Today

- Primal and dual problems
- Convex envelope and tight relaxations
- Subgradients and subgradient descent

Convex optimization – standard form

The problem

$$\begin{aligned} \underset{\mathbf{x}\in\mathcal{D}}{\text{minimize}}\ & f_0(\mathbf{x}) \\ \text{subject to}\ & f_i(\mathbf{x}) \leq 0, i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, i = 1, \ldots, p \end{aligned}$$

where $\mathcal{D} = \bigcap_{i=0}^{p} \mathbf{dom}(f_i) \cap \bigcap_{i=1}^{m} \mathbf{dom}(h_i)$.
If f_i are all convex and h_i are all affine then the above problem is convex.

Convex optimization

A point $\mathbf{x}$ for which all constraints are satisfied ($f_i(\mathbf{x}) \leq 0, i = 1, \ldots, m$ and $h_i(\mathbf{x}) = 0, i = 1, \ldots, p$) is called **feasible**.

If the set of feasible points is $\mathcal{C} \subset \mathcal{D}$ then the optimal value of the problem is

$$p^* = \inf_{\mathbf{x} \in \mathcal{C}} f_0(\mathbf{x}).$$

The set of points for $\{\mathbf{x} | \mathbf{x} \in \mathcal{C}, f_0(\mathbf{x}) = p^*\}$ is called the **optimal set** and any point in that set is an **optimal point**.

Duality

An optimization problem (**primal problem**)

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

induces a Lagrangian $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$

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

The $\mathbf{x}$ is a vector of **primal variables**.

The $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ are called **dual variables** or **Lagrange multipliers**.

The dual problem

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

is the **dual function**.

The dual lower bounds the optimal value of the primal problem $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq p^*$ when $\boldsymbol{\lambda} \succeq 0$.

Why $\boldsymbol{\lambda} \succeq 0$?

To see why $\boldsymbol{\lambda} \succeq 0$ let us assume a simple problem

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \; f_0(\mathbf{x}) \\ &\text{subject to } f_1(\mathbf{x}) \leq 0 \end{aligned}$$

where f_1 is unbounded from above, that is to say we can find an $\mathbf{x}$ such that $f_1(\mathbf{x}) = \infty$.

The dual

$$g(\lambda_1) = \inf_{\mathbf{x}} f_0(\mathbf{x}) + \lambda_1 f_1(\mathbf{x})$$

for $\lambda_1 < 0$ can be made $-\infty$.

Hence for $\lambda_1 < 0$ we get a trivial lower bound $(-\infty)$ on the optimum of the original problem.

Hence, in a general convex optimization problem, to get a sensible bound on p^* we must insist on $\boldsymbol{\lambda} \succeq 0$.

Example of a dual problem

Primal problem (linear program)

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} && \mathbf{c}^T\mathbf{x} \\ &\text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \succeq 0 \end{aligned}$$

Note that $f_i(\mathbf{x}) = -x_i, i = 1, \ldots, n$ and $h_i(\mathbf{x}) = \mathbf{a}_i^T x - b_i, i = 1, \ldots, m$

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) &= \mathbf{c}^T\mathbf{x} - \sum_{i=1} \lambda_i x_i + \boldsymbol{\nu}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \\ &= -\mathbf{b}^T\boldsymbol{\nu} + (\mathbf{c} + \mathbf{A}^T\boldsymbol{\nu} - \boldsymbol{\lambda})^T\mathbf{x} \end{aligned}$$

Example of a dual problem

The dual function is

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = -\mathbf{b}^T \boldsymbol{\nu} + \inf_{\mathbf{x}} (\mathbf{c} + \mathbf{A}^T \boldsymbol{\nu} - \boldsymbol{\lambda})^T \mathbf{x}$$

and unless $\mathbf{c} + \mathbf{A}^T \boldsymbol{\nu} - \boldsymbol{\lambda} = 0$ the above expression is $-\infty$ so we can write

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \begin{cases} -\mathbf{b}^T \boldsymbol{\nu}, & \mathbf{c} + \mathbf{A}^T \boldsymbol{\nu} - \boldsymbol{\lambda} = 0 \\ -\infty, & \text{otherwise} \end{cases}$$

which gives us the following dual problem

$$\begin{aligned} \underset{\boldsymbol{\lambda}, \boldsymbol{\nu}}{\text{maximize}} \quad & -\mathbf{b}^T \boldsymbol{\nu} \\ \text{subject to} \quad & \mathbf{c} + \mathbf{A}^T \boldsymbol{\nu} - \boldsymbol{\lambda} = 0 \\ & \boldsymbol{\lambda} \succeq 0 \end{aligned}$$

Dual problem

$$\begin{array}{ll} \underset{\boldsymbol{\lambda},\boldsymbol{\nu}}{\text{maximize}} & g(\boldsymbol{\lambda},\boldsymbol{\nu}) \\ & \boldsymbol{\lambda} \succeq 0 \end{array}$$

and in the context of this problem a point $(\boldsymbol{\lambda},\boldsymbol{\nu})$ is **dual feasible** if $\boldsymbol{\lambda} \succeq 0$ and $g(\boldsymbol{\lambda},\boldsymbol{\nu}) > -\infty$

Convex Conjugate

Convex conjugate is an important concept in convex analysis and very useful in machine learning.

The two key applications of the convex conjugates are

- deriving duals
- deriving convex envelopes, tightest convex relaxations

Convex Conjugate

Given a function $f : \mathbf{R}^N \to \mathbf{R}$ its convex conjugate is

$$f^*(\boldsymbol{\lambda}) = \sup_{\mathbf{x}} \langle \mathbf{x}, \boldsymbol{\lambda} \rangle - f(\mathbf{x})$$

Convex Conjugate

And to put this in context, assume an optimization problem

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\ f(\mathbf{x}) \\ &\text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned}$$

With Lagrangian

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_i \lambda_i (\mathbf{a}_i^T \mathbf{x} - b_i) = f(\mathbf{x}) + \langle \boldsymbol{\lambda}, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle$$

and dual

$$\begin{aligned} g(\boldsymbol{\lambda}) &= \inf_{\mathbf{x}} f(\mathbf{x}) + \langle \boldsymbol{\lambda}, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle = -\langle \boldsymbol{\lambda}, \mathbf{b} \rangle - \sup_{\mathbf{x}} \left\langle -\mathbf{A}^T \boldsymbol{\lambda}, \mathbf{x} \right\rangle - f(\mathbf{x}) \\ &= -\langle \boldsymbol{\lambda}, \mathbf{b} \rangle - f^*(-\mathbf{A}^T \boldsymbol{\lambda}) \end{aligned}$$

where we used $\inf_{\mathbf{x}} f(\mathbf{x}) = -\sup_{\mathbf{x}} -f(\mathbf{x})$.

Convex Conjugates

	Name	$f(\mathbf{x})$	convex conjugate $f^*(\mathbf{u})$
Losses	Linear	$-\left[\langle \mathbf{y}, \mathbf{x}\rangle - (1/2)\,\|\mathbf{x}\|_2^2\right]$	$(1/2)\,\|\mathbf{u}+\mathbf{y}\|_2^2$
	Logistic	$-\left[\langle \mathbf{y}, \mathbf{x}\rangle - \sum_i \log\{1+e^{x_i}\}\right]$	$\langle \mathbf{u}+\mathbf{y}, \log(\mathbf{u}+\mathbf{y})\rangle + \langle \mathbf{1}-\mathbf{u}-\mathbf{y}, \log(\mathbf{1}-\mathbf{u}-\mathbf{y})\rangle$
	Poisson	$-\left[\langle \mathbf{y}, \mathbf{x}\rangle - \sum_i e^{x_i}\right]$	$\langle \mathbf{u}+\mathbf{y}, 1-\log(\mathbf{u}+\mathbf{y})\rangle$
	Neg. bin.	$-\left[\langle \mathbf{y}, \mathbf{x}\rangle + \kappa \sum_i \log\{1-e^{x_i}\}\right]$	$\left\langle \mathbf{u}+\mathbf{y}, \log \frac{\mathbf{u}+\mathbf{y}}{\kappa+\mathbf{u}+\mathbf{y}}\right\rangle + \left\langle \kappa\mathbf{1}, \log \frac{\kappa}{\kappa+\mathbf{u}+\mathbf{y}}\right\rangle$
	Exp. Family	$-\left[\langle T(\mathbf{y}), \mathbf{x}\rangle - A(\mathbf{x}) + \log h(\mathbf{y})\right]$	$A^*(\mathbf{u}+T(\mathbf{y})) + \log h(\mathbf{y})$
Norm Penalties	ℓ_2	$\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$	$f^*(\mathbf{u}) = \begin{cases} 0, & \|\mathbf{u}\|_2 \le 1 \\ \infty, & \text{otherwise} \end{cases}$
	ℓ_1	$\|\mathbf{x}\|_1 = \sum_i \|x_i\|$	$f^*(\mathbf{u}) = \begin{cases} 0, & \|\mathbf{u}\|_\infty \le 1 \\ \infty, & \text{otherwise} \end{cases}$
	ℓ_∞	$\|\mathbf{x}\|_\infty = \max_{1\le i\le n} \|x_i\|$	$f^*(\mathbf{u}) = \begin{cases} 0, & \|\mathbf{u}\|_1 \le 1 \\ \infty, & \text{otherwise} \end{cases}$
	ℓ_1/ℓ_2	$\|\mathbf{x}\|_{1/2} = \sum_i \left\|x_{G_i}\right\|$	$f^*(\mathbf{u}) = \begin{cases} 0, & \max_i \left\|\mathbf{u}_{G_i}\right\| \\ \infty, & \text{otherwise} \end{cases}$
	nuclear	$\|\mathbf{X}\|_{\text{nn}} = \sum_i \sigma_i(\mathbf{X})$	$f^*(\mathbf{U}) = \begin{cases} 0, & \|\mathbf{U}\|_{\text{op}} \le 1 \\ \infty, & \text{otherwise} \end{cases}$
	operator	$\|\mathbf{X}\|_{\text{op}} = \max_{1\le i\le k} \sigma_i(\mathbf{X})$	$f^*(\mathbf{U}) = \begin{cases} 0, & \|\mathbf{U}\|_{\text{nn}} \le 1 \\ \infty, & \text{otherwise} \end{cases}$

Dual norm balls

Table: Convex conjugates of some frequently used losses and norms. κ denotes a known constant. The exponential family loss subsumes the other losses above but we keep them in their explicit form (without the log base measure term $\log h(y)$).

Convex envelopes

By construction, convex conjugate is a convex function regardless of whether the original function was convex.

Taking the convex conjugate twice yields a convex lower bound on the original function

$$f^{**}(\mathbf{x}) \leq f(\mathbf{x}).$$

Further, f^{**} is the tightest convex relaxation of the original function. The second convex conjugate is called **convex envelope**.

Example of convex envelopes – cardinality

Function which measures number of non-zero entries in a vector is usually called cardinality

$$\mathrm{card}(\mathbf{x}) = \sum_i [x_i \neq 0]$$

This is a non-smooth and a non-convex function.

$$\mathrm{card}^{**}(\mathbf{x}) = \|\mathbf{x}\|_1$$

The tightest convex relaxation of cardinality is ℓ_1 norm.

Example of convex envelopes – rank

Function which measures rank of a matrix can be seen as counting number of non-zero singular values. Let $\sigma_i(\mathbf{A})$ denote i^{th} singular value of a matrix

$$\mathrm{rank}(\mathbf{A}) = \sum_i [\sigma_i(\mathbf{A}) \neq 0]$$

This is a non-smooth and a non-convex function.

$$\mathrm{rank}^{**}(\mathbf{A}) = \|\mathbf{A}\|_* = \sum_i \sigma_i(\mathbf{A})$$

The tightest convex relaxation of rank is nuclear norm – sum of singular values.

Weak duality

The optimal value d^* of the Lagrange dual is guaranteed to be the best lower bound on p^* optimal value of the primal

$$d^* \leq p^*$$

This will hold even if the primal problem was not convex (e.g. integer program) and the gap

$$p^* - d^*$$

is called the **optimal duality gap**.

Strong duality

If the equality

$$d^* = p^*$$

holds then we say **strong duality** holds.

Conditions that are required for the strong duality to hold are called **constraint qualifications**

The most popular one is **Slater's condition** that simply requires that there is a point $\mathbf{x}$ in the relative interior of $\mathcal{D}$

$$\{\mathbf{x} \in \mathcal{D} | \forall y \in \mathcal{D}\ \exists z \in \mathcal{D}\ \exists \boldsymbol{\lambda} \in (0,1) \mathbf{x} = \boldsymbol{\lambda} y + (1 - \boldsymbol{\lambda}) z\}$$

such that $f_i(\mathbf{x}) < 0$ for $i = 1, \ldots, m$ and $h_i(\mathbf{x}) = 0, i = 1, \ldots, p$.

Optimality conditions – complementary slackness

If for a problem with strong duality $\mathbf{x}^*$ is a primal optimal and $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ is a dual optimal point then the **complementary slackness** conditions

$$\lambda_i^* f_i(\mathbf{x}^*) = 0, i = 1, \ldots, m$$

hold.

Stated differently

$$\lambda_i^* > 0 \rightarrow f_i(\mathbf{x}^*) = 0.$$

And also

$$f_i(\mathbf{x}^*) < 0 \rightarrow \lambda_i^* = 0.$$

Karush-Kuhn-Tucker optimality conditions

Assume $\mathbf{x}^*$ is a primal optimal and $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ is a dual optimal point of a problem with Lagrangian

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

The KKT optimality conditions are

$$\begin{aligned}
\nabla f_0(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(\mathbf{x}^*) + \sum_{i=1}^{p} \nu_i^* \nabla h_i(\mathbf{x}^*) &= 0 \\
f_i(\mathbf{x}^*) &\leq 0, i = 1, \ldots, m \\
h_i(\mathbf{x}^*) &= 0, i = 1, \ldots, p \\
\lambda_i^* &\geq 0, i = 1, \ldots, m \\
\lambda_i^* f_i(\mathbf{x}^*) &= 0, i = 1, \ldots, m
\end{aligned}$$

Duality, KKT conditions and algorithms

For a convex problem, if $(\mathbf{x}^*, \boldsymbol{\lambda}^* \boldsymbol{\nu}^*)$ satisfy the KKT conditions then these are primal and dual optimal points.

Hence a convex optimization algorithm can be constructed from KKT conditions.

Methods that work directly on both primal and dual problems are called primal-dual algorithms.

Sometimes dual problem is easier than the primal. From dual optimal $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ we can obtain primal optimal $\mathbf{x}^*$ by solving $\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = 0$

Subgradients

A vector $\mathbf{g} : \mathbf{R}^n \times 1$ is a **subgradient** of a convex $f : \mathbf{R}^n \to \mathbf{R}$ at $\mathbf{x} \in \mathbf{dom}\, f$ if for all $\mathbf{z} \in \mathbf{dom}\, f$

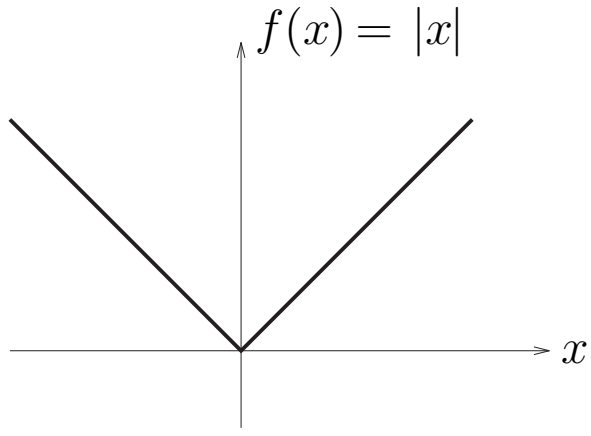
$$f(\mathbf{z}) \geq f(\mathbf{x}) + (\mathbf{z} - \mathbf{x})^T \mathbf{g}$$

If f is also differentiable at $\mathbf{x}$ then $\nabla f(\mathbf{x})$ is a subgradient.

A function is called **subdifferentiable** at $\mathbf{x}$ if there exists at least one subgradient.

Subdifferential

The set of subgradients of a function f at a point $\mathbf{x}$ is called **subdifferential** and is denoted $\partial f(\mathbf{x})$.



For $f(\mathbf{x}) = |\mathbf{x}|$

- $\mathbf{x} > 0 : \partial f(\mathbf{x}) = \{1\}$
- $\mathbf{x} < 0 : \partial f(\mathbf{x}) = \{1\}$
- $\mathbf{x} = 0 : \partial f(\mathbf{x}) = [-1, 1]$

Minimum of nondifferentiable functions

A point $\mathbf{x}^*$ is a minimizer of a convex function f if and only if $0 \in \partial f(\mathbf{x}^*)$

If function f is differentiable at $\mathbf{x}^*$ there is only one subgradient (the gradient) and it has to be 0 for $\mathbf{x}^*$ to be a minimizer.

Calculating subgradients

$$
\begin{aligned}
g = \alpha f, \alpha \geq 0 \quad &: \quad \partial g(\mathbf{x}) = \alpha \partial f(\mathbf{x}) \\
f = f_1 + \cdots + f_m \quad &: \quad \partial f = \partial f_1 + \cdots + \partial f_m \\
g(\mathbf{x}) = f(\mathbf{A}\mathbf{x} + \mathbf{b}) \quad &: \quad \partial g(\mathbf{x}) = \mathbf{A}^T \partial f(\mathbf{A}\mathbf{x} + \mathbf{b}) \\
f(\mathbf{x}) = \max_i f_i(\mathbf{x}) \quad &: \quad \partial f(\mathbf{x}) = \mathbf{Co} \cup \{\partial f_i(\mathbf{x}) | f_i(\mathbf{x}) = f(\mathbf{x})\}
\end{aligned}
$$

where

$$
\mathbf{Co}\{\mathbf{x}_i | i = 1, \ldots, n\} = \left\{ \sum_i \theta_i \mathbf{x}_i | \sum \theta = 1, \theta \geq 0 \right\}
$$

Subgradient optimization

A problem of form

$$\underset{\mathbf{x}\in\mathcal{D}}{\text{minimize}}\ f_0(\mathbf{x})$$

can be optimized by an algorithm that iterates updates

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{g}^{(k)}$$

where k denotes the iteration number and α_k is a step size.

Step sizes for subgradient optimization

- Constant step size: $\alpha_k = c$
- Constant step length: $\alpha_k = \frac{c}{\left\|\mathbf{g}^{(k)}\right\|_2}$
- Square summable but not summable: $\sum_{k=1}^{\infty} \alpha_k^2 < \infty, \sum_{k=1}^{\infty} \alpha_k = \infty$, for example $\alpha_k = \frac{a}{b+k}$ where $a > 0, b \geq 0$
- Nonsummable diminishing: $\lim_{k\to\infty} \alpha_k = 0, \sum_{k=1}^{\infty} \alpha_k = \infty$ for example $\alpha_k = \frac{a}{\sqrt{k}}$, where $a > 0$

Convergence rate

Subgradient algorithms are relatively slow. It takes $k = O(1/\epsilon^2)$ iterations to achieve $|f^{(k)} - f^*| < \epsilon$.

On the plus side, it is relatively easy to implement.

Projected subgradients

For a constrained problem

$$\begin{aligned} &\underset{\mathbf{x}\in\mathcal{D}}{\text{minimize}}\ f_0(\mathbf{x}) \\ &\text{subject to } \mathbf{x} \in \mathcal{C} \end{aligned}$$

a different update is used

$$\mathbf{x}^{(k+1)} = \Pi_{\mathcal{C}}[\mathbf{x}^{(k)} - \alpha_k \mathbf{g}^{(k)}]$$

where $\Pi_{\mathcal{C}}$ is a Euclidean projection operator.

$$\Pi_{\mathcal{C}}[\mathbf{x}] := \underset{y\in\mathcal{C}}{\operatorname{argmin}} \sum_i (y_i - x_i)^2$$

Performance of subgradient methods

If a single iteration is very fast it can be competitive with heavier methods.

The ADMM methods we saw earlier in class have comparable or better performance, but require derivation.

It minimizes the distance from the current solution to the optimal set – function value may oscillate.

Gaussian MRFs

One of the tractable versions of MRFs that can have an arbitrarily complicated graph.

Instead of working with a covariance matrix, we will work with the inverse covariance matrix, also called a **precision** matrix.

$$p(\mathbf{x}) = (2\pi)^{-d/2}(\det P)^{1/2} \exp\left\{-(1/2)(\mathbf{x}-\mu)^T P(\mathbf{x}-\mu)\right\}$$

Precision matrix and conditional independencies

It can be shown that

$$x_i \perp x_j | x_{\{k|k\neq i, k\neq j\}} \Leftrightarrow P_{ij} = 0.$$

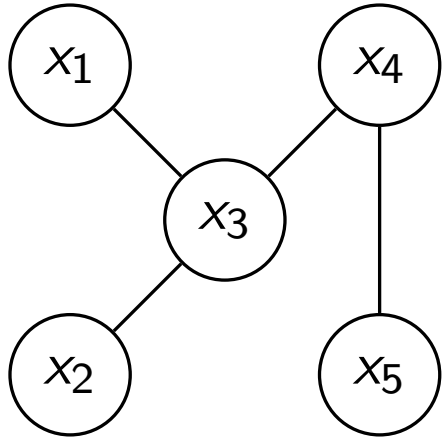
And as a consequence

$$x_i \perp x_j | x_{B(i)}, \forall j \notin B(i)$$

where $B(i) = \{k | P(i,k) \neq 0\}$.

For example gene i's expression is conditionally independent from all other genes given genes in $B(i)$.

Precision matrix and GMRF structure



$$P = \begin{bmatrix} \bullet & & \bullet & & \\ & \bullet & \bullet & & \\ \bullet & \bullet & \bullet & \bullet & \\ & & \bullet & \bullet & \bullet \\ & & & \bullet & \bullet \end{bmatrix}$$

Learning a precision matrix

The log likelihood function

$$\mathrm{LL}(P, \mu) = n/2 \log\det P - (1/2) \sum_{t=1}^{n} (\mathbf{x}^t - \mu)^T P (\mathbf{x}^t - \mu)$$

which yields the maximum likelihood estimate for P

$$P = \underset{Q \in \mathbf{S}_+^d}{\mathrm{argmax}} \log\det Q - \mathbf{Tr}(Cov(\mathbf{x})Q)$$

Learning a sparse precision matrix

$$P = \underset{Q \in \mathbf{S}_+^d}{\operatorname{argmax}} \log\det Q - \mathbf{Tr}(Cov(\mathbf{x})Q) - \boldsymbol{\lambda} \|Q\|_1$$

This is sometimes called graphical lasso as well.

The dual problem

The primal

$$\underset{Q \in \mathbf{S}_+^d}{\text{minimize}} \; -\log\det Q + \mathbf{Tr}(Cov(\mathbf{x})Q) + \sum_{ij} \lambda |Q_{ij}|$$

We will use an auxiliary variable trick (like we did in ADMM)

$$\underset{Q,Z \in \mathbf{S}_+^d}{\text{minimize}} \; -\log\det Q + \mathbf{Tr}(Cov(\mathbf{x})Q) + \sum_{ij} \lambda |Z_{ij}|$$

$$\text{subject to } Q - Z = 0$$

The dual problem

Lagrangian

$$L(Q, Z, W) = -\log\det Q + \mathbf{Tr}(Cov(\mathbf{x})Q) + \sum_{ij} \lambda |Z_{ij}| + \mathbf{Tr}(W^T(Q - Z))$$

The dual

$$\begin{aligned} g(W) &= \inf_Q \inf_Z L(Q, Z, W) \\ &= \left(\inf_Q -\log\det Q + \mathbf{Tr}(Cov(\mathbf{x})Q) + \mathbf{Tr}(W^T Q) \right) \\ &+ \left(\inf_Z \sum_{ij} \lambda |Z_{ij}| - \mathbf{Tr}(W^T Z) \right) \end{aligned}$$

and in standard form after simplification

$$\begin{aligned} &\underset{W}{\text{minimize}} \ \log\det(W + Cov(\mathbf{x})) \\ &\text{subject to } |W_{ij}| \leq \lambda, \forall i, j. \end{aligned}$$

Projected subgradients for sparse precision matrix learning

Subgradient

$$\nabla \log\det(W + Cov(\mathbf{x})) = (W + Cov(\mathbf{x}))^{-1}$$

Projection operator

$$(\Pi[W])_{ij} := \begin{cases} W_{ij}, & |W_{ij}| \leq \lambda \\ \mathrm{sign}(W_{ij})\lambda, & \mathrm{otherwise} \end{cases}$$

Hence the optimization proceeds by using update

$$W^{k+1} = \Pi[W^k + \alpha_k (W + Cov(\mathbf{x}))^{-1}]$$

Upon termination we have dual optimal variable W^* we can solve $\nabla_Q L(Q, W^*, Z) = 0$ and obtain $Q = (W^*)^{-1}$.

Stare at some code ...

Book and Matlab package

Boyd S. and Vandenberghe L., Convex Optimization
`http://www.stanford.edu/~boyd/cvxbook/`

Matlab optimization package: CVX
`http://cvxr.com/cvx/`

Today

- Primal and dual problems
- Convex envelope and tight relaxations
- Subgradients and subgradient descent