CSE6243: Advanced Machine Learning **Fall 2024**

Lecture 2: Convex Preliminary

Lecturer: Bo Dai *Scribes: Eric Chen, Rishit Ahuja*

Note: *LaTeX template courtesy of UC Berkeley EECS Department.*

1.1 Recap: ML paradigms

1. **Supervised Learning**

   - **Data:** The dataset $\mathcal{D}$ consists of pairs (x_i, y_i), where each x_i is an input and y_i is the corresponding label or output. This is common in tasks like classification or regression.
   - **Algorithm:** The goal of the algorithm (denoted by Alg) is to learn a function $f(\cdot)$ that maps inputs from the input space $\mathcal{X}$ to the output space $\mathcal{Y}$.

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$$
$$\text{Alg}(\mathcal{D}) \Rightarrow f(\cdot) : \mathcal{X} \to \mathcal{Y}$$

2. **Unsupervised Learning**

   - **Data:** The dataset $\mathcal{D}$ consists only of inputs x_i, without any associated labels. This is used in tasks like clustering or dimensionality reduction.
   - **Algorithm:** The algorithm seeks to learn a function $f(\cdot)$ that maps the input space $\mathcal{X}$ to some latent space $\mathcal{Z}$ which represents the underlying structure in the data.

$$\mathcal{D} = \{x_i\}_{i=1}^n \in \mathcal{X}$$
$$\text{Alg}(\mathcal{D}) \Rightarrow f(\cdot) : \mathcal{X} \to \mathcal{Z}$$

3. **Reinforcement Learning**

   - **Data:** The dataset $\mathcal{D}$ consists of sequences of state (s_i), action (a_i), reward (r_i), and next state (s_i') tuples, collected over time as the agent interacts with an environment.
   - **Algorithm:** The goal is to learn a policy $\pi(\cdot|s)$ that maps states s to a probability distribution over actions $\Delta(\mathcal{A})$, aiming to maximize cumulative rewards.

$$\text{Alg(Env)} \Rightarrow (\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^T, \pi(\cdot|s) : \mathcal{S} \to \Delta(\mathcal{A}))$$

The first two (Supervised and Unsupervised) are considered passive paradigms. RL is considered active because it must gather its own data used to improve its policy by interacting with the environment.

1.2 New Content

1.2.1 Optimization Motivation

Within the paradigms of ML, optimization can be used to find the appropriate Alg which best satisfies problems in each paradigm. Optimization Problem:

$$\min_{x \in \Omega} f(x)$$
$$\text{s.t. } g(x) \geq 0$$

Where $f(x)$, sometimes also denoted $\mathcal{L}(x)$, is a loss (objective) function subject to constraint $g(x) \geq 0$, which can also be optional. $f(x)$ is a convex function, Ω is some feasible domain (a convex set), and $g(x)$ is a convex function. Convex optimization has become a popular tool for solving ML problems recently because of known polynomial algorithms.

1.2.1.1 Examples

These are some of the ML algorithms framed as equivalent optimization problems:

1. Regression: For $y \in \mathbb{R}$, find $f(\cdot) : X \to \mathbb{R}$ such that $f = \underset{f}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$

2. (Binary) Classification: For $y \in \{0, 1\}$, find $f(\cdot) : X \to \{0, 1\}$ such that

$$f = \underset{f}{\text{argmin}} \sum_{i=1}^{n} y_i \log(P(y = 1|x_i)) + (1 - y_i) \log(1 - P(y_i = 1|x_i))$$

   where $P(y_i = 1|x_i) = \frac{1}{1+\exp(-f(x_i))}$

3. Dimension Reduction: Dimensionality reduction is a type of unsupervised learning problem.

   Generally, we can frame the problem as:

   Let, x_i be a datapoint in the original dataset and z_i be a datapoint in the modified dataset, then $x_i \in \mathbb{R}^{d \times 1}$, $z_i \in \mathbb{R}^{p \times 1}$ where $p < d$. One of the ways to solve this problem is to use PCA, where we do a linear transformation, specifically we find a matrix A such that $A \in \mathbb{R}^{d \times p}$ and $x = Az$ and then we solve the optimization problem $\min_{A,z} ||x_i - Az_i||^2$.

   Note how if we restrict z such that $z \in \{0, 1\}^p$ and $||z||_2 = 1$, i.e. restricting z to be a binary vector containing a 1 in only one entry and zeros everywhere else, then this becomes a clustering problem with p clusters. In this case, each of the p entries of the z vector correspond to one of p possible clusters.

4. Bayesian Regression: Professor emphasized that for Bayesian Linear Regression, the optimization occurs at distribution level, as opposed to the above examples, where the optimization occurs only at the parameters/weights level assuming a fixed distribution as in Ordinary Least Squares Regression. Please refer to the Appendix at the end of the PDF for details of Bayesian Linear Regression that were talked about in class at a high level.

It is also possible to take discrete (and often times intractable) problems to the continuous space and frame them as optimization problems. Using known efficient optimization techniques can reveal insights or be used to approximate the true optimal in the original discrete setting.

Ex: For some reward function $R(\cdot)$, find a binary (integer) vector x such that $\min_{x \in \{0,1\}^d} R(x)$. (NP Hard) If we relax the discrete restriction on x and treat it as a probability vector instead, the objective of finding x such that $\min_{x \in [0,1]^d} \mathbb{E}[R(x)] = \sum p(x_i) \cdot R(x_i)$ becomes a tractable optimization problem.

1.2.2 Properties of Convex Optimization

1.2.2.1 Definitions

1. Convex Set - Ω denotes a convex set if $\forall x, y \in \Omega$ and for any $t \in [0,1]$, $tx + (1-t)y \in \Omega$. In other words, if $x, y \in \Omega$, all the points along their interpolation (line) must also be a member of the convex set.

2. Convex Function - $f(\cdot) : \Omega \to \mathbb{R}$ is a convex function if the domain is a convex set and $\forall x, y \in \Omega$ and $t \in [0,1]$, $f\left(tx + (1-t)y\right) \leq tf(x) + (1-t)f(y)$. If $t = 1/2$, what the definition implies is that the value of the average of two points is at most the average of the values of the two points within the convex set. This idea can be extended for any $t \in [0,1]$, which essentially states that the interpolation of the value between two points is always at least the value of the interpolation.

3. Local minimum - Let $\omega \in \Omega$ denote a local minimum for convex function $f(\cdot)$ if $\forall u, ||\omega - u|| \leq \rho$ and $f(u) \geq f(\omega)$. ρ describes a small threshold distance which defines a region around ω, and u represents points in the convex set domain which fall inside that region.

With these definitions, they imply an important conclusion which helps to motivate convex optimization: *every local minimum is a global minimum for a convex function.*

1.2.2.2 Propositions

Proposition 1. If $f(\cdot)$ is convex and ω^* is a local minimum, then ω^* is a global minimum of f.

Proof:

1. Let $\omega_\circ$ denote a "true" global minimum such that $f(\omega_\circ) < f(\omega^*)$, where ω^* is a local minimum of convex function f.

2. By convexity of f, we have that

$$f(t\omega_\circ + (1-t)\omega^*) \leq tf(\omega_\circ) + (1-t)f(\omega^*)$$

3. Continuing with the assumption that $f(\omega_\circ) < f(\omega^*)$:

$$\begin{aligned} &f(t\omega_\circ + (1-t)\omega^*) \leq tf(\omega_\circ) + (1-t)f(\omega^*) \\ &\leq tf(\omega^*) + (1-t)f(\omega^*) = f(\omega^*) \end{aligned}$$

4. Note how by the definition of the local minimum ω^*, there exists some threshold ρ which describes a region where $f(\omega^*) \leq f(u)$, for all u such that $||\omega^* - u|| \leq \rho$. However, from step 3, we know that there is some $t \in [0,1]$ which satisfies the inequality and implies $f(t\omega_\circ + (1-t)\omega^*) \leq f(\omega^*)$.

5. This results in a contradiction, because it is now possible to define a t which describes a point $u = (t\omega_\circ + (1-t)\omega^*)$ such that $||\omega^* - u|| \leq \rho$. By the convexity definition described in step 3, $f(u) < f(\omega^*)$, but by the local minimum definition, $f(u) \geq f(\omega^*)$.This region would now contain a point lower than the local minimum and violate the given condition of ω^* being a local minimum. Therefore, the assumption that there exists some $\omega_\circ$ such that $f(\omega_\circ) < f(\omega^*)$ must be false, implying that a local minimum for a convex function must also be a global minimum.

   For completeness in the contradiction argument, technically the point u could be such that $f(u) = f(\omega^*)$ for the specific local minimum defined ω^*. However, this would imply that u is also a local minimum and the same argument would apply that the ρ_u neighborhood around u must contain no points with $f(\cdot)$ value greater than $f(u)$. Since ρ for the local minimum definition is always nonzero, there will eventually be a point along the interpolation between $\omega_\circ$ and ω^* such which creates a contradiction.

Proposition 2. First Order Conditions for local minimums:

- x is a local minimum *iff* $\nabla f(x) = 0$ (1st order check for local minima) and there are no other constraints. **Note:** this necessary and sufficient condition for local minimum is specifically for the convex setting. In general, $\nabla f(x) = 0$ is only a necessary condition for local minima.
- More generally, (i.e. considering constraints), x is a local minimum iff $\forall y \in \Omega$, $\nabla f(y)^\top (y - x) \geq 0$. Intuitively, this indicates that moving from x to any other feasible point y does not decrease the value of the function.

Example: As an example of unconstrained optimization, in linear regression, we seek to find ω to minimize $||Y - \omega^\top X||^2$.

The solution to the unconstrained optimization of the least squares linear regression is defined as:

$$f(x) = w^T x, \quad \min_w ||Y - w^T X|| = l(w)$$

is given by $w^* = (XX^T)^{-1}XY^T$.

Proof: Here is a proof using the propositions above. Another, more commonly used, linear algebraic proof exists but this proof particularly emphasizes the propositions above and uses the fact that the loss function is convex.

$$\begin{aligned} &\text{Thus, } \nabla l(w^*) = 0 \quad (w^* \text{ is a local/global minimum}) \\ &\implies (Y - w^T X)X^T = 0 \implies (YX^T) = (w^T XX^T) \\ &\implies w^* = (XX^T)^{-1}XY^T \quad (\text{assumption: } X^T X \text{ is a full rank matrix}) \end{aligned}$$

This concludes the proof.

1.3 Appendix

1.3.1 Bayesian Linear Regression

Bayesian linear regression is a probabilistic approach to linear regression where model parameters are treated as random variables characterized by probability distributions rather than fixed values. Generally speaking,

in Bayesian Learning algorithms, including linear regression we integrate over the posterior of weights as opposed to using a point estimate of the parameters as in MLE, LSE, etc.

- Prior Distribution and Likelihood: Prior distribution captures the prior belief about the weights. Assuming Gaussian for the weights, $\theta = (\mathbf{w}, \sigma^2)$:

  $$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

  where $\mathbf{m}_0$ and $\mathbf{S}_0$ represent the mean and covariance matrix of the prior.

  Assuming Gaussian noise, the likelihood is given by:

  $$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) = \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I})$$

  where $\mathbf{X}$ is the design matrix, and $\mathbf{y}$ is the vector of observations.

- Posterior Distribution: Using prior and likelihood, the posterior is computed using Bayes' theorem:

  $$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})}$$

- Optimization at distribution level using KL Divergence: The Kullback-Leibler (KL) divergence is used to measure the difference between two distributions. In the context of Bayesian linear regression, it measures how close an approximate distribution $q(\mathbf{w})$ is to the true posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$:

  $$\mathrm{KL}(q(\mathbf{w})\|p(\mathbf{w}|\mathbf{X}, \mathbf{y})) = \int q(\mathbf{w}) \log \frac{q(\mathbf{w})}{p(\mathbf{w}|\mathbf{X}, \mathbf{y})}\, d\mathbf{w}$$

  Minimizing the KL divergence ensures that $q(\mathbf{w})$ is a good approximation of the posterior distribution.

This example shows in detail how we use optimization at distribution level in Bayesian Linear Regression algorithm.