Optimization for Machine Learning (CSL4010)

Dr. Md Abu Talhamainuddin Ansary
Department of Mathematics, IIT Jodhpur

Optimization problems

- A generalized optimization problem is of the form

$$
\begin{aligned}
(P): \quad \min_{x \in \mathbb{R}^n} \max \ f(x) & & \\
s.\,t.\ \ g_i(x) &\leq& 0 \ \ i = 1, 2, \ldots, m \\
h_j(x) &=& 0 j = 1, 2, \ldots, p
\end{aligned}
$$

- g_i, h_j are constraints.
- Set of feasible solutions

$$X = \{x \in \mathbb{R}^n | g_i(x) \leq 0, \ \ \forall \ \ i, \ \ h_j(x) = 0 \ \ \forall \text{ `} j\}$$

- If there is no constraints then it is called an unconstrained optimization problem.

- An organization has an abundance of two types of crude oil, called light crude and dark crude. It also has a refinery in which it can process light crude for $ 25 per barrel and dark crude for $17 per barrel. Processing yields fuel oil, gasoline, and jet fuel as indicated in the following table.

Output	Light crude	dark crude
fuel oil	0.21	0.55
gasoline	0.5	0.3
jet fuel	.25	0.1

  The organization requires 30 barrels of fuel oil, 70 barrels of gasoline, and 50 barrels of jet fuel. Construct optimization problem for minimizing processing cost.

Optimization problem construction

- An airline offers coach and first-class tickets. For the airline to be profitable, it must sell a minimum of 25 first-class tickets and a minimum of 40 coach tickets. The company makes a profit of \$225 for each coach ticket and \$200 for each first-class ticket. At most, the plane has a capacity of 150 travelers. How many of each ticket should be sold in order to maximize profits?

Optimization problem construction

- consider a discrete probability density corresponding to a measured value taking one of n values $x_1, x_2, \ldots, x_n$. The probability associated with x_i is p_i. The p_i's satisfy $p_i > 0$ and $\sum_{i=1}^{n} p_i = 1$. The entropy of such a density is $E = -\sum_{i=1}^{n} p_i \log(p_i)$. The mean value of the density is $\sum_{i=1}^{n} x_i p_i$. Construct an optimization problem to find maximum entropy when mean value of density is m.

- The data set $\mathcal{D}$ consists of m objects:

$$\mathcal{D} := \{(a^j, y_j), j = 1, 2, \ldots, m\} \tag{1}$$

where a^j is a vector (matrix) of features and y_j is an observation or level.

- The data set $\mathcal{D}$ consists of m objects:

$$\mathcal{D} := \{(a^j, y_j), j = 1, 2, \ldots, m\} \tag{1}$$

  where a^j is a vector (matrix) of features and y_j is an observation or level.
- The Data analysis task is to discover a function ϕ such that $\phi(a^j) \approx y_j$ for all $j = 1, 2, \ldots, m$.

- The data set $\mathcal{D}$ consists of m objects:

$$\mathcal{D} := \{(a^j, y_j), j = 1, 2, \ldots, m\} \tag{1}$$

where a^j is a vector (matrix) of features and y_j is an observation or level.

- The Data analysis task is to discover a function ϕ such that $\phi(a^j) \approx y_j$ for all $j = 1, 2, \ldots, m$.
- The process of discovering ϕ is often called “learning” or “training”.

- The data set $\mathcal{D}$ consists of m objects:

$$\mathcal{D} := \{(a^j, y_j), j = 1, 2, \ldots, m\} \tag{1}$$

  where a^j is a vector (matrix) of features and y_j is an observation or level.
- The Data analysis task is to discover a function ϕ such that $\phi(a^j) \approx y_j$ for all $j = 1, 2, \ldots, m$.
- The process of discovering ϕ is often called "learning" or "training".
- In practice, we use certain functions with some parameter x or X as ϕ.

- The data set $\mathcal{D}$ consists of m objects:

$$\mathcal{D} := \{(a^j, y_j), j = 1, 2, \ldots, m\} \tag{1}$$

  where a^j is a vector (matrix) of features and y_j is an observation or level.
- The Data analysis task is to discover a function ϕ such that $\phi(a^j) \approx y_j$ for all $j = 1, 2, \ldots, m$.
- The process of discovering ϕ is often called “learning” or “training”.
- In practice, we use certain functions with some parameter x or X as ϕ.
- With this parametrization, the problem identifying ϕ becomes a traditional data-fitting problem: i.e. *Find the parameter x defining ϕ such that $\phi(a^j) \approx y_j \; \forall \; j$ in some optimum sense.*

- The data set $\mathcal{D}$ consists of m objects:

$$\mathcal{D} := \{(a^j, y_j), j = 1, 2, \ldots, m\} \tag{1}$$

where a^j is a vector (matrix) of features and y_j is an observation or level.
- The Data analysis task is to discover a function ϕ such that $\phi(a^j) \approx y_j$ for all $j = 1, 2, \ldots, m$.
- The process of discovering ϕ is often called "learning" or "training".
- In practice, we use certain functions with some parameter x or X as ϕ.
- With this parametrization, the problem identifying ϕ becomes a traditional data-fitting problem: i.e. *Find the parameter x defining ϕ such that $\phi(a^j) \approx y_j \ \forall\ j$ in some optimum sense.*
- Thus we need to construct and solve an optimization problem.

- Frequently this optimization problem has objective function

$$\mathcal{L}_{\mathcal{D}}(x) := \frac{1}{m}\sum_{j=1}^{m} l(a^j, y_j; x) = \frac{1}{2m}\sum_{j=1}^{m}\left(\phi(a^j; x) - y_j\right)^2$$

- Frequently this optimization problem has objective function

$$\mathcal{L}_{\mathcal{D}}(x) := \frac{1}{m}\sum_{j=1}^{m} l(a^j, y_j; x) = \frac{1}{2m}\sum_{j=1}^{m}\left(\phi(a^j; x) - y_j\right)^2 \tag{2}$$

- The function l here represents a "loss" due to improper aligning $\phi(a)$ with y.

- Frequently this optimization problem has objective function

$$\mathcal{L}_{\mathcal{D}}(x) := \frac{1}{m}\sum_{j=1}^{m} l(a^j, y_j; x) = \frac{1}{2m}\sum_{j=1}^{m}\left(\phi(a^j; x) - y_j\right)^2 \tag{2}$$

- The function l here represents a "loss" due to improper aligning $\phi(a)$ with y.
- Thus the objective function $\mathcal{L}_{\mathcal{D}}$ represents the average loss accrued over the entire data set.

- Frequently this optimization problem has objective function

$$\mathcal{L}_{\mathcal{D}}(x) := \frac{1}{m}\sum_{j=1}^{m} l(a^j, y_j; x) = \frac{1}{2m}\sum_{j=1}^{m}\left(\phi(a^j; x) - y_j\right)^2 \qquad (2)$$

- The function l here represents a “loss” due to improper aligning $\phi(a)$ with y.
- Thus the objective function $\mathcal{L}_{\mathcal{D}}$ represents the average loss accrued over the entire data set.
- We have to solve the problem

$$\min_{x \in X \subseteq \mathbb{R}^n} \mathcal{L}_{\mathcal{D}}(x)$$

- Suppose x^* is the optimal solution of the problem (2), then we define $y_j \approx \phi(a^j : x^*)$.

- Frequently this optimization problem has objective function

$$\mathcal{L}_{\mathcal{D}}(x) := \frac{1}{m}\sum_{j=1}^{m} l(a^j, y_j; x) = \frac{1}{2m}\sum_{j=1}^{m}\left(\phi(a^j; x) - y_j\right)^2 \qquad (2)$$

- The function l here represents a “loss” due to improper aligning $\phi(a)$ with y.
- Thus the objective function $\mathcal{L}_{\mathcal{D}}$ represents the average loss accrued over the entire data set.
- We have to solve the problem

$$\min_{x\in X\subseteq\mathbb{R}^n} \quad \mathcal{L}_{\mathcal{D}}(x)$$

- Suppose x^* is the optimal solution of the problem (2), then we define $y_j \approx \phi(a^j : x^*)$.
- Given an unseen data $\hat{a}$ of type a^j, we predict $\hat{y} = \phi(\hat{a}) = \phi(\hat{a}; x^*)$.

- Apart from optimizing average loss $\mathcal{L}_{\mathcal{D}}$, we need to avoid *overfitting* the model to the data set $\mathcal{D}$.

- Apart from optimizing average loss $\mathcal{L}_{\mathcal{D}}$, we need to avoid *overfitting* the model to the data set $\mathcal{D}$.
- The data set $\mathcal{D}$ is a finite sample drawn from some underlying collection of possible data points.

- Apart from optimizing average loss $\mathcal{L}_{\mathcal{D}}$, we need to avoid *overfitting* the model to the data set $\mathcal{D}$.
- The data set $\mathcal{D}$ is a finite sample drawn from some underlying collection of possible data points.
- We wish to ϕ to perform well in unobserved data set as well as $\mathcal{D}$.

- Apart from optimizing average loss $\mathcal{L}_{\mathcal{D}}$, we need to avoid *overfitting* the model to the data set $\mathcal{D}$.
- The data set $\mathcal{D}$ is a finite sample drawn from some underlying collection of possible data points.
- We wish to ϕ to perform well in unobserved data set as well as $\mathcal{D}$.
- That is, ϕ should not be to sensitive to the sample $\mathcal{D}$.

- Apart from optimizing average loss $\mathcal{L}_{\mathcal{D}}$, we need to avoid *overfitting* the model to the data set $\mathcal{D}$.
- The data set $\mathcal{D}$ is a finite sample drawn from some underlying collection of possible data points.
- We wish to ϕ to perform well in unobserved data set as well as $\mathcal{D}$.
- That is, ϕ should not be to sensitive to the sample $\mathcal{D}$.
- One possible way to avoid this is to modify the objective function by adding some penalty function or constraints in a way that limits 'complexity' of ϕ.

- Apart from optimizing average loss $\mathcal{L}_{\mathcal{D}}$, we need to avoid *overfitting* the model to the data set $\mathcal{D}$.
- The data set $\mathcal{D}$ is a finite sample drawn from some underlying collection of possible data points.
- We wish to ϕ to perform well in unobserved data set as well as $\mathcal{D}$.
- That is, ϕ should not be to sensitive to the sample $\mathcal{D}$.
- One possible way to avoid this is to modify the objective function by adding some penalty function or constraints in a way that limits 'complexity' of ϕ.
- This process is called *regularization*.

- Apart from optimizing average loss $\mathcal{L}_{\mathcal{D}}$, we need to avoid *overfitting* the model to the data set $\mathcal{D}$.
- The data set $\mathcal{D}$ is a finite sample drawn from some underlying collection of possible data points.
- We wish to ϕ to perform well in unobserved data set as well as $\mathcal{D}$.
- That is, ϕ should not be to sensitive to the sample $\mathcal{D}$.
- One possible way to avoid this is to modify the objective function by adding some penalty function or constraints in a way that limits 'complexity' of ϕ.
- This process is called *regularization*.
- A modified optimization model that balances fit to the data set, model complexity,

$$\min_{x \in \Omega} \ \mathcal{L}_{\mathcal{D}}(x) + \lambda pen(x)$$

- Apart from optimizing average loss $\mathcal{L}_{\mathcal{D}}$, we need to avoid *overfitting* the model to the data set $\mathcal{D}$.
- The data set $\mathcal{D}$ is a finite sample drawn from some underlying collection of possible data points.
- We wish to ϕ to perform well in unobserved data set as well as $\mathcal{D}$.
- That is, ϕ should not be to sensitive to the sample $\mathcal{D}$.
- One possible way to avoid this is to modify the objective function by adding some penalty function or constraints in a way that limits 'complexity' of ϕ.
- This process is called *regularization*.
- A modified optimization model that balances fit to the data set, model complexity,

$$\min_{x \in \Omega} \; \mathcal{L}_{\mathcal{D}}(x) + \lambda pen(x) \tag{3}$$

where Ω is the set of allowable values for x, $pen(.)$ is the regularization function or regularizer, and $\lambda \geq 0$ is the regularization parameter.

Example 1

- We have collected the memory size, Ram and price of some smartphones from amazon.com. Details are given int the table:

Memory (GB)	Ram (Ram)	Price (thousand rs)
32	2	7
32	3	8
64	4	10
128	8	19.5
256	8	25

Table 1: Data set based on mobile quality and price

Example 1

- We have collected the memory size, Ram and price of some smartphones from amazon.com. Details are given int the table:

Memory (GB)	Ram (Ram)	Price (thousand rs)
32	2	7
32	3	8
64	4	10
128	8	19.5
256	8	25

Table 1: Data set based on mobile quality and price

- Thus we have a data set $\mathcal{D} := \{(a^j, y_j), j = 1, 2, 3, 4\}$, where $a^j = (Memory,\ Ram)^T$, and $y_j = Price$.

Example 1

- We have collected the memory size, Ram and price of some smartphones from amazon.com. Details are given int the table:

Memory (GB)	Ram (Ram)	Price (thousand rs)
32	2	7
32	3	8
64	4	10
128	8	19.5
256	8	25

Table 1: Data set based on mobile quality and price

- Thus we have a data set $\mathcal{D} := \{(a^j, y_j), j = 1, 2, 3, 4\}$, where $a^j = (Memory,\ Ram)^T$, and $y_j = Price$.
- We have to predict the price of a mobile of 512 GB memory and 12 GB Ram.

Example 1

- We have collected the memory size, Ram and price of some smartphones from amazon.com. Details are given int the table:

Memory (GB)	Ram (Ram)	Price (thousand rs)
32	2	7
32	3	8
64	4	10
128	8	19.5
256	8	25

Table 1: Data set based on mobile quality and price

- Thus we have a data set $\mathcal{D} := \{(a^j, y_j), j = 1, 2, 3, 4\}$, where $a^j = (Memory,\ Ram)^T$, and $y_j = Price$.
- We have to predict the price of a mobile of 512 GB memory and 12 GB Ram.
- We need to construct a function ϕ such that $\phi(a^j) \approx\ y_j, \forall\ j$.

Example 1

- We have collected the memory size, Ram and price of some smartphones from amazon.com. Details are given int the table:

Memory (GB)	Ram (Ram)	Price (thousand rs)
32	2	7
32	3	8
64	4	10
128	8	19.5
256	8	25

Table 1: Data set based on mobile quality and price

- Thus we have a data set $\mathcal{D} := \{(a^j, y_j), j = 1, 2, 3, 4\}$, where $a^j = (Memory,\ Ram)^T$, and $y_j = Price$.
- We have to predict the price of a mobile of 512 GB memory and 12 GB Ram.
- We need to construct a function ϕ such that $\phi(a^j) \approx\ y_j, \forall\ j$.
- Using this function we can find the value $\hat{y} = \phi(\hat{a})$ where $\hat{a} = (512, 12)^T$.

Example 1

- We have collected the memory size, Ram and price of some smartphones from amazon.com. Details are given int the table:

Memory (GB)	Ram (Ram)	Price (thousand rs)
32	2	7
32	3	8
64	4	10
128	8	19.5
256	8	25

Table 1: Data set based on mobile quality and price

- Thus we have a data set $\mathcal{D} := \{(a^j, y_j), j = 1, 2, 3, 4\}$, where $a^j = (Memory,\ Ram)^T$, and $y_j = Price$.
- We have to predict the price of a mobile of 512 GB memory and 12 GB Ram.
- We need to construct a function ϕ such that $\phi(a^j) \approx\ y_j, \forall\, j$.
- Using this function we can find the value $\hat{y} = \phi(\hat{a})$ where $\hat{a} = (512, 12)^T$.
- Since data size 5 is very less than the number of mobiles available in this cite, we need to add some regularization to predict the price more accurately.

Least Square Problem

- Least square solution is the oldest data analysis technique.

Least Square Problem

- Least square solution is the oldest data analysis technique.
- Define $\phi(a;x) = a^T x$ and $l(a^j, y_j; x) = \frac{1}{2}|{a^j}^T x - y_j|^2$. Then from (2),

$$\mathcal{L}_{\mathcal{D}}(x) := \frac{1}{2m}\sum_{j=1}^{m} |{a^j}^T x - y_j|^2$$

Least Square Problem

- Least square solution is the oldest data analysis technique.
- Define $\phi(a; x) = a^T x$ and $l(a^j, y_j; x) = \frac{1}{2}|{a^j}^T x - y_j|^2$. Then from (2),

$$\mathcal{L}_{\mathcal{D}}(x) := \frac{1}{2m}\sum_{j=1}^{m}|{a^j}^T x - y_j|^2$$

- Define $A = \begin{bmatrix} {a^1}^T \\ {a^2}^T \\ \dots \\ {a^m}^T \end{bmatrix}$ and $y = (y_1, y_2, \dots, y_m)^T$. Then the optimization problem becomes

$$\min_{x\in\mathbb{R}^n} \frac{1}{2m}\|Ax - y\|^2 \tag{4}$$

- The above problem is an unconstrained optimization problem.

- Introducing penalty function $p(x) = \sum_{i=1}^{n} |x_i| = \|x\|_1$ with penalty parameter λ, the least square problem can be revised as

$$\min_{x \in \mathbb{R}^n} \frac{1}{2m}\|Ax - y\|^2 + \frac{\lambda}{2}\|x\|_1$$

- Introducing penalty function $p(x) = \sum_{i=1}^{n} |x_i| = \|x\|_1$ with penalty parameter λ, the least square problem can be revised as

$$\min_{x \in \mathbb{R}^n} \frac{1}{2m} \|Ax - y\|^2 + \frac{\lambda}{2} \|x\|_1$$

- The above problem is non-differentiable.

Support vector machine

- Consider a data set $\mathcal{D} = \{(x^i, y_i), i = 1, 2, \ldots, N\}$ where $y_i \in \{-1, 1\}$.
- Best on performance in recent matches some cricketers are identified as batsman/bowler.

Total runs	Strike rate	Wicket	Batsman/Bowler
276	147	0	Batsman
15	214	2	Bowler
79	144	8	Bowler
111	139	2	Batsman

- We have to identify a new cricketer with either as bowler or as batsman.
- We have to solve the following optimization problem

$$(P_{svm}): \quad \min_{w,b} \quad \frac{1}{2}\|w\|^2$$
$$s.t. \;\; 1 - y_i(w^T x^i + b) \leq \quad 0 \quad \forall \;\; i$$

- This is a constrained nonlinear programming problem.
- Suppose (w^*, b^*) is the optimal solution of the problem.
- For $\hat{a}$, we can predict $\hat{y} = sgn({w^*}^T \hat{a} + b^*)$.

Logistic regression

- Logistic regression can be vowed as soften form of binary support vector machine.
- We seek an "odd function" p parametrized by $x \in \mathbb{R}^n$,

$$p(a, x) := \frac{1}{1 + e^{a^T x}}$$

- Our aim to find x such that

$$\begin{aligned} p(a^j; x) &\approx 1 \quad \text{when } y_j = +1 \\ p(a^j; x) &\approx 0 \quad \text{when } y_j = -1 \end{aligned}$$

- The optimal value of x can be found by minimizing negative-log-likelihood function

$$L(x) := -\frac{1}{m}\left[\sum_{j: y_j = -1} \log(1 - p(a^j; x)) + \sum_{j: y_j = +1} \log(p(a^j; x))\right]$$

- Adding penalty function $\|x\|_1$ the optimization problem can be written as

$$L(x) := \min \; -\frac{1}{m}\left[\sum_{j: y_j = -1} \log(1 - p(a^j; x)) + \sum_{j: y_j = +1} \log(p(a^j; x))\right] + \frac{\lambda}{2}\|x\|_1$$

- Boyd, S. and Vandenberghe, L.: Convex optimization. Cambridge university press, 2004.
- Boyd, S. and Vandenberghe L.: Introduction to applied linear algebra: vectors, matrices, and least squares. Cambridge university press, 2018.