Optimization Theory and its Applications to Machine Learning

Thirumulanathan D
Assistant Professor
Indian Institute of Technology, Kanpur.

25-Feb-2021

Outline

1. Motivation
2. Optimizing one variable
3. Optimizing multiple variables
4. Constrained optimization
5. Application to machine learning problems

Overview

1. Motivation
2. Optimizing one variable
3. Optimizing multiple variables
4. Constrained optimization
5. Application to machine learning problems

Some Practical Examples

- What is the need for optimization theory? Where do we use it?

Some Practical Examples

- What is the need for optimization theory? Where do we use it?
- **Time management:** I have to finish the following by today – eight hours of work in the office, two hours of meeting, buying groceries/ medicines for the home, take the child to the hospital for a checkup, prepare for the meetings tomorrow, have food, complete my personal deeds, spend at least an hour with the family, and have at least six hours of sleep.

Some Practical Examples

- What is the need for optimization theory? Where do we use it?
- **Time management:** I have to finish the following by today – eight hours of work in the office, two hours of meeting, buying groceries/ medicines for the home, take the child to the hospital for a checkup, prepare for the meetings tomorrow, have food, complete my personal deeds, spend at least an hour with the family, and have at least six hours of sleep.
- I optimize the time of 24 hours to maximize the utility value. In other words, I solve the optimization problem:

$$\max_{t_1,\ldots,t_n} \; u_1(t_1) + \ldots + u_n(t_n)$$
$$\text{subject to } t_1 + \ldots + t_n \leq 24.$$

Some Practical Examples

- What is the need for optimization theory? Where do we use it?
- **Time management:** I have to finish the following by today – eight hours of work in the office, two hours of meeting, buying groceries/ medicines for the home, take the child to the hospital for a checkup, prepare for the meetings tomorrow, have food, complete my personal deeds, spend at least an hour with the family, and have at least six hours of sleep.
- I optimize the time of 24 hours to maximize the utility value. In other words, I solve the optimization problem:

$$\max_{t_1,\ldots,t_n} \; u_1(t_1) + \ldots + u_n(t_n)$$
$$\text{subject to } t_1 + \ldots + t_n \leq 24.$$

- Optimization problem involves either utility maximization, or cost minimization.
  - Maximize earnings, subject to a maximum limit on work hours.
  - Minimize work hours, subject to a minimum limit on salary.

Applications

- **Hypothesis testing:** Consider the data obtained from a radar as $Y = X + N$. The signal from the radar is $X = \{-1, 1\}$, but we receive it with a Gaussian noise N. We need to decide whether to raise an alarm or not, based on Y. Our objective is to minimize the probabilities of both miss detection (MD) and false alarm (FA).

Applications

- **Hypothesis testing:** Consider the data obtained from a radar as $Y = X + N$. The signal from the radar is $X = \{-1, 1\}$, but we receive it with a Gaussian noise N. We need to decide whether to raise an alarm or not, based on Y. Our objective is to minimize the probabilities of both miss detection (MD) and false alarm (FA).
- We solve the one of following problems:
  - $\min P(FA)$ subject to $P(MD) \leq \epsilon$.
  - $\min P(MD)$ subject to $P(FA) \leq \epsilon'$.

Applications

- **Hypothesis testing:** Consider the data obtained from a radar as $Y = X + N$. The signal from the radar is $X = \{-1, 1\}$, but we receive it with a Gaussian noise N. We need to decide whether to raise an alarm or not, based on Y. Our objective is to minimize the probabilities of both miss detection (MD) and false alarm (FA).
- We solve the one of following problems:
  - $\min P(FA)$ subject to $P(MD) \leq \epsilon$.
  - $\min P(MD)$ subject to $P(FA) \leq \epsilon'$.
- **Microeconomics:** Consider a consumer buying n products with an income of w. He obtains a utility of $u(x_1, x_2, \ldots, x_n)$ by buying x_i quantities of product i. But he needs to pay p_i to buy a unit of product i. He then maximizes his utility by solving the following problem:

$$
\begin{aligned}
&\max_{x_1,\ldots,x_n} u(x_1, \ldots, x_n) \\
\text{subject to} \quad &\text{(i) } p_1 x_1 + \ldots + p_n x_n \leq w, \\
&\text{(ii) } x_i \geq 0,\ i = 1, \ldots, n.
\end{aligned}
$$

Applications (contd...)

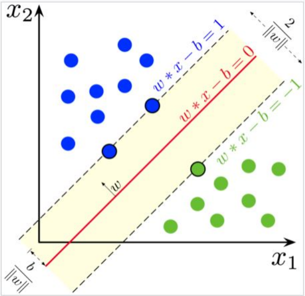
- **Support Vector Machines:** Consider the training data

$$(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_n, y_n)$$

for classifying the data as $+1$ or -1. The classification occurs by a hyperplane vector w, which is found by solving

$$\min_{\boldsymbol{w}, b} ||\boldsymbol{w}||^2$$

$$\text{subject to } y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1,\ i = 1, \dots, n.$$

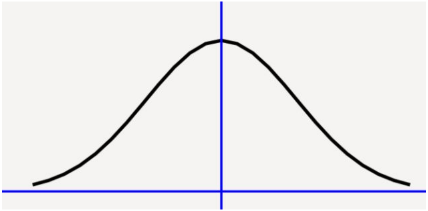


Overview

1. Motivation
2. Optimizing one variable
3. Optimizing multiple variables
4. Constrained optimization
5. Application to machine learning problems

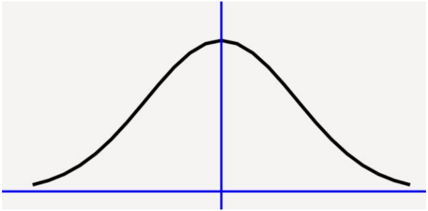
Maximizing a function

- Consider $\max_x e^{-x^2/2}$. The maximum occurs at $x = 0$.



Maximizing a function

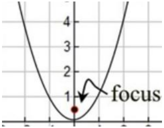
- Consider $\max_x e^{-x^2/2}$. The maximum occurs at $x = 0$.



- Let $f(x) = e^{-x^2/2}$. Then, $f'(x) = -xe^{-x^2/2} = 0$ implies $x = 0$. So the maximum of a function can be found by computing x^* that satisfies $f'(x^*) = 0$.

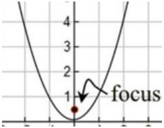
Minimizing a function

- Consider $\min_x x^2$. The minimum occurs at $x = 0$.



Minimizing a function

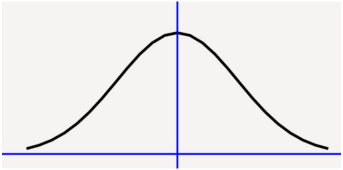
- Consider $\min_x x^2$. The minimum occurs at $x = 0$.

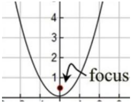


- Let $f(x) = x^2$. Then, $f'(x) = 2x = 0$ implies $x = 0$. So the minimum of a function can be found by computing x^* that satisfies $f'(x^*) = 0$.

Critical Point

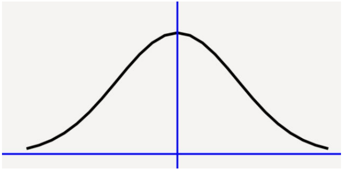
- Both the maximum and the minimum of a function can be found by equating $f'(x) = 0$.
  - $f'(x) > 0 \Rightarrow f(x^-) < f(x)$, $f(x^+) > f(x)$.
  - $f'(x) < 0 \Rightarrow f(x^-) > f(x)$, $f(x^+) < f(x)$.

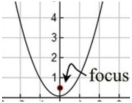




Critical Point

- Both the maximum and the minimum of a function can be found by equating $f'(x) = 0$.
  - $f'(x) > 0 \Rightarrow f(x^-) < f(x)$, $f(x^+) > f(x)$.
  - $f'(x) < 0 \Rightarrow f(x^-) > f(x)$, $f(x^+) < f(x)$.

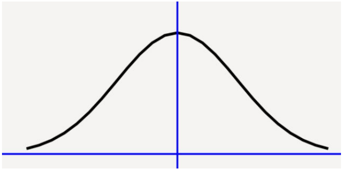


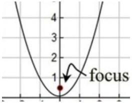


- Define *critical points* of a function as the set $\{x^* : f'(x^*) = 0\}$. Any extremum point is a critical point.

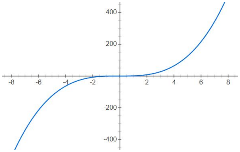
Critical Point

- Both the maximum and the minimum of a function can be found by equating $f'(x) = 0$.
  - $f'(x) > 0 \Rightarrow f(x^-) < f(x)$, $f(x^+) > f(x)$.
  - $f'(x) < 0 \Rightarrow f(x^-) > f(x)$, $f(x^+) < f(x)$.



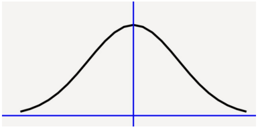


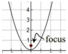
- Define *critical points* of a function as the set $\{x^* : f'(x^*) = 0\}$. Any extremum point is a critical point.
- A critical point can also be a saddle point. Let $f(x) = x^3$, so $f'(x) = 3x^2 = 0$ implies $x = 0$. But not an extremum point.

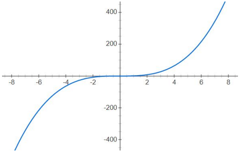


Second Order Conditions

- Given the set of critical points, how do we find whether it is a maximum, a minimum, or a saddle point?
  - Maximum occurs when $f'(x^-) > 0$, $f'(x) = 0$, and $f'(x^+) < 0$. So $f''(x) \leq 0$.
  - Minimum occurs when $f'(x^-) < 0$, $f'(x) = 0$, and $f'(x^+) > 0$. So $f''(x) \geq 0$.
  - Saddle point occurs when $f'(x^-)f'(x^+) > 0$. So we have $f''(x) = 0$.

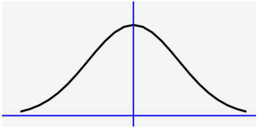


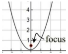


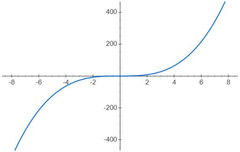


Second Order Conditions

- Given the set of critical points, how do we find whether it is a maximum, a minimum, or a saddle point?
  - Maximum occurs when $f'(x^-) > 0$, $f'(x) = 0$, and $f'(x^+) < 0$. So $f''(x) \leq 0$.
  - Minimum occurs when $f'(x^-) < 0$, $f'(x) = 0$, and $f'(x^+) > 0$. So $f''(x) \geq 0$.
  - Saddle point occurs when $f'(x^-)f'(x^+) > 0$. So we have $f''(x) = 0$.



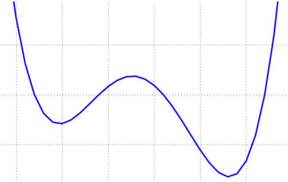




- The second order conditions are
  - $\{f'(x) = 0, f''(x) > 0\} \Rightarrow x$ is the minimizer.
  - $\{f'(x) = 0, f''(x) < 0\} \Rightarrow x$ is the maximizer.
  - $\{f'(x) = 0, f''(x) = 0\} \Rightarrow$ further probe is required.

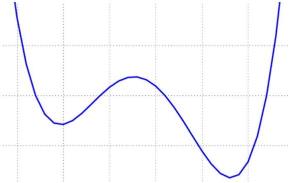
Local and global extremum

- Consider $\min_x (x-\alpha)(x-\beta)(x-\gamma)(x-\delta)$. The curve looks as follows:



Local and global extremum

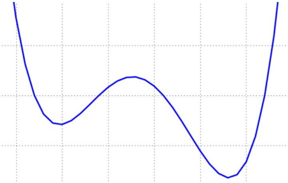
- Consider $\min_x (x-\alpha)(x-\beta)(x-\gamma)(x-\delta)$. The curve looks as follows:



- We have three critical points, two of which satisfying the condition for minimizer. But only one of the two is the minimum.

Local and global extremum

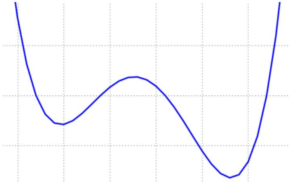
- Consider $\min_x (x-\alpha)(x-\beta)(x-\gamma)(x-\delta)$. The curve looks as follows:



- We have three critical points, two of which satisfying the condition for minimizer. But only one of the two is the minimum.
- The conditions we derived until now, helps us only to compute the local extremum, and NOT the global extremum.

Local and global extremum

- Consider $\min_x (x-\alpha)(x-\beta)(x-\gamma)(x-\delta)$. The curve looks as follows:



- We have three critical points, two of which satisfying the condition for minimizer. But only one of the two is the minimum.
- The conditions we derived until now, helps us only to compute the local extremum, and NOT the global extremum.
- The global extremum needs to be computed by comparing the local extremum points.

So far...

- Consider the problem $\max_x f(x)$ or $\min_x f(x)$. The local extremum of this function can be computed by the following method:
  - Compute the set of critical points $\{x^* : f'(x^*) = 0\}$.
  - Among the critical points, $f''(x^*) > 0 \Rightarrow x^*$ is the (local) minimizer; $f''(x^*) < 0 \Rightarrow x^*$ is the (local) maximizer; $f''(x^*) = 0 \Rightarrow$ further probe is required.

So far...

- Consider the problem $\max_x f(x)$ or $\min_x f(x)$. The local extremum of this function can be computed by the following method:
  - Compute the set of critical points $\{x^* : f'(x^*) = 0\}$.
  - Among the critical points, $f''(x^*) > 0 \Rightarrow x^*$ is the (local) minimizer; $f''(x^*) < 0 \Rightarrow x^*$ is the (local) maximizer; $f''(x^*) = 0 \Rightarrow$ further probe is required.
- Global extremum can be computed by finding the extremum among the local extrema.

Overview

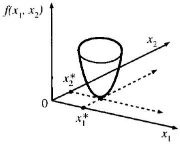
1. Motivation
2. Optimizing one variable
3. Optimizing multiple variables
4. Constrained optimization
5. Application to machine learning problems

Many variables

- How do we optimize over multiple variables?

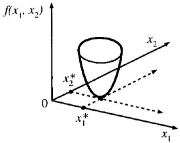
Many variables

- How do we optimize over multiple variables?
- Consider $\min_{x_1, x_2} (x_1 - x_1^*)^2 + (x_2 - x_2^*)^2$. The minimum occurs at (x_1^*, x_2^*).



Many variables

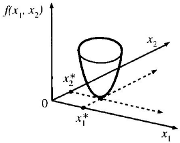
- How do we optimize over multiple variables?
- Consider $\min_{x_1,x_2}(x_1 - x_1^*)^2 + (x_2 - x_2^*)^2$. The minimum occurs at (x_1^*, x_2^*).



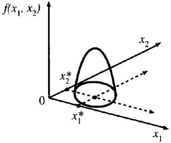
- The gradient of $f(x_1, x_2)$, $\nabla f = \begin{bmatrix} 2(x_1 - x_1^*) \\ 2(x_2 - x_2^*) \end{bmatrix}$. Equating $\nabla f = 0$, we obtain the minimizer (x_1^*, x_2^*).

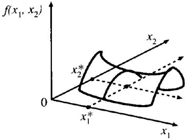
Many variables

- How do we optimize over multiple variables?
- Consider $\min_{x_1,x_2}(x_1 - x_1^*)^2 + (x_2 - x_2^*)^2$. The minimum occurs at (x_1^*, x_2^*).



- The gradient of $f(x_1, x_2)$, $\nabla f = \begin{bmatrix} 2(x_1 - x_1^*) \\ 2(x_2 - x_2^*) \end{bmatrix}$. Equating $\nabla f = 0$, we obtain the minimizer (x_1^*, x_2^*).
- When $f(x_1, x_2) = -(x_1 - x_1^*)^2 - (x_2 - x_2^*)^2$, we have a maximizer at (x_1^*, x_2^*). Similarly, we have a saddle point at (x_1^*, x_2^*) when $f(x_1, x_2) = (x_1 - x_1^*)^2 - (x_2 - x_2^*)^2$.





Critical Point

- The set of critical points are computed in a similar manner: $\{(x_1^*, \ldots, x_n^*) : \nabla f(x_1^*, \ldots, x_n^*) = 0\}$.

Critical Point

- The set of critical points are computed in a similar manner: $\{(x_1^*, \ldots, x_n^*) : \nabla f(x_1^*, \ldots, x_n^*) = 0\}$.
- But a critical point could either be a local maximum, a local minimum, or a saddle point. Thus second-order conditions are required.

Critical Point

- The set of critical points are computed in a similar manner: $\{(x_1^*, \ldots, x_n^*) : \nabla f(x_1^*, \ldots, x_n^*) = 0\}$.
- But a critical point could either be a local maximum, a local minimum, or a saddle point. Thus second-order conditions are required.
- Define Hessian matrix $\nabla_2 f$ as follows.

$$\nabla_2 f = \begin{bmatrix} \frac{\partial}{\partial x_1^2} f & \frac{\partial}{\partial x_1 x_2} f & \cdots & \frac{\partial}{\partial x_1 x_n} f \\ \frac{\partial}{\partial x_2 x_1} f & \frac{\partial}{\partial x_2^2} f & \cdots & \frac{\partial}{\partial x_2 x_n} f \\ \vdots & \vdots & & \vdots \\ \frac{\partial}{\partial x_n x_1} f & \frac{\partial}{\partial x_n x_2} f & \cdots & \frac{\partial}{\partial x_n^2} f \end{bmatrix}$$

Critical Point

- The set of critical points are computed in a similar manner: $\{(x_1^*, \ldots, x_n^*) : \nabla f(x_1^*, \ldots, x_n^*) = 0\}$.
- But a critical point could either be a local maximum, a local minimum, or a saddle point. Thus second-order conditions are required.
- Define Hessian matrix $\nabla_2 f$ as follows.

$$\nabla_2 f = \begin{bmatrix} \frac{\partial}{\partial x_1^2} f & \frac{\partial}{\partial x_1 x_2} f & \cdots & \frac{\partial}{\partial x_1 x_n} f \\ \frac{\partial}{\partial x_2 x_1} f & \frac{\partial}{\partial x_2^2} f & \cdots & \frac{\partial}{\partial x_2 x_n} f \\ \vdots & \vdots & & \vdots \\ \frac{\partial}{\partial x_n x_1} f & \frac{\partial}{\partial x_n x_2} f & \cdots & \frac{\partial}{\partial x_n^2} f \end{bmatrix}$$

- A critical point turns out to be a
  - (local) maximum when $\nabla_2 f(x_1^*, \ldots, x_n^*) \preceq 0$,
  - (local) minimum when $\nabla_2 f(x_1^*, \ldots, x_n^*) \succeq 0$,
  - saddle point when $\nabla_2 f(x_1^*, \ldots, x_n^*) \not\prec 0$ and $\nabla_2 f(x_1^*, \ldots, x_n^*) \not\succ 0$,

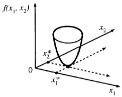
Definiteness of a matrix

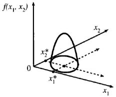
- A matrix Q is
  - *positive definite* ($Q \succ 0$), if $\boldsymbol{x}^T Q\boldsymbol{x} > 0$ for all $\boldsymbol{x} \neq 0$,
  - *positive semi-definite* ($Q \succeq 0$), if $\boldsymbol{x}^T Q\boldsymbol{x} \geq 0$ for all $\boldsymbol{x} \neq 0$,
  - *negative definite* ($Q \prec 0$), if $\boldsymbol{x}^T Q\boldsymbol{x} < 0$ for all $\boldsymbol{x} \neq 0$,
  - *negative semi-definite* ($Q \preceq 0$), if $\boldsymbol{x}^T Q\boldsymbol{x} \leq 0$ for all $\boldsymbol{x} \neq 0$,
  - *indefinite* if $\boldsymbol{x}^T Q\boldsymbol{x} > 0$ for some $\boldsymbol{x}$, and $\boldsymbol{x}^T Q\boldsymbol{x} < 0$ for some $\boldsymbol{x}$.

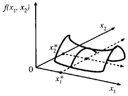
Definiteness of a matrix

- A matrix Q is
  - *positive definite* ($Q \succ 0$), if $\boldsymbol{x}^T Q \boldsymbol{x} > 0$ for all $\boldsymbol{x} \neq 0$,
  - *positive semi-definite* ($Q \succeq 0$), if $\boldsymbol{x}^T Q \boldsymbol{x} \geq 0$ for all $\boldsymbol{x} \neq 0$,
  - *negative definite* ($Q \prec 0$), if $\boldsymbol{x}^T Q \boldsymbol{x} < 0$ for all $\boldsymbol{x} \neq 0$,
  - *negative semi-definite* ($Q \preceq 0$), if $\boldsymbol{x}^T Q \boldsymbol{x} \leq 0$ for all $\boldsymbol{x} \neq 0$,
  - *indefinite* if $\boldsymbol{x}^T Q \boldsymbol{x} > 0$ for some $\boldsymbol{x}$, and $\boldsymbol{x}^T Q \boldsymbol{x} < 0$ for some $\boldsymbol{x}$.
- Examples are as follows:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \succ 0, \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \prec 0, \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \not\succeq 0, \not\preceq 0.$$



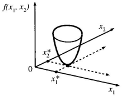


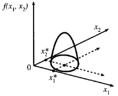


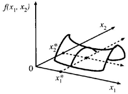
Definiteness of a matrix

- A matrix Q is
  - *positive definite* ($Q \succ 0$), if $\boldsymbol{x}^T Q \boldsymbol{x} > 0$ for all $\boldsymbol{x} \neq 0$,
  - *positive semi-definite* ($Q \succeq 0$), if $\boldsymbol{x}^T Q \boldsymbol{x} \geq 0$ for all $\boldsymbol{x} \neq 0$,
  - *negative definite* ($Q \prec 0$), if $\boldsymbol{x}^T Q \boldsymbol{x} < 0$ for all $\boldsymbol{x} \neq 0$,
  - *negative semi-definite* ($Q \preceq 0$), if $\boldsymbol{x}^T Q \boldsymbol{x} \leq 0$ for all $\boldsymbol{x} \neq 0$,
  - *indefinite* if $\boldsymbol{x}^T Q \boldsymbol{x} > 0$ for some $\boldsymbol{x}$, and $\boldsymbol{x}^T Q \boldsymbol{x} < 0$ for some $\boldsymbol{x}$.
- Examples are as follows:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \succ 0, \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \prec 0, \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \not\succeq 0, \not\preceq 0.$$







- So the second order conditions are
  - $\{\nabla f(\boldsymbol{x}^*) = 0, \nabla_2 f(\boldsymbol{x}^*) \succ 0\} \Rightarrow \boldsymbol{x}^*$ is the minimizer.
  - $\{\nabla f(\boldsymbol{x}^*) = 0, \nabla_2 f(\boldsymbol{x}^*) \prec 0\} \Rightarrow \boldsymbol{x}^*$ is the maximizer.
  - $\{\nabla f(\boldsymbol{x}^*) = 0, \boldsymbol{d}^T \nabla_2 f(\boldsymbol{x}^*) \boldsymbol{d} = 0$ for $\boldsymbol{d} \neq 0\} \Rightarrow$ further probe is required.

Overview

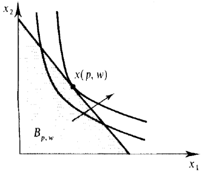
1. Motivation
2. Optimizing one variable
3. Optimizing multiple variables
4. Constrained optimization
5. Application to machine learning problems

Constraint set

- *Recall:* All practical examples that we considered have some constraints on the variables to be optimized.

Constraint set

- *Recall:* All practical examples that we considered have some constraints on the variables to be optimized.
- **Microeconomics:** There are n products with price $(p_1, \ldots, p_n)$. A consumer with an income of w wants to maximize his utility $x_1 x_2 \ldots x_n$. We have the following problem:



$$
\begin{aligned}
&\max_{x_1, \ldots, x_n} (x_1 x_2 \ldots x_n) \\
\text{s.t.} \quad &\text{(i) } p_1 x_1 + \ldots + p_n x_n \leq w, \\
&\text{(ii) } x_i \geq 0, \; i = 1, \ldots, n.
\end{aligned}
$$

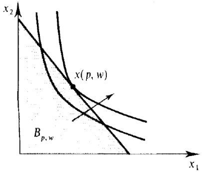
Constraint set

- *Recall:* All practical examples that we considered have some constraints on the variables to be optimized.
- **Microeconomics:** There are n products with price $(p_1, \ldots, p_n)$. A consumer with an income of w wants to maximize his utility $x_1 x_2 \ldots x_n$. We have the following problem:

$$\max_{x_1,\ldots,x_n} (x_1 x_2 \ldots x_n)$$

$$\text{s.t.} \quad \text{(i) } p_1 x_1 + \ldots + p_n x_n \leq w,$$

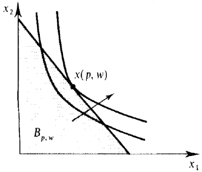
$$\text{(ii) } x_i \geq 0,\ i = 1, \ldots, n.$$



- The curve $x_1 x_2 = k$ is a rectangular hyperbola. So we need to find the value of k that just touches the boundary of the constraint set. When $k = \frac{w^2}{4p_1p_2}$, the curve touches the boundary only at $(\frac{w}{2p_1}, \frac{w}{2p_2})$. So consumer buys this quantity.

Constraint set

- *Recall:* All practical examples that we considered have some constraints on the variables to be optimized.
- **Microeconomics:** There are n products with price $(p_1, \ldots, p_n)$. A consumer with an income of w wants to maximize his utility $x_1 x_2 \ldots x_n$. We have the following problem:



$$\begin{aligned} &\max_{x_1,\ldots,x_n} (x_1 x_2 \ldots x_n) \\ \text{s.t.} \quad &\text{(i) } p_1 x_1 + \ldots + p_n x_n \leq w, \\ &\text{(ii) } x_i \geq 0, \; i = 1, \ldots, n. \end{aligned}$$

- The curve $x_1 x_2 = k$ is a rectangular hyperbola. So we need to find the value of k that just touches the boundary of the constraint set. When $k = \frac{w^2}{4p_1p_2}$, the curve touches the boundary only at $(\frac{w}{2p_1}, \frac{w}{2p_2})$. So consumer buys this quantity.
- How do we compute the answer for n variables?

Lagrangian method

- Consider the Lagrangian function

$$\mathcal{L} = x_1 \dots x_n + \lambda\Big(\sum_i p_i x_i - w\Big) - \sum_j \mu_j x_j.$$

Lagrangian method

- Consider the Lagrangian function

$$\mathcal{L} = x_1 \dots x_n + \lambda(\sum_i p_i x_i - w) - \sum_j \mu_j x_j.$$

- We now maximize this function (instead of $x_1 \dots x_n$), subject to the following conditions:

$$\lambda(\sum_i p_i x_i - w) = 0, \quad \mu_j x_j = 0, j = 1, \dots, n, \quad \lambda, \mu_j \geq 0.$$

Lagrangian method

- Consider the Lagrangian function

$$\mathcal{L} = x_1 \dots x_n + \lambda(\sum_i p_i x_i - w) - \sum_j \mu_j x_j.$$

- We now maximize this function (instead of $x_1 \dots x_n$), subject to the following conditions:

$$\lambda(\sum_i p_i x_i - w) = 0, \quad \mu_j x_j = 0, j = 1, \dots, n, \quad \lambda, \mu_j \geq 0.$$

- $(\nabla L)_i = \frac{x_1 \dots x_n}{x_i} + \lambda p_i - \mu_i$. So $(\nabla \mathcal{L})_i = 0$ for all i implies $\lambda p_i x_i - \mu_i x_i = k'$ (constant) for all i.
- Choose $\mu_i = 0$. We also need $\sum_i p_i x_i = w$. So $\lambda = \frac{nk'}{w}$ which implies $x_i = \frac{w}{np_i}$. We have $(x_1^*, \dots, x_n^*) = (\frac{w}{np_1}, \dots, \frac{w}{np_n})$.

Lagrangian method

- Consider the Lagrangian function
$$\mathcal{L} = x_1 \dots x_n + \lambda(\sum_i p_i x_i - w) - \sum_j \mu_j x_j.$$
- We now maximize this function (instead of $x_1 \dots x_n$), subject to the following conditions:
$$\lambda(\sum_i p_i x_i - w) = 0, \quad \mu_j x_j = 0, j = 1, \dots, n, \quad \lambda, \mu_j \geq 0.$$
- $(\nabla L)_i = \frac{x_1 \dots x_n}{x_i} + \lambda p_i - \mu_i$. So $(\nabla \mathcal{L})_i = 0$ for all i implies $\lambda p_i x_i - \mu_i x_i = k'$ (constant) for all i.
- Choose $\mu_i = 0$. We also need $\sum_i p_i x_i = w$. So $\lambda = \frac{nk'}{w}$ which implies $x_i = \frac{w}{np_i}$. We have $(x_1^*, \dots, x_n^*) = (\frac{w}{np_1}, \dots, \frac{w}{np_n})$.
- The buyer thus shares his income equally on all the products.

Karush-Kuhn-Tucker (KKT) Conditions

- Consider the following optimization problem:

$$\min_{x_1,\ldots,x_n} f(x_1, x_2, \ldots, x_n)$$

$$\text{s.t.} \quad \text{(i) } g_i(x_1, \ldots, x_n) \leq 0,\ i = 1, \ldots, k,$$

$$\text{(ii) } h_j(x_1, \ldots, x_n) = 0,\ j = 1, \ldots, m.$$

Karush-Kuhn-Tucker (KKT) Conditions

- Consider the following optimization problem:

$$\min_{x_1,\dots,x_n} f(x_1, x_2, \dots, x_n)$$

$$\text{s.t.} \quad \text{(i) } g_i(x_1,\dots,x_n) \le 0,\ i = 1,\dots,k,$$

$$\text{(ii) } h_j(x_1,\dots,x_n) = 0,\ j = 1,\dots,m.$$

- The Lagrangian function can be written as

$$\mathcal{L}(x_1,\dots,x_n,\lambda_1,\dots,\lambda_{,k},\mu_1,\dots,\mu_m)$$

$$= f(x_1,\dots,x_n) + \sum_{i=1}^{k} \lambda_i g_i(x_1,\dots,x_n) + \sum_{j=1}^{m} \mu_j h_j(x_1,\dots,x_n).$$

Karush-Kuhn-Tucker (KKT) Conditions

- Consider the following optimization problem:

$$\begin{aligned} &\min_{x_1,\ldots,x_n} f(x_1, x_2, \ldots, x_n) \\ \text{s.t.} \quad &\text{(i) } g_i(x_1, \ldots, x_n) \le 0,\ i = 1, \ldots, k, \\ &\text{(ii) } h_j(x_1, \ldots, x_n) = 0,\ j = 1, \ldots, m. \end{aligned}$$

- The Lagrangian function can be written as

$$\begin{aligned} &\mathcal{L}(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_{,k}, \mu_1, \ldots, \mu_m) \\ &= f(x_1, \ldots, x_n) + \sum_{i=1}^{k} \lambda_i g_i(x_1, \ldots, x_n) + \sum_{j=1}^{m} \mu_j h_j(x_1, \ldots, x_n). \end{aligned}$$

- The set of points $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ satisfying the following *Karush-Kuhn-Tucker (KKT)* conditions are the critical points:

$$\begin{aligned} &\nabla \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = 0, \\ &\lambda_i^* g_i(\boldsymbol{x}^*) = 0,\ i = 1, \ldots, k, \quad \lambda_i^* \ge 0,\ i = 1, \ldots, k, \\ &g_i(\boldsymbol{x}^*) \le 0,\ i = 1, \ldots, n, \quad h_j(\boldsymbol{x}^*) = 0,\ j = 1, \ldots, m. \end{aligned}$$

Second Order Conditions

- We use second order conditions to verify whether the critical point so obtained is a local maximum, a local minimum, or a saddle point.

Second Order Conditions

- We use second order conditions to verify whether the critical point so obtained is a local maximum, a local minimum, or a saddle point.
- Define the tangent space at a critical point $\boldsymbol{x}^*$ as

$$T_{\boldsymbol{x}^*} = \{\boldsymbol{d} : \nabla h_j(\boldsymbol{x}^*)\boldsymbol{d} = 0,\ \forall j, \nabla g_i(\boldsymbol{x}^*)\boldsymbol{d} = 0,\ \forall i \in \mathcal{A}_{\boldsymbol{x}^*}\}$$

  where $\mathcal{A}_{\boldsymbol{x}^*}$ is the set of inequality constraints that satisfy with equality at $\boldsymbol{x} = \boldsymbol{x}^*$.

Second Order Conditions

- We use second order conditions to verify whether the critical point so obtained is a local maximum, a local minimum, or a saddle point.
- Define the tangent space at a critical point $\boldsymbol{x}^*$ as

$$T_{\boldsymbol{x}^*} = \{\boldsymbol{d} : \nabla h_j(\boldsymbol{x}^*)\boldsymbol{d} = 0,\ \forall j, \nabla g_i(\boldsymbol{x}^*)\boldsymbol{d} = 0,\ \forall i \in \mathcal{A}_{\boldsymbol{x}^*}\}$$

  where $\mathcal{A}_{\boldsymbol{x}^*}$ is the set of inequality constraints that satisfy with equality at $\boldsymbol{x} = \boldsymbol{x}^*$.
- The critical point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ is a (local) maximum if $\boldsymbol{d}^T \nabla_2 \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\boldsymbol{d} \leq 0\, \forall \boldsymbol{d} \in T_{\boldsymbol{x}^*}$,
- (local) minimum if $\boldsymbol{d}^T \nabla_2 \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\boldsymbol{d} \geq 0\, \forall \boldsymbol{d} \in T_{\boldsymbol{x}^*}$,
- saddle point if $\boldsymbol{d}^T \nabla_2 \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\boldsymbol{d} > 0$ for some $\boldsymbol{d} \in T_{\boldsymbol{x}^*}$, and < 0 for some $\boldsymbol{d} \in T_{\boldsymbol{x}^*}$.

Second Order Conditions

- We use second order conditions to verify whether the critical point so obtained is a local maximum, a local minimum, or a saddle point.
- Define the tangent space at a critical point $\boldsymbol{x}^*$ as

$$T_{\boldsymbol{x}^*} = \{\boldsymbol{d} : \nabla h_j(\boldsymbol{x}^*)\boldsymbol{d} = 0, \forall j, \nabla g_i(\boldsymbol{x}^*)\boldsymbol{d} = 0, \forall i \in \mathcal{A}_{\boldsymbol{x}^*}\}$$

  where $\mathcal{A}_{\boldsymbol{x}^*}$ is the set of inequality constraints that satisfy with equality at $\boldsymbol{x} = \boldsymbol{x}^*$.
- The critical point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ is a (local) maximum if $\boldsymbol{d}^T \nabla_2 \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\boldsymbol{d} \leq 0 \forall \boldsymbol{d} \in T_{\boldsymbol{x}^*}$,
- (local) minimum if $\boldsymbol{d}^T \nabla_2 \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\boldsymbol{d} \geq 0 \forall \boldsymbol{d} \in T_{\boldsymbol{x}^*}$,
- saddle point if $\boldsymbol{d}^T \nabla_2 \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\boldsymbol{d} > 0$ for some $\boldsymbol{d} \in T_{\boldsymbol{x}^*}$, and < 0 for some $\boldsymbol{d} \in T_{\boldsymbol{x}^*}$.
- So $\boldsymbol{d}^T \nabla_2 \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\boldsymbol{d} > 0$ for all $\boldsymbol{d}$ implies $\boldsymbol{x}^*$ is a local minimizer, < 0 for all $\boldsymbol{d}$ implies that $\boldsymbol{x}^*$ is a local maximizer, and $= 0$ for some $\boldsymbol{d}$ implies that a further probe is required.

Convex Optimization

$$\begin{aligned}
&\min_{x_1,\ldots,x_n} f(x_1, x_2, \ldots, x_n) \\
\text{s.t.} \quad &\text{(i) } g_i(x_1, \ldots, x_n) \leq 0,\ i = 1, \ldots, k, \\
&\text{(ii) } h_j(x_1, \ldots, x_n) = 0,\ j = 1, \ldots, m.
\end{aligned}$$

- The optimization problem given above is convex, if $f(x_1, \ldots, x_n)$ is a convex function, and the set defined by the constraints $\{(g_i)_{i=1}^k, (h_j)_{j=1}^m\}$ is a convex set.

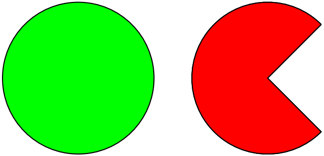
Convex Optimization

$$\min_{x_1,\ldots,x_n} f(x_1, x_2, \ldots, x_n)$$

$$\text{s.t.} \quad \text{(i) } g_i(x_1, \ldots, x_n) \leq 0,\ i = 1, \ldots, k,$$

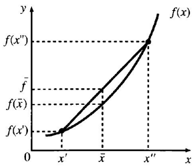
$$\text{(ii) } h_j(x_1, \ldots, x_n) = 0,\ j = 1, \ldots, m.$$

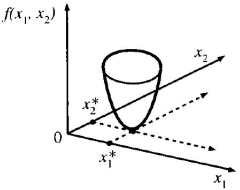
- The optimization problem given above is convex, if $f(x_1, \ldots, x_n)$ is a convex function, and the set defined by the constraints $\{(g_i)_{i=1}^k, (h_j)_{j=1}^m\}$ is a convex set.
- A set is convex if the line segment connecting two points in the set lies within the set. The green set below is convex, but the red set is not.



Convex Optimization (contd...)

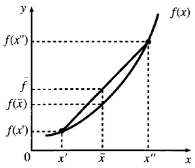
- A function is convex if the line segment between two points of the function lies above the curve.

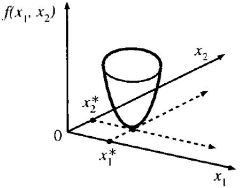




Convex Optimization (contd...)

- A function is convex if the line segment between two points of the function lies above the curve.

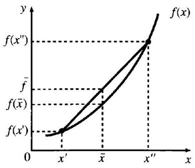


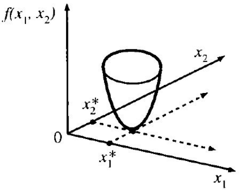


- The advantage with convex optimization is that the local minimum turns out to be the global minimum as well.

Convex Optimization (contd...)

- A function is convex if the line segment between two points of the function lies above the curve.

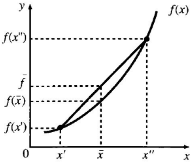


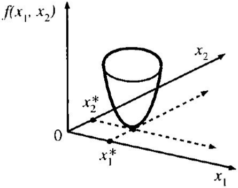


- The advantage with convex optimization is that the local minimum turns out to be the global minimum as well.
- Well-established algorithms available for solving convex optimization problems: Steepest-descent, Newton's method, gradient descent, ...

Convex Optimization (contd...)

- A function is convex if the line segment between two points of the function lies above the curve.





- The advantage with convex optimization is that the local minimum turns out to be the global minimum as well.
- Well-established algorithms available for solving convex optimization problems: Steepest-descent, Newton's method, gradient descent, ...
- Non-convex optimization is computationally hard.

Linear and Non-linear Optimization

$$\min_{x_1,\ldots,x_n} f(x_1, x_2, \ldots, x_n)$$
$$\text{s.t.} \quad \text{(i) } g_i(x_1, \ldots, x_n) \leq 0,\ i = 1, \ldots, k,$$
$$\text{(ii) } h_j(x_1, \ldots, x_n) = 0,\ j = 1, \ldots, m.$$

- **Linear Optimization:** The optimization problem given above is linear, if f, g_i, h_j are all linear functions of $(x_1, \ldots, x_n)$. It can thus be rewritten as

$$\min_{x_1,\ldots,x_n} \boldsymbol{w}^T \boldsymbol{x}$$
$$\text{s.t.} \quad \text{(i) } A\boldsymbol{x} \leq 0,$$
$$\text{(ii) } B\boldsymbol{x} = 0,$$

where A is a $k \times n$ matrix, and B is an $m \times n$ matrix.

Linear and Non-linear Optimization

$$\begin{aligned} &\min_{x_1,\ldots,x_n} f(x_1, x_2, \ldots, x_n) \\ \text{s.t.} \quad &\text{(i) } g_i(x_1, \ldots, x_n) \leq 0,\ i = 1, \ldots, k, \\ &\text{(ii) } h_j(x_1, \ldots, x_n) = 0,\ j = 1, \ldots, m. \end{aligned}$$

- **Linear Optimization:** The optimization problem given above is linear, if f, g_i, h_j are all linear functions of $(x_1, \ldots, x_n)$. It can thus be rewritten as

$$\begin{aligned} &\min_{x_1,\ldots,x_n} \boldsymbol{w}^T \boldsymbol{x} \\ \text{s.t.} \quad &\text{(i) } A\boldsymbol{x} \leq 0, \\ &\text{(ii) } B\boldsymbol{x} = 0, \end{aligned}$$

  where A is a $k \times n$ matrix, and B is an $m \times n$ matrix.
- A subset of the set of convex optimization problems.

Linear and Non-linear Optimization

$$\min_{x_1,\dots,x_n} f(x_1, x_2, \dots, x_n)$$
$$\text{s.t.} \quad \text{(i) } g_i(x_1, \dots, x_n) \le 0,\ i = 1, \dots, k,$$
$$\text{(ii) } h_j(x_1, \dots, x_n) = 0,\ j = 1, \dots, m.$$

- **Linear Optimization:** The optimization problem given above is linear, if f, g_i, h_j are all linear functions of $(x_1, \dots, x_n)$. It can thus be rewritten as

$$\min_{x_1,\dots,x_n} \boldsymbol{w}^T \boldsymbol{x}$$
$$\text{s.t.} \quad \text{(i) } A\boldsymbol{x} \le 0,$$
$$\text{(ii) } B\boldsymbol{x} = 0,$$

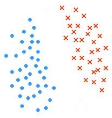
  where A is a $k \times n$ matrix, and B is an $m \times n$ matrix.
- A subset of the set of convex optimization problems.
- The other optimization problems are termed to be non-linear.

Overview

1. Motivation
2. Optimizing one variable
3. Optimizing multiple variables
4. Constrained optimization
5. Application to machine learning problems

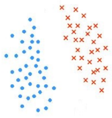
Binary Classification

- Consider the binary classification problem under supervised learning. As a practical example, consider that the machine needs to learn whether a given mail is a spam or not. It is given some characteristic vectors $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$, and a classification of each vector on whether it is a spam ($y_i = 1$) or not ($y_i = -1$).



Binary Classification

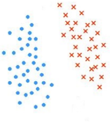
- Consider the binary classification problem under supervised learning. As a practical example, consider that the machine needs to learn whether a given mail is a spam or not. It is given some characteristic vectors $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$, and a classification of each vector on whether it is a spam ($y_i = 1$) or not ($y_i = -1$).



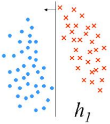
- We now construct a linear classifier. The spam and not non-spam vectors must be separated by a plane $\boldsymbol{w}^T \boldsymbol{x} + b = 0$.

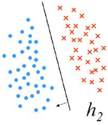
Binary Classification

- Consider the binary classification problem under supervised learning. As a practical example, consider that the machine needs to learn whether a given mail is a spam or not. It is given some characteristic vectors $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$, and a classification of each vector on whether it is a spam $(y_i = 1)$ or not $(y_i = -1)$.



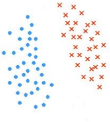
- We now construct a linear classifier. The spam and not non-spam vectors must be separated by a plane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$.
- But we can construct many such lines for the given example.



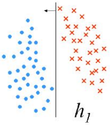


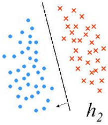
Binary Classification

- Consider the binary classification problem under supervised learning. As a practical example, consider that the machine needs to learn whether a given mail is a spam or not. It is given some characteristic vectors $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$, and a classification of each vector on whether it is a spam $(y_i = 1)$ or not $(y_i = -1)$.



- We now construct a linear classifier. The spam and not non-spam vectors must be separated by a plane $\boldsymbol{w}^T \boldsymbol{x} + b = 0$.
- But we can construct many such lines for the given example.

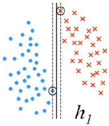


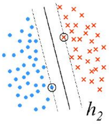


- But h_2 seems a better-fit than h_1. What is the reason?

Support Vector Machine

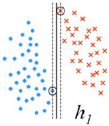
- The plane h_2 maximizes the minimum distance between the plane and the characteristic vector.

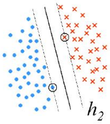




Support Vector Machine

- The plane h_2 maximizes the minimum distance between the plane and the characteristic vector.

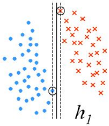


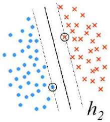


- How do we compute the best-fit plane given a set of arbitrary characteristic vectors along with their classification?

Support Vector Machine

- The plane h_2 maximizes the minimum distance between the plane and the characteristic vector.

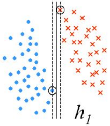


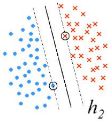


- How do we compute the best-fit plane given a set of arbitrary characteristic vectors along with their classification?
- The distance between the plane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ and the point $\boldsymbol{x}_i$ is $\gamma_i = \frac{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)}{||\boldsymbol{w}||}$. We must maximize $\min_{i=1}^{n} \gamma_i$.

Support Vector Machine

- The plane h_2 maximizes the minimum distance between the plane and the characteristic vector.





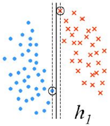
- How do we compute the best-fit plane given a set of arbitrary characteristic vectors along with their classification?
- The distance between the plane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ and the point $\boldsymbol{x}_i$ is $\gamma_i = \frac{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)}{||\boldsymbol{w}||}$. We must maximize $\min_{i=1}^n \gamma_i$.
- The values of $\boldsymbol{w}$ and b can be normalized to have $\min_{i=1}^n \gamma_i = 1$. So we must maximize $\frac{1}{||\boldsymbol{w}||}$, or minimize $||\boldsymbol{w}||^2$.

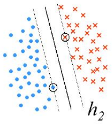
$$\min_{\boldsymbol{w},b} ||\boldsymbol{w}||^2$$

$$\text{subject to } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1,\ i = 1, \ldots, n.$$

Support Vector Machine

- The plane h_2 maximizes the minimum distance between the plane and the characteristic vector.





- How do we compute the best-fit plane given a set of arbitrary characteristic vectors along with their classification?
- The distance between the plane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ and the point $\boldsymbol{x}_i$ is $\gamma_i = \frac{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)}{||\boldsymbol{w}||}$. We must maximize $\min_{i=1}^{n} \gamma_i$.
- The values of $\boldsymbol{w}$ and b can be normalized to have $\min_{i=1}^{n} \gamma_i = 1$. So we must maximize $\frac{1}{||\boldsymbol{w}||}$, or minimize $||\boldsymbol{w}||^2$.

$$\min_{\boldsymbol{w}, b} ||\boldsymbol{w}||^2$$

$$\text{subject to } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1, \; i = 1, \ldots, n.$$

- A convex optimization problem. The algorithm to compute the best-fit plane is called the *support vector machine* (SVM).

Solution

- Lagrangian: $\mathcal{L} = ||\boldsymbol{w}||^2 - \sum_{i=1}^{n} \lambda_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1)$. KKT conditions are

$$\frac{1}{2}\sum_{i=1}^{n} \lambda_i y_i \boldsymbol{x}_i = \boldsymbol{w}, \tag{1}$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0, \tag{2}$$

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1,\ i = 1, \ldots, n, \tag{3}$$

$$\lambda_i(y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1) = 0,\ i = 1, \ldots, n. \tag{4}$$

Solution

- Lagrangian: $\mathcal{L} = ||\boldsymbol{w}||^2 - \sum_{i=1}^{n} \lambda_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1)$. KKT conditions are

$$\frac{1}{2}\sum_{i=1}^{n} \lambda_i y_i \boldsymbol{x}_i = \boldsymbol{w}, \tag{1}$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0, \tag{2}$$

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1,\ i = 1, \ldots, n, \tag{3}$$

$$\lambda_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1) = 0,\ i = 1, \ldots, n. \tag{4}$$

- Consider (1) and (3). We must find those characteristic vectors for which $y_i(\sum_j \lambda_j y_j (\boldsymbol{x}_j^T \boldsymbol{x}_i)/2 + b) = 1$.
- By (4), only those $\lambda_i > 0$. Every other $\lambda_i = 0$. These are the vectors that are closest to the best-fit plane, and are called *support vectors*.

Solution

- Lagrangian: $\mathcal{L} = ||\boldsymbol{w}||^2 - \sum_{i=1}^{n} \lambda_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1)$. KKT conditions are

$$\frac{1}{2}\sum_{i=1}^{n} \lambda_i y_i \boldsymbol{x}_i = \boldsymbol{w}, \tag{1}$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0, \tag{2}$$

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1,\ i = 1, \ldots, n, \tag{3}$$

$$\lambda_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1) = 0,\ i = 1, \ldots, n. \tag{4}$$

- Consider (1) and (3). We must find those characteristic vectors for which $y_i(\sum_j \lambda_j y_j (\boldsymbol{x}_j^T \boldsymbol{x}_i)/2 + b) = 1$.
- By (4), only those $\lambda_i > 0$. Every other $\lambda_i = 0$. These are the vectors that are closest to the best-fit plane, and are called *support vectors*.
- Given that λ's are found, we can now compute $\boldsymbol{w}$ by (1), and b from (3).

Other applications in machine learning

- **Soft SVM margin:** The samples obtained may not be linearly separable. We consider a soft SVM margin in such cases, and solve the following problem:

$$\min_{\boldsymbol{w},b} ||\boldsymbol{w}||^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \quad \text{(i) } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i,\ i = 1,\ldots,n,$$

$$\text{(ii) } \xi_i \geq 0,\ i = 1,\ldots,n.$$

Other applications in machine learning

- **Soft SVM margin:** The samples obtained may not be linearly separable. We consider a soft SVM margin in such cases, and solve the following problem:

$$\min_{\boldsymbol{w},b} ||\boldsymbol{w}||^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \quad \text{(i) } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i,\ i = 1, \ldots, n,$$
$$\text{(ii) } \xi_i \geq 0,\ i = 1, \ldots, n.$$

- **Regression:** We want to fit the best linear plane to a set of characteristic vectors $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. We then minimize the mean square error between the vectors and the points on the plane.

$$\min_{\boldsymbol{w},b} \sum_{i=1}^{n} ||\boldsymbol{w}^T\boldsymbol{x}_i + b||^2$$

$$\text{subject to } ||\boldsymbol{w}|| = 1.$$

Summary

- Optimization involves minimizing/ maximizing a function with respect to some constraints on the variables.

Summary

- Optimization involves minimizing/ maximizing a function with respect to some constraints on the variables.
- **Unconstrained:** Compute the critical points by equating the gradient of the function to 0, and then decide whether those points correspond to maximum, minimum, or saddle point, using the “definiteness” of the Hessian matrix.

Summary

- Optimization involves minimizing/ maximizing a function with respect to some constraints on the variables.
- **Unconstrained:** Compute the critical points by equating the gradient of the function to 0, and then decide whether those points correspond to maximum, minimum, or saddle point, using the "definiteness" of the Hessian matrix.
- **Constrained:** Compute the critical points using the Karush-Kuhn-Tucker (KKT) conditions, and then decide if those points are maximum, minimum, or saddle point, using second-order conditions.

Summary

- Optimization involves minimizing/ maximizing a function with respect to some constraints on the variables.
- **Unconstrained:** Compute the critical points by equating the gradient of the function to 0, and then decide whether those points correspond to maximum, minimum, or saddle point, using the "definiteness" of the Hessian matrix.
- **Constrained:** Compute the critical points using the Karush-Kuhn-Tucker (KKT) conditions, and then decide if those points are maximum, minimum, or saddle point, using second-order conditions.
- Convex optimization, linear optimization, and non-linear optimization.

Summary

- Optimization involves minimizing/ maximizing a function with respect to some constraints on the variables.
- **Unconstrained:** Compute the critical points by equating the gradient of the function to 0, and then decide whether those points correspond to maximum, minimum, or saddle point, using the "definiteness" of the Hessian matrix.
- **Constrained:** Compute the critical points using the Karush-Kuhn-Tucker (KKT) conditions, and then decide if those points are maximum, minimum, or saddle point, using second-order conditions.
- Convex optimization, linear optimization, and non-linear optimization.
- Many applications of optimization theory in machine learning problems: support vector machine, soft SVM margin, regression, ...

References

- "Convex Optimization" by Boyd and Vandenberghe.
- "Linear and Non-linear Programming", by Luenberger.
- "Practical Methods of Optimization", by Fletcher.