CS 769: Optimization in Machine Learning

COURSE INTRO

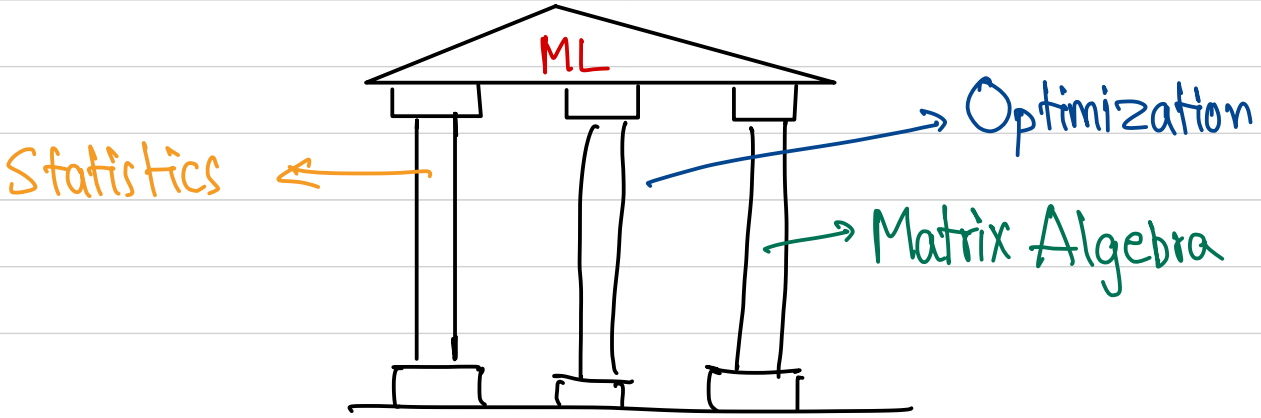
Why take this course?

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

Basics of Continuous Optimization and Convexity

Notation: Vectors and Matrices

- n denotes the number of training instances. m denotes the number of features.
- $\mathbf{x}_i \in \mathbf{R}^m$ as m-dimensional feature vector
- $\mathbf{x}_i[j]$ as the j^{th} feature
- y_i as the label of the i^{th} instance
- Given two vectors $\mathbf{w}, \mathbf{x} \in \mathbf{R}^m$, $\langle \mathbf{w}, \mathbf{x} \rangle = \sum_{j=1}^{m} \mathbf{x}[j]\mathbf{w}[j]$
- L1 norm $||\mathbf{w}||_1 = \sum_{i=1}^{m} |\mathbf{w}[i]|$
- The squared L2 norm $||\mathbf{w}||_2^2 = \sum_{i=1}^{m} |\mathbf{w}[i]|^2$
- $\mathbf{y} = A\mathbf{x}$, $A \in \mathbf{R}^{m \times n}$, $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{y} \in \mathbf{R}^m$
- $C = AB = [Ab_1 | Ab_2 | \cdots | Ab_p]$, $A \in \mathbf{R}^{m \times n}$, $B \in \mathbf{R}^{n \times p}$, $C \in \mathbf{R}^{m \times p}$

Derivatives:

- $y = f(x) \Rightarrow \frac{dy}{dx} = \frac{df(x)}{dx} = f'(x)$
- $y = c \Rightarrow y' = 0$
- $(cf)' = cf'$ for a constant c
- $y = x^a \Rightarrow y' = a x^{a-1}$
- $y = f \pm g \Rightarrow y' = f' \pm g'$
- $y = fg \Rightarrow y' = fg' + f'g$
- $y = f \circ g \Rightarrow y' = f'(g)\, g'$
- $y = \frac{f}{g} \Rightarrow y' = \frac{gf' - fg'}{g^2}$
- Subgradient Calculus

Gradient:

- For ease of notation, we define vector $w = [w_1 \cdots w_m]^T$
- The loss function denoted as $L(w) = L(w_1, w_2, \cdots, w_m)$
- The gradient $\nabla L(w)$ is defined as:

$$\nabla L(w) = \left[\frac{\partial L}{\partial w_1} \quad \frac{\partial L}{\partial w_2} \cdots \frac{\partial L}{\partial w_m} \right]$$

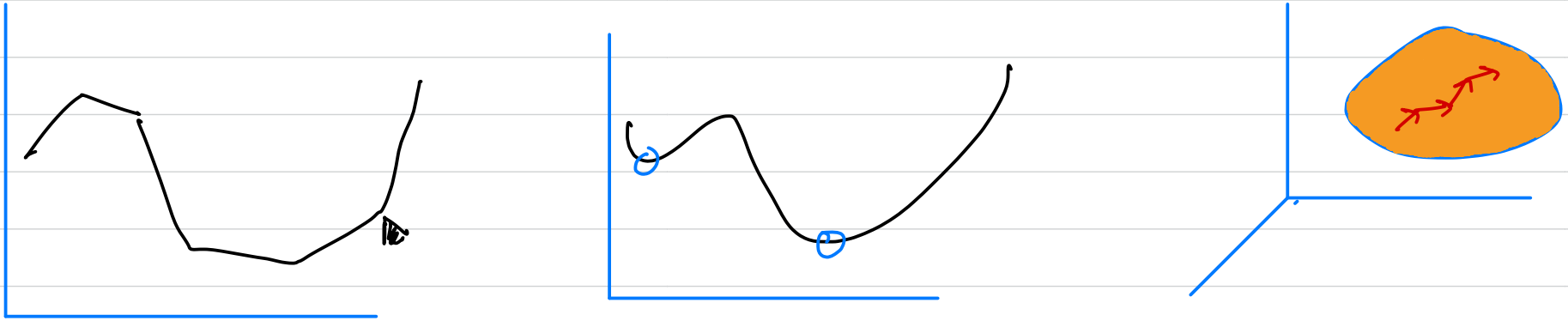
Hessian

$$\nabla^2 f(w) = \begin{bmatrix} \frac{\partial^2 f}{\partial w_1^2} & \frac{\partial^2 f}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 f}{\partial w_1 \partial w_m} \\ \frac{\partial^2 f}{\partial w_2 \partial w_1} & \frac{\partial^2 f}{\partial w_2^2} & \cdots & \frac{\partial^2 f}{\partial w_2 \partial w_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial w_m \partial w_1} & \frac{\partial^2 f}{\partial w_m \partial w_2} & \cdots & \frac{\partial^2 f}{\partial w_m^2} \end{bmatrix}$$

- Hessian matrix is symmetric when the mixed partial derivatives are continuous.
- $$\nabla^2 f(w) = \left[\nabla \frac{\partial f}{\partial w_1} \quad \nabla \frac{\partial f}{\partial w_2} \cdots \nabla \frac{\partial f}{\partial w_m} \right]$$

Convexity: Properties, Applications in ML

- Why Convexity: Every function has convex regions
- Even if a function is non-convex, convergence of algorithms is generally in terms of convergence in convex regions.
- For convex regions of functions, at the points of non-differentiability we use subdifferentials / subgradients.
- Most deep learning models exhibit convex losses / have convex regions.



- Further, the domain/constraints of optimization functions we discuss will be typically convex sets so that algorithm based updates are guaranteed to lie within the set.
- Especially important for projection/Frank Wolfe/interior point/Lagrange multiplier based methods

Convex Sets:

A set C is a convex set if the line segment between any two points of C lies in C, i.e. if for any $x, y \in C$ and for any $0 < \lambda < 1$, we have that $\lambda x + (1-\lambda) y \in C$

- If $\lambda_1 x + \lambda_2 y \in C \quad \forall \lambda_1 + \lambda_2 = 1 \;\&\; x, y \in C$, then C is called an affine set. It is also convex, obtained by relaxing positivity of lambdas.

- If $\lambda_1 x + \lambda_2 y \in C \quad \forall \lambda_1, \lambda_2 \geq 0 \;\&\; x, y \in C$, then C is called a Convex Cone (and it is a convex set), obtained by relaxing $\lambda_1 + \lambda_2 = 1$.

Properties of Convex Sets:

- Intersections of convex sets are convex.
- Union of convex sets is not always convex. It can have disconnected regions.
- Projections onto convex sets are unique and often efficient to compute. If set S is not convex, then the projection of x onto S need not be unique.

$$P_C(x) = \underset{y \in C}{\operatorname{argmin}} \, \| y - x \|$$

- Examples:
  - Norm Ball: $C = \{x \in \mathbb{R}^n : \|x\| \leq k\}$
  - Half Space: $C = \{x \in \mathbb{R}^n : w^T x \leq k\}$
  - Sublevel Set: Given a convex function f, the associated set $C_f = \{x \in \mathbb{R}^n : f(x) \leq k\}$ is convex

Convex Combination and Convex Hull

- Convex Combination: For a set of points $x_1, x_2, \cdots, x_k$; it is any point x of the form

$$x = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k = \text{conv}(\{x_1, x_2, \cdots, x_k\})$$
$$\text{with } \theta_1 + \theta_2 + \cdots + \theta_k = 1, \ \theta_i \geq 0$$

- Convex Hull or conv(S): It is the set of all convex combinations of point in the set S. The set S need not be convex, but the convex hull is always convex.

- Any convex set can be thought of as convex hull of all the points (i.e., canonical points) lying on its boundary.

Euclidean Balls and Ellipsoids

- A Euclidean ball with center $\boldsymbol{x}_c$ and radius r is given by

$$B(\boldsymbol{x}_c, r) = \{\boldsymbol{x} \mid \|\boldsymbol{x}-\boldsymbol{x}_c\|_2 \leq r\} = \{\boldsymbol{x}_c + r\boldsymbol{u} \mid \|\boldsymbol{u}\|_2 \leq 1\}$$

- An ellipsoid is a **set** of form:

$$\{\boldsymbol{x} \mid (\boldsymbol{x}-\boldsymbol{x}_c)^T P^{-1}(\boldsymbol{x}-\boldsymbol{x}_c) \leq 1\}, \text{ where } P \in \mathbb{S}^n_{++} \text{ i.e. } P \text{ is SPD matrix}$$

  - Alternate representation: $\{\boldsymbol{x}_c + A\boldsymbol{u} \mid \|\boldsymbol{u}\|_2 \leq 1\}$ s.t. A is square and non-singular.

- When P is the covariance matrix, the distance $(x-x_c)^T P^{-1}(x-x_c)$ is called the Mahalanobis distance.
- P and its inverse induces rotation and scaling of points within a Euclidean ball.

Norm Balls

- Norm: A function $\|\cdot\|$ that satisfies

  1. $\|x\| \geq 0$ and $\|x\| = 0$ iff $x = 0$

  2. $\|\alpha x\| = |\alpha| \, \|x\|$ for any scalar $\alpha \in \mathbb{R}$

  3. $\|x_1 + x_2\| \leq \|x_1\| + \|x_2\|$ for any vectors x_1 and x_2

- Norm ball with center x_c and radius γ: $\{x \mid \|x - x_c\| \leq \gamma\}$ is a convex set.
  Eg: Ellipsoid is defined using $\|x\|_P^2 = x^T P x$ Matrix induced vector norm
  Euclidean ball is defined using L_2 norm.

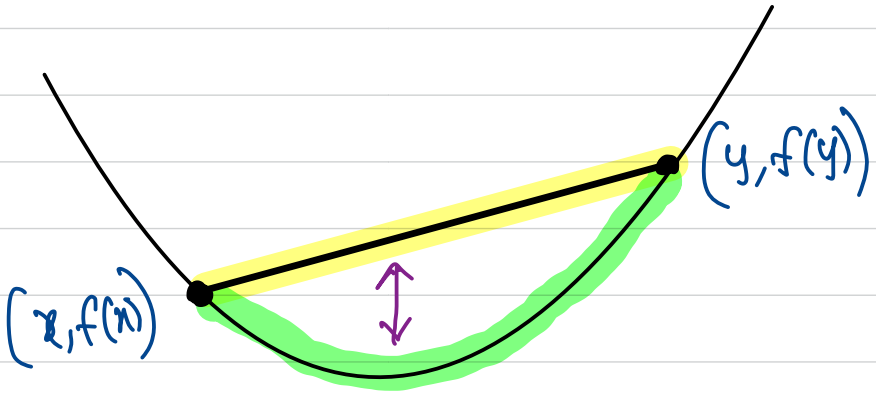
There is a dual view involving sets and functions in the context of convex optimization

Convex Functions

- A function $f: \mathbb{R}^d \to \mathbb{R}$ is convex if:
  1. $dom(f)$ is a convex set
  2. $\forall x, y \in dom(f)$ and $\lambda: 0 < \lambda < 1$, we have $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$

- Function at convex combination $\leq$ convex combination of function values

- Geometrically, the line segment between $(x, f(x))$ and $(y, f(y))$ lies above the graph of f.



- f is strictly convex if $\forall x, y \in dom(f)$ and $\lambda: 0 < \lambda < 1$, we have:

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y)$$

Gap is strictly enforced

Strongly Convex Functions

- A function $f: \mathbb{R}^d \to \mathbb{R}$ is strongly convex if there exists a $\mu > 0$ such that the function g given by $g(x) = f(x) - \frac{\mu}{2}\|x\|^2$ is convex.

- μ: Strong Convexity Parameter

- Strong convexity means that there exists a quadratic lower bound on the growth of the function.
- Strong convexity $\Rightarrow$ strict convexity
- Strong convexity $\not\Rightarrow$ function is differentiable
- If f is strongly convex and g is convex, then $f+g$ is strongly convex.
- $\|x\|^2$ is strongly convex
- Hence, for any convex function f, the function $f(x) + \frac{\lambda}{2}\|x\|^2$ is strongly convex.

- Strict convexity (non-zero gap) and strong convexity (function based gap) are both special cases of convex functions.
- Other function based gaps are also possible.

$f: \mathbb{R}^d \to \mathbb{R}$ is strongly convex for all $x_1, x_2 \in \text{dom}(f)$ and $\theta: 0 < \theta < 1 \Rightarrow$

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2) - \frac{1}{2}\mu\theta(1-\theta)\underbrace{\|x_1 - x_2\|^2}_{\text{quadratic gap}}$$

- Either we can add the quadratic gap to a convex function to make it strongly convex, or
- We can remove that quadratic gap from a strong convex function to make it convex only.
- Examples:
  1. Linear Functions: $f(x) = a^T x$
  2. Affine Functions: $f(x) = a^T x + b$
  3. Exponential: $f(x) = e^{ax}$
  4. Every norm

Properties of Convex Functions

- Non-negative weighted sum: $f = \sum_{i=1}^{n} \alpha_i f_i$ is convex if each f_i is convex and $\alpha_i \geq 0 \ \forall i$.

- Composition with affine function: $f(Ax+b)$ is convex if f is convex.

Eg: 1. The log barrier for linear inequalities: $f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$ is convex since $-\log x$ is convex.

2. Any norm of an affine function: $f(x) = \|Ax+b\|$.

- This is closely related to the fact that ellipsoids (as affine transformation of euclidean balls) are also convex.

- Composition with scalar functions: $g: \mathbb{R}^n \to \mathbb{R}$, $h: \mathbb{R} \to \mathbb{R}$, $f(x) = h(g(x))$ f is convex if
  a) g is convex, h is convex and non-decreasing OR
  b) g is concave, h is convex and non-increasing

- Composition with vector functions: $g: \mathbb{R}^n \to \mathbb{R}^k$ and $h: \mathbb{R}^k \to \mathbb{R}$, $f(x) = h(g(x)) = h(g_1(x), \cdots, g_k(x))$
f is convex if
  a) g_i's are convex, h is convex and non-decreasing in each argument OR
  b) g_i's are concave, h is convex and non-increasing in each argument