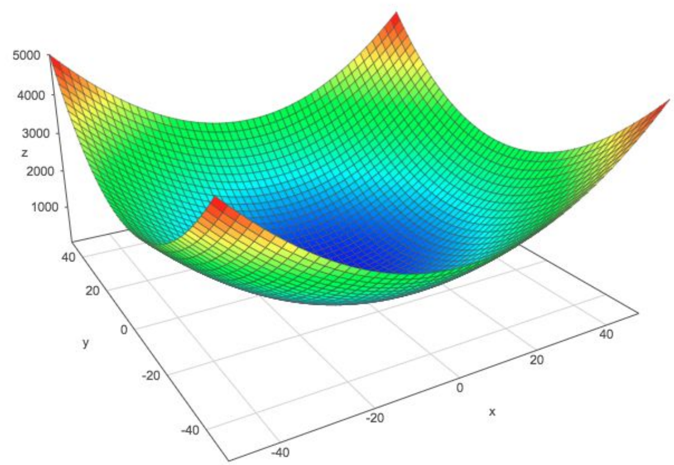
# Optimization in General and Convex Optimization




**Sunghee Yun**

**Chief Applied Scientist**
**Gauss Labs**

# Today

- Optimization
  - mathematical optimization problem formulation
  - optimization problem examples
  - solving optimization problems
- Convex Optimization
  - why convex optimization?
  - convex optimization problem examples
  - convex optimization and machine learning
- Duality
  - Lagrangian, Lagrange dual function, dual problem, optimality certificate
  - weak and strong duality
  - duality examples
  - Karush-Kuhn-Tucker (KKT) conditions
  - SVM & KKT

# Prerequisite for this talk

This talk will assume the audience

- has been exposed to basic linear algebra and calculus
- knows what function from $\mathbf{R}^n$ to $\mathbf{R}$, *i.e.*, $f : \mathbf{R}^n \to \mathbf{R}$, means

$$f(x) = f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = f(x_1, \ldots, x_n)$$

- knows what gradient is

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$

  - example: $g : \mathbf{R}^3 \to \mathbf{R}$

$$g(x_1, x_2, x_3) = x_1^2 + 1.2 x_2 x_3 - 0.5 x_1 x_3^3 + e^{x_2}$$

$$\nabla g(x) = \begin{bmatrix} 2x_1 - 0.5x_3^3 \\ 1.2x_3 + e^{x_2} \\ 1.2x_2 - 1.5x_1 x_3^2 \end{bmatrix}$$

- can distinguish componentwise inequality from that for positive semidefiniteness, $i.e.$,

$$Ax \leq b \Leftrightarrow \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} x \leq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \Leftrightarrow a_i^T x \leq b_i \text{ for } i = 1, \ldots, m,$$

  for $A \in \mathbf{R}^{m \times n}$, $x \in \mathbf{R}^n$, and $b \in \mathbf{R}^m$

- but, for $A \in \mathbf{R}^{n \times n}$

$$A \succeq 0 \Leftrightarrow A = A^T \text{ and } x^T A x \geq 0 \text{ for all } x \in \mathbf{R}^n$$

$$A \succ 0 \Leftrightarrow A = A^T \text{ and } x^T A x > 0 \text{ for all nonzero } x \in \mathbf{R}^n$$

# Mathematical optimization

- mathematical optimization problem:

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq 0, \;\; i = 1, \ldots, m \\
& h_i(x) = 0, \;\; i = 1, \ldots, p
\end{array}
$$

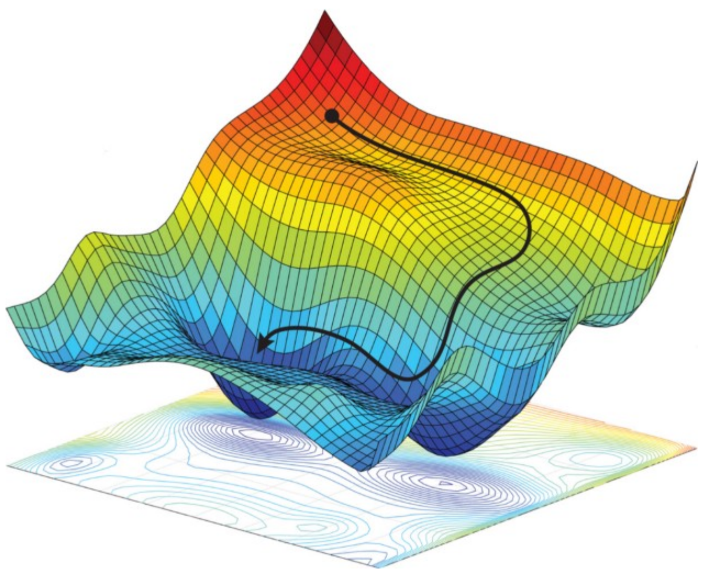
  - $x = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T \in \mathbf{R}^n$ is (vector) *optimization variable*
  - $f_0 : \mathbf{R}^n \to \mathbf{R}$ is *objective function*
  - $f_i : \mathbf{R}^n \to \mathbf{R}$ are *inequality constraint functions*
  - $h_i : \mathbf{R}^n \to \mathbf{R}$ are *equality constraint functions*

# Optimization problem examples

- circuit optimization
  - optimization variables: transistor widths, resistances, capacitances, inductances
  - objective: operating speed (or equivalently, maximum delay)
  - constraints: area, power consumption
- portfolio optimization
  - optimization variables: amounts invested in different assets
  - objective: expected return, overall risk, return variance
  - constraints: budget

# Optimization problem examples

- neural network training
  - optimization variables: neural net weights
  - objective: loss function
  - constraints: network architecture



# Solving optimization problems

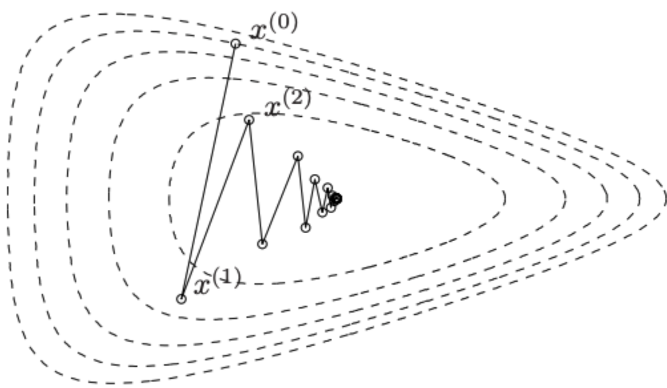
- for general optimization problems
  - extremely difficult to solve
  - lots of times, impossible to solve, $e.g.$, TSP
  - most methods try to find (good) suboptimal solutions, $e.g.$, using heuristics
- some exceptions: we can solve this problems
  - least-squares (LS), liner program (LP)
  - quadratic program (QP), quadratically constrained quadratic program (QCQP)
  - cone programming (CP), semidefinite programming (SDP)
  - optimization problems for logistic regression, support vector machine (SVM), $etc.$

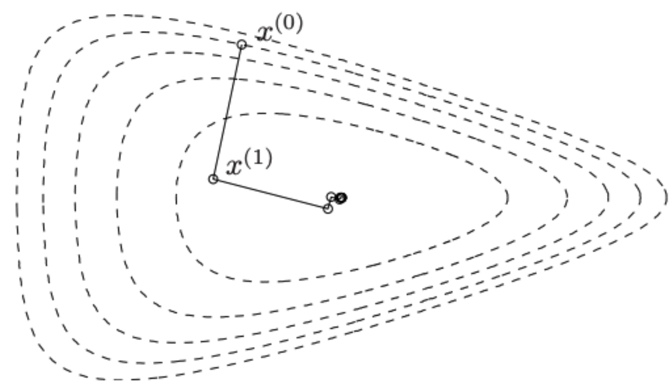
# What makes them exceptions

- they are convex optimization problems; thus, we can solve them
- what do you mean being able to solve them?
  - **polynomial-time algorithms** exist
  - for unconstrained optimization problem
    - gradient descent method, steepest descent method (first-order methods), Newton's method (second-order method), quasi-Newtons's methods, $e.g.$, BFGS
  - for constrained optimization problem
    - Newton's method with equality constraints, infeasible start Newton method
    - interior-point methods: barrier method, primal-dual method,
- what do you mean being really able to solve them?
  - can provide **optimality certificate** (or infeasibility certificate)

# (BTW, difference between gradient descent and Newton's methods)

- trajectories of two methods for a convex function
- can you guess which one is which?



# What is convex optimization?

- convex optimization problem:

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq 0,\ i = 1, \ldots, m \\
& Ax = b \Leftrightarrow a_j^T x = b_j,\ j = 1, \ldots, p
\end{array}
$$

where

- $f_i$ are convex functions ($i = 0, \ldots, m$), *i.e.*, for all $x, y \in \mathcal{D}$ and $0 \leq \lambda \leq 1$,

$$
f_i(\lambda x + (1 - \lambda) y) \leq \lambda f_i(x) + (1 - \lambda) f_i(y)
$$

(when $f_i$ are twice differentiable, equivalent to $\nabla^2 f_i(x) \succeq 0$ for all $x \in \mathcal{D}$)
- all equality constraints are linear

# General description: convex programming

- convex optimization:

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \preceq_{K_i} 0, \;\; i = 1, \ldots, m \\
& Ax = b
\end{array}
$$

  where

  - $f_0(\lambda x + (1-\lambda) y) \leq \lambda f_0(x) + (1-\lambda) f_0(y)$ for all $x, y \in \mathbf{R}^n$ and $0 \leq \lambda \leq 1$
  - $f_i : \mathbf{R}^n \to \mathbf{R}^{k_i}$ are $K_i$-convex w.r.t. proper cone $K_i \subseteq \mathbf{R}^{k_i}$
  - all equality constraints are linear

# Why convex optimization?

- many machine learning algorithms (inherently) depend on convex optimization
- quite a few optimization problems can (actually) be solved
- many engineering and scientific problems can be cast into convex optimization problems
- many more can be approximated to convex optimization
- convex optimization sheds lights on understanding intrinsic property and structure of all optimization problems

# Why convex optimization?

- many machine learning algorithms (inherently) depend on convex optimization
- quite a few optimization problems can (actually) be solved
- many engineering and scientific problems can be cast into convex optimization problems
- many more can be approximated to convex optimization
- convex optimization sheds lights on understanding intrinsic property and structure of all optimization problems

# Why convex optimization?

- many machine learning algorithms (inherently) depend on convex optimization
- quite a few optimization problems can (actually) be solved
- many engineering and scientific problems can be cast into convex optimization problems
- many more can be approximated to convex optimization
- convex optimization sheds lights on understanding intrinsic property and structure of all optimization problems

# Why convex optimization?

- many machine learning algorithms (inherently) depend on convex optimization
- quite a few optimization problems can (actually) be solved
- many engineering and scientific problems can be cast into convex optimization problems
- many more can be approximated to convex optimization
- convex optimization sheds lights on understanding intrinsic property and structure of all optimization problems

# Why convex optimization?

- many machine learning algorithms (inherently) depend on convex optimization
- quite a few optimization problems can (actually) be solved
- many engineering and scientific problems can be cast into convex optimization problems
- many more can be approximated to convex optimization
- convex optimization sheds lights on understanding intrinsic property and structure of all optimization problems

# Why convex optimization?

- many machine learning algorithms (inherently) depend on convex optimization
- quite a few optimization problems can (actually) be solved
- many engineering and scientific problems can be cast into convex optimization problems
- many more can be approximated to convex optimization
- convex optimization sheds lights on understanding intrinsic property and structure of all optimization problems

# Algorithms for convex optimization problems

- algorithms
  - classical algorithms like simplex method still work very well for many LPs
  - many state-of-the-art algorithms develoled for large-scale convex optimization problems
    * barrier methods
    * primal-dual interior-point methods

# Convex optimization example: least-squares (LS)

- LS problem

$$\text{minimize} \quad \|Ax - b\|_2^2 = \sum_{i=1}^m (a_i^T x - b_i)^2$$

  - analytic solution: any solution satisfying $(A^T A)x^* = A^T b$
  - extremely reliable and efficient algorithms
  - has been there at least since Gauss

- applications
  - LS problems are easy to recognize
  - has huge number of applications, $e.g.$, line fitting

## Convex optimization example: linear programming (LP)

- LP

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Ax \leq b\end{array}$$

  - no analytic solution
  - reliable and efficient algorithms exist, $e.g.$, simplex method, interiorpoint method
  - has been there at least since Fourier
  - used during World War II

- applications
  - less obvious to recognize (than LS)
  - lots of problems can be cast into LP, $e.g.$, network flow problem

# Convex optimization example: quadratic programming (QP)

- QP - assuming $P \in \mathbf{S}^n_{++}$, *i.e.*, $P \succ 0$

$$
\begin{array}{ll}
\mbox{minimize} & x^T P x + q^T x \\
\mbox{subject to} & Ax \leq b
\end{array}
$$

  - no analytic solution
  - reliable and efficient algorithms exist, *e.g.*, interiorpoint method

- applications
  - less obvious to recognize (than LP)
  - lots of problems can be cast into QP, *e.g.*, model preditive control (MPC), signal and image processing, optimal portfolio, *etc.*

## Convex optimization example: semidefinite programming (SDP)

- SDP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F_0 + x_1 F_1 + \cdots + x_n F_n \succeq 0 \end{array}$$

  - again, no analytic solution
  - again, reliable and efficient algorithms exist, *e.g.*, interior-point method

- applications
  - never easy to recognize
  - lots of problems, *e.g.*, optimal control theory, can be cast into SDP
  - extremely non-obvious, but convex, hence global optimality easily achieved!

# Convex optimization example: max-det problem

- max-det program:

$$\begin{array}{ll} \text{minimize} & c^T x + \log\det(F_0 + x_1 F_1 + \cdots + x_n F_n) \\ \text{subject to} & G_0 + x_1 G_1 + \cdots + x_n G_n \succeq 0 \\ & F_0 + x_1 F_1 + \cdots + x_n F_n \succ 0 \end{array}$$

  - again, no analytic solution
  - again, reliable and efficient algorithms exist, $e.g.$, interior-point method
  - recent technology
- applications
  - never easy to recognize
  - lots of stochastic optimization problems, $e.g.$, every covariance matrix is positive semidefinite
  - again convex, hence global optimality (relatively) easily achieved!

# Properties convex optimization enjoys!

- convex optimization problems can be solved extremely reliably and fast
- a local minimum is a global minimum, which is implied by

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

because Taylor theorem implies

$$f(y) \simeq f(x) + \nabla f(x)^T (y - x) + (y - x)^T \nabla^2 f(x)(y - x)/2$$

- nice theoretical property, $e.g.$, self-concordance implies complexity bound with Newton's method

$$\frac{f(x_0) - p^*}{\gamma} + \log_2 \log_2(1/\epsilon)$$

- even better pratical performance!

# Mathematical formulation for supervised learning

- given training set, $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, where $x^{(i)} \in \mathbf{R}^p$ and $y^{(i)} \in \mathbf{R}^q$
- want to find function $g_\theta : \mathbf{R}^p \to \mathbf{R}^q$ parameterized by learning parameter, $\theta \in \mathbf{R}^n$
  - $g_\theta(x)$ desired to be as close as possible to $y$ for future/unseen data $(x, y) \in \mathbf{R}^p \times \mathbf{R}^q$
  - *i.e.*, $g_\theta(x) \sim y$
- define a loss function $l : \mathbf{R}^q \times \mathbf{R}^q \to \mathbf{R}_+$
- solve the optimization problem:

$$\begin{array}{ll} \mbox{minimize} & f(\theta) = \frac{1}{m} \sum_{i=1}^m l(g_\theta(x^{(i)}), y^{(i)}) \\ \mbox{subject to} & \theta \in \Theta \end{array}$$

# Linear regression

- (simple) linear regression is a supervised learning problem when
  - $q = 1$, *i.e.*, the output is scalar
  - $g_\theta(x) = \theta^T \begin{bmatrix} 1 \\ x \end{bmatrix} = \theta_0 + \theta_1 x_1 + \cdots + \theta_p x_p$, *i.e.*, $n = p + 1$
  - $l : \mathbf{R} \times \mathbf{R} \to \mathbf{R}_+$ is defined by $l(y_1, y_2) = (y_1 - y_2)^2$
  - $\Theta = \mathbf{R}^{p+1}$, *i.e.*, parameter domain is the set of all real numbers
- formulation

$$\text{minimize} \quad f(\theta) = \frac{1}{m} \sum_{i=1}^m \left( \theta^T \begin{bmatrix} 1 \\ x^{(i)} \end{bmatrix} - y^{(i)} \right)^2$$

# Solution method for linear regression

- linear regression is nothing but LS since

$$
\begin{aligned}
mf(\theta) &= \sum_{i=1}^{m} \left( \theta^T \left[ \begin{array}{c} 1 \\ x^{(i)} \end{array} \right] - y^{(i)} \right)^2 = \left\| \left[ \begin{array}{cc} 1 & {x^{(1)}}^T \\ \vdots & \vdots \\ 1 & {x^{(m)}}^T \end{array} \right] \theta - \left[ \begin{array}{c} y^{(1)} \\ \vdots \\ y^{(m)} \end{array} \right] \right\|_2^2 \\
&= \|X\theta - y\|_2^2
\end{aligned}
$$

- just another LS problem
- thus, analytic solution exists; solve the normal equation:

$$
(X^T X)\theta = X^T y
$$

## How can we solve linear regression with constraints?

- what if we have one constraint?

$$\begin{array}{ll} \text{minimize} & f(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left( \theta^T \left[ \begin{array}{c} 1 \\ x^{(i)} \end{array} \right] - y^{(i)} \right)^2 \\ \text{subject to} & \theta_1 \geq 0 \end{array}$$

- no analytic solution exists (with only one constraint) in general
- however, convex optimization algorithms can solve it as easily as original problem
- actually, with *any* number of convex constraints

$$\begin{array}{ll} \text{minimize} & f(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left( \theta^T \left[ \begin{array}{c} 1 \\ x^{(i)} \end{array} \right] - y^{(i)} \right)^2 \\ \text{subject to} & h_i(\theta) \leq 0 \text{ for } i = 1, \ldots, l \\ & A\theta = b \end{array}$$

## How can we solve linear regression with constraints?

- what if we have one constraint?

$$\begin{array}{ll} \text{minimize} & f(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\theta^T \left[\begin{array}{c} 1 \\ x^{(i)} \end{array}\right] - y^{(i)}\right)^2 \\ \text{subject to} & \theta_1 \geq 0 \end{array}$$

- no analytic solution exists (with only one constraint) in general
- however, convex optimization algorithms can solve it as easily as original problem
- actually, with *any* number of convex constraints

$$\begin{array}{ll} \text{minimize} & f(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\theta^T \left[\begin{array}{c} 1 \\ x^{(i)} \end{array}\right] - y^{(i)}\right)^2 \\ \text{subject to} & h_i(\theta) \leq 0 \text{ for } i = 1, \ldots, l \\ & A\theta = b \end{array}$$

## How can we solve linear regression with constraints?

- what if we have one constraint?

$$\begin{array}{ll} \text{minimize} & f(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\theta^T \left[\begin{array}{c} 1 \\ x^{(i)} \end{array}\right] - y^{(i)}\right)^2 \\ \text{subject to} & \theta_1 \geq 0 \end{array}$$

- no analytic solution exists (with only one constraint) in general
- however, convex optimization algorithms can solve it as easily as original problem
- actually, with *any* number of convex constraints

$$\begin{array}{ll} \text{minimize} & f(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\theta^T \left[\begin{array}{c} 1 \\ x^{(i)} \end{array}\right] - y^{(i)}\right)^2 \\ \text{subject to} & h_i(\theta) \leq 0 \text{ for } i = 1, \ldots, l \\ & A\theta = b \end{array}$$

# How can we solve linear regression with constraints?

- what if we have one constraint?

$$
\begin{array}{ll}
\text{minimize} & f(\theta) = \frac{1}{m}\sum_{i=1}^m \left(\theta^T \begin{bmatrix} 1 \\ x^{(i)} \end{bmatrix} - y^{(i)}\right)^2 \\
\text{subject to} & \theta_1 \geq 0
\end{array}
$$

- no analytic solution exists (with only one constraint) in general
- however, convex optimization algorithms can solve it as easily as original problem
- actually, with *any* number of convex constraints

$$
\begin{array}{ll}
\text{minimize} & f(\theta) = \frac{1}{m}\sum_{i=1}^m \left(\theta^T \begin{bmatrix} 1 \\ x^{(i)} \end{bmatrix} - y^{(i)}\right)^2 \\
\text{subject to} & h_i(\theta) \leq 0 \text{ for } i = 1, \ldots, l \\
& A\theta = b
\end{array}
$$

# Ridge regression

- Ridge regression solves the following problem: (for some $\lambda > 0$)

$$\text{minimize} \quad f_0(x) = \|Ax - b\|_2^2 + \lambda \|x\|_2^2$$

  - with regularization to preventing overfitting
- can be reformulated as

$$\text{minimize} \quad f_0(x) = \left\| \begin{bmatrix} A \\ \sqrt{\lambda} I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2$$

- yet another LS, hence solve the following normal equation:

$$\begin{bmatrix} A^T & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} A \\ \sqrt{\lambda} I \end{bmatrix} x = (A^T A + \lambda I) x = A^T b$$

# Least absolute shrinkage & selection operator (lasso)

- Lasso solves (a problem equivalent to) the following problem:

$$\text{minimize} \quad f_0(x) = \|Ax - b\|^2 + \lambda \|x\|_1$$

  - 1-norm penalty term for parameter selection
  - similar to drop-out technique for regularization

- However, the objective funtion is *not* smooth.

- simple trick resolves this smoothness problem
  - with additional convex inequality constraints and affine equality constraints

$$\begin{array}{ll} \text{minimize} & f_0(x) = \|Ax - b\|^2 + \lambda \sum_{i=1}^n z_i \\ \text{subject to} & -z_i \leq x_i \leq z_i, \ i = 1, \ldots, n \end{array}$$

# Support vector machine (SVM)

- problem definition:
  - given $x^{(i)} \in \mathbf{R}^p$: input data, and $y^{(i)} \in \{-1, 1\}$: output labels
  - find hyperplane which separates two different classes as distinctively as possible (in some measure)
- (typical) formulation:

$$
\begin{array}{ll}
\text{minimize} & \|a\|_2^2 + \gamma \sum_{i=1}^m u_i \\
\text{subject to} & y^{(i)}(a^T x^{(i)} + b) \geq 1 - u_i, \ i = 1, \ldots, m \\
& u \succeq 0
\end{array}
$$

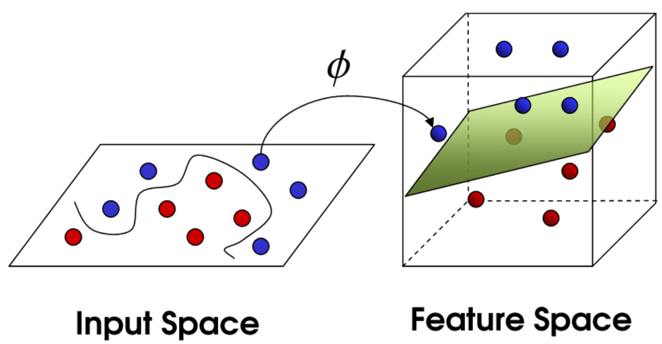
  - optimization variables: $a \in \mathbf{R}^n$, $b \in \mathbf{R}$, $u \in \mathbf{R}^m$
  - convex optimization problem, hence stable and efficient algorithms exist even for very large problems
  - has worked extremely well in practice

# SVM using kernels

- use feature transformation $\phi : \mathbf{R}^p \to \mathbf{R}^q$ (with $q > p$)
- formulation:

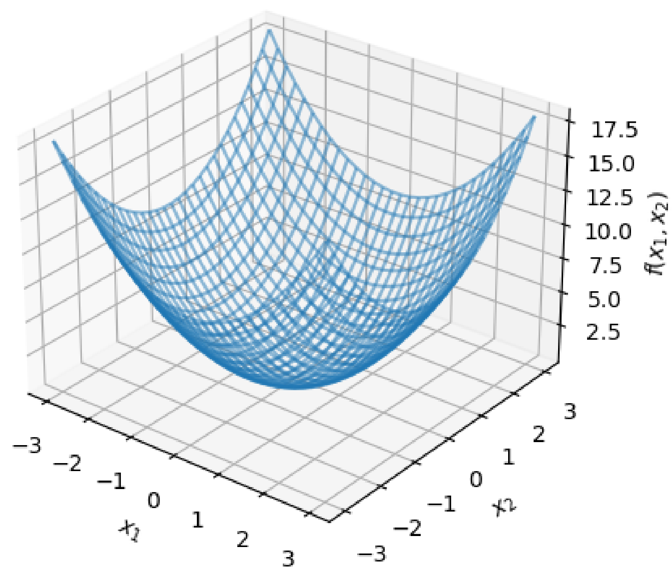
$$
\begin{array}{ll}
\mbox{minimize} & \|\tilde{a}\|_2^2 + \gamma \sum_{i=1}^m \tilde{u}_i \\
\mbox{subject to} & y^{(i)}(\tilde{a}^T \phi(x^{(i)}) + \tilde{b}) \geq 1 - \tilde{u}_i, \;\; i = 1, \ldots, m \\
& \tilde{u} \succeq 0
\end{array}
$$

- still convex optimization problem



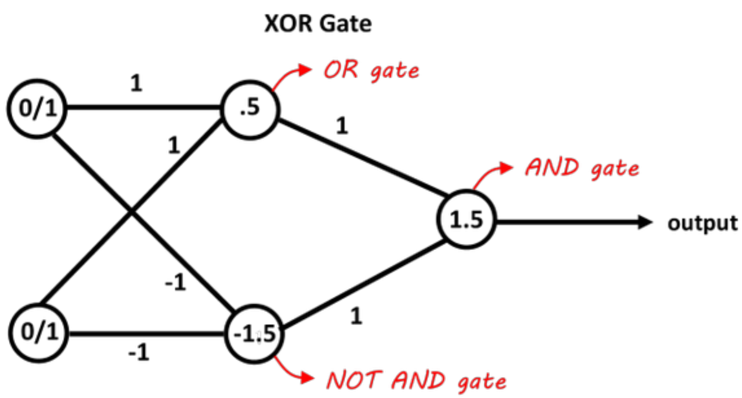
# Why NN is not a convex function?

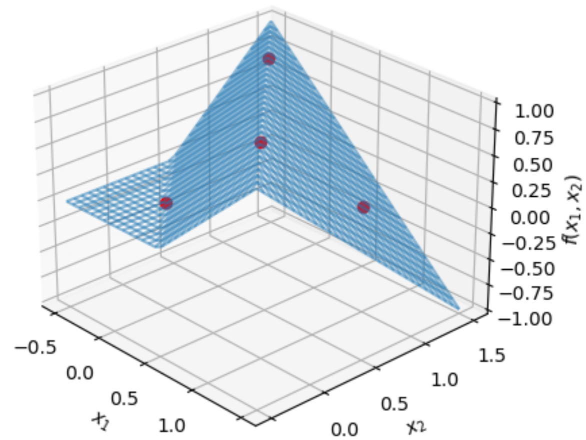
- graph of a convex function



# Why NN is not a convex function?

- graph of a very simple neural network with one hidden layer

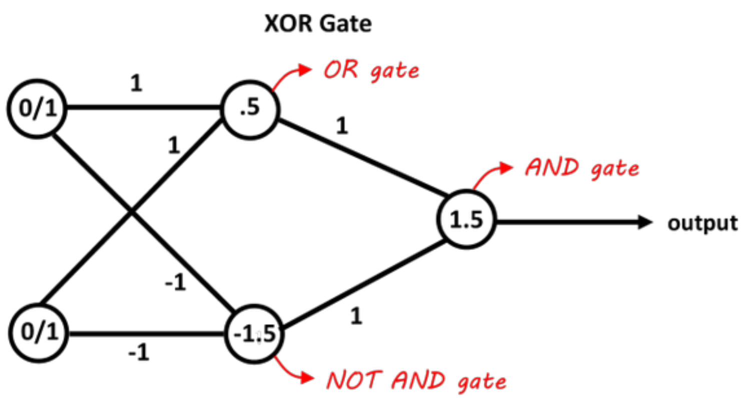


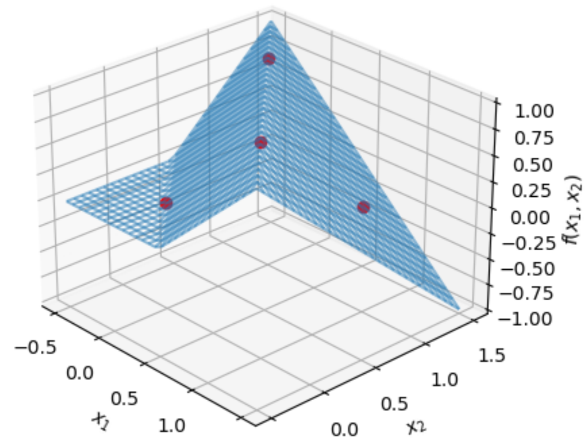




- What is wrong with this argument?
- Yes, we should look at error function with respect to *weights*

# Why NN is not a convex function?

- graph of a very simple neural network with one hidden layer

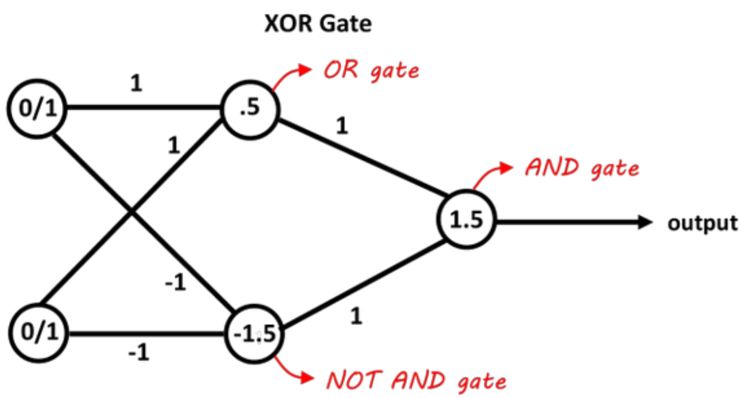


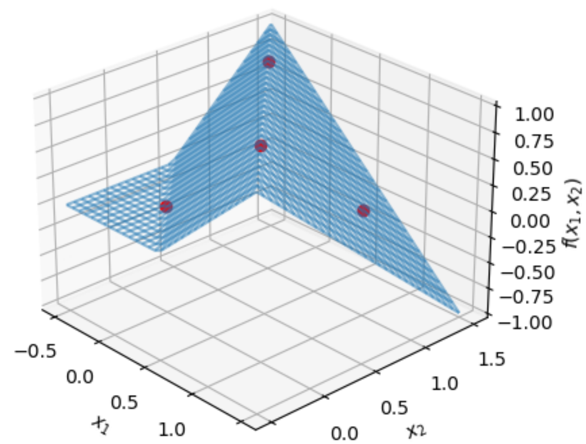




- What is wrong with this argument?
- Yes, we should look at error function with respect to *weights*

# Why NN is not a convex function?

- graph of a very simple neural network with one hidden layer

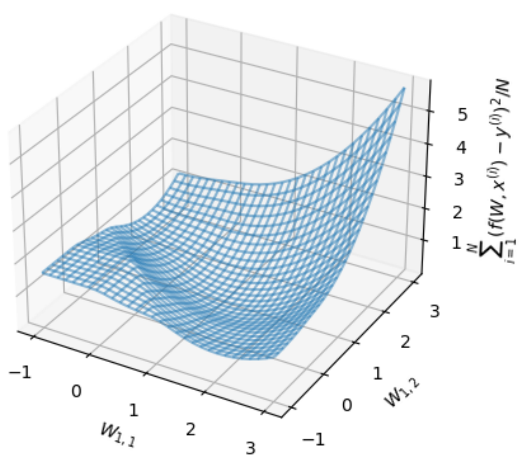


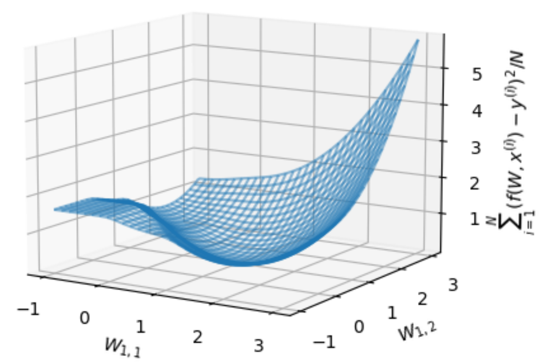




- What is wrong with this argument?
- Yes, we should look at error function with respect to *weights*

# Why NN is not a convex function?

- graph of the error function as a function of *weights*




- this is *why* NN's error function is not a convex function in weights (parameters)

# Duality

- every (constrained) optimization problem has a *dual problem* (whether or not it's a convex optimization problem)

- every dual problem is a *convex optimization problem* (whether or not it's a convex optimization problem)

- duality provides *optimality certificate*, hence plays *central role* for modern optimization and some machine learning algorithm implementation

- (usually) solving one readily solves the other!

# Lagrangian

- standard form problem:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \ i = 1, \ldots, m \\ & h_i(x) = 0, \ i = 1, \ldots, p \end{array}$$

  where $x \in \mathbf{R}^n$ is optimization variable, $\mathcal{D}$ is domain, $p^*$ is optimal value

- Lagrangian: $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ with $\mathbf{dom}\, L = \mathcal{D} \times \mathbf{R}^m \times \mathbf{R}^p$ defined by

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x)$$

  - $\lambda_i$: Lagrange multiplier associated with $f_i(x) \leq 0$
  - $\nu_i$: Lagrange multiplier associated with $h_i(x) = 0$

# Lagrange dual function

- Lagrange dual function: $g : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ defined by

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} \left( f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x) \right)$$

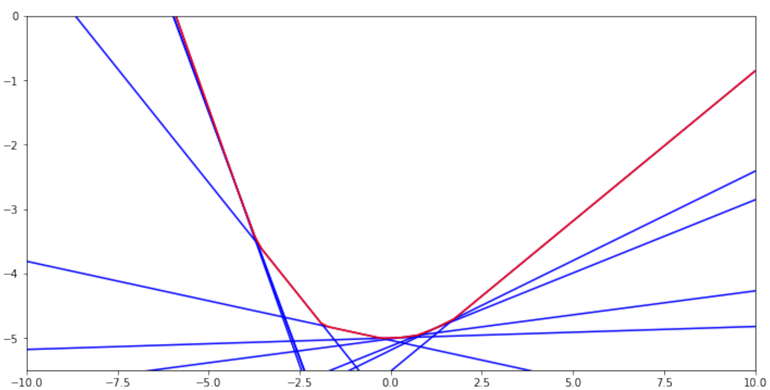
  - $g$ is *always* concave
  - $g(\lambda, \nu)$ can be $-\infty$

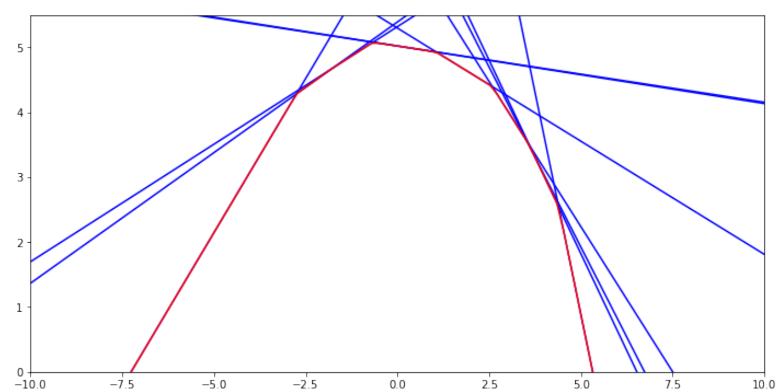
- lower bound property: if $\lambda \succeq 0$, then $g(\lambda, \nu) \leq p^*$

# Sup of convex functions and inf of concave functions

- $\sup_{\alpha \in \mathcal{A}} f_\alpha(x)$ is convex if $f_\alpha(x)$ is convex for all $\alpha \in \mathcal{A}$

- $\inf_{\alpha \in \mathcal{A}} f_\alpha(x)$ is concave if $f_\alpha(x)$ is concave for all $\alpha \in \mathcal{A}$



# Proof

- let $g(x) = \sup_{\alpha \in \mathcal{A}} f_\alpha(x)$ where $f_\alpha(x)$ is a convex function for all $\alpha \in \mathcal{A}$
- then for any $0 < \lambda < 1$

$$
\begin{aligned}
g(\lambda x + (1-\lambda)y) &= \sup_{\alpha \in \mathcal{A}} f_\alpha(\lambda x + (1-\lambda)y) \leq \sup_{\alpha \in \mathcal{A}} (\lambda f_\alpha(x) + (1-\lambda) f_\alpha(y)) \\
&\leq \sup_{\alpha \in \mathcal{A}} \lambda f_\alpha(x) + \sup_{\alpha \in \mathcal{A}} (1-\lambda) f_\alpha(y) = \lambda g(x) + (1-\lambda) g(y)
\end{aligned}
$$

- thus, $g(x)$ is a convex function
- concavity of the latter can be proved similarly

# Why dual function is lower bound for the optimal value?

- you only need middle school math!
- for any (primal) feasible $\tilde{x}$, any $\lambda \geq 0$, and $\nu$

$$
\begin{aligned}
g(\lambda, \nu) &= \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \leq L(\tilde{x}, \lambda, \nu) = f_0(\tilde{x}) + \sum_{i=1}^m \lambda_i f_i(\tilde{x}) + \sum_{i=1}^p \nu_i h_i(\tilde{x}) \\
&= f_0(\tilde{x}) + \sum_{i=1}^m \lambda_i f_i(\tilde{x}) \leq f_0(\tilde{x})
\end{aligned}
$$

because feasibility implies $\lambda_i f_i(\tilde{x}) \leq 0$ $(1 \leq i \leq m)$ and $\nu_i h_i(\tilde{x})$ $(1 \leq i \leq p)$

- thus, for any $\lambda \geq 0$ and $\nu$,

$$
g(\lambda, \nu) \leq p^* = \inf_{x \in \mathcal{D}} f_0(x)
$$

# Dual problem

- Lagrange dual problem:

$$\begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \geq 0 \end{array}$$

  - convex optimization problem (because $-g(\lambda, \nu)$ is a convex function)
  - provides a lower bound on $p^*$

- let $d^*$ denote the optimal value for the dual problem
  - week duality: $d^* \leq p^*$
  - strong duality: $d^* = p^*$

# Weak duality

- weak duality implies $d^* \leq p^*$
  - always true
  - provides *nontrivial* lower bounds, especially, for difficult problems, $e.g.$, solving the following SDP:

$$\begin{array}{ll} \text{maximize} & -\mathbf{1}^T \nu \\ \text{subject to} & W + \mathbf{diag}(\nu) \succeq 0 \end{array}$$

gives a lower bound for max-cut problem (NP-complete)

$$\begin{array}{ll} \text{minimize} & x^T W x \\ \text{subject to} & x_i^2 = 1, \ i = 1, \ldots, n \end{array}$$

# Strong duality

- strong duality implies $d^* = p^*$
  - not necessarily hold; does not hold in general
  - *usually* holds for convex optimization problems
  - conditions which guarantee strong duality in convex problems called *constraint qualifications*

# Slater's constraint qualification

- strong duality holds for a convex optimization problem:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \ i = 1, \ldots, m \\ & Ax = b \end{array}$$

  - if it is strictly feasible, *i.e.*, there exists $x \in \mathbf{R}^n$ such that

$$f_i(x) < 0, \ i = 1, \ldots, m, \ Ax = b$$

- Slater's condition
  - also guarantees the dual optimum is attained (if $p^* > -\infty$)
  - linear inequalities do not need to hold with strict inequalities

# Duality example: LP

- primal problem:

$$\begin{array}{ll}\text{minimize} & c^T x\\ \text{subject to} & Ax \leq b\end{array}$$

- dual function:

$$g(\lambda) = \inf_x \left(\left(c + A^T\lambda\right)^T x - b^T\lambda\right) = \begin{cases} -b^T\lambda & \text{if } A^T\lambda + c = 0\\ -\infty & \text{otherwise}\end{cases}$$

- dual problem:

$$\begin{array}{ll}\text{maximize} & -b^T\lambda\\ \text{subject to} & A^T\lambda + c = 0\\ & \lambda \geq 0\end{array}$$

  - Slater's condition implies that $p^* = d^*$ if $A\tilde{x} < b$ for some $\tilde{x}$
  - truth is, $p^* = d^*$ except when both primal and dual are infeasible

# Duality example: QP

- primal problem (assuming $P \in \mathbf{S}^n_{++}$):

$$\begin{array}{ll} \text{minimize} & x^T P x \\ \text{subject to} & Ax \preceq b \end{array}$$

- dual function:

$$g(\lambda) = \inf_x \left( x^T P x + \lambda^T (Ax - b) \right) = -\frac{1}{4} \lambda^T A P^{-1} A^T \lambda - b^T \lambda$$

- dual problem:

$$\begin{array}{ll} \text{maximize} & -\lambda^T A P^{-1} A^T \lambda / 4 - b^T \lambda \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

  - Slater's condition implies that $p^* = d^*$ if $A\tilde{x} \prec b$ for some $\tilde{x}$
  - truth is, $p^* = d^*$ always!

# Dual problem provides optimality certificate!

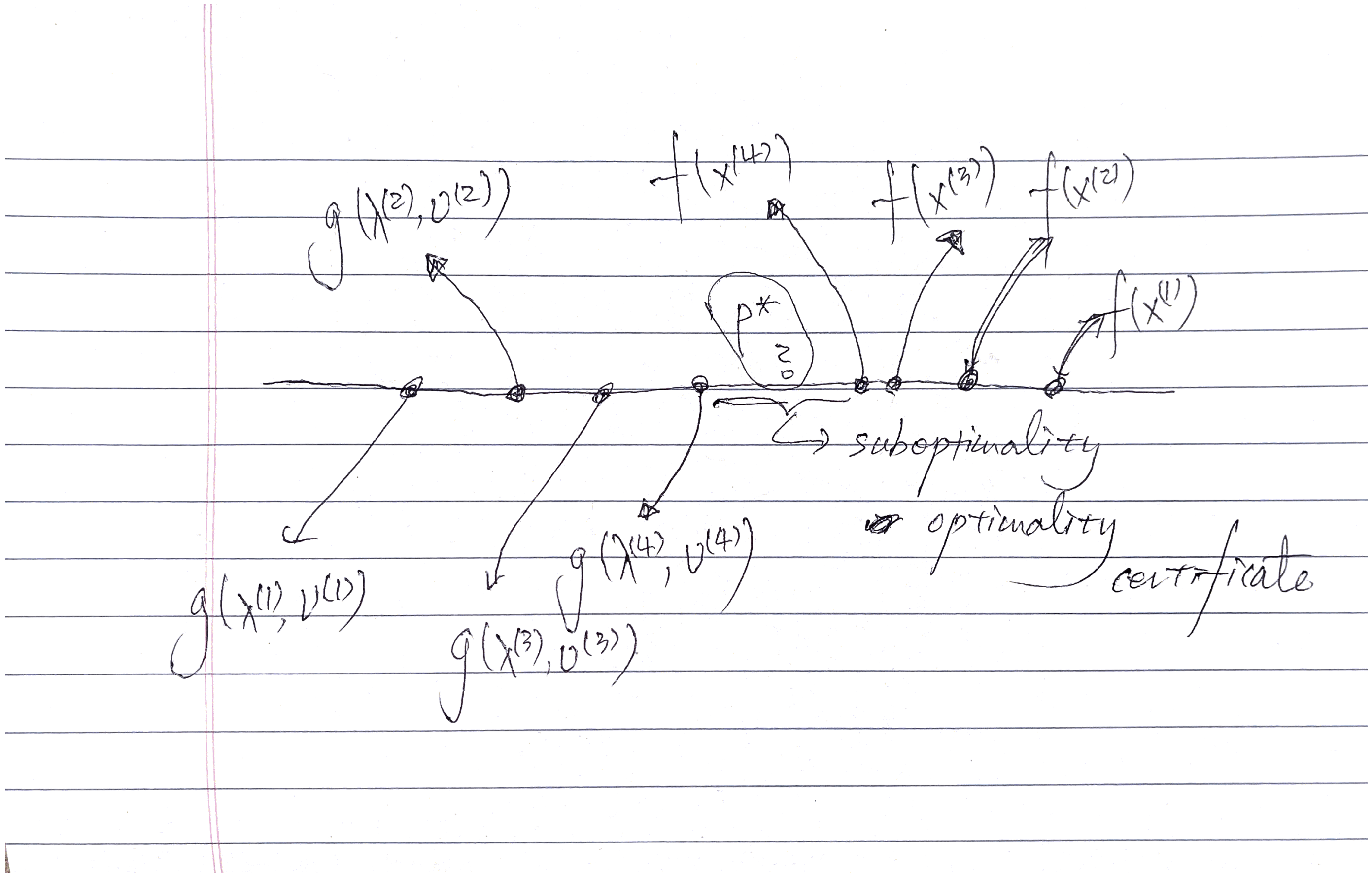
- many algorithms solves the dual problem simultaneously
- (sometimes) Lagrangian dual variables obtained with no additional cost, $e.g.$, barrior method for inequality constrained problems
- if iterative algorithm generates solution sequence,

$$(x^{(1)},\lambda^{(1)},\nu^{(1)}) \to (x^{(2)},\lambda^{(2)},\nu^{(2)}) \to (x^{(3)},\lambda^{(3)},\nu^{(3)}) \to \cdots$$

then, we have an optimality certificate:

$$\begin{array}{rcl} f(x^{(k)}) \geq p^* \text{ and } g(\lambda^{(k)},\nu^{(k)}) \leq p^* & \Leftrightarrow & f(x^{(k)}) - p^* \leq f(x^{(k)}) - g(\lambda^{(k)},\nu^{(k)}) \\ & \Leftrightarrow & g(\lambda^{(k)},\nu^{(k)}) \leq p^* \leq f(x^{(k)}) \end{array}$$

# Optimality certificate with primal and dual paths



# Newton's method for analytic centering problem

- each curve corresponds to four different initial points

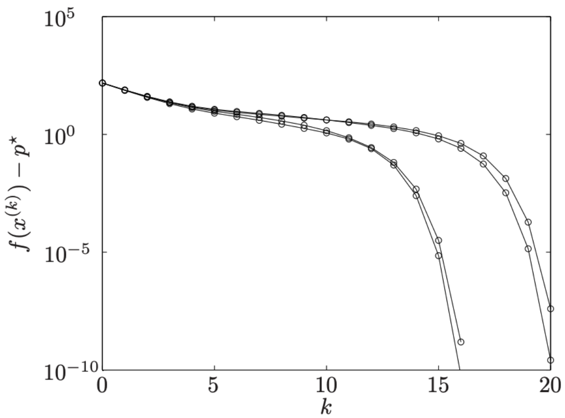



**Figure 10.6** Error $f(x^{(k)}) - p^\star$ in Newton's method, applied to an equality constrained analytic centering problem of size $p = 100$, $n = 500$. The different curves correspond to four different starting points. Final quadratic convergence is clearly evident.

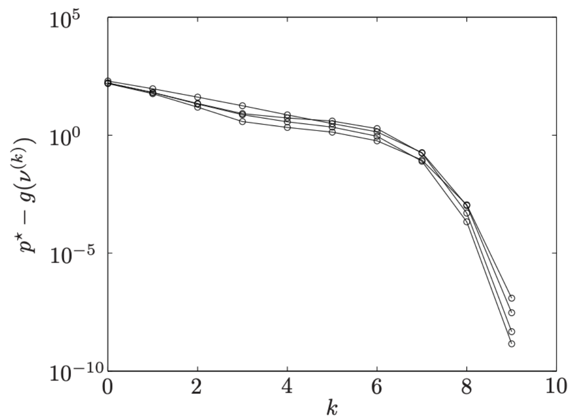
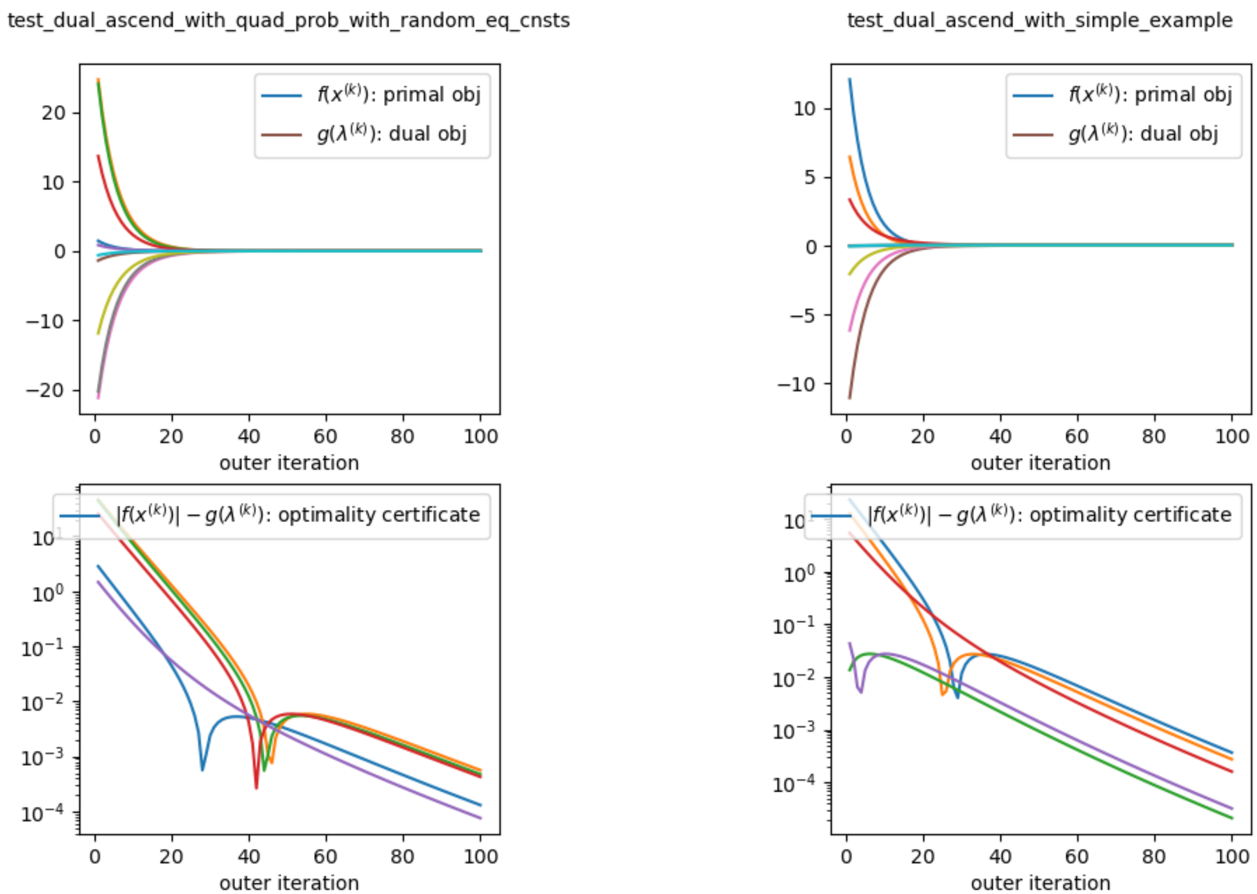



**Figure 10.7** Error $|g(\nu^{(k)}) - p^\star|$ in Newton's method, applied to the dual of the equality constrained analytic centering problem.

(Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, NY, USA, 2004.)

# Dual ascend method for QP



## Complementary slackness

- again, you only need middle school math!
- assume strong dualtiy holds, $x^*$ is primal optimal, and $(\lambda^*, \nu^*)$ is dual optimal

$$
\begin{aligned}
f_0(x^*) &= g(\lambda^*, \nu^*) = \inf_{x \in \mathcal{D}} L(x, \lambda^*, \nu^*) \\
&\leq L(x^*, \lambda^*, \nu^*) \\
&= f_0(x^*) + \sum_{i=1}^m \lambda_i^* f_i(x^*) + \sum_{i=1}^p \nu_i^* h_i(x^*) \\
&\leq f_0(x^*)
\end{aligned}
$$

because $\lambda^* \geq 0$, $f_i(x^*) \leq 0$, and $h_i(x^*) = 0$

- note if $a \leq b \leq a$, $a = b$

- thus, all inequalities are tight, $i.e.$, they hold with equalities
  - $x^*$ minimizes $L(x, \lambda^*, \nu^*)$, thus

$$\nabla_x L(x^*, \lambda^*, \nu^*) = \nabla f_0(x^*) + \sum_{i=1}^m \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^p \nu_i^* \nabla h_i(x^*) = 0$$

    when the functions are differentiable
  - $\sum_{i=1}^m \lambda_i^* f_i(x^*) = 0$ where $\lambda_i^* f_i(x^*) \geq 0$ for all $i$, thus

$$\lambda_i^* f_i(x^*) = 0 \text{ for all } i$$

    known as *complementary slackness*

$$\lambda_i^* > 0 \Rightarrow f_i(x^*) = 0, \quad f_i(x^*) < 0 \Rightarrow \lambda_i^* = 0$$

# Karush-Kuhn-Tucker (KKT) conditions

- KKT (optimality) conditions consist of
  - primal feasibility: $f_i(x) \leq 0$ for all $1 \leq i \leq m$, $h_i(x) = 0$ for all $1 \leq i \leq p$
  - dual feasibility: $\lambda \succeq 0$
  - complementary slackness: $\lambda_i f_i(x) = 0$
  - zero gradient of Lagrangian: $\nabla f_0(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) + \sum_{i=1}^p \nu_i \nabla h_i(x) = 0$
- if strong daulity holds and $x^*$, $\lambda^*$, and $\nu^*$ are optimal, they satisfy KKT condtions!

# KKT conditions for convex optimization problem

- if $\tilde{x}$, $\tilde{\lambda}$, and $\tilde{\nu}$ satisfy KKT for convex optimization problem, then they are optimal!
  - complementary slackness implies $f_0(\tilde{x}) = L(\tilde{x}, \tilde{\lambda}, \tilde{\nu})$
  - last conidtion together with convexity implies $g(\tilde{\lambda}, \tilde{\nu}) = L(\tilde{x}, \tilde{\lambda}, \tilde{\nu})$
- thus, for example, if Slater's condition is satisfied, $x$ is optimal if and only if there exist $\lambda$, $\nu$ that satisfy KKT conditions
  - Slater's condition implies strong dualtiy, hence dual optimum is attained
  - this generalizes optimality condition $\nabla f_0(x) = 0$ for unconstrained problem

# Dual problem of SVM problem

- optimization problem for SVM:

$$
\begin{array}{ll}
\text{minimize} & \frac{1}{2}\|a\|_2^2 + \gamma \sum_{i=1}^m u_i \\
\text{subject to} & y^{(i)}(a^T x^{(i)} + b) \geq 1 - u_i, \ i = 1, \ldots, m \\
& u \succeq 0
\end{array}
$$

- Lagrangian:

$$
\begin{aligned}
& L(a, b, u, \lambda, \nu) \\
& \quad = \frac{1}{2}\|a\|_2^2 + \gamma \sum_{i=1}^m u_i + \sum_{i=1}^m \lambda_i (1 - u_i - y^{(i)}(a^T x^{(i)} + b)) + \sum_{i=1}^m \nu_i(-u_i) \\
& \quad = \frac{1}{2}\|a\|_2^2 - \left( \sum_{i=1}^m \lambda_i y^{(i)} x^{(i)} \right)^T a - b \sum_{i=1}^m \lambda_i y^{(i)} + \sum_{i=1}^m u_i(\gamma - \lambda_i - \nu_i) + \sum_{i=1}^m \lambda_i
\end{aligned}
$$

- dual function

$$g(\lambda, \nu) = \left\{ \begin{array}{ll} -\frac{1}{2} \left\| \sum_{i=1}^m \lambda_i y^{(i)} x^{(i)} \right\|_2^2 + \sum_{i=1}^m \lambda_i & \text{if } \sum_{i=1}^m \lambda_i y^{(i)} = 0, \lambda_i + \nu_i = \gamma \\ -\infty & \text{otherwise} \end{array} \right.$$

- dual problem

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^m \lambda_i - \frac{1}{2} \left\| \sum_{i=1}^m \lambda_i y^{(i)} x^{(i)} \right\|_2^2 \\ \text{subject to} & \sum_{i=1}^m \lambda_i y^{(i)} = 0 \\ & \lambda_i + \nu_i = \gamma \text{ for } i = 1, \ldots, m \end{array}$$

- or equivalently,

$$
\begin{array}{ll}
\text{maximize} & \sum_{i=1}^m \lambda_i - \frac{1}{2} \sum_{1 \leq i,j \leq m} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} \\
\text{subject to} & \sum_{i=1}^m \lambda_i y^{(i)} = 0 \\
& \lambda_i + \nu_i = \gamma \text{ for } i = 1, \ldots, m
\end{array}
$$

- yet again, equivalently,

$$
\begin{array}{ll}
\text{maximize} & \sum_{i=1}^m \lambda_i - \frac{1}{2} \lambda^T P \lambda \\
\text{subject to} & \sum_{i=1}^m \lambda_i y^{(i)} = 0 \\
& \lambda_i + \nu_i = \gamma \text{ for } i = 1, \ldots, m
\end{array}
$$

where $P = X^T X \succeq 0$ and $X = \left[ \begin{array}{ccc} y^{(1)} x^{(1)} & \cdots & y^{(m)} x^{(m)} \end{array} \right] \in \mathbf{R}^{n \times m}$

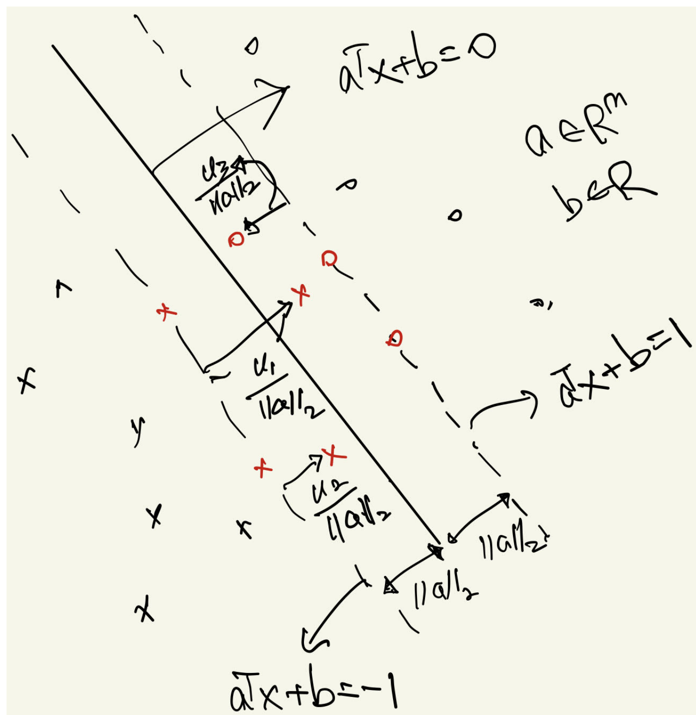
- $i.e.$, dual problem is *quadratic program*

# KKT conditions for SVM problem

- assume that $a^*$, $b^*$, $u^*$ are primal optimal and $\lambda^*$ and $\nu^*$ are dual optimal, then KKT conditions imply
  - $y^{(i)}({a^*}^T x^{(i)} + b^*) \geq 1 - u_i^*$ for $i = 1, \ldots, m$
  - $u_i^* \geq 0, \lambda_i^* \geq 0, \nu_i^* \geq 0, \lambda_i^* + \nu_i^* = \gamma$ for $i = 1, \ldots, m$
  - $\nu_i^* u_i^* = 0$ for $i = 1, \ldots, m$
  - $\lambda_i^*(1 - u_i^* - y^{(i)}({a^*}^T x^{(i)} + b^*)) = 0$ for $i = 1, \ldots, m$
  - $\sum_{i=1}^m \lambda_i^* y^{(i)} = 0$
  - $a^* = \sum_{i=1}^m \lambda_i^* y^{(i)} x^{(i)}$
- $x^{(i)}$ with $\lambda_i^* > 0$ are called *support vectors*!
  - those with positive slacks ($u_i^* > 0$), $\lambda_i^* = \gamma$
  - those on the edge ($u_i^* = 0$), $0 < \lambda_i^* \leq \gamma$
- then the boundary can be characterized by $\sum_{i=1}^m \lambda_i^* y^{(i)} {x^{(i)}}^T x + b^*$
  - with kernel, the boundary is $\sum_{i=1}^m \lambda_i^* y^{(i)} K(x, x^{(i)}) + b^*$

# Graphical representation of support vectors

- red circles and crosses indicate the support vectors



# Next time

- we can discuss
  - sensitivity analysis using Lagrange dual variables
  - various interpretations for dual problems and dual variables
  - some algorithms for convex optimization, $e.g.$, gradient descent, Newton's method
  - their convergence analysis
  - various applications in approximation, fitting, statistical estimation, geometric problems, $etc.$

## References

[1] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

**Thank you!** for watching lots of equations during your lunch time. ☺