMPUTATIONAL OPTIMIZATION**: Optimal or near optimal algorithms to solve

ML and AI are embedded in practically every spear of our life. Democratization of AI has made way to create ML applications with a few lines of code by any one. Optimization is one of the important backbones of ML. This course helps in understanding the math behind the existing ML problems, and equips one with the skills needed to solve real-life problems using different optimization algorithms; and also to carry out research work in this area.



Types of Optimization

1. **CONTINUOUS OPTIMIZATION:** It often appears as relaxations of risk/error minimization problem. The Learning problem in many parametrized models (supervised or unsupervised or semi-supervised or RL) involves continuous optimization.
   Eg: Logistic loss, Listwise loss

2. **DISCRETE OPTIMIZATION:** It occurs in Inference problems in structured spaces, certain learning problems and auxiliary problems such as Feature Selection, Data subset selection, Data summarization, Architecture search etc.

3. **MIXED CONTINUOUS AND DISCRETE:** Clustering, feature selection, structured sparsity etc

Philosophy of this Course

- Algorithmic aspects of optimization, not so much on modeling.
- Flavor of proofs and proof techniques
- Implementational aspects

Continuous Optimization

- Basics of Continuous Optimization
- Convexity
- Gradient Descent
- Projected/Proximal GD
- Subgradient Descent
- Accelerated GD
- Newton & Quasi Newton
- Duality: Lagrange, Fenchel
- Coordinate Descent
- Frank Wolfe
- Optimization in Practice

Discrete Optimization

- Linear Cost Problems
- Matroids, Spanning trees
- s-t paths, s-t cuts
- Matchings
- Covers (Set Covers, Vertex Covers, Edge Covers)
- Optimal Transport
- Non-linear Discrete Optimization
- Submodular Functions
- Submodularity and Convexity
- Submodular Minimization
- Submodular Maximization
- Optimization in Practice

Convex Optimization in ML: Applications

Application 1: Supervised Learning

- Data: Given training examples $\{(x_1, y_1), \dots, (x_n, y_n)\}$ where $x_i \in \mathbb{R}^m$ is the feature vectors and y_i is the label
- Applications: Email Spam Filtering, Handwritten Digit Recognition, Housing Price Prediction

Supervised Learning: Modeling

- Data: Given training examples $\{(x_1, y_1), \dots, (x_n, y_n)\}$ where $x_i \in \mathbb{R}^m$ is the feature vectors and y_i is the label
- Model: It is denoted by $F_\theta(x)$ with θ denoting the parameters of the model. $F_\theta(x)$ can be anything from a simple linear model to deep recursive functions.
- Loss Functions: It is denoted by L. It is a function that tries to measure the distance between $F_\theta(x_i)$ and y_i

Supervised Learning: Optimization Problem

- Loss + Regularizer Framework:

$$\min_{\theta} G(\theta) = \sum_{i=1}^{n} L(F_\theta(x_i), y_i) + \lambda \Omega(\theta)$$

Purposes of a Regularizer:

1) Increasing the bias of the model to yield simpler parameters. This is same as adding a prior to the model.
2) Makes the optimization problem well-behaved through properties such as strong local/global convexity resulting in faster convergence rates of optimization algorithms.

- Examples of L:

Logistic Loss: $\log(1+\exp(-y_i F_\theta(x_i)))$
Hinge Loss: $\max\{0, 1-y_i F_\theta(x_i)\}$
Softmax Loss: $-F_{\theta_{y_i}}(x_i) + \log\left(\sum_{c=1}^{k} \exp(F_{\theta_c}(x_i))\right)$
Absolute Error: $|F_\theta(x_i) - y_i|$
Least Squares: $(F_\theta(x_i) - y_i)^2$
Boosting: $e^{-F_\theta(x_i) y_i}$

- Examples of Ω:

L1 Regularizer: $\sum_{i=1}^{m} |\theta[i]|$
L2 Regularizer: $\sum_{i=1}^{m} \theta^2[i]$

Application 2: Clustering

- This is an instance of unsupervised learning. This also has an discrete optimization counterpart.
- The continuous clustering formulations often yield convex clusters (unless we employ kernels).

- **Data:** Given unlabeled (unsupervised) data $\{x_1, \ldots, x_n\}$ where $x_i \in \mathbb{R}^m$ is a feature vector.

- **Goal:** Find clusters (sets) $C_1, C_2, \ldots, C_k$ with each cluster consisting of similar instances. Denote $V = \{1, \ldots, n\}$. Then $\bigcup_{i=1}^{k} C_i = V$.

- Similarity could be measured as decreasing functions of distances.
- Distance d should preferably satisfy the triangle inequality.
- Likewise, if similarity is defined directly, then preferably it should be a positive semi-definite kernel.

Eg: cosine similarity, polynomial functions of cosine, Eucledian distance, L1 distance

- **Optimization Problem:** The k-means optimization problem is

$$\min_{C_1, \ldots C_k} \sum_{i=1}^{k} \sum_{x \in C_i} \left\| x - \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \right\|_2^2$$

Application 3: Principal Component Analysis

Unsupervised Learning

- **Data:** Given unlabeled data $\{x_1, x_2, \cdots, x_n\}$ where $x_i \in \mathbf{R}^m$ are the feature vectors

- **Goal:** Identify a lower dimensional meaningful representations such that you capture the directions of maximum spread of the data. Here spread = variance

  Find compressors $U \in \mathbf{R}^{m \times k}$ and correspondingly decompressors $V \in \mathbf{R}^{k \times m}$ such that x_i is close to UVx_i. The compressor must also satisfy $V = U^T$ and $U^TU = I$

- **Optimization Problem:**

$$\min_{U : U^TU = I} \sum_{i=1}^{n} \| x_i - UU^Tx_i \|_2^2$$

Application 4: Matrix Completion

- **Data:** Given observations $y_1, \dots, y_n$ such that each $y_j = A_j(X)$ where A_j could be a single element or a combination of elements in $X \in \mathbf{R}^{m \times n}$. A common example for X is the product recommendation matrix.

- **Goal:** Find the simplest matrix X s.t. $A_j(X) \approx y_j \ \forall j = 1, 2, \dots, n$

- **Optimization Problem:**

$$\min_X \sum_{i=1}^{n} \| y_i - A_j(X) \|_2^2 + \| X \|_*$$

$\| \cdot \|_*$ denotes the nuclear norm.

- This problem can be solved using Iterative Singular Value Thresholding (ISTA)

Application 5: Low Rank and Non Negative Matrix Factorization

- Goal: Find matrices L, R with $L \in \mathbf{R}^{m \times k}$ and $R \in \mathbf{R}^{k \times n}$ s.t. $A_j(LR) \approx y_j \ \forall j \in \{1, 2, \dots, n\}$ so that X has low rank.

- Optimization Problem:

$$\min_{L,R} \sum_{i=1}^{n} \| y_i - A_i(LR) \|_2^2 \qquad \text{Rank}(LR) \leq k$$

No need of matrix regularization.

- We can also add non-negativity constraints and this becomes non-negative Matrix Factorization

$$\min_{\substack{L,R: \\ L \geq 0, R \geq 0}} \sum_{i=1}^{n} \| y_i - A_i(LR) \|_2^2$$

- Can be solved using the Iterative Projection Algorithm.
- Sometimes Y is fully observed and we want a NNMF + Low Rank $Y \approx LR$

$$\min_{L,R: L \geq 0, R \geq 0} \sum_{i=1}^{n} \| Y - LR \|_2^2$$

Discrete Optimization in ML

MAP inference in Probabilistic Models: Ising Models, DPPs

Feature Subset Selection

Data Partitioning

Data Subset Selection

Data Summarization: Text, Images, Video Summarization

Social Networks, Influence Maximization

NLP: words, phrases, n-grams, syntax trees, semantic structures

CV: Image segmentation, Image Correspondence

Genomics and Computational Biology: cell types or assay selection, selecting peptides and proteins

Application 1: Image Segmentation and Correspondance

- Formulating image segmentation as a mincut problem.
- Formulating image correspondance as a bipartite matching problem.

TODO: Add Pics

Application 2 : Feature Selection

- **Data :** Given random variables $X_1, X_2, \ldots, X_n$ as features of a given ML task. Denote $I(X_1, X_2)$ as the Mutual Information between variables X_1 and X_2

- **Goal:** Select a subset of features $A \subseteq \{1, \ldots, n\}$ such that the subset of features are as good as the original set.

- **Optimization Problem:** Maximize the Mutual Information between the set of features and the label Y

$$\max_{A : |A| \leq k} I(X_A ; Y)$$

- This is a constrained submodular maximization problem.
- Entropy H is a submodular function and it can be replaced by several other submodular functions to get different measures of Mutual Information.

Application 3: Training Data Subset Selection

- **Data:** Given a training dataset $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ and a budget k.
- **Goal:** Select a subset of data-points $A \subseteq \{1, \cdots, n\}$ such that the model trained on the subset of data is as good as the entire dataset.
- This setting makes even more sense when the labels are missing on some or all of the given data-points. It is useful in identifying the subset worth labelling.
- **Optimization Problem:** If $D(\cdot)$ denotes the distance (such as KL div.) between the two datasets (e.g. distributions $P(\cdot)$ or gradients), the optimization problem can be cast as:

$$\max_{A:\,|A| \leq k} \; -D(X_A, X_V)$$

- Sometimes, diversity within X_A (possibly relative to X_V) is also considered.

Application 4: k-Mediods Clustering

- **Data:** Given a set of datapoints $\{x_1, x_2, \dots, x_n\}$, a similarity function s_{ij}, $i,j \in \{1,\dots,n\}$ and a budget k.

- **Goal:** Select a subset of data-points $A \subseteq \{1,\dots,n\}$ which can act as k-mediods (similar to k-means except that the means are a part of the original set of points).

- **Optimization Problem:**

$$\max_{A:\,|A|\leq k} \sum_{i=1}^{n} \max_{j\in A} s_{ij}$$

- Any point i is assigned to the mediod j which is most similar to it.
- Search space is the power set
- Similarities s_{ij} can be chosen to be rich, including gradients of loss functions