Convex Optimization: Old Tricks for New Problems

Ryota Tomioka[1]

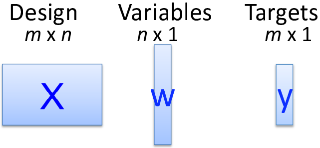
[1]The University of Tokyo

2011-08-26 @ DTU PhD Summer Course

Why care about convex optimization (and sparsity)?

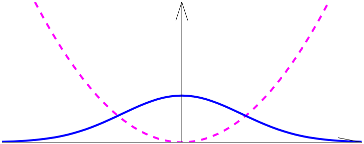
A typical machine learning problem (1/2)

$$\underset{\boldsymbol{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{\frac{1}{2}\|\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y}\|_2^2}_{\text{data-fit}} + \underbrace{\phi_\lambda(\boldsymbol{w})}_{\text{Regularization}}$$



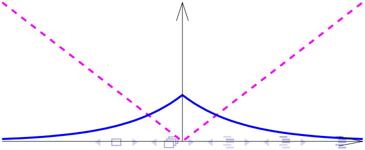
Ridge penalty

$$\phi_\lambda = \frac{\lambda}{2}\sum_{j=1}^{n} w_j^2.$$



L1 penalty

$$\phi_\lambda = \lambda \sum_{j=1}^{n} |w_j|.$$



A typical machine learning problem (2/2)

Logistic regression for binary ($y_i \in \{-1, +1\}$) classification:

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

The logistic loss function

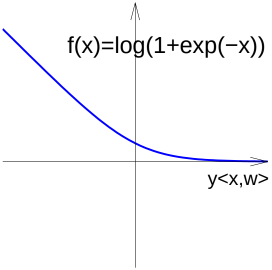
$$\log(1 + e^{-yz}) = -\log P(Y = y|z)$$

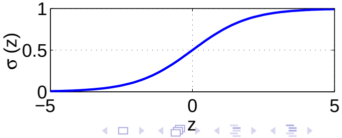
negative log-likelihood

where

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

logistic function





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

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

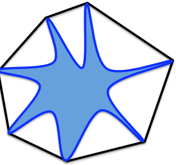
$$\text{s.t.} \quad q(w) \geq 0, \quad \int q(w)\mathrm{d}w = 1$$

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

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

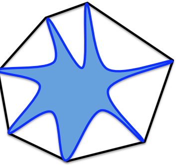
$$\text{s.t.} \quad q(w) \geq 0, \quad \int q(w)\mathrm{d}w = 1$$

where

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

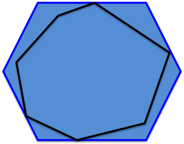
$$\Rightarrow \quad q(w) = \frac{1}{Z} e^{-f(w)} \quad \text{(Bayesian posterior)}$$

Inner approximations



- Variational Bayes
- Empirical Bayes

Outer approximations



- Belief propagation

See Wainwright & Jordan 08.

Convex optimization = standard forms (boring?)

Example: Linear Programming (LP)

Primal problem

$$(\text{P}) \quad \min \quad \boldsymbol{c}^\top \boldsymbol{x}, \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq 0.$$

Dual problem

$$(\text{D}) \quad \max \quad \boldsymbol{b}^\top \boldsymbol{y}, \quad \text{s.t.} \quad \boldsymbol{A}^\top \boldsymbol{y} \leq \boldsymbol{c}.$$

Quadratic Programming (QP), Second Order Cone Programming (SOCP), Semidefinite Programming (SDP), etc...

Convex optimization = standard forms (boring?)

Example: Linear Programming (LP)

Primal problem

$$(\text{P}) \quad \min \quad \boldsymbol{c}^\top \boldsymbol{x}, \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq 0.$$

Dual problem

$$(\text{D}) \quad \max \quad \boldsymbol{b}^\top \boldsymbol{y}, \quad \text{s.t.} \quad \boldsymbol{A}^\top \boldsymbol{y} \leq \boldsymbol{c}.$$

Quadratic Programming (QP), Second Order Cone Programming (SOCP), Semidefinite Programming (SDP), etc...

- Pro: "Efficient" (but complicated) solvers are already available.
- Con: Have to *rewrite* your problem into one of them.

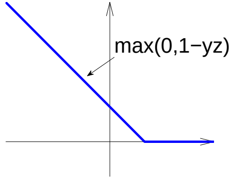
Easy problems (that we don't discuss)

- Objective f is differentiable & no constraint
  - L-BFGS quasi-Newton method
    - ★ requires only gradient.
    - ★ scales well.
  - Newton's method
    - ★ requires also Hessian.
    - ★ very accurate.
    - ★ for medium sized problems.
- Differentiable f & simple box constraint
  - L-BFGS-B quasi-Newton method

Non-differentiability is everywhere

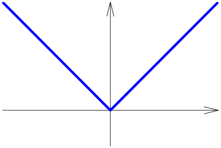
- Support Vector Machine

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad C\sum_{i=1}^{m} \ell_H(y_i \langle \boldsymbol{x}_i, \boldsymbol{w} \rangle) + \frac{1}{2}\|\boldsymbol{w}\|^2$$



- Lasso (least absolute shrinkage and selection operator)

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad L(\boldsymbol{w}) + \lambda \sum_{j=1}^{n} |w_j|$$



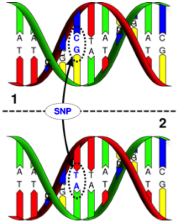
⇒ Leads to sparse (most of w_j will be zero) solutions

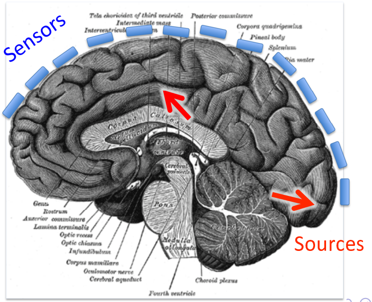
Why we need sparsity

- Genome-wide association studies
  - Hundreds of thousands of genetic variations (SNPs), small number of participants (samples).
  - Number of genes responsible for the disease is small.
  - Solve classification problem (disease/healthy) with sparsity constraint.
- EEG/MEG source localization
  - Number of possible sources $\gg$ number of sensors
  - Needs sparsity at a group level

$$\phi_\lambda(\boldsymbol{w}) = \lambda \sum_{\mathfrak{g} \in \mathfrak{G}} \|\boldsymbol{w}_\mathfrak{g}\|_2$$

$$(\boldsymbol{w}_\mathfrak{g} \in \mathbb{R}^3)$$



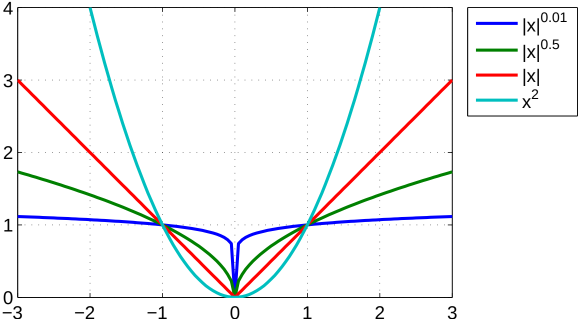


L1-regularization and sparsity

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

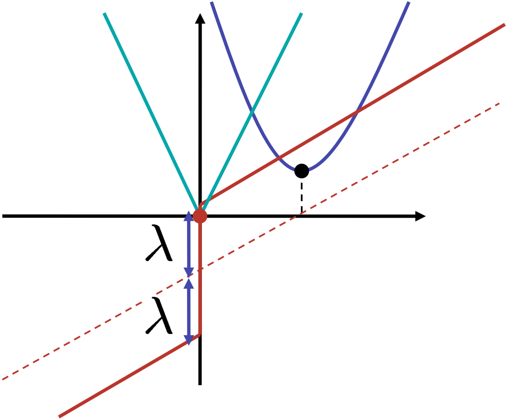
L1-regularization and sparsity

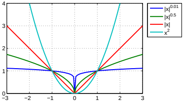
- Best convex approximation of $\|\boldsymbol{w}\|_0$.



L1-regularization and sparsity

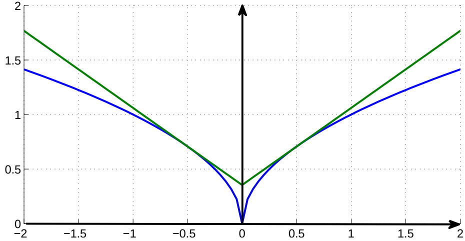
- Best convex approximation of $\|\boldsymbol{w}\|_0$.
- Threshold occurs for finite λ.

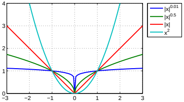


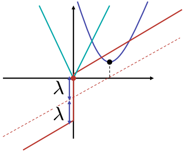


L1-regularization and sparsity

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







Multiple kernels & multiple tasks

- Multiple kernel learning [Lanckriet et al., 04; Bach et al., 04;...]
  - Given: kernel functions $k_1(x, x'), \ldots, K_M(x, x')$
  - How do we optimally select and combine “good“ kernels?

$$\underset{\substack{f_1 \in \mathcal{H}_1, \\ f_2 \in \mathcal{H}_2, \\ \ldots, f_M \in \mathcal{H}_M}}{\text{minimize}} \quad C \sum_{i=1}^{N} \ell \left(y_i \sum_{m=1}^{M} f_m(x_i) \right) + \lambda \sum_{m=1}^{M} \|f_m\|_{\mathcal{H}_m}$$

Multiple kernels & multiple tasks

- Multiple kernel learning [Lanckriet et al., 04; Bach et al., 04;...]
  - Given: kernel functions $k_1(x, x'), \ldots, K_M(x, x')$
  - How do we optimally select and combine “good“ kernels?

$$\underset{\substack{f_1 \in \mathcal{H}_1,\\ f_2 \in \mathcal{H}_2,\\ \ldots, f_M \in \mathcal{H}_M}}{\text{minimize}} \quad C \sum_{i=1}^{N} \ell\left(y_i \sum_{m=1}^{M} f_m(x_i)\right) + \lambda \sum_{m=1}^{M} \|f_m\|_{\mathcal{H}_m}$$

- Multiple task learning [Evgeniou et al 05]
  - Given: two learning tasks.
  - Can we do better than solving them individually?

$$\underset{\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_{12}}{\text{minimize}} \quad \underbrace{L_1(\boldsymbol{w}_1 + \boldsymbol{w}_{12})}_{\text{Task 1 loss}} + \underbrace{L_2(\boldsymbol{w}_2 + \boldsymbol{w}_{12})}_{\text{Task 2 loss}} + \lambda(\|\boldsymbol{w}_1\| + \|\boldsymbol{w}_2\| + \|\boldsymbol{w}_{12}\|)$$

$\boldsymbol{w}_{12}$: shared component, $\boldsymbol{w}_1$: Task 1 only component, $\boldsymbol{w}_2$: Task 2 only component.

Estimation of low-rank matrices (1/2)

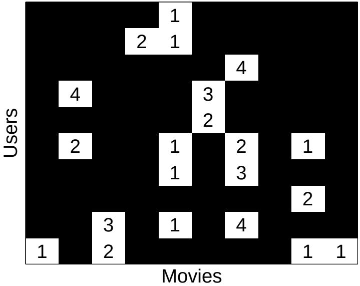
- Completion of partially observed low-rank matrix

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

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

Linear sum of singular-values ⇒ sparsity in the singular-values.

- Collaborative filtering (netflix)
- Sensor network localization



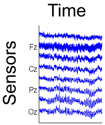
Estimation of low-rank matrices (2/2)

- Classification of matrix shaped data $\boldsymbol{X}$.

$$f(\boldsymbol{X}) = \langle \boldsymbol{W}, \boldsymbol{X} \rangle + b$$

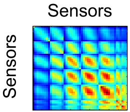
- Multivariate Time Series

$$X =$$



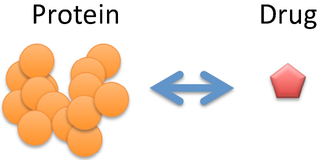
- Second order statistics

$$X =$$



- Classification of binary relationship between two objects (e.g., protein and drug)

$$f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{W} \boldsymbol{y} + b$$



Agenda

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

Convexity

Learning objectives

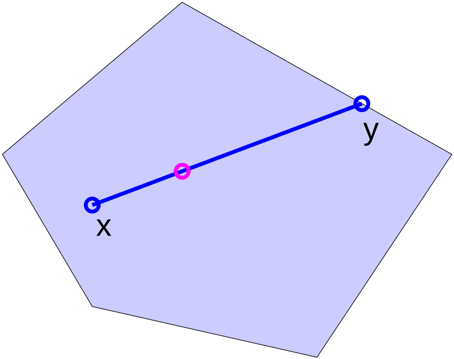
- Convex sets
- Convex function
- Conditions that guarantee convexity
- Convex optimization problem

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,

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



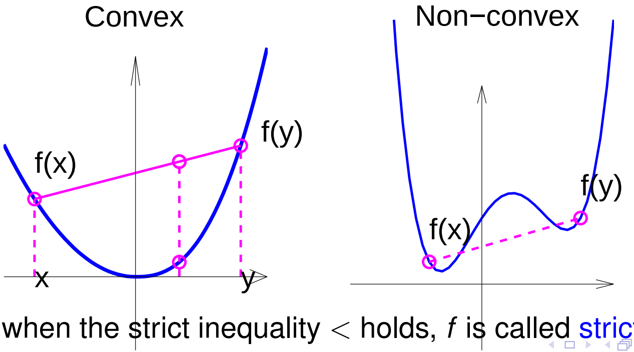
Convex function

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

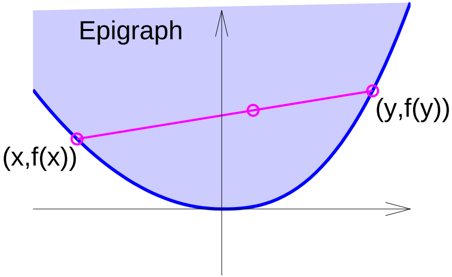
NB: when the strict inequality $<$ holds, f is called strictly convex.

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

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



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

Exercise

- Show that the indicator function $\delta_C(\boldsymbol{x})$ of a convex set C is a convex function. Here

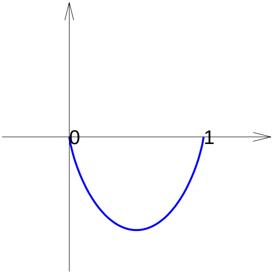
$$\delta_C(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

Conditions that guarantee convexity (1/3)

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

  Examples
  - (Negative) entropy is a convex function.

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



Conditions that guarantee convexity (1/3)

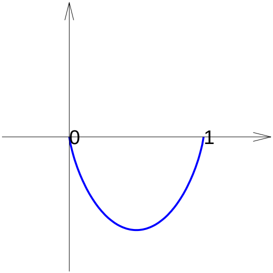
- Hessian $\nabla^2 f(\boldsymbol{x})$ is positive semidefinite (if f is differentiable)

  Examples

  - (Negative) entropy is a convex function.

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

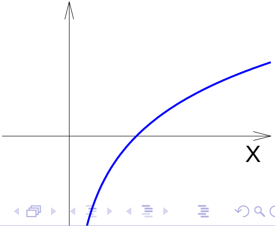
$$\nabla^2 f(p) = \mathrm{diag}(1/p_1, \ldots, 1/p_n) \succeq 0.$$



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

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

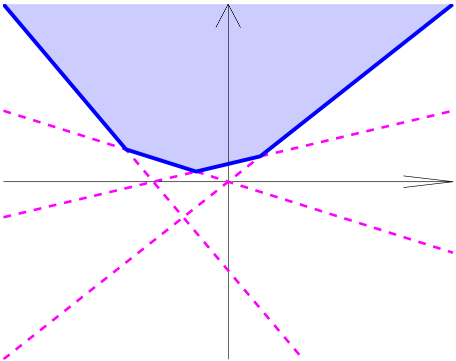
$$\nabla^2 f(\boldsymbol{X}) = -\boldsymbol{X}^{-\top} \otimes \boldsymbol{X}^{-1} \preceq 0$$



Conditions that guarantee convexity (2/3)

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

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

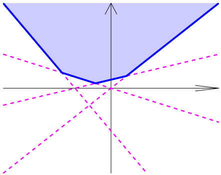


The same as saying "intersection of convex sets is a convex set"

Conditions that guarantee convexity (2/3)

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

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



Example

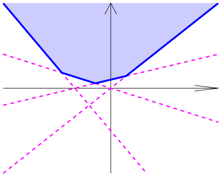
- Quadratic over linear is a convex function

$$f(\boldsymbol{y},\boldsymbol{\Sigma}) = \max_{\boldsymbol{\alpha}} \left[-\frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{\Sigma}\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \boldsymbol{y} \right] \quad (\boldsymbol{\Sigma} \succ 0)$$

Conditions that guarantee convexity (2/3)

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

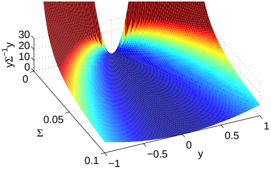
$$f(\boldsymbol{x}) := \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} f(\boldsymbol{x};\boldsymbol{\alpha})$$



Example

- Quadratic over linear is a convex function

$$f(\boldsymbol{y},\boldsymbol{\Sigma}) = \max_{\boldsymbol{\alpha}} \left[-\frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{\Sigma}\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top\boldsymbol{y} \right] \quad (\boldsymbol{\Sigma} \succ 0)$$
$$= \frac{1}{2}\boldsymbol{y}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{y}$$



Conditions that guarantee convexity (3/3)

- Minimum of jointly convex function $f(x, y)$

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

Examples

- Hierarchical prior minimization

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

Conditions that guarantee convexity (3/3)

- Minimum of jointly convex function $f(x, y)$

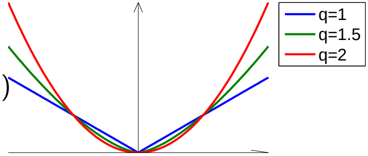
$$f(x) := \min_{y \in \mathbb{R}^n} f(x, y) \quad \text{is convex.}$$

Examples

- Hierarchical prior minimization

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

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



Conditions that guarantee convexity (3/3)

- Minimum of jointly convex function $f(x, y)$

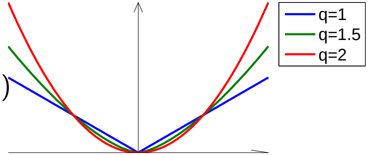
$$f(x) := \min_{y \in \mathbb{R}^n} f(x, y) \quad \text{is convex.}$$

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)

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

Conditions that guarantee convexity (3/3)

- Minimum of jointly convex function $f(x, y)$

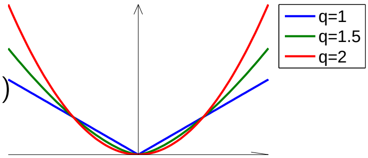
$$f(x) := \min_{y \in \mathbb{R}^n} f(x, y) \quad \text{is convex.}$$

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)

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

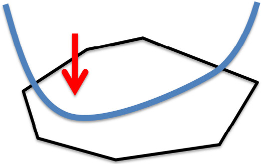
$$= \mathrm{Tr}\left((\boldsymbol{X}^\top \boldsymbol{X})^{1/2} \right) = \sum_{j=1}^{r} \sigma_j(\boldsymbol{X}).$$

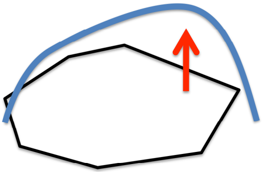
Convex optimization problem

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!

Proximity operators and IST methods

Learning objectives

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

Proximity view on gradient descent

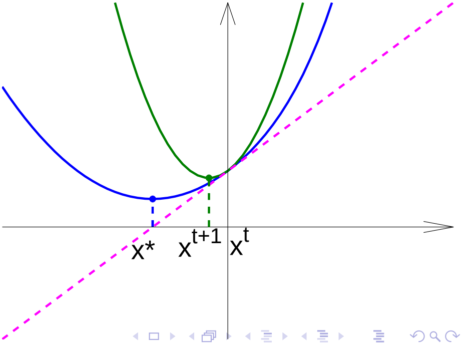
"Linearize and Prox"

$$\boldsymbol{x}^{t+1} = \operatorname*{argmin}_{\boldsymbol{x}} \left(\nabla f(\boldsymbol{x}^t)(\boldsymbol{x} - \boldsymbol{x}^t) + \frac{1}{2\eta_t}\|\boldsymbol{x} - \boldsymbol{x}^t\|^2 \right)$$
$$= \boldsymbol{x}^t - \eta_t \nabla f(\boldsymbol{x}^t)$$

- 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?

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

Constraint minimization problem

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

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

- can be equivalently written as

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

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

Projected gradient method (Bertsekas 99; Nesterov 03)

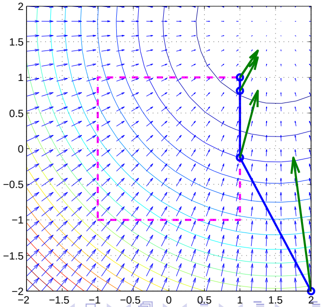
Linearize the objective f, δ_C is the indicator of the constraint C

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

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

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

- Need the projection proj_C to be easy to compute



Ideas for regularized minimization

Constrained minimization problem

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

⇒ need to compute the projection

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

Regularized minimization problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{x}) + \phi_\lambda(\boldsymbol{x})$$

⇒ need to compute the proximity operator

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

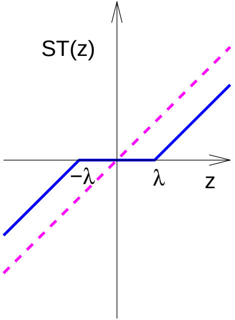
Proximal Operator: generalization of projection

$$\mathrm{prox}_{\phi_\lambda}(\boldsymbol{z}) = \operatorname*{argmin}_{\boldsymbol{x}} \left(\phi_\lambda(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 \right)$$

- $\phi_\lambda = \delta_C$: Projection onto a convex set $\mathrm{prox}_{\delta_C}(\boldsymbol{z}) = \mathrm{proj}_C(\boldsymbol{z})$.
- $\phi_\lambda(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1$: Soft-Threshold

$$\mathrm{prox}_\lambda(\boldsymbol{z}) = \operatorname*{argmin}_{\boldsymbol{x}} \left(\lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 \right)$$

$$= \begin{cases} z_j + \lambda & (z_j < -\lambda), \\ 0 & (-\lambda \le z_j \le \lambda), \\ z_j - \lambda & (z_j > \lambda). \end{cases}$$



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

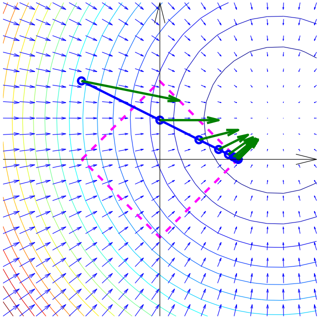
Iterative Shrinkage Thresholding (IST)

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

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

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

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



IST summary

Solve minimization problem

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

by iteratively computing

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

Exercise: Derive prox operator for

- Ridge regularization

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

- Elastic-net regularization

$$\phi_\lambda(\boldsymbol{w}) = \lambda \sum_{j=1}^{n} \left((1-\theta)|w_j| + \theta w_j^2\right).$$

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 hints

1. Compute the gradient of the loss term

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

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:

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

Regularization:

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

Matrix completion via IST (Mazumder et al. 10)

Loss function:

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

gradient:

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

Regularization:

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

Matrix completion via IST (Mazumder et al. 10)

Loss function:

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

gradient:

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

Regularization:

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

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:

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

gradient:

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

Regularization:

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

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.

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

1. Initialize $\boldsymbol{x}^0$ appropriately, $\boldsymbol{y}^1 = \boldsymbol{x}^0, \;\; s_1 = 1$.
2. Update $\boldsymbol{x}^t$:

$$\boldsymbol{x}^t = \mathrm{prox}_{\lambda\eta_t}(\boldsymbol{y}^t - \eta_t \nabla L(\boldsymbol{y}^t)).$$

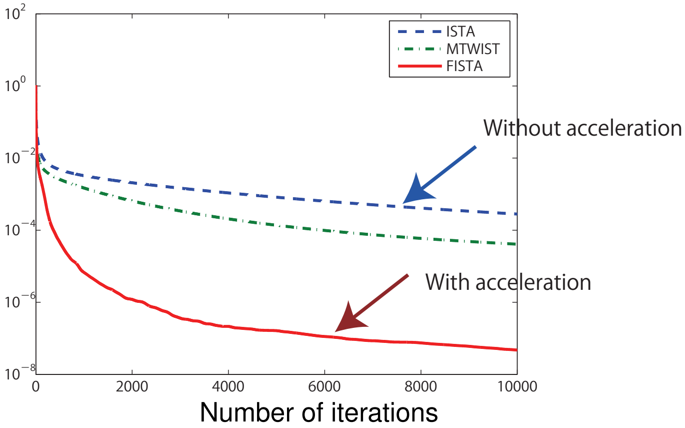
3. Update $\boldsymbol{y}^t$:

$$\boldsymbol{y}^{t+1} = \boldsymbol{x}^t + \left(\frac{s_t - 1}{s_{t+1}}\right)(\boldsymbol{x}^t - \boldsymbol{x}^{t-1}),$$

where $s_{t+1} = (1 + \sqrt{1 + 4s_t^2})/2$.

- The same per iteration complexity. Converges as $O(1/k^2)$.
- Roughly speaking, $\boldsymbol{y}^t$ 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

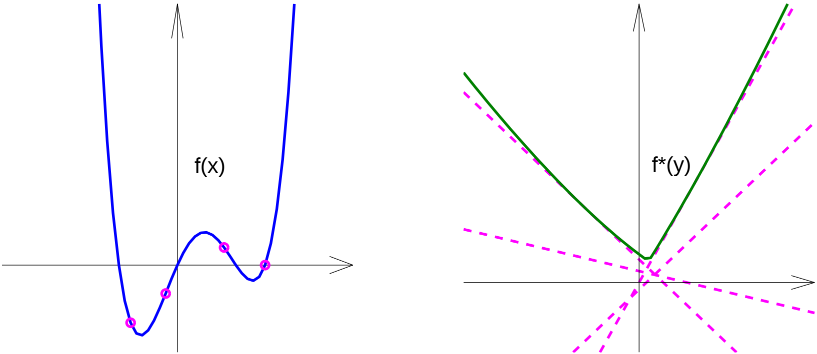
Conjugate duality and dual ascent

- Convex conjugate function
- Lagrangian relaxation and dual problem
- Dual ascent

Conjugate duality

The convex conjugate f^* of a function f:

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



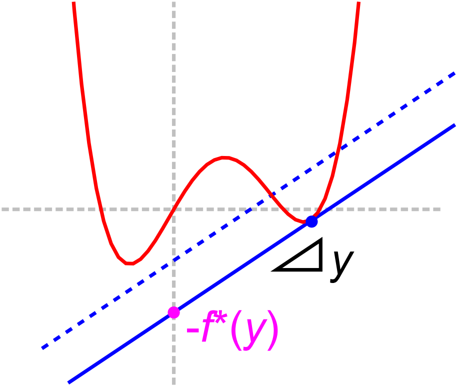
Since the maximum over linear functions is always convex, f need not be convex.

Conjugate duality (dual view)

Convex conjugate function

$-f^*(\boldsymbol{y})$ is the minimum y-intercept of the hyperplanes that has slope $\boldsymbol{y}$ and have intersection with the graph of $f(\boldsymbol{x})$.

$$
\begin{aligned}
f^*(\boldsymbol{y}) &= \sup_{\boldsymbol{x}} \left(\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x})\right) \\
\Leftrightarrow -f^*(\boldsymbol{y}) &= \inf_{\boldsymbol{x}} \left(f(\boldsymbol{x}) - \langle \boldsymbol{x}, \boldsymbol{y}\rangle\right) \\
&= \inf_{\boldsymbol{x}, b} b, \\
&\text{s.t.} \quad f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{y}\rangle + b.
\end{aligned}
$$

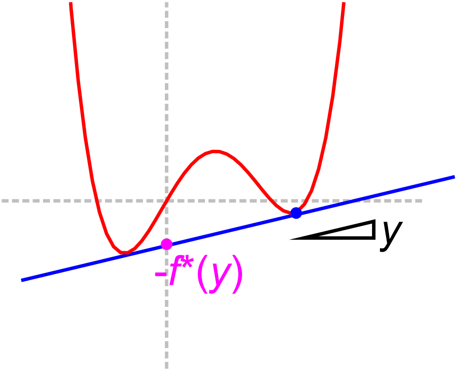


Conjugate duality (dual view)

Convex conjugate function

$-f^*(\boldsymbol{y})$ is the minimum y-intercept of the hyperplanes that has slope $\boldsymbol{y}$ and have intersection with the graph of $f(\boldsymbol{x})$.

$$
\begin{aligned}
f^*(\boldsymbol{y}) &= \sup_{\boldsymbol{x}} \left(\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x})\right)\\
\Leftrightarrow -f^*(\boldsymbol{y}) &= \inf_{\boldsymbol{x}} \left(f(\boldsymbol{x}) - \langle \boldsymbol{x}, \boldsymbol{y}\rangle\right)\\
&= \inf_{\boldsymbol{x}, b} b,\\
&\text{s.t.} \quad f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{y}\rangle + b.
\end{aligned}
$$

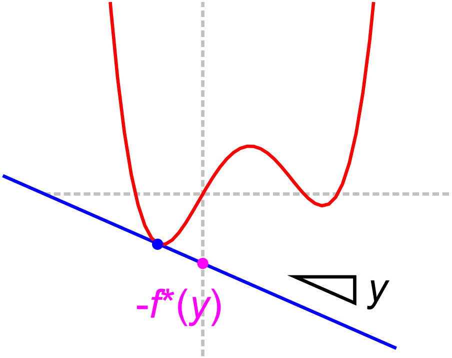


Conjugate duality (dual view)

Convex conjugate function

$-f^*(\boldsymbol{y})$ is the minimum y-intercept of the hyperplanes that has slope $\boldsymbol{y}$ and have intersection with the graph of $f(\boldsymbol{x})$.

$$
\begin{aligned}
f^*(\boldsymbol{y}) &= \sup_{\boldsymbol{x}} \left(\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x})\right) \\
\Leftrightarrow -f^*(\boldsymbol{y}) &= \inf_{\boldsymbol{x}} \left(f(\boldsymbol{x}) - \langle \boldsymbol{x}, \boldsymbol{y}\rangle\right) \\
&= \inf_{\boldsymbol{x}, b} b, \\
\text{s.t.} \quad & f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{y}\rangle + b.
\end{aligned}
$$

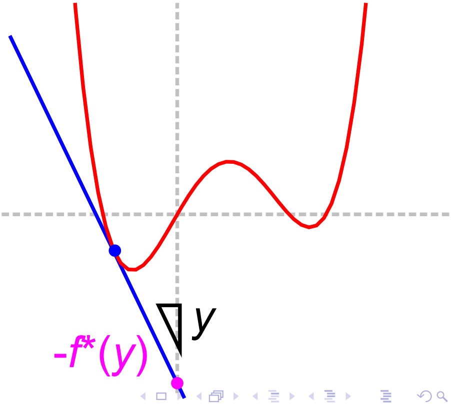


Conjugate duality (dual view)

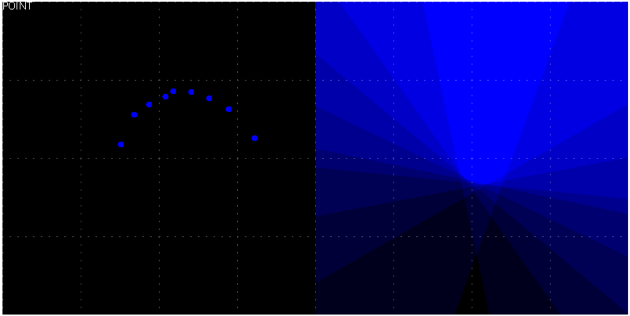
Convex conjugate function

$-f^*(\boldsymbol{y})$ is the minimum y-intercept of the hyperplanes that has slope $\boldsymbol{y}$ and have intersection with the graph of $f(\boldsymbol{x})$.

$$
\begin{aligned}
f^*(\boldsymbol{y}) &= \sup_{\boldsymbol{x}} \left(\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x})\right) \\
\Leftrightarrow -f^*(\boldsymbol{y}) &= \inf_{\boldsymbol{x}} \left(f(\boldsymbol{x}) - \langle \boldsymbol{x}, \boldsymbol{y}\rangle\right) \\
&= \inf_{\boldsymbol{x}, b} b, \\
&\text{s.t.} \quad f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{y}\rangle + b.
\end{aligned}
$$



Demo

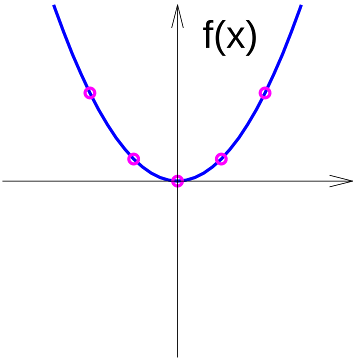


http://www.ibis.t.u-tokyo.ac.jp/ryotat/applets/pld/

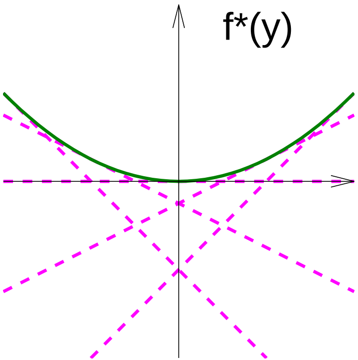
Example of conjugate duality $f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \mathbb{R}^n} (\langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x}))$

- Quadratic function

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



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



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

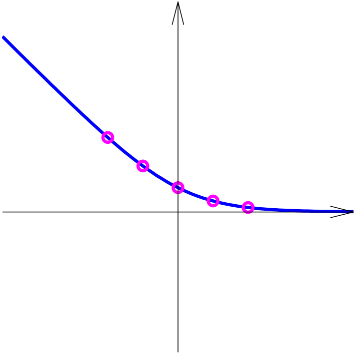
- Logistic loss function

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

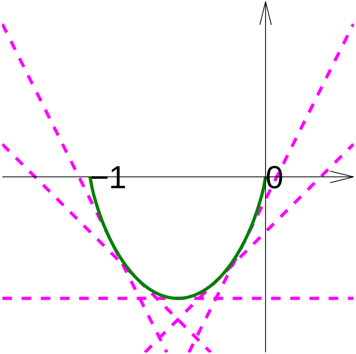
Example of conjugate duality $f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n} (\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f(\boldsymbol{x}))$

- Logistic loss function

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



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



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

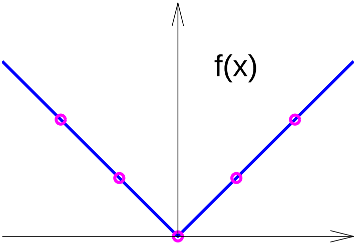
- L1 regularizer

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

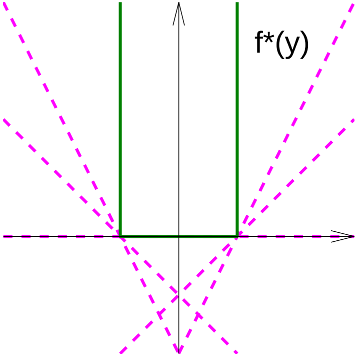
Example of conjugate duality $f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \mathbb{R}^n} (\langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x}))$

- L1 regularizer

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

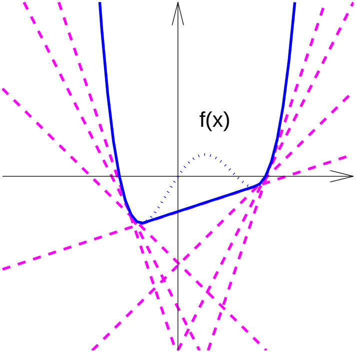


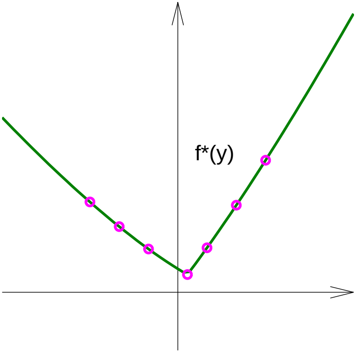
$$f^*(y) = \begin{cases} 0 & (-1 \leq y \leq 1) \\ +\infty & (\text{otherwise}) \end{cases}$$



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:

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

(For example $f(\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|_2^2$ (squared loss))

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

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

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

Weak duality

$$\inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z},\boldsymbol{w},\alpha) \leq p^* \quad \text{(primal optimal)}$$

proof

$$\inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z},\boldsymbol{w},\alpha) = \inf\left(\inf_{\boldsymbol{z}=\boldsymbol{A}\boldsymbol{w}} \mathcal{L}(\boldsymbol{z},\boldsymbol{w},\alpha),\ \inf_{\boldsymbol{z}\neq\boldsymbol{A}\boldsymbol{w}} \mathcal{L}(\boldsymbol{z},\boldsymbol{w},\alpha)\right)$$

Weak duality

$$\inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha) \leq p^* \quad \text{(primal optimal)}$$

proof

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

Weak duality

$$\inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}, \alpha) \leq p^* \quad \text{(primal optimal)}$$

proof

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

Dual problem

From the above argument

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

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

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 $\boldsymbol{w}$?

Dual problem

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

Note

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

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

Dual problem

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

Dual problem

$$
\begin{aligned}
d(\alpha) &= \inf_{\boldsymbol{z},\boldsymbol{w}} \mathcal{L}(\boldsymbol{z},\boldsymbol{w},\alpha) \quad (\leq p^*) \\
&= \inf_{\boldsymbol{z},\boldsymbol{w}} \Big(f(\boldsymbol{z}) + g(\boldsymbol{w}) + \alpha^\top (\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}) \Big)
\end{aligned}
$$

Dual problem

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

Dual problem

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

Dual problem

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

Fenchel's duality

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

M. W. Fenchel

Examples

- Logistic regression with L1 regularization

$$f(\boldsymbol{z}) = \sum_{i=1}^{m} \log(1 + \exp(-z_i)), \quad g(\boldsymbol{w}) = \lambda\|\boldsymbol{w}\|_1.$$

- Support vector machine (SVM)

$$f(\boldsymbol{z}) = C\sum_{i=1}^{m} \max(0, 1 - z_i), \quad g(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2.$$

Example 1: Logistic regression with L1 regularization

Primal

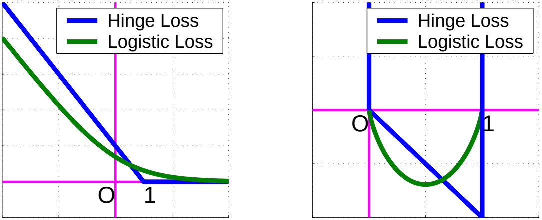
$$\min_{\boldsymbol{w}} \quad f(\boldsymbol{y} \circ \boldsymbol{X}\boldsymbol{w}) + \phi_\lambda(\boldsymbol{w})$$

$$\begin{cases} f(\boldsymbol{z}) = \sum_{i=1}^{m} \log(1 + \exp(-z_i)), \\ \phi_\lambda(\boldsymbol{w}) = \lambda \|\boldsymbol{w}\|_1. \end{cases}$$

Dual

$$\max_{\boldsymbol{\alpha}} \quad -f^*(-\boldsymbol{\alpha}) - \phi_\lambda^*(\boldsymbol{X}^\top(\boldsymbol{\alpha} \circ \boldsymbol{y}))$$

$$\begin{cases} f^*(-\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i \log(\alpha_i) \\ \qquad\qquad + (1-\alpha_i)\log(1-\alpha_i), \\ \phi_\lambda^*(\boldsymbol{v}) = \begin{cases} 0 & (\|\boldsymbol{w}\|_\infty \le \lambda), \\ +\infty & (\text{otherwise}). \end{cases} \end{cases}$$



(a) primal losses

(b) dual losses

Example 2: Support vector machine

Primal

$$\min_{\boldsymbol{w}} \quad f(\boldsymbol{y} \circ \boldsymbol{X}\boldsymbol{w}) + \phi_\lambda(\boldsymbol{w})$$

$$\begin{cases} f(\boldsymbol{z}) = C\sum_{i=1}^{m} \max(0, 1 - z_i), \\ \phi_\lambda(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2. \end{cases}$$

Dual

$$\max_{\boldsymbol{\alpha}} \quad -f^*(-\boldsymbol{\alpha}) - \phi_\lambda^*(\boldsymbol{X}^\top(\boldsymbol{\alpha} \circ \boldsymbol{y}))$$

$$\begin{cases} f^*(-\boldsymbol{\alpha}) = \begin{cases} \sum_{i=1}^{m} -\alpha_i & (0 \leq \boldsymbol{\alpha} \leq C), \\ +\infty & \text{(oterwise)}, \end{cases} \\ \phi_\lambda^*(\boldsymbol{v}) = \frac{1}{2}\|\boldsymbol{v}\|^2. \end{cases}$$

Dual ascent

Assume for a moment that the dual $d(\alpha)$ is differentiable.

For a given α^t

$$d(\alpha^t) = \inf_{\boldsymbol{z},\boldsymbol{w}} \left(f(\boldsymbol{z}) + g(\boldsymbol{w}) + \langle \alpha^t, \boldsymbol{z} - \boldsymbol{A}\boldsymbol{w} \rangle \right)$$

and one can show that (Chapter 6, Bertsekas 99)

$$\nabla_{\alpha} d(\alpha^t) = \boldsymbol{z}^{t+1} - \boldsymbol{A}\boldsymbol{w}^{t+1}$$

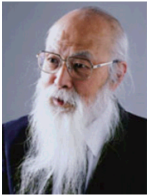
where

$$\boldsymbol{z}^{t+1} = \operatorname*{argmin}_{\boldsymbol{z}} \left(f(\boldsymbol{z}) + \langle \alpha^t, \boldsymbol{z} \rangle \right)$$

$$\boldsymbol{w}^{t+1} = \operatorname*{argmin}_{\boldsymbol{w}} \left(g(\boldsymbol{w}) - \left\langle \boldsymbol{A}^\top \alpha^t, \boldsymbol{w} \right\rangle \right)$$

Dual ascent (Uzawa's method)

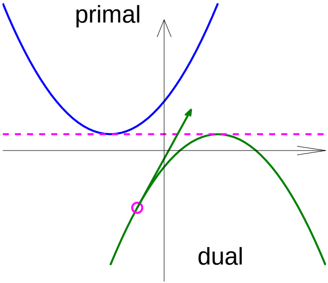
$$\begin{cases} \text{Minimize the Lagrangian wrt } \boldsymbol{x} \text{ and } \boldsymbol{z}\text{:} \\ \boldsymbol{z}^{t+1} = \mathrm{argmin}_{\boldsymbol{z}} \left(f(\boldsymbol{z}) + \langle \boldsymbol{\alpha}^t, \boldsymbol{z} \rangle \right), \\ \boldsymbol{w}^{t+1} = \mathrm{argmin}_{\boldsymbol{w}}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



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.



Exercise 2: Matrix completion via dual ascent (Cai et al. 08)

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad \underbrace{\frac{1}{2\lambda}\|\boldsymbol{z}-\boldsymbol{y}\|^2}_{\text{Strictly convex}} + \underbrace{\left(\tau\|\boldsymbol{X}\|_{\text{tr}} + \frac{1}{2}\|\boldsymbol{X}\|^2\right)}_{\text{Strictly convex}},$$

$$\text{s.t.} \quad \Omega(\boldsymbol{X}) = \boldsymbol{z}.$$

Exercise 2: Matrix completion via dual ascent (Cai et al. 08)

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad \underbrace{\frac{1}{2\lambda}\|\boldsymbol{z}-\boldsymbol{y}\|^2}_{\text{Strictly convex}} + \underbrace{\left(\tau\|\boldsymbol{X}\|_{\text{tr}} + \frac{1}{2}\|\boldsymbol{X}\|^2\right)}_{\text{Strictly convex}},$$

$$\text{s.t.} \quad \Omega(\boldsymbol{X}) = \boldsymbol{z}.$$

$$\Downarrow$$

Lagrangian:

$$\mathcal{L}(\boldsymbol{X},\boldsymbol{z},\boldsymbol{\alpha}) = \underbrace{\frac{1}{2\lambda}\|\boldsymbol{z}-\boldsymbol{y}\|^2}_{=f(\boldsymbol{z})} + \underbrace{\left(\tau\|\boldsymbol{X}\|_{S_1} + \frac{1}{2}\|\boldsymbol{X}\|^2\right)}_{=g(\boldsymbol{x})} + \boldsymbol{\alpha}^\top(\boldsymbol{z}-\Omega(\boldsymbol{X})).$$

Dual ascent

$$\begin{cases} \boldsymbol{X}^{t+1} = \text{prox}_\tau\left(\Omega^\top(\boldsymbol{\alpha}^t)\right) & \text{(Singular-Value Thresholding)} \\ \boldsymbol{z}^{t+1} = \boldsymbol{y} - \lambda\boldsymbol{\alpha}^t \\ \boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + \eta_t(\boldsymbol{z}^{t+1} - \Omega(\boldsymbol{X}^{t+1})) \end{cases}$$

Augmented Lagrangian and ADMM

Learning objectives

- Structured sparse estimation
- Augmented Lagrangian
- Alternating direction method of multipliers

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

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

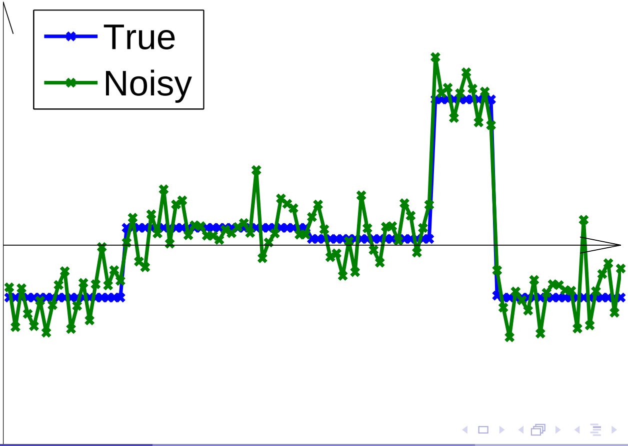
Original X_0

Observed Y

In one dimension

- Fused lasso [Tibshirani et al. 05]

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda \sum_{j=1}^{n-1} |x_{j+1} - x_j|$$



Structured sparsity estimation

- TV denoising

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

- Fused lasso

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \sum_{j=1}^{n-1} |x_{j+1} - x_j|$$

Structured sparsity estimation

- TV denoising

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

- Fused lasso

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

Structured sparse estimation problem

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

Structured sparse estimation problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{f(\boldsymbol{x})}_{\text{data-fit}} + \underbrace{\phi_\lambda(\boldsymbol{Ax})}_{\text{regularization}}$$

- Not easy to compute prox operator (because it is non-separable)
  $\Rightarrow$ difficult to apply IST-type methods.
- Dual is not necessarily differentiable
  $\Rightarrow$ difficult to apply dual ascent.

Forming the *augmented* Lagrangian

Structured sparsity problem

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

Equivalently written as

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

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

Forming the *augmented* Lagrangian

Structured sparsity problem

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

Equivalently written as

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

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

Augmented Lagrangian function

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

Augmented Lagrangian Method

Augmented Lagrangian function

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

Augmented Lagrangian method (Hestenes 69, Powell 69)

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

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

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

$$\begin{cases} \text{Minimize the AL function } \mathcal{L}_\eta(\boldsymbol{x}, \boldsymbol{z}^t, \alpha^t) \text{ wrt } \boldsymbol{x}\text{:} \\ \\ \text{Minimize the AL function } \mathcal{L}_\eta(\boldsymbol{x}^{t+1}, \boldsymbol{z}, \alpha^t) \text{ wrt } \boldsymbol{z}\text{:} \\ \\ \text{Update the Lagrangian multiplier:} \\ \alpha^{t+1} = \alpha^t + \eta(\boldsymbol{z}^{t+1} - \boldsymbol{A}\boldsymbol{x}^{t+1}). \end{cases}$$

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

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

Minimize the AL function $\mathcal{L}_\eta(\boldsymbol{x}, \boldsymbol{z}^t, \alpha^t)$ wrt $\boldsymbol{x}$:

$$\boldsymbol{x}^{t+1} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \left(f(\boldsymbol{x}) - \alpha^{t\top}\boldsymbol{A}\boldsymbol{x} + \frac{\eta}{2}\|\boldsymbol{z}^t - \boldsymbol{A}\boldsymbol{x}\|_2^2 \right).$$

Minimize the AL function $\mathcal{L}_\eta(\boldsymbol{x}^{t+1}, \boldsymbol{z}, \alpha^t)$ wrt $\boldsymbol{z}$:

Update the Lagrangian multiplier:

$$\alpha^{t+1} = \alpha^t + \eta(\boldsymbol{z}^{t+1} - \boldsymbol{A}\boldsymbol{x}^{t+1}).$$

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

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

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

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

Exercise: implement an ADMM for fused lasso

Fused lasso

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{A}\boldsymbol{x}\|_1$$

- What is the loss function f?
- What is the regularizer g?
- What is the matrix $\boldsymbol{A}$ for fused lasso?
- What is the prox operator for the regularizer g?

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

- Stopping criterion
  - Care must be taken when making a comparison.
- 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
  - Dual problem is always convex. It provides a lower-bound of the original problem. If $p^* = d^*$, you are done!
  - Dual ascent (or dual decomposition) for sequence labeling in natural language processing; see [Wainwright, Jaakkola, Willsky 05; Koo et al. 10]
  - Difference of convex (DC) programming.
  - Eigenvalue problem.
- Stochastic optimization
  - Good tutorial by Nathan Srebro (ICML2010)

A new book “Optimization for Machine Learning” is coming out from the MIT press.

Contributed authors including: A. Nemirovksi, D. Bertsekas, L. Vandenberghe, and more.

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.