# Convex Optimization: Old Tricks for New Problems

Ryota Tomioka

The University of Tokyo

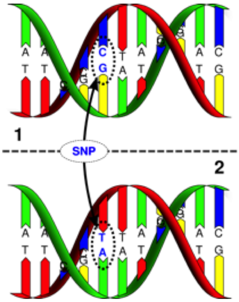
2012-08-15 @ DTU PhD Summer Course

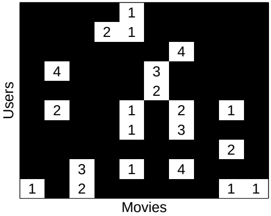
# Introduction

## Why care about convex optimization (and sparsity)?

# Why do we care about optimization — sparse estimation

- High dimensional problems (dimension $\gg$ # samples)
  - Bioinformatics (microarray, SNP analysis, etc)
  - Text-mining (POS tagging, )
  - Magnetic resonance imaging — compressed sensing
- Structure inference
  - Collaborative filtering — low-rank structure
  - Graphical model inference— sparse graph structure



# Ex. 1: SNP (single nucleotide polymorphism) analysis

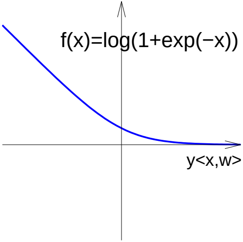
$\boldsymbol{x}_i$: input (SNP), $y_i = 1$: has the illness, $y_i = -1$: healthy

Goal: Infer the association from genetic variability $\boldsymbol{x}_i$ to the illness $y_i$.

Logistic regression

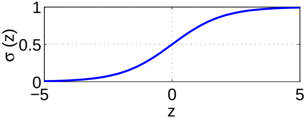
$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{\sum_{i=1}^{m} \log(1+\exp(-y_i \langle \boldsymbol{x}_i, \boldsymbol{w}\rangle))}_{\text{data-fit}} \quad + \quad \underbrace{\lambda \|\boldsymbol{w}\|_1}_{\text{Regularization}}$$

- E.g., # SNPs $n = 500,000$, # subjects $m = 5,000$
- MAP etimation with the *logistic loss f*.




$$\log(1+e^{-yz}) = -\log P(Y = y|z)$$

where $P(Y = +1|z) = \frac{e^z}{1+e^z}$.

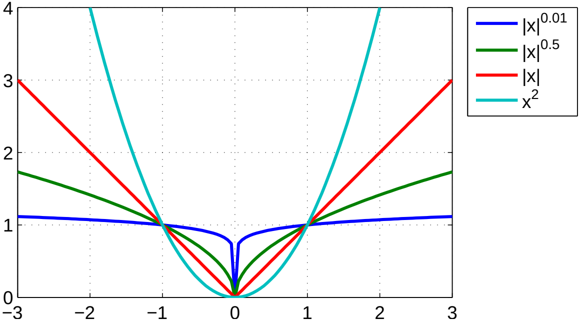


# L1-regularization and sparsity

- Best convex approximation of $\|\boldsymbol{w}\|_0$.

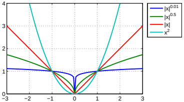
# L1-regularization and sparsity

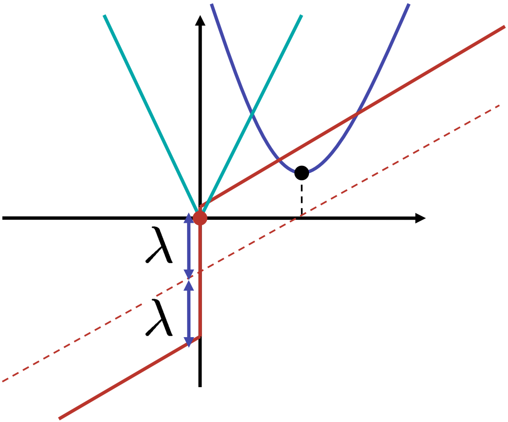
- Best convex approximation of $\|\boldsymbol{w}\|_0$.



# L1-regularization and sparsity

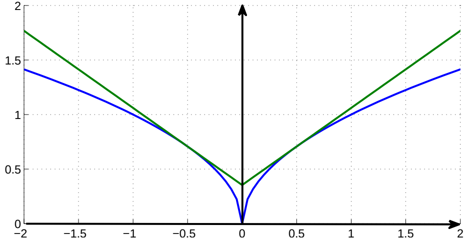
- Best convex approximation of $\|\boldsymbol{w}\|_0$.
- Threshold occurs for finite $\lambda$.

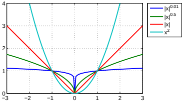


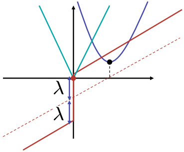



# L1-regularization and sparsity

- Best convex approximation of $\|\boldsymbol{w}\|_0$.
- Threshold occurs for finite $\lambda$.
- Non-convex cases ($p < 1$) can be solved by re-weighted L1 minimization



# Ex. 2: Compressed sensing [Candes, Romberg, & Tao 06]

Signal (MRI image) recovery from (noisy) low-dimensional measurements.

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \qquad \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Omega}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{\Phi}\boldsymbol{w}\|_1$$

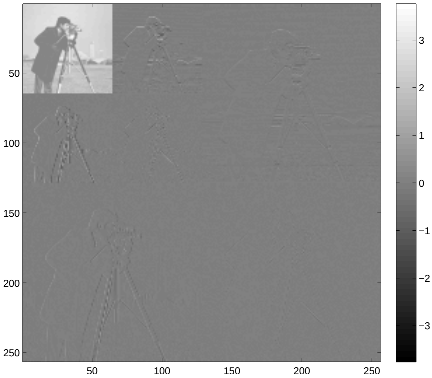
- $\boldsymbol{y}$: Noisy signal
- $\boldsymbol{w}$: Original signal
- $\boldsymbol{\Omega}$: $\mathbb{R}^n \to \mathbb{R}^m$: Observation matrix (random, fourier transform)
- $\boldsymbol{\Phi}$: Trnasformation s.t. the original signal is sparse

NB: If $\boldsymbol{\Phi}^{-1}$ exists, we can solve instead

$$\underset{\tilde{\boldsymbol{w}}\in\mathbb{R}^n}{\text{minimize}} \qquad \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{w}}\|_2^2 + \lambda\|\tilde{\boldsymbol{w}}\|_1,$$

where $\boldsymbol{A} = \boldsymbol{\Omega}\boldsymbol{\Phi}^{-1}$.


50
100
150
200
250
50
100
150
200
250

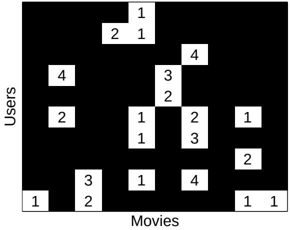

50
100
150
200
250
50
100
150
200
250
3
2
1
0
−1
−2
−3

# Ex. 3: Estimation of a low-rank matrix [Fazel+ 01; Srebro+ 05]

Goal: Recover a low-rank matrix $\boldsymbol{X}$ from partial (noisy) measurement $\boldsymbol{Y}$

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad \frac{1}{2}\|\Omega(\boldsymbol{X} - \boldsymbol{Y})\|^2 + \lambda\|\boldsymbol{X}\|_{S_1}$$

$$\text{where} \quad \|\boldsymbol{X}\|_{S_1} := \sum_{j=1}^{r} \sigma_j(\boldsymbol{X}) \quad \text{(Schatten 1-norm)}$$

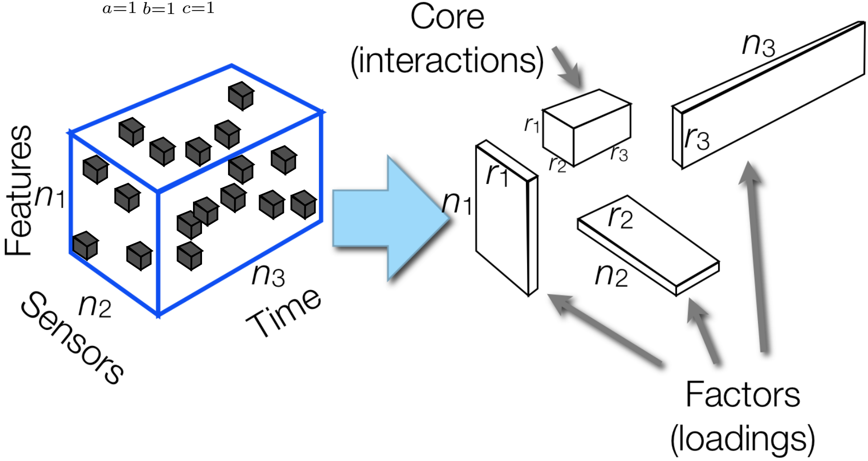



Aka trace norm, nuclear norm
$\Rightarrow$ Linear sum of singular values
$\Rightarrow$ Sparsity in the SV spectrum
$\Rightarrow$ Low-rank

# Ex. 4: Low-rank tensor completion [Tomioka+11]

$$X_{ijk} = \sum_{a=1}^{r_1}\sum_{b=1}^{r_2}\sum_{c=1}^{r_3} C_{abc} U^{(1)}_{ia} U^{(2)}_{jb} U^{(c)}_{kc}$$

Tucker decomposition



# Simple vs. structured sparse estimation problems

- Simple sparse estimation problem

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad L(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1$$

  - SNP analysis
  - Compressed sensing with $\boldsymbol{\Phi}^{-1}$ (e.g., wavelet)
  - Collaborative filtering (matrix completion)

- Structured sparse estimation problem

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad L(\boldsymbol{w}) + \lambda \|\boldsymbol{\Phi}\boldsymbol{w}\|_1$$

  - Compressed sensing without $\boldsymbol{\Phi}^{-1}$ (e.g., total variation)
  - Low-rank tensor completion

# Common criticisms

- Convex optimization is another developed field (and it is boring). We can just use it as a black box.
  - Yes, but we can do much better by *knowing the structure of our problems*.

# Common criticisms

- Convex optimization is another developed field (and it is boring). We can just use it as a black box.
  - Yes, but we can do much better by *knowing the structure of our problems*.
- Convexity is too restrictive.
  - Convexity depends on parametrization. A seemingly non-convex problem could be *reformulated into a convex problem*.

# Common criticisms

- Convex optimization is another developed field (and it is boring). We can just use it as a black box.
  - Yes, but we can do much better by *knowing the structure of our problems*.
- Convexity is too restrictive.
  - Convexity depends on parametrization. A seemingly non-convex problem could be *reformulated into a convex problem*.
- I am only interested in making things work.
  - Yes, convex optimization works. But it can also be used for *analyzing how algorithms perform at the end*.

# Bayesian inference as a convex optimization

$$\underset{q}{\text{minimize}} \quad \underbrace{\mathbb{E}_q[f(w)]}_{\text{average energy}} + \underbrace{\mathbb{E}_q[\log q(w)]}_{\text{entropy}}$$

$$\text{s.t.} \quad q(w) \geq 0, \quad \int q(w) \mathrm{d}w = 1$$

where

$$f(w) = \underbrace{-\log P(D|w)}_{\text{neg. log likelihood}} \underbrace{-\log P(w)}_{\text{neg. log prior}}$$

# Bayesian inference as a convex optimization

$$\underset{q}{\text{minimize}} \quad \underbrace{\mathbb{E}_q[f(w)]}_{\text{average energy}} + \underbrace{\mathbb{E}_q[\log q(w)]}_{\text{entropy}}$$

$$\text{s.t.} \quad q(w) \geq 0, \quad \int q(w)\mathrm{d}w = 1$$

where

$$f(w) = \underbrace{-\log P(D|w)}_{\text{neg. log likelihood}} \underbrace{-\log P(w)}_{\text{neg. log prior}}$$

$$\Rightarrow \quad q(w) = \frac{1}{Z}e^{-f(w)} \quad \text{(Bayesian posterior)}$$

# Bayesian inference as a convex optimization

$$\underset{q}{\text{minimize}} \quad \underbrace{\mathbb{E}_q[f(w)]}_{\text{average energy}} + \underbrace{\mathbb{E}_q[\log q(w)]}_{\text{entropy}}$$

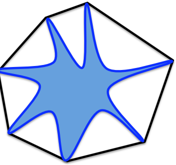
$$\text{s.t.} \quad q(w) \geq 0, \quad \int q(w) \mathrm{d}w = 1$$

where

$$f(w) = \underbrace{-\log P(D|w)}_{\text{neg. log likelihood}} \underbrace{-\log P(w)}_{\text{neg. log prior}}$$

$$\Rightarrow \quad q(w) = \frac{1}{Z} e^{-f(w)} \quad \text{(Bayesian posterior)}$$

## Inner approximations



- Variational Bayes
- Empirical Bayes

# Bayesian inference as a convex optimization

$$\underset{q}{\text{minimize}} \quad \underbrace{\mathbb{E}_q[f(w)]}_{\text{average energy}} + \underbrace{\mathbb{E}_q[\log q(w)]}_{\text{entropy}}$$

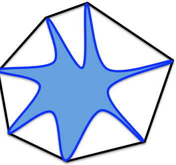
$$\text{s.t.} \quad q(w) \geq 0, \quad \int q(w)\mathrm{d}w = 1$$

where

$$f(w) = \underbrace{-\log P(D|w)}_{\text{neg. log likelihood}} \underbrace{-\log P(w)}_{\text{neg. log prior}}$$

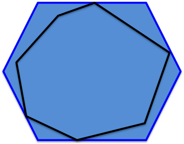
$$\Rightarrow \quad q(w) = \frac{1}{Z}e^{-f(w)} \quad \text{(Bayesian posterior)}$$

Inner approximations



- Variational Bayes
- Empirical Bayes

Outer approximations



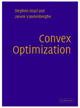
- Belief propagation

See Wainwright & Jordan 08.

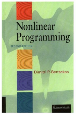
# Overview

- Convex optimization basics
  - Convex sets
  - Convex function
  - Conditions that guarantee convexity
  - Convex optimization problem
- Looking into more structures
  - Proximity operators
  - Conjugate duality and dual ascent
  - Augmented Lagrangian and ADMM

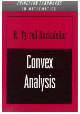
References:




Boyd & Vandenberghe. (2004) Convex optimization.




Bertsekas (1999) Nonlinear Programming.




Rockafellar (1970) Convex Analysis.

Moreau (1965) Proximité et dualité dans un espace Hilbertien.

# Convexity

Learning objectives

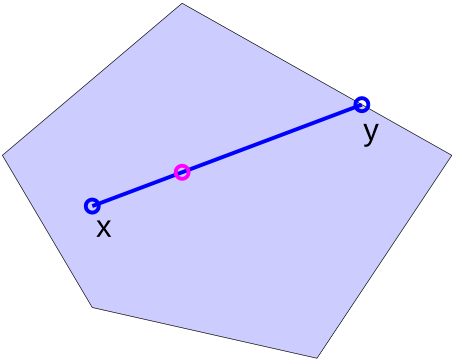
- Convex sets
- Convex function
- Conditions that guarantee convexity
- Convex optimization problem

# Convex set

A subset $V \subseteq \mathbb{R}^n$ is a convex set

$\Leftrightarrow$ line segment between two arbitrary points $\boldsymbol{x}, \boldsymbol{y} \in V$ is included in $V$; that is,

$$\forall \boldsymbol{x}, \boldsymbol{y} \in V, \forall \lambda \in [0, 1], \quad \lambda \boldsymbol{x} + (1 - \lambda) \boldsymbol{y} \in V.$$



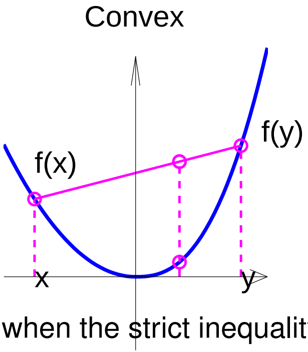
# Convex function

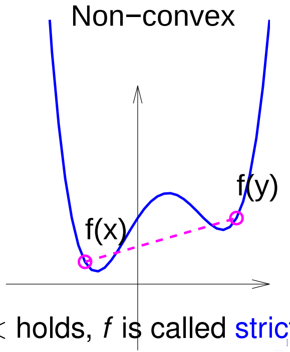
A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a convex function

$\Leftrightarrow$ the function $f$ is below any line segment between two points on $f$; that is,

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \forall \lambda \in [0, 1], \quad f((1-\lambda)\boldsymbol{x} + \lambda \boldsymbol{y}) \leq (1-\lambda) f(\boldsymbol{x}) + \lambda f(\boldsymbol{y})$$

(Jensen's inequality)




Johan Jensen

1859 – 1925

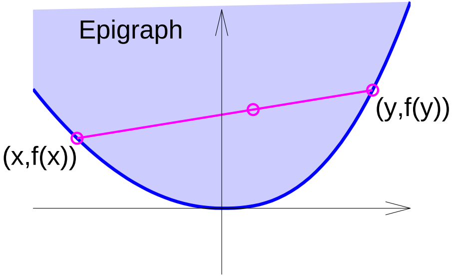
NB: when the strict inequality $<$ holds, $f$ is called strictly convex.

# Convex function

A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a convex function

$\Leftrightarrow$ the epigraph of $f$ is a convex set; that is

$$V_f := \{(t, \boldsymbol{x}) : (t, \boldsymbol{x}) \in \mathbb{R}^{n+1}, t \geq f(\boldsymbol{x})\} \text{ is convex.}$$

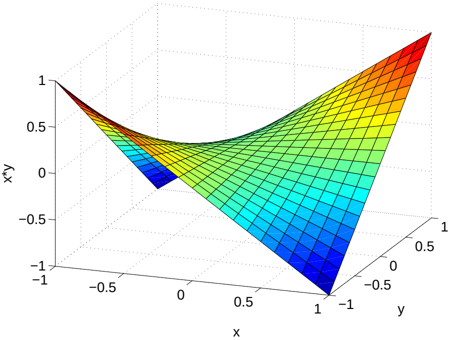


# Jointly convex

- A function $f(x, y)$ can be convex wrt $x$ ($y$) for any fixed $y$ ($x$), respectively, zbut can fail to be convex for $x$ and $y$ simultaneously.

# Jointly convex

- A function $f(x, y)$ can be convex wrt $x$ ($y$) for any fixed $y$ ($x$), respectively, zbut can fail to be convex for $x$ and $y$ simultaneously.




$$f(x, y) \text{ is convex} \quad \Rightarrow (\not\Leftarrow) \quad f(x, y) \text{ is convex for } x \text{ and } y \text{ individually}$$

- To be more explicit, we sometimes say jointly convex.

# Why do we allow infinity?

- $f(x) = 1/x$ is convex for $x > 0$.

$$f(x) = \begin{cases} 1/x & \text{if } x > 0, \\ +\infty & \text{otherwise.} \end{cases}$$

and we can forget about the domain.

# Why do we allow infinity?

- $f(x) = 1/x$ is convex for $x > 0$.

$$f(x) = \begin{cases} 1/x & \text{if } x > 0, \\ +\infty & \text{otherwise.} \end{cases}$$

  and we can forget about the domain.
- The indicator function $\delta_C(\boldsymbol{x})$ of a set $C$:

$$\delta_C(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

  Is this a convex function? (consider the epigraph)

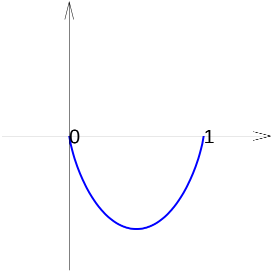
## Condition #1: Hessian

Hessian $\nabla^2 f(\boldsymbol{x})$ is positive semidefinite (if $f$ is differentiable)

### Examples

- (Negative) entropy is a convex function.

$$f(p) = \sum_{i=1}^{n} p_i \log p_i,$$



# Condition #1: Hessian

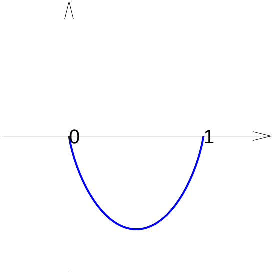
Hessian $\nabla^2 f(\boldsymbol{x})$ is positive semidefinite (if $f$ is differentiable)

## Examples

- (Negative) entropy is a convex function.

$$f(p) = \sum_{i=1}^{n} p_i \log p_i,$$

$$\nabla^2 f(p) = \mathrm{diag}(1/p_1, \ldots, 1/p_n) \succeq 0.$$



## Condition #1: Hessian

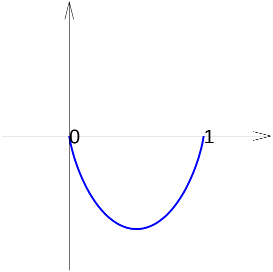
Hessian $\nabla^2 f(\boldsymbol{x})$ is positive semidefinite (if $f$ is differentiable)

### Examples

- (Negative) entropy is a convex function.

$$f(p) = \sum_{i=1}^{n} p_i \log p_i,$$

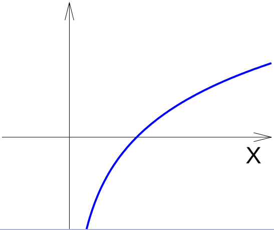
$$\nabla^2 f(p) = \mathrm{diag}(1/p_1, \ldots, 1/p_n) \succeq 0.$$




- log determinant is a *concave* ($-f$ is convex) function

$$f(\boldsymbol{X}) = \log|\boldsymbol{X}| \quad (\boldsymbol{X} \succeq 0),$$

$$\nabla^2 f(\boldsymbol{X}) = -\boldsymbol{X}^{-\top} \otimes \boldsymbol{X}^{-1} \preceq 0$$

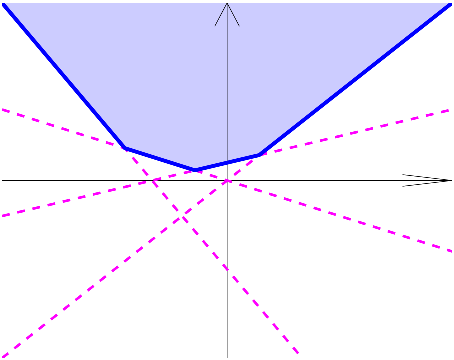


## Condition #2: Maximum over convex functions

Maximum over convex functions $\{f_j(x)\}_{j=1}^{\infty}$

$$f(\boldsymbol{x}) := \max_j f_j(\boldsymbol{x}) \qquad (f_j(\boldsymbol{x}) \text{ is convex for all } j)$$

is convex.



The same as saying "intersection of convex sets is a convex set"

## Condition #2: Maximum over convex functions

Maximum over convex functions $\{f(x;\boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \mathbb{R}^m\}$

$$f(\boldsymbol{x}) := \sup_{\boldsymbol{\alpha}\in\mathbb{R}^m} f(\boldsymbol{x};\boldsymbol{\alpha})$$

is convex.

### Example

- Quadratic over linear is a convex function

$$f(x,y) = \sup_{\alpha\in\mathbb{R}} \left(-\frac{\alpha^2}{2}x + \alpha y\right) \qquad (x > 0)$$

## Condition #2: Maximum over convex functions

Maximum over convex functions $\{f(x; \boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \mathbb{R}^m\}$

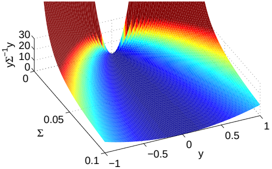
$$f(\boldsymbol{x}) := \sup_{\boldsymbol{\alpha} \in \mathbb{R}^m} f(\boldsymbol{x}; \boldsymbol{\alpha})$$

is convex.

### Example

- Quadratic over linear is a convex function

$$f(x, y) = \sup_{\alpha \in \mathbb{R}} \left( -\frac{\alpha^2}{2} x + \alpha y \right) \qquad (x > 0)$$
$$= \frac{y^2}{2x}$$



## Condition #2: Maximum over convex functions

Maximum over convex functions $\{f(x;\boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \mathbb{R}^m\}$

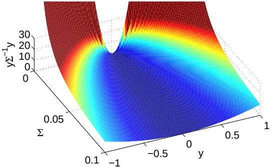
$$f(\boldsymbol{x}) := \sup_{\boldsymbol{\alpha}\in\mathbb{R}^m} f(\boldsymbol{x};\boldsymbol{\alpha})$$

is convex.

### Example

- Quadratic over linear is a convex function

$$\begin{aligned} f(x,y) &= \sup_{\alpha\in\mathbb{R}} \left(-\frac{\alpha^2}{2}x + \alpha y\right) \qquad (x>0) \\ &= \frac{y^2}{2x} \end{aligned}$$




- Similarly

$$f(\boldsymbol{\Sigma},\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{y} \qquad (\boldsymbol{\Sigma}\succ 0) \qquad \text{is a convex function (show it!)}$$

## Condition #3: Partial minimum

Partial minimum of a convex function $f(x, y)$

$$f(x) := \min_{y \in \mathbb{R}^n} f(x, y) \qquad \text{is convex.}$$

### Examples

- Hierarchical prior minimization

$$f(\boldsymbol{x}) = \min_{d_1, \ldots, d_n \geq 0} \frac{1}{2} \sum_{j=1}^{n} \left( \frac{x_j^2}{d_j} + \frac{d_j^p}{p} \right) \quad (p \geq 1)$$

## Condition #3: Partial minimum

Partial minimum of a convex function $f(x, y)$

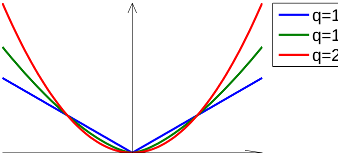
$$f(x) := \min_{y \in \mathbb{R}^n} f(x, y) \qquad \text{is convex.}$$

### Examples

- Hierarchical prior minimization

$$f(\boldsymbol{x}) = \min_{d_1,\dots,d_n \geq 0} \frac{1}{2} \sum_{j=1}^{n} \left( \frac{x_j^2}{d_j} + \frac{d_j^p}{p} \right) \quad (p \geq 1)$$

$$= \frac{1}{q} \sum_{j=1}^{n} |x_j|^q \quad (q = \frac{2p}{1+p})$$



## Condition #3: Partial minimum

Partial minimum of a convex function $f(x, y)$

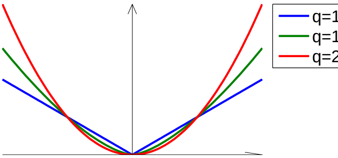
$$f(x) := \min_{y \in \mathbb{R}^n} f(x, y) \qquad \text{is convex.}$$

### Examples

- Hierarchical prior minimization

$$f(\boldsymbol{x}) = \min_{d_1,\ldots,d_n \geq 0} \frac{1}{2} \sum_{j=1}^{n} \left( \frac{x_j^2}{d_j} + \frac{d_j^p}{p} \right) \quad (p \geq 1)$$

$$= \frac{1}{q} \sum_{j=1}^{n} |x_j|^q \quad (q = \frac{2p}{1+p})$$




- Schatten 1- norm (sum of singularvalues)

$$f(\boldsymbol{X}) = \min_{\boldsymbol{\Sigma} \succeq 0} \frac{1}{2} \left( \mathrm{Tr}\left( \boldsymbol{X} \boldsymbol{\Sigma}^{-1} \boldsymbol{X}^\top \right) + \mathrm{Tr}\left( \boldsymbol{\Sigma} \right) \right)$$

## Condition #3: Partial minimum

Partial minimum of a convex function $f(x, y)$

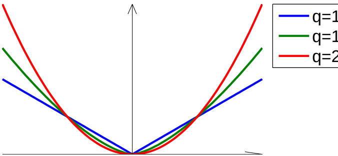
$$f(x) := \min_{y \in \mathbb{R}^n} f(x, y) \qquad \text{is convex.}$$

### Examples

- Hierarchical prior minimization

$$f(\boldsymbol{x}) = \min_{d_1,\ldots,d_n \geq 0} \frac{1}{2} \sum_{j=1}^{n} \left( \frac{x_j^2}{d_j} + \frac{d_j^p}{p} \right) \quad (p \geq 1)$$

$$= \frac{1}{q} \sum_{j=1}^{n} |x_j|^q \quad (q = \frac{2p}{1+p})$$




- Schatten 1- norm (sum of singularvalues)

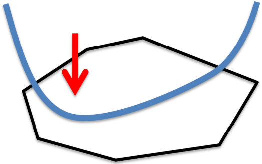
$$f(\boldsymbol{X}) = \min_{\boldsymbol{\Sigma} \succeq 0} \frac{1}{2} \left( \mathrm{Tr}\left( \boldsymbol{X} \boldsymbol{\Sigma}^{-1} \boldsymbol{X}^\top \right) + \mathrm{Tr}\left( \boldsymbol{\Sigma} \right) \right) = \mathrm{Tr}\left( (\boldsymbol{X}^\top \boldsymbol{X})^{1/2} \right) = \sum_{j=1}^{r} \sigma_j(\boldsymbol{X}).$$

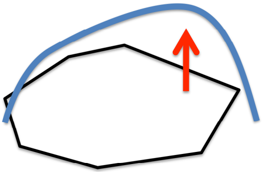
# Convex optimization problem

$f$: convex function, $g$: concave function ($-g$ is convex), $C$: convex set.

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}), \qquad \text{s.t.} \quad \boldsymbol{x} \in C.$$

$$\underset{\boldsymbol{y}}{\text{maximize}} \quad g(\boldsymbol{y}), \qquad \text{s.t.} \quad \boldsymbol{y} \in C.$$





Why?

- local optimum $\Rightarrow$ global optimum
- duality (later) can be used to check convergence

$\Rightarrow$ We can be *sure* that we are doing the right thing!

# Coming up next:

- Gradient descent:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta_t \nabla f(\boldsymbol{w}^t)$$

- What do we do if we have
  - Constraints
  - Non-differentiable terms, like $\|\boldsymbol{w}\|_1$

$\Rightarrow$ projection/proximity operator

# Proximity operators and iterative shrinkage/thresholding methods

Learning objectives

- (Projected) gradient method
- Iterative shrinkage/thresholding (IST) method
- Acceleration

# Proximity view on gradient descent

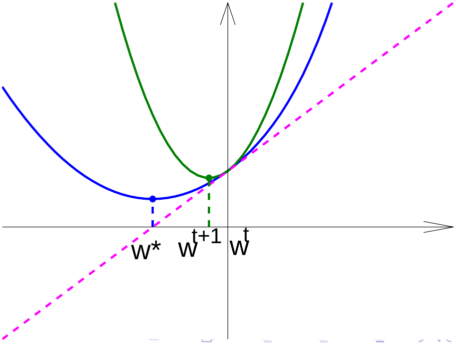
"Linearize and Prox"

$$\begin{aligned}\boldsymbol{w}^{t+1} &= \underset{\boldsymbol{w}}{\mathrm{argmin}}\left(\nabla f(\boldsymbol{w}^t)(\boldsymbol{w}-\boldsymbol{w}^t) + \frac{1}{2\eta_t}\|\boldsymbol{w}-\boldsymbol{w}^t\|^2\right)\\ &= \boldsymbol{w}^t - \eta_t \nabla f(\boldsymbol{w}^t)\end{aligned}$$

- Step-size should satisfy $\eta_t \leq 1/L(f)$.
- $L(f)$: the Lipschitz constant

$$\|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\| \leq L(f)\|\boldsymbol{y}-\boldsymbol{x}\|.$$

- $L(f)$=upper bound on the maximum eigenvalue of the Hessian



# Constraint minimization problem

- What do we do, if we have a constraint?

$$\begin{aligned} \underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\boldsymbol{w}), \\ \text{s.t.} \quad & \boldsymbol{w} \in C. \end{aligned}$$

# Constraint minimization problem

- What do we do, if we have a constraint?

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{w}), \qquad \text{s.t.} \quad \boldsymbol{w} \in C.$$

- can be equivalently written as

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{w}) + \delta_C(\boldsymbol{w}),$$

where $\delta_C(\boldsymbol{w})$ is the indicator function of the set $C$.

# Projected gradient method (Bertsekas 99; Nesterov 03)

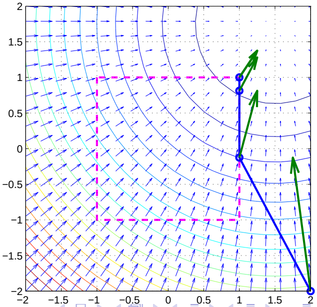
Linearize the objective $f$, $\delta_C$ is the indicator of the constraint $C$

$$\begin{aligned}
\boldsymbol{w}^{t+1} &= \operatorname*{argmin}_{\boldsymbol{w}} \left( \nabla f(\boldsymbol{w}^t)(\boldsymbol{w} - \boldsymbol{w}^t) + \delta_C(\boldsymbol{w}) + \frac{1}{2\eta_t}\|\boldsymbol{w} - \boldsymbol{w}^t\|_2^2 \right) \\
&= \operatorname*{argmin}_{\boldsymbol{w}} \left( \delta_C(\boldsymbol{w}) + \frac{1}{2\eta_t}\|\boldsymbol{w} - (\boldsymbol{w}^t - \eta_t \nabla f(\boldsymbol{w}^t))\|_2^2 \right) \\
&= \mathrm{proj}_C(\boldsymbol{w}^t - \eta_t \nabla f(\boldsymbol{w}^t)).
\end{aligned}$$

- Requires $\eta_t \leq 1/L(f)$.
- Convergence rate

$$f(\boldsymbol{w}^k) - f(\boldsymbol{w}^*) \leq \frac{L(f)\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_2^2}{2k}$$

- Need the projection $\mathrm{proj}_C$ to be easy to compute



# Ideas for regularized minimization

## Constrained minimization problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{w}) + \delta_C(\boldsymbol{w}).$$

$\Rightarrow$ need to compute the projection

$$\boldsymbol{w}^{t+1} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left( \delta_C(\boldsymbol{w}) + \frac{1}{2\eta_t}\|\boldsymbol{w} - \boldsymbol{y}\|_2^2 \right)$$

# Ideas for regularized minimization

## Constrained minimization problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{w}) + \delta_C(\boldsymbol{w}).$$

$\Rightarrow$ need to compute the projection

$$\boldsymbol{w}^{t+1} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left( \delta_C(\boldsymbol{w}) + \frac{1}{2\eta_t}\|\boldsymbol{w} - \boldsymbol{y}\|_2^2 \right)$$

## Regularized minimization problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_1$$

$\Rightarrow$ need to compute the proximity operator

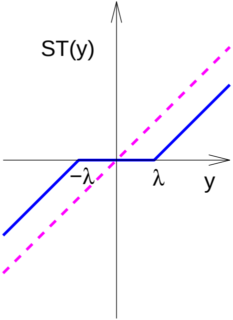
$$\boldsymbol{w}^{t+1} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left( \lambda\|\boldsymbol{w}\|_1 + \frac{1}{2\eta_t}\|\boldsymbol{w} - \boldsymbol{y}\|_2^2 \right)$$

# Proximal Operator: generalization of projection

$$\mathrm{prox}_g(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{w}} \left( g(\boldsymbol{w}) + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{y}\|_2^2 \right)$$

- $g = \delta_C$: Projection onto a convex set $\mathrm{proj}_C(\boldsymbol{y})$.
- $g(\boldsymbol{w}) = \lambda\|\boldsymbol{w}\|_1$: Soft-Threshold

$$\mathrm{prox}_\lambda(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{w}} \left( \lambda\|\boldsymbol{w}\|_1 + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{y}\|_2^2 \right)$$
$$= \begin{cases} y_j + \lambda & (y_j < -\lambda), \\ 0 & (-\lambda \le y_j \le \lambda), \\ y_j - \lambda & (y_j > \lambda). \end{cases}$$




- Prox can be computed easily for a separable $f$.
- Non-differentiability is OK.

# Exercise

Derive prox operator $\text{prox}_g$ for

- Ridge regularization

$$g(\boldsymbol{w}) = \lambda \sum_{j=1}^{n} w_j^2$$

- Group lasso regularization [Yuan & Lin 2006]

$$g(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n) = \lambda \sum_{j=1}^{n} \|\boldsymbol{w}_j\|_2$$

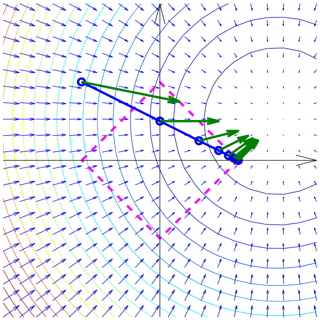
# Iterative Shrinkage Thresholding (IST)

$$\begin{aligned}\boldsymbol{w}^{t+1} &= \underset{\boldsymbol{w}}{\mathrm{argmin}}\left(\nabla f(\boldsymbol{w}^t)(\boldsymbol{w}-\boldsymbol{w}^t) + \lambda\|\boldsymbol{w}\|_1 + \frac{1}{2\eta_t}\|\boldsymbol{w}-\boldsymbol{w}^t\|_2^2\right)\\ &= \underset{\boldsymbol{w}}{\mathrm{argmin}}\left(\lambda\|\boldsymbol{w}\|_1 + \frac{1}{2\eta_t}\|\boldsymbol{w}-(\boldsymbol{w}^t-\eta_t\nabla f(\boldsymbol{w}^t))\|_2^2\right)\\ &= \mathrm{prox}_{\lambda\eta_t}(\boldsymbol{w}^t-\eta_t\nabla f(\boldsymbol{w}^t)).\end{aligned}$$

- The same condition for $\eta_t$, the same $O(1/k)$ convergence (Beck & Teboulle 09)

$$f(\boldsymbol{w}^k) - f(\boldsymbol{w}^*) \leq \frac{L(f)\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|^2}{2k}$$

- If the Prox operator $\mathrm{prox}_\lambda$ is easy, it is simple to implement.
- AKA Forward-Backward Splitting (Lions & Mercier 76)



# IST summary

Solve minimization problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_1$$

by iteratively computing

$$\boldsymbol{w}^{t+1} = \text{prox}_{\lambda\eta_t}(\boldsymbol{w}^t - \eta_t\nabla f(\boldsymbol{w}^t)),$$

where

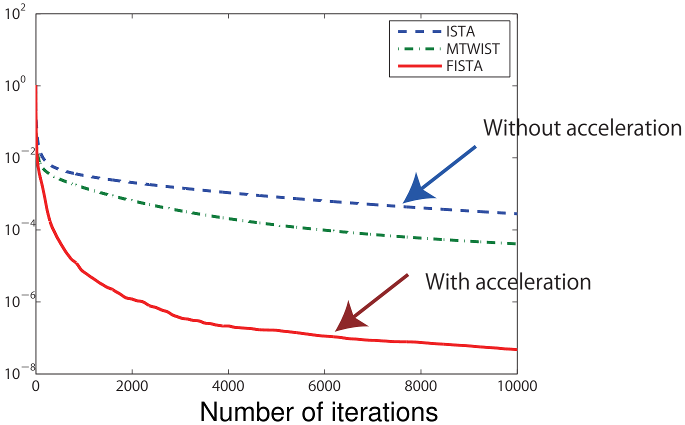
$$\text{prox}_{\lambda}(\boldsymbol{y}) = \underset{\boldsymbol{w}}{\text{argmin}} \left(\lambda\|\boldsymbol{w}\|_1 + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{y}\|_2^2\right).$$

# FISTA: accelerated version of IST (Beck & Teboulle 09; Nesterov 07)

1. Initialize $\boldsymbol{w}^0$ appropriately, $\boldsymbol{z}^1 = \boldsymbol{w}^0, \;\; s_1 = 1$.
2. Update $\boldsymbol{w}^t$:
$$\boldsymbol{w}^t = \mathrm{prox}_{\lambda\eta_t}(\boldsymbol{z}^t - \eta_t \nabla f(\boldsymbol{z}^t)).$$
3. Update $\boldsymbol{z}^t$:
$$\boldsymbol{z}^{t+1} = \boldsymbol{w}^t + \left(\frac{s_t - 1}{s_{t+1}}\right)(\boldsymbol{w}^t - \boldsymbol{w}^{t-1}),$$
where $s_{t+1} = (1 + \sqrt{1 + 4s_t^2})/2$.

- The same per iteration complexity. Converges as $O(1/k^2)$.
- Roughly speaking, $\boldsymbol{z}^t$ predicts where the IST step should be computed.

# Effect of acceleration




From Beck & Teboulle 2009 SIAM J. IMAGING SCIENCES

Vol. 2, No. 1, pp. 183-202

# MATLAB Exercise 1: implement an L1 regularized logistic regression via IST

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{\sum_{i=1}^{m}\log(1+\exp(-y_i\langle\boldsymbol{x}_i,\boldsymbol{w}\rangle))}_{\text{data-fit}} + \underbrace{\lambda\sum_{j=1}^{n}|w_j|}_{\text{Regularization}}$$

Hint: define

$$f_\ell(\boldsymbol{z}) = \sum_{i=1}^{m}\log(1+\exp(-z_i)).$$

Then the problem is

$$\text{minimize} \quad f_\ell(\boldsymbol{A}\boldsymbol{w}) + \lambda\sum_{j=1}^{n}|w_j| \quad \text{where} \quad \boldsymbol{A} = \begin{pmatrix} y_1\boldsymbol{x}_1{}^\top \\ y_2\boldsymbol{x}_2{}^\top \\ \vdots \\ y_m\boldsymbol{x}_m{}^\top \end{pmatrix}$$

# Some more hints

1. Compute the gradient of the loss term

$$\nabla_{\boldsymbol{w}} f_\ell(\boldsymbol{A}\boldsymbol{w}) = -\boldsymbol{A}^\top \left(\frac{\exp(-z_i)}{1+\exp(-z_i)}\right)_{i=1}^m \quad (\boldsymbol{z} = \boldsymbol{A}\boldsymbol{w})$$

2. The gradient step becomes

$$\boldsymbol{w}^{t+\frac{1}{2}} = \boldsymbol{w}^t + \eta_t \boldsymbol{A}^\top \left(\frac{\exp(-z_i)}{1+\exp(-z_i)}\right)_{i=1}^m$$

3. Then compute the proximity operator

$$\begin{aligned}\boldsymbol{w}^{t+1} &= \mathrm{prox}_{\lambda\eta_t}(\boldsymbol{w}^{t+\frac{1}{2}}) \\ &= \begin{cases} w_j^{t+\frac{1}{2}} + \lambda\eta_t & (w_j^{t+\frac{1}{2}} < -\lambda\eta_t), \\ 0 & (-\lambda\eta_t \le w_j^{t+\frac{1}{2}} \le \lambda\eta_t), \\ w_j^{t+\frac{1}{2}} - \lambda\eta_t & (w_j^{t+\frac{1}{2}} > \lambda\eta_t). \end{cases}\end{aligned}$$

# Matrix completion via IST (Mazumder et al. 10)

**Loss function:**

$$f(\boldsymbol{X}) = \frac{1}{2}\|\Omega(\boldsymbol{X} - \boldsymbol{Y})\|^2.$$

**Regularization:**

$$g(\boldsymbol{X}) = \lambda \sum_{j=1}^{r} \sigma_j(\boldsymbol{X}) \quad (S_1\text{-norm}).$$

# Matrix completion via IST (Mazumder et al. 10)

Loss function:

$$f(\boldsymbol{X}) = \frac{1}{2}\|\Omega(\boldsymbol{X} - \boldsymbol{Y})\|^2.$$

gradient:

$$\nabla f(\boldsymbol{X}) = \Omega^\top(\Omega(\boldsymbol{X} - \boldsymbol{Y}))$$

Regularization:

$$g(\boldsymbol{X}) = \lambda \sum_{j=1}^{r} \sigma_j(\boldsymbol{X}) \quad (S_1\text{-norm}).$$

# Matrix completion via IST (Mazumder et al. 10)

Loss function:

$$f(\boldsymbol{X}) = \frac{1}{2}\|\Omega(\boldsymbol{X} - \boldsymbol{Y})\|^2.$$

gradient:

$$\nabla f(\boldsymbol{X}) = \Omega^\top(\Omega(\boldsymbol{X} - \boldsymbol{Y}))$$

Regularization:

$$g(\boldsymbol{X}) = \lambda \sum_{j=1}^{r} \sigma_j(\boldsymbol{X}) \quad (S_1\text{-norm}).$$

Prox operator (Singular Value Thresholding):

$$\mathrm{prox}_\lambda(\boldsymbol{Z}) = \boldsymbol{U} \max(\boldsymbol{S} - \lambda \boldsymbol{I}, 0) \boldsymbol{V}^\top.$$

# Matrix completion via IST (Mazumder et al. 10)

Loss function:

$$f(\boldsymbol{X}) = \frac{1}{2}\|\Omega(\boldsymbol{X} - \boldsymbol{Y})\|^2.$$

gradient:

$$\nabla f(\boldsymbol{X}) = \Omega^\top(\Omega(\boldsymbol{X} - \boldsymbol{Y}))$$

Regularization:

$$g(\boldsymbol{X}) = \lambda \sum_{j=1}^{r} \sigma_j(\boldsymbol{X}) \quad (S_1\text{-norm}).$$

Prox operator (Singular Value Thresholding):

$$\mathrm{prox}_\lambda(\boldsymbol{Z}) = \boldsymbol{U}\max(\boldsymbol{S} - \lambda \boldsymbol{I}, 0)\boldsymbol{V}^\top.$$

Iteration:

$$\boldsymbol{X}^{t+1} = \mathrm{prox}_{\lambda\eta_t}\Big(\underbrace{(\boldsymbol{I} - \eta_t\Omega^\top\Omega)(\boldsymbol{X}^t)}_{\text{fill in missing}} + \underbrace{\eta_t\Omega^\top\Omega(\boldsymbol{Y}^t)}_{\text{observed}}\Big)$$

- When $\eta_t = 1$, fill missings with predicted values $\boldsymbol{X}^t$, overwrite the observed with observed values, then soft-threshold.

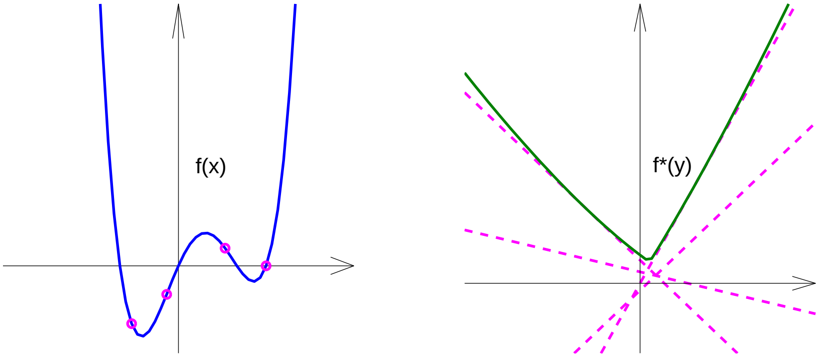
# Conjugate duality and dual ascent

- Convex conjugate function
- Lagrangian relaxation and dual problem

# Conjugate duality

The convex conjugate $f^*$ of a function $f$:

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \mathbb{R}^n} \left( \langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x}) \right)$$




Since the maximum over linear functions is always convex, $f$ need not be convex.

# Demo

Try

- `demo_conjugate(@(x)x.^2/2,-5:0.1:5);`
- `demo_conjugate(@(x)abs(x),-5:0.1:5);`
- ```
  demo_conjugate(@(x)x.*log(x)+(1-x).*log(1-x),...
                 0.001:0.001:0.999);
  ```
  ```

# Conjugate duality (dual view)

## Convex conjugate function

Every pair $(\boldsymbol{y}, f^*(\boldsymbol{y}))$ corresponds to a tangent line $\langle \boldsymbol{x}, \boldsymbol{y} \rangle - f^*(\boldsymbol{y})$ of the original function $f(\boldsymbol{x})$.

Because
$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}} (\langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x}))$
implies

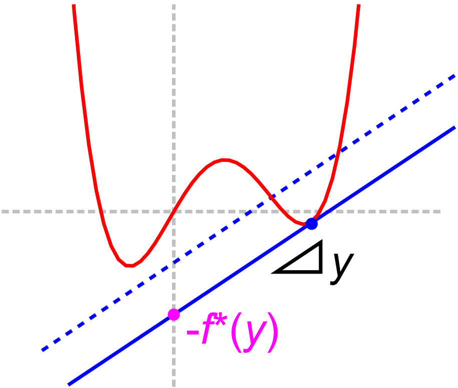
- If $t < f^*(\boldsymbol{y})$, there is a $\boldsymbol{x}$ s.t.

$$f(\boldsymbol{x}) < \langle \boldsymbol{x}, \boldsymbol{y} \rangle - t.$$

- If $t \geq f^*(\boldsymbol{y})$,

$$f(\boldsymbol{x}) \geq \langle \boldsymbol{x}, \boldsymbol{y} \rangle - t$$

  for every $\boldsymbol{x}$.



# Conjugate duality (dual view)

## Convex conjugate function

Every pair $(\boldsymbol{y}, f^*(\boldsymbol{y}))$ corresponds to a tangent line $\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f^*(\boldsymbol{y})$ of the original function $f(\boldsymbol{x})$.

Because
$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}} \left(\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x})\right)$
implies

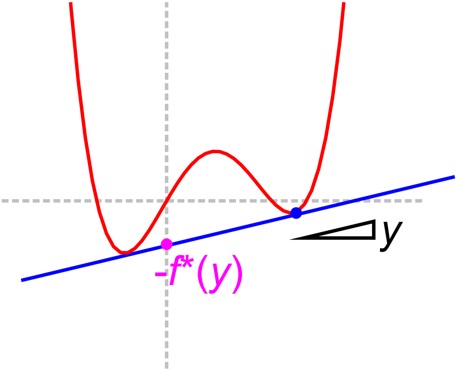
- If $t < f^*(\boldsymbol{y})$, there is a $\boldsymbol{x}$ s.t.

$$f(\boldsymbol{x}) < \langle \boldsymbol{x}, \boldsymbol{y}\rangle - t.$$

- If $t \geq f^*(\boldsymbol{y})$,

$$f(\boldsymbol{x}) \geq \langle \boldsymbol{x}, \boldsymbol{y}\rangle - t$$

  for every $\boldsymbol{x}$.



# Conjugate duality (dual view)

## Convex conjugate function

Every pair $(\boldsymbol{y}, f^*(\boldsymbol{y}))$ corresponds to a tangent line $\langle \boldsymbol{x}, \boldsymbol{y} \rangle - f^*(\boldsymbol{y})$ of the original function $f(\boldsymbol{x})$.

Because
$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}} \left( \langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x}) \right)$
implies

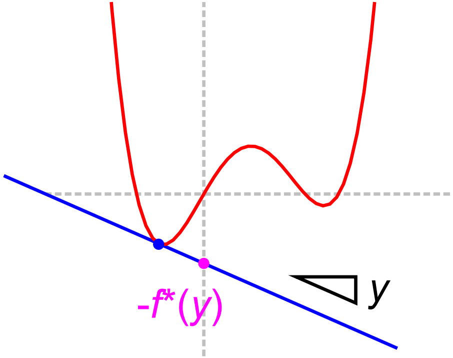
- If $t < f^*(\boldsymbol{y})$, there is a $\boldsymbol{x}$ s.t.

$$f(\boldsymbol{x}) < \langle \boldsymbol{x}, \boldsymbol{y} \rangle - t.$$

- If $t \geq f^*(\boldsymbol{y})$,

$$f(\boldsymbol{x}) \geq \langle \boldsymbol{x}, \boldsymbol{y} \rangle - t$$

for every $\boldsymbol{x}$.



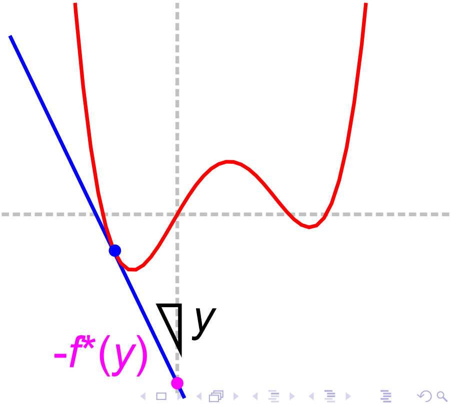
# Conjugate duality (dual view)

## Convex conjugate function

Every pair $(\boldsymbol{y}, f^*(\boldsymbol{y}))$ corresponds to a tangent line $\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f^*(\boldsymbol{y})$ of the original function $f(\boldsymbol{x})$.

Because
$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}} \left(\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x})\right)$
implies

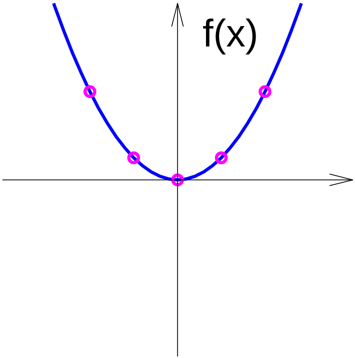
- If $t < f^*(\boldsymbol{y})$, there is a $\boldsymbol{x}$ s.t.
$$f(\boldsymbol{x}) < \langle \boldsymbol{x}, \boldsymbol{y}\rangle - t.$$
- If $t \geq f^*(\boldsymbol{y})$,
$$f(\boldsymbol{x}) \geq \langle \boldsymbol{x}, \boldsymbol{y}\rangle - t$$
for every $\boldsymbol{x}$.



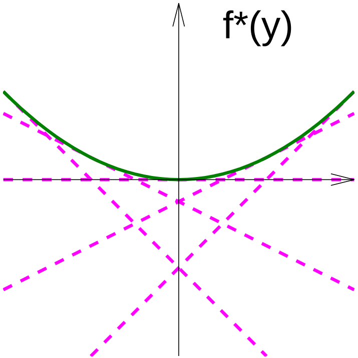
# Example of conjugate duality $f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n} (\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x}))$

- Quadratic function

$$f(x) = \frac{x^2}{2\sigma^2}$$




$$f^*(y) = \frac{\sigma^2 y^2}{2}$$



# Example of conjugate duality $f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \mathbb{R}^n} (\langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x}))$

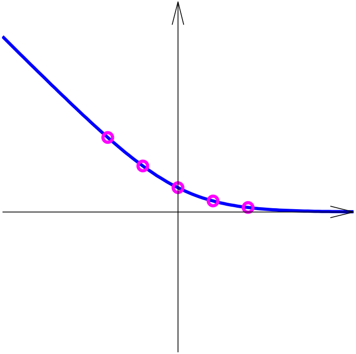
- Logistic loss function

$$f(x) = \log(1 + \exp(-x))$$

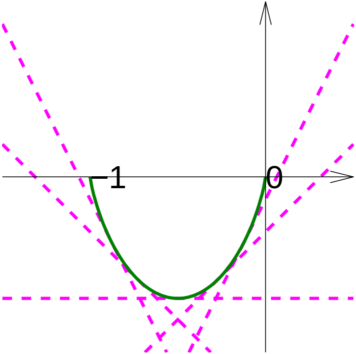
# Example of conjugate duality $f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n}(\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x}))$

- Logistic loss function

$f(x) = \log(1 + \exp(-x))$



$f^*(-y) = y\log(y) + (1-y)\log(1-y)$



# Example of conjugate duality $f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n} \left(\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x})\right)$

- L1 regularizer

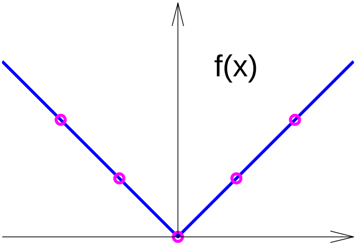
$$f(x) = |x|$$

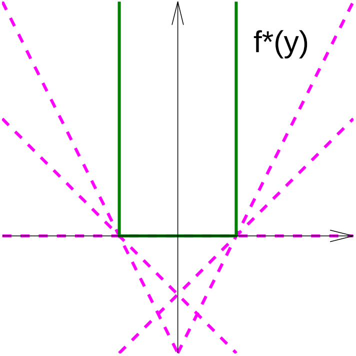
# Example of conjugate duality $f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n}\left(\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x})\right)$

- L1 regularizer

$$f(x) = |x|$$

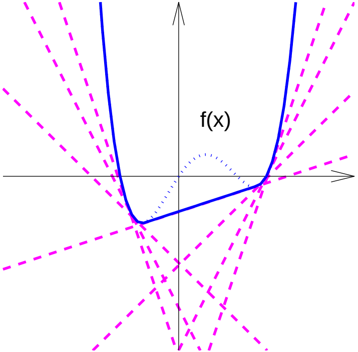
$$f^*(y) = \begin{cases} 0 & (-1 \leq y \leq 1) \\ +\infty & (\text{otherwise}) \end{cases}$$

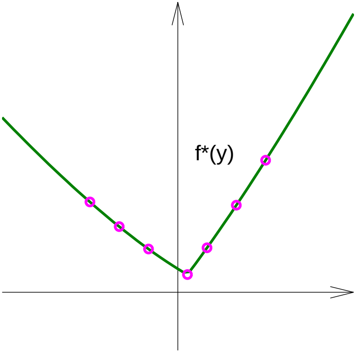


# Bi-conjugate $f^{**}$ may be different from $f$

For nonconvex $f$,



# Lagrangian relaxation

## Our optimization problem:

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{A}\boldsymbol{w}) + g(\boldsymbol{w})$$

$$\left(\begin{array}{l}\text{For example} \\ f(\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{z}-\boldsymbol{y}\|_2^2 \\ \text{(squared loss)}\end{array}\right)$$

# Lagrangian relaxation

Our optimization problem:

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{A}\boldsymbol{w}) + g(\boldsymbol{w})$$

$$\left( \begin{array}{l} \text{For example} \\ f(\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|_2^2 \\ \text{(squared loss)} \end{array} \right)$$

Equivalently written as

$$\underset{\boldsymbol{z}\in\mathbb{R}^m, \boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{z}) + g(\boldsymbol{w}),$$

$$\text{s.t.} \quad \boldsymbol{z} = \boldsymbol{A}\boldsymbol{w} \qquad \text{(equality constraint)}$$

# Lagrangian relaxation

Our optimization problem:

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{A}\boldsymbol{w}) + g(\boldsymbol{w})$$

$$\left(\begin{array}{l}\text{For example} \\ f(\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|_2^2 \\ \text{(squared loss)}\end{array}\right)$$

Equivalently written as

$$\underset{\boldsymbol{z}\in\mathbb{R}^m, \boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{z}) + g(\boldsymbol{w}),$$

$$\text{s.t.} \quad \boldsymbol{z} = \boldsymbol{A}\boldsymbol{w} \qquad \text{(equality constraint)}$$

Lagrangian relaxation

$$\underset{\boldsymbol{z},\boldsymbol{w}}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha) = f(\boldsymbol{z}) + g(\boldsymbol{w}) + \alpha^\top(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w})$$

# Lagrangian relaxation

Our optimization problem:

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{A}\boldsymbol{w}) + g(\boldsymbol{w})$$

$$\left( \begin{array}{l} \text{For example} \\ f(\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{z}-\boldsymbol{y}\|_2^2 \\ \text{(squared loss)} \end{array} \right)$$

Equivalently written as

$$\begin{aligned} \underset{\boldsymbol{z}\in\mathbb{R}^m, \boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\boldsymbol{z}) + g(\boldsymbol{w}), \\ \text{s.t.} \quad & \boldsymbol{z} = \boldsymbol{A}\boldsymbol{w} \qquad \text{(equality constraint)} \end{aligned}$$

Lagrangian relaxation

$$\underset{\boldsymbol{z},\boldsymbol{w}}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha) = f(\boldsymbol{z}) + g(\boldsymbol{w}) + \alpha^\top(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w})$$

- As long as $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{w}$, the relaxation is exact.
- $\sup_\alpha \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha)$ recovers the original problem.
- Minimum of $\mathcal{L}$ is no greater than the minimum of the original.

# Weak duality

$$\inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha) \leq \inf_{\boldsymbol{w}} \left( f(\boldsymbol{A}\boldsymbol{w}) + g(\boldsymbol{w}) \right) =: p^*$$

proof

$$\begin{aligned}
\inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha) &= \min\left( \inf_{\boldsymbol{z}=\boldsymbol{A}\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha), \ \inf_{\boldsymbol{z}\neq\boldsymbol{A}\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha) \right) \\
&= \min\left( p^*, \inf_{\boldsymbol{z}\neq\boldsymbol{A}\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha) \right) \\
&\leq p^*
\end{aligned}$$

# Dual problem

From the above argument

$$d(\boldsymbol{\alpha}) := \inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\alpha})$$

is a lower bound for $p^*$ for any $\boldsymbol{\alpha}$. Why don't we maximize over $\boldsymbol{\alpha}$?

# Dual problem

From the above argument

$$d(\boldsymbol{\alpha}) := \inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\alpha})$$

is a lower bound for $p^*$ for any $\boldsymbol{\alpha}$. Why don't we maximize over $\boldsymbol{\alpha}$?

Dual problem

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^m}{\text{maximize}} \quad d(\boldsymbol{\alpha})$$

Note

$$\sup_{\boldsymbol{\alpha}} \inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\alpha}) = d^* \leq p^* = \inf_{\boldsymbol{z},\boldsymbol{w}} \sup_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\alpha})$$

If $d^* = p^*$, strong duality holds. This is the case if $f$ and $g$ both closed and convex.

# Fenchel's duality

For convex[1] functions $f$ and $g$, and a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$

$$\sup_{\boldsymbol{\alpha} \in \mathbb{R}^m} \left( -f^*(-\boldsymbol{\alpha}) - g^*(\boldsymbol{A}^\top \boldsymbol{\alpha}) \right) = \inf_{\boldsymbol{w} \in \mathbb{R}^n} \left( f(\boldsymbol{A}\boldsymbol{w}) + g(\boldsymbol{w}) \right)$$

Werner Fenchel

1905 – 1988

- Only need conjugate functions $f^*$ and $g^*$ to compute the dual.
- We can make a list of them (like Laplace transform)

MATLAB Exercise 1.5:

- Compute the Fenchel dual of L1-logistic regression problem in Ex.1 and implement the stopping criterion: stop optimization if

$$(\mathrm{obj}_{\mathrm{prim}} - \mathrm{obj}_{\mathrm{dual}})/\mathrm{obj}_{\mathrm{prim}} < \epsilon \quad \text{(relative duality gap)}.$$

[1] More precisely, proper, closed, and convex.

# Derivation of Fenchel's duality theorem

$$
\begin{aligned}
d(\alpha) &= \inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha) \\
&= \inf_{\boldsymbol{z},\boldsymbol{w}} \left( f(\boldsymbol{z}) + g(\boldsymbol{w}) + \alpha^\top (\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}) \right) \\
&= \inf_{\boldsymbol{z}} \left( f(\boldsymbol{z}) + \langle \alpha, \boldsymbol{z} \rangle \right) + \inf_{\boldsymbol{w}} \left( g(\boldsymbol{w}) - \left\langle \boldsymbol{A}^\top \alpha, \boldsymbol{w} \right\rangle \right) \\
&= -\sup_{\boldsymbol{z}} \left( \langle -\alpha, \boldsymbol{z} \rangle - f(\boldsymbol{z}) \right) - \sup_{\boldsymbol{w}} \left( \left\langle \boldsymbol{A}^\top \alpha, \boldsymbol{w} \right\rangle - g(\boldsymbol{w}) \right) \\
&= -f^*(-\alpha) - g^*(\boldsymbol{A}^\top \alpha)
\end{aligned}
$$

# Augmented Lagrangian and ADMM

Learning objectives

- Structured sparse estimation
- Augmented Lagrangian
- Alternating direction method of multipliers (ADMM)

# Recap: Simple vs. structured sparse estimation problems

- Simple sparse estimation problem

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad f(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1$$

  - SNP analysis
  - Compressed sensing with $\boldsymbol{\Phi}^{-1}$ (e.g., wavelet)
  - Collaborative filtering (matrix completion)

- Structured sparse estimation problem

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad f(\boldsymbol{w}) + \lambda \|\boldsymbol{\Phi}\boldsymbol{w}\|_1$$

  - Compressed sensing without $\boldsymbol{\Phi}^{-1}$ (e.g., total variation)
  - Low-rank tensor completion

# Total Variation based image denoising [Rudin, Osher, Fatemi 92]

$$\underset{W}{\text{minimize}} \quad \frac{1}{2}\|W - M\|_F^2 + \lambda \sum_{i,j} \left\| \begin{pmatrix} \partial_x W_{ij} \\ \partial_y W_{ij} \end{pmatrix} \right\|_2$$

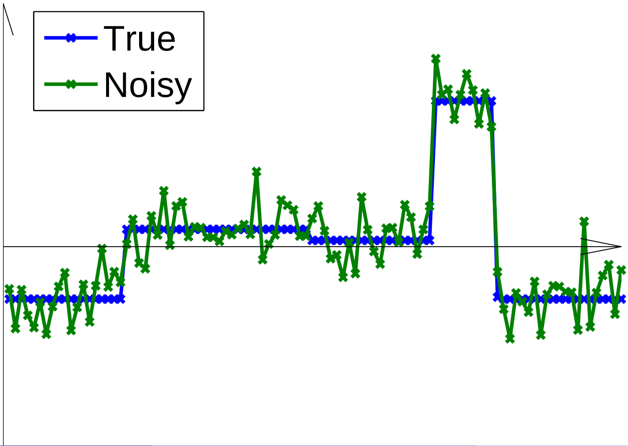
Original $W_0$

Observed $M$

# In one dimension

- Fused lasso [Tibshirani et al. 05]

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w}-\boldsymbol{y}\|_2^2 + \lambda\sum_{j=1}^{n-1}|w_{j+1}-w_j|$$



# Structured sparse estimation

- TV denoising

$$\underset{W}{\text{minimize}} \quad \frac{1}{2}\|W - M\|_F^2 + \lambda \sum_{i,j} \left\| \begin{pmatrix} \partial_x W_{ij} \\ \partial_y W_{ij} \end{pmatrix} \right\|_2$$

- Fused lasso

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \lambda \sum_{j=1}^{n-1} |w_{j+1} - w_j|$$

# Structured sparse estimation

- TV denoising

$$\underset{W}{\text{minimize}} \quad \frac{1}{2}\|W - M\|_F^2 + \lambda \sum_{i,j} \left\| \begin{pmatrix} \partial_x W_{ij} \\ \partial_y W_{ij} \end{pmatrix} \right\|_2$$

- Fused lasso

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \lambda \sum_{j=1}^{n-1} |w_{j+1} - w_j|$$

## Structured sparse estimation problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{f(\boldsymbol{w})}_{\text{data-fit}} + \underbrace{\lambda\|\boldsymbol{A}\boldsymbol{w}\|_1}_{\text{regularization}}$$

# Structured sparse estimation problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{f(\boldsymbol{w})}_{\text{data-fit}} + \underbrace{\lambda\|\boldsymbol{A}\boldsymbol{w}\|_1}_{\text{regularization}}$$

- Not easy to compute prox operator (because it is non-separable)
  $\Rightarrow$ difficult to apply IST-type methods.

# Structured sparse estimation problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{f(\boldsymbol{w})}_{\text{data-fit}} + \underbrace{\lambda\|\boldsymbol{A}\boldsymbol{w}\|_1}_{\text{regularization}}$$

- Not easy to compute prox operator (because it is non-separable)
  $\Rightarrow$ difficult to apply IST-type methods.

Can we use the Lagrangian relaxation trick?

# Forming the Lagrangian

## Structured sparsity problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{f(\boldsymbol{w})}_{\text{data-fit}} + \underbrace{\lambda\|\boldsymbol{A}\boldsymbol{w}\|_1}_{\text{regularization}}$$

## Equivalently written as

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{w}) + \underbrace{\lambda\|\boldsymbol{z}\|_1}_{\text{separable!}},$$

$$\text{s.t.} \quad \boldsymbol{z} = \boldsymbol{A}\boldsymbol{w} \qquad \text{(equality constraint)}$$

# Forming the Lagrangian

## Structured sparsity problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{f(\boldsymbol{w})}_{\text{data-fit}} + \underbrace{\lambda\|\boldsymbol{A}\boldsymbol{w}\|_1}_{\text{regularization}}$$

## Equivalently written as

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{w}) + \underbrace{\lambda\|\boldsymbol{z}\|_1}_{\text{separable!}},$$

$$\text{s.t.} \quad \boldsymbol{z} = \boldsymbol{A}\boldsymbol{w} \qquad \text{(equality constraint)}$$

### Lagrangian function

$$\mathcal{L}(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\alpha}) = f(\boldsymbol{w}) + \lambda\|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^\top(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}).$$

$\boldsymbol{\alpha}$: Lagrangian multiplier vector.

# Dual ascent

## Dual problem

$$\max_{\boldsymbol{\alpha}} \inf_{\boldsymbol{z},\boldsymbol{w}} \left( f(\boldsymbol{w}) + \lambda \|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^\top (\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}) \right)$$

We can compute the dual objective $d(\boldsymbol{\alpha})$ by separately minimizing

$$(1) \qquad \min_{\boldsymbol{w}} \left( f(\boldsymbol{w}) - \boldsymbol{\alpha}^\top \boldsymbol{A}\boldsymbol{w} \right)$$

$$(2) \qquad \min_{\boldsymbol{z}} \left( \lambda \|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^\top \boldsymbol{z} \right)$$

# Dual ascent

## Dual problem

$$\max_{\boldsymbol{\alpha}} \inf_{\boldsymbol{z},\boldsymbol{w}} \left( f(\boldsymbol{w}) + \lambda \|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^\top (\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}) \right)$$

We can compute the dual objective $d(\boldsymbol{\alpha})$ by separately minimizing

$$(1) \qquad \min_{\boldsymbol{w}} \left( f(\boldsymbol{w}) - \boldsymbol{\alpha}^\top \boldsymbol{A}\boldsymbol{w} \right) = -f^*(\boldsymbol{A}^\top \boldsymbol{\alpha}),$$

$$(2) \qquad \min_{\boldsymbol{z}} \left( \lambda \|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^\top \boldsymbol{z} \right) = -(\lambda \| \cdot \|_1)^*(-\boldsymbol{\alpha}).$$

# Dual ascent

## Dual problem

$$\max_{\boldsymbol{\alpha}} \inf_{\boldsymbol{z},\boldsymbol{w}} \left( f(\boldsymbol{w}) + \lambda \|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^\top (\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}) \right)$$

We can compute the dual objective $d(\boldsymbol{\alpha})$ by separately minimizing

$$(1) \qquad \min_{\boldsymbol{w}} \left( f(\boldsymbol{w}) - \boldsymbol{\alpha}^\top \boldsymbol{A}\boldsymbol{w} \right) = -f^*(\boldsymbol{A}^\top \boldsymbol{\alpha}),$$

$$(2) \qquad \min_{\boldsymbol{z}} \left( \lambda \|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^\top \boldsymbol{z} \right) = -(\lambda \| \cdot \|_1)^*(-\boldsymbol{\alpha}).$$

But also we get the gradient of $d(\boldsymbol{\alpha})$ (for free) as follows:

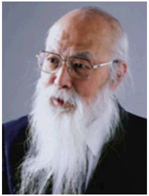
$$\nabla_{\boldsymbol{\alpha}} d(\boldsymbol{\alpha}) = \boldsymbol{z}^* - \boldsymbol{A}\boldsymbol{w}^*,$$

where $\boldsymbol{w}^*$: argmin of (1), $\boldsymbol{z}^*$: argmin of (2). See Chapter 6, Bertsekas 1999.

Gradient ascent (in the dual)!

# Dual ascent (Arrow, Hurwicz, & Uzawa 1958)

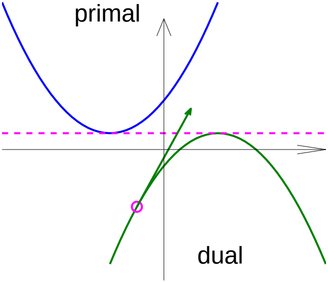
$$
\begin{cases}
\text{Minimize the Lagrangian wrt } \boldsymbol{x} \text{ and } \boldsymbol{z}\text{:} \\
\boldsymbol{w}^{t+1} = \mathrm{argmin}_{\boldsymbol{w}} \left( f(\boldsymbol{w}) - \boldsymbol{\alpha}^\top \boldsymbol{A}\boldsymbol{w} \right) . \\
\\
\boldsymbol{z}^{t+1} = \mathrm{argmin}_{\boldsymbol{z}} \left( \lambda \|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^\top \boldsymbol{z} \right) , \\
\text{Update the Lagrangian multiplier } \boldsymbol{\alpha}^t\text{:} \\
\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \eta_t (\boldsymbol{z}^{t+1} - \boldsymbol{A}\boldsymbol{w}^{t+1}).
\end{cases}
$$



H. Uzawa

- Pro: Very simple.
- Con: When $f^*$ or $g^*$ is non-differentiable, it is a dual subgradient method (convergence more tricky)

NB: $f^*$ is differentiable $\Leftrightarrow$ $f$ is strictly convex.



# Forming the *augmented* Lagrangian

## Structured sparsity problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{f(\boldsymbol{w})}_{\text{data-fit}} + \underbrace{\lambda\|\boldsymbol{A}\boldsymbol{w}\|_1}_{\text{regularization}}$$

Equivalently written as (for any $\eta > 0$)

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{w}) + \underbrace{\lambda\|\boldsymbol{z}\|_1}_{\text{separable!}} + \underbrace{\frac{\eta}{2}\|\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}\|_2^2}_{\text{penalty term}},$$

$$\text{s.t.} \quad \boldsymbol{z} = \boldsymbol{A}\boldsymbol{w} \qquad \text{(equality constraint)}$$

# Forming the *augmented* Lagrangian

## Structured sparsity problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{f(\boldsymbol{w})}_{\text{data-fit}} + \underbrace{\lambda\|\boldsymbol{A}\boldsymbol{w}\|_1}_{\text{regularization}}$$

Equivalently written as (for any $\eta > 0$)

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{w}) + \underbrace{\lambda\|\boldsymbol{z}\|_1}_{\text{separable!}} + \underbrace{\frac{\eta}{2}\|\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}\|_2^2}_{\text{penalty term}},$$

$$\text{s.t.} \quad \boldsymbol{z} = \boldsymbol{A}\boldsymbol{w} \qquad \text{(equality constraint)}$$

### Augmented Lagrangian function

$$\mathcal{L}_\eta(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\alpha}) = f(\boldsymbol{w}) + \lambda\|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^\top(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}) + \frac{\eta}{2}\|\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}\|_2^2$$

$\boldsymbol{\alpha}$: Lagrangian multiplier, $\eta$: penalty parameter

# Augmented Lagrangian Method

## Augmented Lagrangian function

$$\mathcal{L}_\eta(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\alpha}) = f(\boldsymbol{w}) + \lambda\|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^\top(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}) + \frac{\eta}{2}\|\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}\|^2.$$

## Augmented Lagrangian method (Hestenes 69, Powell 69)

$$\begin{cases} \text{Minimize the AL function wrt } \boldsymbol{w} \text{ and } \boldsymbol{z}\text{:} \\ (\boldsymbol{w}^{t+1}, \boldsymbol{z}^{t+1}) = \underset{\boldsymbol{w}\in\mathbb{R}^n, \boldsymbol{z}\in\mathbb{R}^m}{\operatorname{argmin}} \mathcal{L}_\eta(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\alpha}^t). \\ \\ \text{Update the Lagrangian multiplier:} \\ \boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \eta(\boldsymbol{z}^{t+1} - \boldsymbol{A}\boldsymbol{w}^{t+1}). \end{cases}$$

- Pro: The dual is always differentiable due to the penalty term.
- Con: Cannot minimize over $\boldsymbol{w}$ and $\boldsymbol{z}$ independently

# Alternating Direction Method of Multipliers (ADMM; Gabay & Mercier 76)

$$\begin{cases} \text{Minimize the AL function } \mathcal{L}_\eta(\boldsymbol{w}, \boldsymbol{z}^t, \boldsymbol{\alpha}^t) \text{ wrt } \boldsymbol{w}\text{:} \\ \boldsymbol{w}^{t+1} = \underset{\boldsymbol{w}\in\mathbb{R}^n}{\operatorname{argmin}} \left( f(\boldsymbol{w}) + \frac{\eta}{2}\|\boldsymbol{z}^t - \boldsymbol{A}\boldsymbol{w} + \boldsymbol{\alpha}^t/\eta\|_2^2 \right). \\ \\ \text{Minimize the AL function } \mathcal{L}_\eta(\boldsymbol{w}^{t+1}, \boldsymbol{z}, \boldsymbol{\alpha}^t) \text{ wrt } \boldsymbol{z}\text{:} \\ \boldsymbol{z}^{t+1} = \underset{\boldsymbol{z}\in\mathbb{R}^m}{\operatorname{argmin}} \left( \lambda\|\boldsymbol{z}\|_1 + \frac{\eta}{2}\|\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}^{t+1} + \boldsymbol{\alpha}^t/\eta\|_2^2 \right). \\ \\ \text{Update the Lagrangian multiplier:} \\ \boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \eta(\boldsymbol{z}^{t+1} - \boldsymbol{A}\boldsymbol{w}^{t+1}). \end{cases}$$

- Looks ad-hoc but convergence can be shown rigorously.
- Stability does not rely on the choice of step-size $\eta$.
- The newly updated $\boldsymbol{w}^{t+1}$ enters the computation of $\boldsymbol{z}^{t+1}$.

# MATLAB Exercise 2: implement an ADMM for fused lasso

## Fused lasso

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \lambda \sum_{j=1}^{n-1} \left| w_{j+1} - w_j \right|$$

- What is the loss function $f$?
- What is the matrix $\boldsymbol{A}$ for fused lasso?
- How does the $\boldsymbol{w}$-update step look?
- How does the $\boldsymbol{z}$-update step look?

# Conclusion

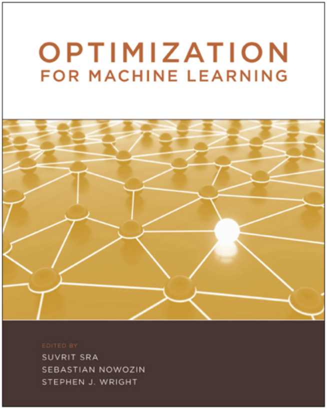
- Three approaches for various sparse estimation problems
  - Iterative shrinkage/thresholding – proximity operator
  - Uzawa's method – convex conjugate function
  - ADMM – combination of the above two
- Above methods go beyond black-box models (e.g., gradient descent or Newton's method) – takes better care of the problem structures.
- These methods are simple enough to be implemented rapidly, but should not be considered as a *silver bullet*.
  $\Rightarrow$ Trade-off between:
  - Quick implementation – test new ideas rapidly
  - Efficient optimization – more inspection/try-and-error/cross validation

# Topics we did not cover

- Beyond polynomial convergence $O(1/k^2)$
  - Dual Augmented Lagrangian (DAL) converges super-linearly $o(\exp(-k))$. Software `http://mloss.org/software/view/183/` (This is limited to non-structured sparse estimation.)
- Beyond convexity
  - Generalized eigenvalue problems.
  - Difference of convex (DC) programming.
  - Dual ascent (or dual decomposition) for sequence labeling in natural language processing; see [Wainwright, Jaakkola, Willsky 05; Koo et al. 10]
- Stochastic optimization
  - Good tutorial by Nathan Srebro (ICML2010)

# Optimization for Machine Learning

A new book "Optimization for Machine Learning" (2011)



# Possible projects

1. Compare the three approaches, namely IST, dual ascent, and ADMM, and discuss empirically (and theoretically) their pros and cons.
2. Apply one of the methods discussed in the lecture to model some real problem with (structured) sparsity or low-rank matrix.

# References


Recent surveys

- Tomioka, Suzuki, & Sugiyama (2011) Augmented Lagrangian Methods for Learning, Selecting, and Combining Features. In Sra, Nowozin, Wright., editors, *Optimization for Machine Learning*, MIT Press.
- Combettes & Pesquet (2010) Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer-Verlag.
- Boyd, Parikh, Peleato, & Eckstein (2010) Distributed optimization and statistical learning via the alternating direction method of multipliers.

Textbooks

- Rockafellar (1970) Convex Analysis. Princeton University Press.
- Bertsekas (1999) Nonlinear Programming. Athena Scientific.
- Nesterov (2003) Introductory Lectures on Convex Optimization: A Basic Course. Springer.
- Boyd & Vandenberghe. (2004) Convex optimization, Cambridge University Press.

# References

IST/FISTA


- Moreau (1965) Proximité et dualité dans un espace Hilbertien. Bul letin de la S. M. F.
- Nesterov (2007) Gradient Methods for Minimizing Composite Objective Function.
- Beck & Teboulle (2009) A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. SIAM J Imag Sci 2, 183–202.


Dual ascent


- Arrow, Hurwicz, & Uzawa (1958) Studies in Linear and Non-Linear Programming. Stanford University Press.
- Chapter 6 in Bertsekas (1999).
- Wainwright, Jaakkola, & Willsky (2005) Map estimation via agreement on trees: message-passing and linear programming. IEEE Trans IT, 51(11).


Augmented Lagrangian


- Rockafellar (1976) Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. of Oper. Res. 1.
- Bertsekas (1982) Constrained Optimization and Lagrange Multiplier Methods. Academic Press.
- Tomioka, Suzuki, & Sugiyama (2011) Super-Linear Convergence of Dual Augmented Lagrangian Algorithm for Sparse Learning. JMLR 12.

# References

ADMM


- Gabay & Mercier (1976) A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput Math Appl 2, 17–40.
- Lions & Mercier (1979) Splitting Algorithms for the Sum of Two Nonlinear Operators. SIAM J Numer Anal 16, 964–979.
- Eckstein & Bertsekas (1992) On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators.


Matrices


- Srebro, Rennie, & Jaakkola (2005) Maximum-Margin Matrix Factorization. Advances in NIPS 17, 1329–1336.
- Cai, Candès, & Shen (2008) A singular value thresholding algorithm for matrix completion.
- Tomioka, Suzuki, Sugiyama, & Kashima (2010) A Fast Augmented Lagrangian Algorithm for Learning Low-Rank Matrices. In ICML 2010.
- Mazumder, Hastie, & Tibshirani (2010) Spectral Regularization Algorithms for Learning Large Incomplete Matrices. JMLR 11, 2287–2322.

# References


Multi-task/Mutliple kernel learning

- Evgeniou, Micchelli, & Pontil (2005) Learning Multiple Tasks with Kernel Methods. JMLR 6, 615–637.
- Lanckriet, Christiani, Bartlett, Ghaoui, & Jordan (2004) Learning the Kernel Matrix with Semidefinite Programming.
- Bach, Thibaux, & Jordan (2005) Computing regularization paths for learning multiple kernels. Advances in NIPS, 73–80.

Structured sparsity

- Tibshirani, Saunders, Rosset, Zhu and Knight. (2005) Sparsity and smoothness via the fused lasso. J. Roy. Stat. Soc. B, 67.
- Rudin, Osher, Fetemi. (1992) Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena, 60.
- Goldstein & Osher (2009) Split Bregman method for L1 regularization problems. SIAM J. Imag. Sci. 2.
- Mairal, Jenatton, Obozinski, & Bach. (2011) Convex and network flow optimization for structured sparsity.

Bayes & Probabilistic Inference

- Wainwright & Jordan (2008) Graphical Models, Exponential Families, and Variational Inference.