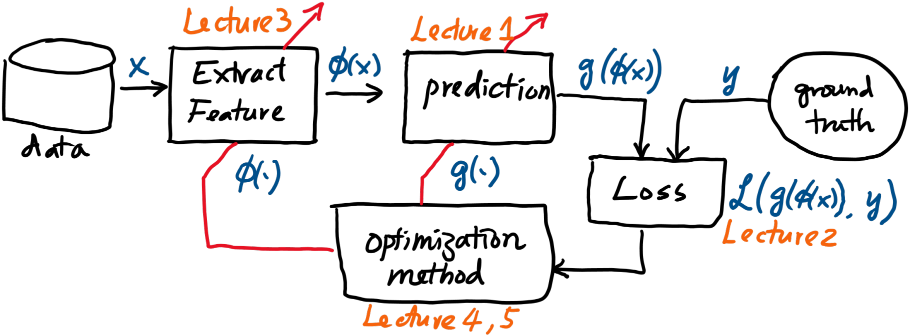
ECE 595: Machine Learning I
Lecture 04 Intro to Optimization

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

Outline


Lecture3
Lecture 1
x
Extract Feature
$\phi(x)$
prediction
$g(\phi(x))$
y
ground truth
data
$\phi(\cdot)$
$g(\cdot)$
Loss
$\mathcal{L}(g(\phi(x)), y)$
Lecture2
optimization method
Lecture 4, 5

Outline

Mathematical Background

- Lecture 4: Intro to Optimization
- Lecture 5: Gradient Descent

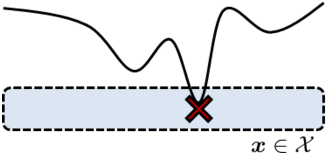
Lecture 4: Intro to Optimization

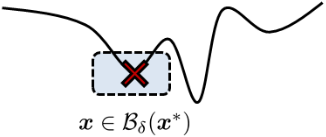
- Unconstrained Optimization
  - First Order Optimality
  - Second Order Optimality
- Convexity
  - What is convexity?
  - Convex optimization
- Constrained Optimization
  - Lagrangian
  - Examples

Unconstrained Optimization

$$\underset{\boldsymbol{x}\in\mathcal{X}}{\text{minimize}} \quad f(\boldsymbol{x})$$

- $\boldsymbol{x}^* \in \mathcal{X}$ is a **global minimizer** if
  - $f(\boldsymbol{x}^*) \le f(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathcal{X}$
- $\boldsymbol{x}^* \in \mathcal{X}$ is a **local minimizer** if
  - $f(\boldsymbol{x}^*) \le f(\boldsymbol{x})$, for any $\boldsymbol{x}$ in a neighborhood $\mathcal{B}_\delta(\boldsymbol{x}^*)$
  - $\mathcal{B}_\delta(\boldsymbol{x}^*) = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \boldsymbol{x}^*\|_2 \le \delta\}$





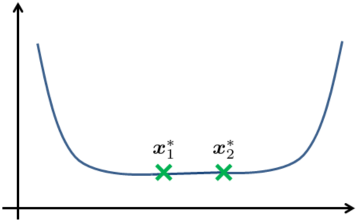
Uniqueness of Global Minimizer

If $\boldsymbol{x}^*$ is global minimizer, then

- Objective value $f(\boldsymbol{x}^*)$ is unique
- Solution $\boldsymbol{x}^*$ is not necessarily unique

Therefore:

- Suppose $f(\boldsymbol{x}) = g(\boldsymbol{x}) + \lambda\|\boldsymbol{x}\|_1$ for some convex g.
- "$\underset{\boldsymbol{x}}{\text{minimize}}\ f(\boldsymbol{x})$" has a global optimal $f(\boldsymbol{x}^*)$.
- But there could be multiple $\boldsymbol{x}^*$'s.
- Some $\boldsymbol{x}^*$ maybe better, but not in the sense of $f(\boldsymbol{x})$.



First and Second Order Optimality

$$\underbrace{\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}}_{\text{First order condition}} \quad \text{and} \quad \underbrace{\nabla^2 f(\boldsymbol{x}^*) \succeq 0}_{\text{Second order condition}}.$$

Necessary Condition:

If $\boldsymbol{x}^*$ is a global (or local) minimizer, then

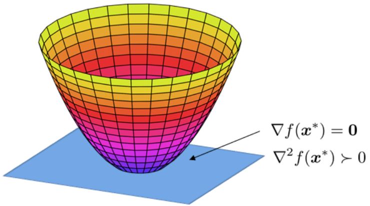
- $\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$.
- $\nabla^2 f(\boldsymbol{x}^*) \succeq 0$.

Sufficient Condition:

If $\boldsymbol{x}^*$ satisfies

- $\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$.
- $\nabla^2 f(\boldsymbol{x}^*) \succ 0$.

then $\boldsymbol{x}^*$ is a global (or local) minimizer.



Why? First Order

- Why is $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$ necessary?
- Suppose $\boldsymbol{x}^*$ is the minimizer.
- Pick any direction $\boldsymbol{d}$, and any step size ϵ. Then

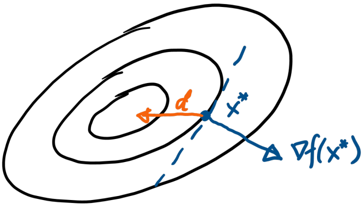
$$f(\boldsymbol{x}^* + \epsilon \boldsymbol{d}) = f(\boldsymbol{x}^*) + \epsilon \nabla f(\boldsymbol{x}^*)^T \boldsymbol{d} + \cancel{\mathcal{O}(\epsilon^2)}.$$

- Rearranging the terms yields

$$\underbrace{\lim_{\epsilon \to 0} \left\{ \frac{f(\boldsymbol{x}^* + \epsilon \boldsymbol{d}) - f(\boldsymbol{x}^*)}{\epsilon} \right\}}_{\geq 0, \ \forall \boldsymbol{d}} = \nabla f(\boldsymbol{x}^*)^T \boldsymbol{d}.$$

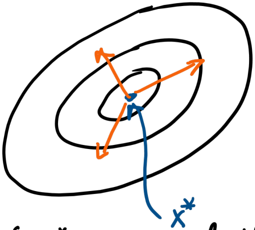
- So $\nabla f(\boldsymbol{x}^*)^T \boldsymbol{d} \geq 0$ for all $\boldsymbol{d}$. True only when $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$.

First Order Condition Illustrated



if x^* is not optimal, then

$\nabla f(x^*)^T d \leq 0$, for <u>some</u> d



if x^* is optimal, then

$\nabla f(x^*)^T d = 0$, for <u>all</u> d

Why? Second Order

Do third order approximation:

$$f(\boldsymbol{x}^* + \epsilon \boldsymbol{d}) = f(\boldsymbol{x}^*) + \epsilon \underbrace{\nabla f(\boldsymbol{x}^*)^T \boldsymbol{d}}_{=0} + \frac{\epsilon^2}{2} \boldsymbol{d}^T \nabla^2 f(\boldsymbol{x}^*) \boldsymbol{d} + \frac{\epsilon^3}{6} \mathcal{O}(\|\boldsymbol{d}\|^3)$$

Therefore,

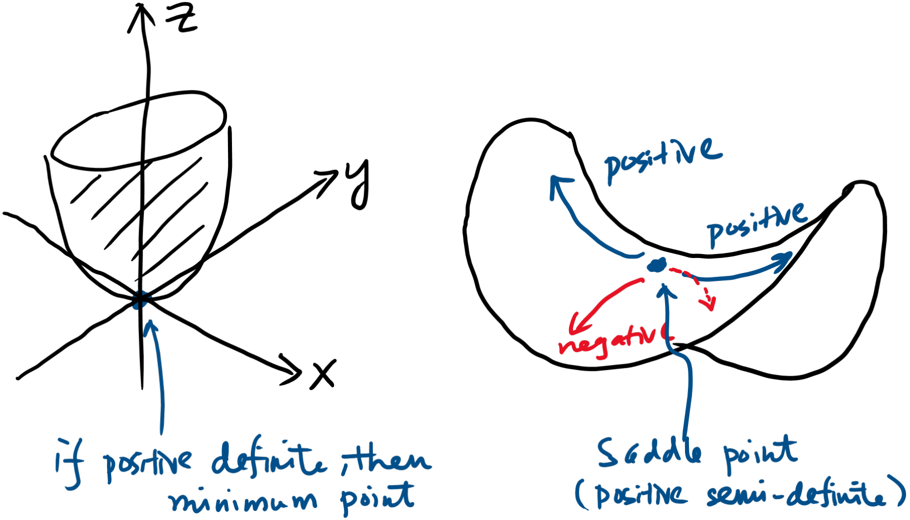
$$\begin{aligned}
\frac{1}{\epsilon^2}\left[f(\boldsymbol{x}^* + \epsilon \boldsymbol{d}) - f(\boldsymbol{x}^*)\right] &= \frac{1}{2} \boldsymbol{d}^T \nabla^2 f(\boldsymbol{x}^*) \boldsymbol{d} + \left[\frac{\epsilon}{6} \mathcal{O}(\|\boldsymbol{d}\|^3)\right] \\
\lim_{\epsilon \to 0} \frac{1}{\epsilon^2} \underbrace{\left[f(\boldsymbol{x}^* + \epsilon \boldsymbol{d}) - f(\boldsymbol{x}^*)\right]}_{\geq 0} &= \frac{1}{2} \boldsymbol{d}^T \nabla^2 f(\boldsymbol{x}^*) \boldsymbol{d} + \lim_{\epsilon \to 0} \left[\frac{\epsilon}{6} \mathcal{O}(\|\boldsymbol{d}\|^3)\right],
\end{aligned}$$

Hence,

$$\frac{1}{2} \boldsymbol{d}^T \nabla^2 f(\boldsymbol{x}^*) \boldsymbol{d} \geq 0, \qquad \forall \boldsymbol{d}.$$

$\Rightarrow$ positive semi-definite!

Second Order Condition Illustrated



Outline

Mathematical Background

- Lecture 4: Intro to Optimization
- Lecture 5: Gradient Descent

Lecture 4: Intro to Optimization

- Unconstrained Optimization
  - First Order Optimality
  - Second Order Optimality
- Convexity
  - What is convexity?
  - Convex optimization
- Constrained Optimization
  - Lagrangian
  - Examples

Most Optimization Problems are Not Easy

Minimize the log-sum-exp function:

$$f(\boldsymbol{x}) = \log\left(\sum_{i=1}^{m} \exp(\boldsymbol{a}_i^T \boldsymbol{x} + b_i)\right)$$

- Gradient is (exercise)

$$\nabla f(\boldsymbol{x}^*) = \frac{1}{\sum_{j=1}^{m} \exp(\boldsymbol{a}_j^T \boldsymbol{x}^* + b_j)} \sum_{i=1}^{m} \exp(\boldsymbol{a}_i^T \boldsymbol{x}^* + b_i)\boldsymbol{a}_i.$$

- Non-linear equation. No closed-form solution.
- Need iterative algorithms, e.g., gradient descent.
- Or off-the-shelf optimization solver, e.g., `CVX`.

CVX Demonstration

- Disciplined optimization: It translates the problem for you.
- Developed by S. Boyd and colleagues (Stanford).
- E.g., Minimize $f(\boldsymbol{x}) = \log\left(\sum_{i=1}^{n}\exp(\boldsymbol{a}_i^T\boldsymbol{x} + b_i)\right) + \lambda\|\boldsymbol{x}\|^2$.

```
import cvxpy as cp
import numpy as np

n = 100
d = 3
A = np.random.randn(n, d)
b = np.random.randn(n)
lambda_ = 0.1

x = cp.Variable(d)
objective = cp.Minimize(cp.log_sum_exp(A*x - b) + lambda_*cp.sum_squares(x))
constraints = []
prob = cp.Problem(objective, constraints)

optimal_objective_value = prob.solve()
print(optimal_objective_value)
print(x.value)
```

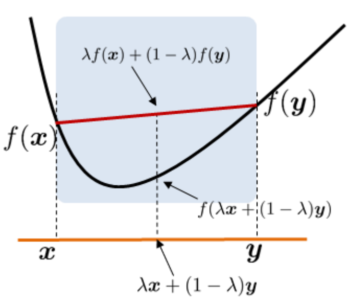
Convex Function

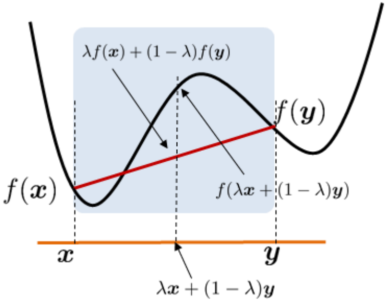
Definition

Let $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{X}$. Let $0 \le \lambda \le 1$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is **convex** over $\mathcal{X}$ if

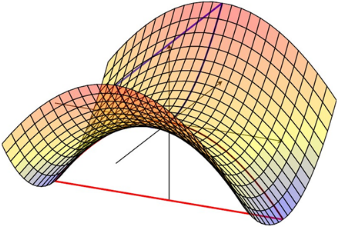
$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \;\le\; \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}).$$

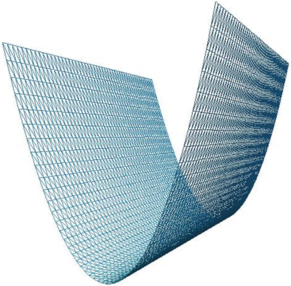
The function is called strictly convex if "$\le$" is replaced by "$<$".

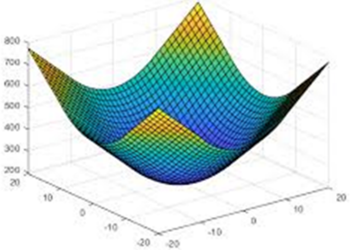


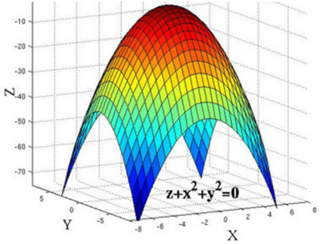


Example: Which one is convex?









Verifying Convexity

Any of the following conditions is **necessary** and **sufficient** for convexity:

1. By definition:

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}).$$

   - Function value is lower than the line.

2. First Order Convexity:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y}-\boldsymbol{x}), \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}.$$

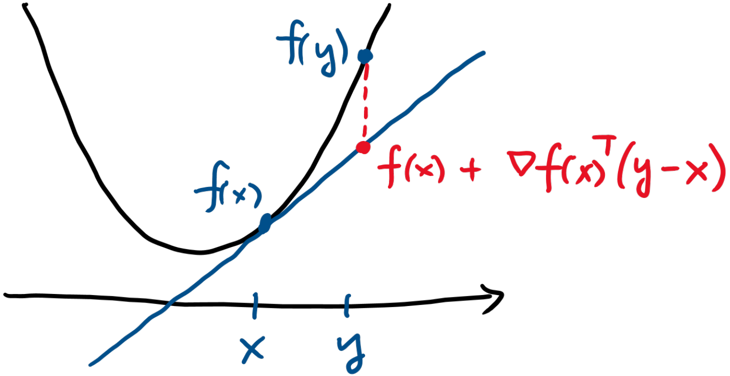
   - Tangent line is always lower than the function

3. Second Order Convexity: f is convex over $\mathcal{X}$ if and only if

$$\nabla^2 f(\boldsymbol{x}) \succeq 0 \quad \forall \boldsymbol{x} \in \mathcal{X}.$$

   - Curvature is positive.

Tangent Line Condition Illustrated



Outline

Mathematical Background

- Lecture 4: Intro to Optimization
- Lecture 5: Gradient Descent

Lecture 4: Intro to Optimization

- Unconstrained Optimization
  - First Order Optimality
  - Second Order Optimality
- Convexity
  - What is convexity?
  - Convex optimization
- Constrained Optimization
  - Lagrangian
  - Examples

Constrained Optimization

Equality Constrained Optimization:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & h_j(\boldsymbol{x}) = 0, \qquad j = 1, \ldots, k. \end{aligned}$$

Requires a function: Lagrangian function

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} f(\boldsymbol{x}) - \sum_{j=1}^{k} \nu_j h_j(\boldsymbol{x}).$$

$\boldsymbol{\nu} = [\nu_1, \ldots, \nu_k]$: **Lagrange multipliers** or the **dual variables**.

Solution $(\boldsymbol{x}^*, \boldsymbol{\nu}^*)$ satisfies

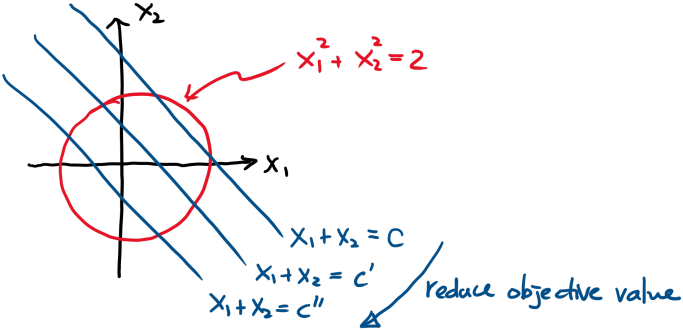
$$\begin{aligned} \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\nu}^*) &= \mathbf{0}, \\ \nabla_{\boldsymbol{\nu}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\nu}^*) &= \mathbf{0}. \end{aligned}$$

Example: Illustrating Lagrangian

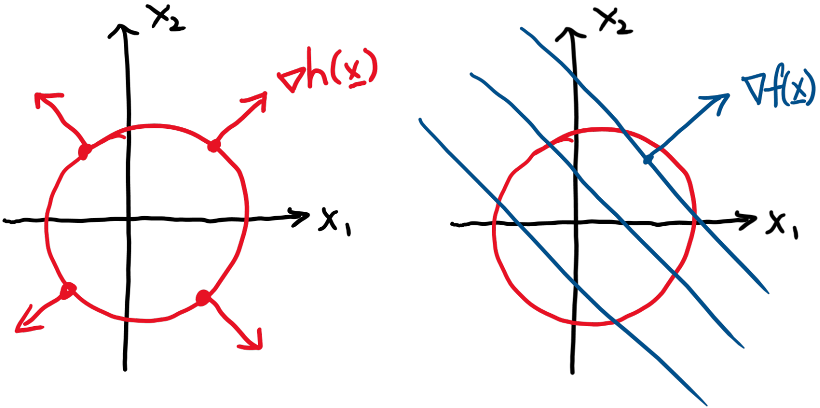
- Consider the problem

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & x_1 + x_2 \\ \text{subject to} \quad & x_1^2 + x_2^2 = 2. \end{aligned}$$

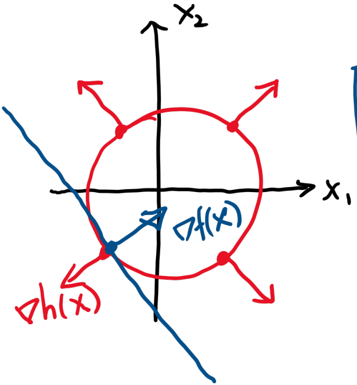
- Minimizer is $\boldsymbol{x} = (-1, -1)$.



Example: Illustrating Lagrangian



Example: Illustrating Lagrangian



$$\boxed{\nabla f(x^*) = \lambda \nabla h(x^*)}$$

$$\Rightarrow \nabla \mathcal{L}(x^*, \lambda^*) = \nabla f(x^*) - \lambda \nabla h(x^*) = 0$$

Example: ℓ_2-minimization with constraint

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}_0\|^2, \qquad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x}=\boldsymbol{y}.$$

The Lagrangian function of the problem is

$$\mathcal{L}(\boldsymbol{x},\boldsymbol{\nu}) = \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}_0\|^2 - \boldsymbol{\nu}^T(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}).$$

The first order optimality condition requires

$$\begin{aligned}
\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x},\boldsymbol{\nu}) &= (\boldsymbol{x}-\boldsymbol{x}_0) - \boldsymbol{A}^T\boldsymbol{\nu} = \boldsymbol{0}\\
\nabla_{\boldsymbol{\nu}}\mathcal{L}(\boldsymbol{x},\boldsymbol{\nu}) &= \boldsymbol{A}\boldsymbol{x}-\boldsymbol{y} = \boldsymbol{0}.
\end{aligned}$$

Multiply the first equation by $\boldsymbol{A}$ on both sides:

$$\begin{aligned}
& & \boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}_0) - \boldsymbol{A}\boldsymbol{A}^T\boldsymbol{\nu} &= \boldsymbol{0}\\
\Rightarrow & & \underbrace{\boldsymbol{A}\boldsymbol{x}}_{=\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{x}_0 &= \boldsymbol{A}\boldsymbol{A}^T\boldsymbol{\nu}\\
\Rightarrow & & \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_0 &= \boldsymbol{A}\boldsymbol{A}^T\boldsymbol{\nu}\\
\Rightarrow & & (\boldsymbol{A}\boldsymbol{A}^T)^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_0) &= \boldsymbol{\nu}
\end{aligned}$$

Example: ℓ_2-minimization with constraint

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}_0\|^2, \qquad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x}=\boldsymbol{y}.$$

The first order optimality condition requires

$$\begin{aligned}
\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x},\boldsymbol{\nu}) &= (\boldsymbol{x}-\boldsymbol{x}_0)-\boldsymbol{A}^T\boldsymbol{\nu}=\boldsymbol{0}\\
\nabla_{\boldsymbol{\nu}}\mathcal{L}(\boldsymbol{x},\boldsymbol{\nu}) &= \boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}=\boldsymbol{0}.
\end{aligned}$$

We just showed: $\boldsymbol{\nu}=(\boldsymbol{A}\boldsymbol{A}^T)^{-1}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}_0)$. Substituting this result into the first order optimality yields

$$\begin{aligned}
\boldsymbol{x} &= \boldsymbol{x}_0+\boldsymbol{A}^T\boldsymbol{\nu}\\
&= \boldsymbol{x}_0+\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}_0)
\end{aligned}$$

Therefore, the solution is $\boldsymbol{x}=\boldsymbol{x}_0+\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}_0)$.

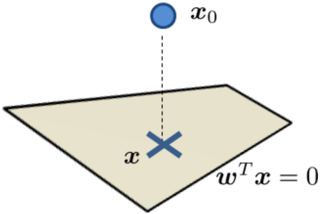
Special Case

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}_0\|^2, \qquad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x}=\boldsymbol{y}.$$

Special case: When $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{y}$ is simplified to $\boldsymbol{w}^T\boldsymbol{x}=0$.

- $\boldsymbol{w}^T\boldsymbol{x}=0$ is a line.
- Find a point $\boldsymbol{x}$ on the line that is closest to $\boldsymbol{x}_0$.
- Solution is

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{x}_0+\boldsymbol{w}(\boldsymbol{w}^T\boldsymbol{w})^{-1}(0-\boldsymbol{w}^T\boldsymbol{x}_0)\\ &= \boldsymbol{x}_0-\left(\frac{\boldsymbol{w}^T\boldsymbol{x}_0}{\|\boldsymbol{w}\|^2}\right)^T\boldsymbol{w}.\end{aligned}$$



In practice ...

- Use CVX to solve problem
- Here is a MATLAB code
- Exercise: Turn it into Python.

```
% MATLAB code: Use CVX to solve min ||x - x0||, s.t. Ax = y
m = 3; n = 2*m;
A      = randn(m,n); xstar = randn(n,1);
y      = A*xstar;
x0     = randn(n,1);
cvx_begin
   variable x(n)
   minimize( norm(x-x0) )
   subject to
        A*x == y;
cvx_end
% you may compare with the solution x0 + A'*inv(A*A')*(y-A*x0).
```

Reading List

Unconstrained Optimality Conditions

- Nocedal-Wright, Numerical Optimization. (Chapter 2.1)
- Boyd-Vandenberghe, Convex Optimization. (Chapter 9.1)

Convexity

- Nocedal-Wright, Numerical Optimization. (Chapter 1)
- Boyd-Vandenberghe, Convex Optimization. (Chapter 2 and 3)
- CMU, Convex Optimization (Lecture 2 and 4)
  https://www.stat.cmu.edu/~ryantibs/convexopt-F18/
- Stanford CS 229 (Tutorial)
  http://cs229.stanford.edu/section/cs229-cvxopt.pdf
- UCSD ECE 273 (Tutorial)
  http://eceweb.ucsd.edu/~gert/ECE273/CvxOptTutPaper.pdf

Constrained Optimization

- Nocedal-Wright, Numerical Optimization. (Chapter 12.1)