COMP 790-124: Goals for today

- More notation: norms and inner products
- Constrained optimization and Lagrange multipliers
- Challenges in combining nonsmooth penalties
- Decomposing problems and Alternating Direction Method of Multipliers (ADMM)
- Fused Lasso illustration of ADMM

Norms

$$
\begin{aligned}
\ell_2 \quad & \|\mathbf{x}\|_2 = \sqrt{\textstyle\sum_i x_i^2} \\
\ell_1 \quad & \|\mathbf{x}\|_1 = \textstyle\sum_i |x_i| \\
\ell_\infty \quad & \|\mathbf{x}\|_\infty = \max_i |x_i| \\
\text{Frobenius} \quad & \|\mathbf{X}\|_F = \sqrt{\textstyle\sum_i \sum_j x_{i,j}^2}
\end{aligned}
$$

Using these we can write various costs more succinctly

$$
\begin{aligned}
\text{Linear Regression} \quad & \tfrac{1}{2}\|\mathbf{y} - \beta_0 - \mathbf{X}\beta\|_2^2 \\
\text{Ridge Regression} \quad & \tfrac{1}{2}\|\mathbf{y} - \beta_0 - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_2^2 \\
\text{Lasso} \quad & \tfrac{1}{2}\|\mathbf{y} - \beta_0 - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1 \\
\text{Elastic Net} \quad & \tfrac{1}{2}\|\mathbf{y} - \beta_0 - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1 + \mu\|\beta\|_2^2
\end{aligned}
$$

Bracket notation for inner products

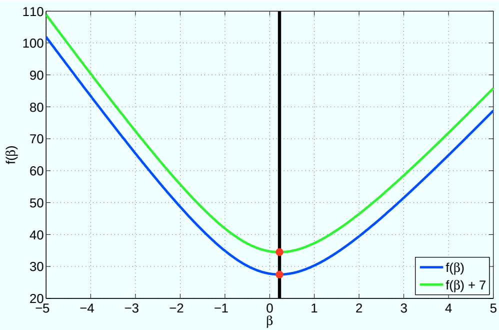
Another convenient piece of notation

$$\mathbf{x}^T\mathbf{y} = \sum_i x_i y_i = \langle \mathbf{x}, \mathbf{y} \rangle$$

and then

$$\begin{aligned} \|\mathbf{x}\|_2^2 &= \langle \mathbf{x}, \mathbf{x} \rangle \\ \|\mathbf{x} - \mathbf{y}\|_2^2 &= \|\mathbf{x}\|_2^2 - 2\langle \mathbf{x}, \mathbf{y} \rangle + \|\mathbf{y}\|_2^2 \end{aligned}$$

Completing the square: couple of observations



$$\min_{\beta} f(\beta) + c = c + \min f(\beta)$$

Adding a constant to an objective changes the optimal value by c.

$$\operatorname*{argmin}_{\beta} f(\beta) + c = \operatorname*{argmin}_{\beta} f(\beta)$$

Adding a constant to an objective does not change the optimal β.

Completing the square

Suppose we have to solve a problem

$$\operatorname*{argmin}_{\mathbf{x}} \underbrace{\frac{\rho}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \langle \mathbf{x}, \mathbf{u} \rangle + f(\mathbf{x})}_{\text{A}}$$

then we can claim that the optimal $\mathbf{x}$ is equal to

$$\operatorname*{argmin}_{\mathbf{x}} \underbrace{\frac{\rho}{2} \left\| \mathbf{x} - \mathbf{y} + \frac{1}{\rho}\mathbf{u} \right\|_2^2 + f(\mathbf{x})}_{\text{B}}$$

because difference between the objectives A and B is $-\frac{1}{2\rho} \|\mathbf{u}\|^2 + \langle \mathbf{u}, \mathbf{y} \rangle$ a constant with respect to $\mathbf{x}$.

Solving linear systems

Frequently problem of solving a linear systems of equations

$$\mathbf{A}\mathbf{x} = \mathbf{y}$$

is cast as optimization problem

$$\underset{\mathbf{x}}{\text{minimize}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$$

And we already know one way of solving this problem.

Coordinate descent using following updates[1] :

$$x_i = \frac{\sum_i y_i^{-j} a_{i,j}}{\sum_i a_{i,j}}$$

This method is also known as Jacobi method.

[1] $y_i^{-j} = y_i - \sum_{k \neq j} a_{i,k} x_k$

Solving linear systems

Frequently problem of solving a linear systems of equations

$$\mathbf{A}\mathbf{x} = \mathbf{y}$$

is cast as optimization problem

$$\underset{\mathbf{x}}{\text{minimize}} \, \frac{1}{2} \left\| \mathbf{A}\mathbf{x} - \mathbf{y} \right\|_2^2$$

and its solution can be obtained by setting gradient of objective to zero

$$\nabla \left\| \mathbf{A}\mathbf{x} - \mathbf{y} \right\|_2^2 = (\mathbf{A}\mathbf{x} - \mathbf{y})^T \mathbf{A} = \mathbf{0}$$

giving us normal equations

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}.$$

In MATLAB this is even more succinctly written as

```
x = A\y
```

Solving pairs of linear systems by stacking

Sometime we might run into an optimization problem

$$\underset{\mathbf{x}}{\text{minimize}}\, \frac{1}{2}\left\|\mathbf{A}\mathbf{x}-\mathbf{y}\right\|_2^2 + \frac{\rho}{2}\left\|\mathbf{B}\mathbf{x}-\mathbf{z}\right\|_2^2$$

and this is equivalent to solving

$$\underset{\mathbf{x}}{\text{minimize}}\, \frac{1}{2}\left\|\begin{bmatrix}\mathbf{A}\\ \sqrt{\rho}\mathbf{B}\end{bmatrix}\mathbf{x} - \begin{bmatrix}\mathbf{y}\\ \sqrt{\rho}\mathbf{z}\end{bmatrix}\right\|_2^2$$

and in Matlab

```
x = [A;sqrt(rho)*B]\[y;sqrt(rho)*z]
```

Constrained optimization and Lagrange multipliers

The problems we looked at so far are unconstrained

$$\underset{\boldsymbol{\theta} \in \mathbf{R}^p}{\text{minimize}} \qquad f(\boldsymbol{\theta}).$$

However sometimes we might want to impose constraints on our problems, for example

$$\begin{aligned} \underset{\boldsymbol{\theta} \in \mathbf{R}^p}{\text{minimize}} \qquad & f(\boldsymbol{\theta}) \\ \text{subject to} \qquad & g(\boldsymbol{\theta}) = 0. \end{aligned}$$

How do we solve such a problem?

Optimization of unconstrained objective

Given a differentiable objective $f(\boldsymbol{\theta})$ a critical point satisfies

$$\nabla f(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_1} \\ \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_2} \\ \dots \\ \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_p} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} = \mathbf{0}_p.$$

One way to find $\boldsymbol{\theta}^*$ such that $\nabla f(\boldsymbol{\theta}^*) = \mathbf{0}_p$ is coordinate descent (ascent if we are maximizing instead of minimizing).

Optimization of a constrained problem

With constrained optimization problem we might not be able to find $\boldsymbol{\theta}^*$, such that $\nabla f(\boldsymbol{\theta}^*) = \mathbf{0}_p$ *and* $g(\boldsymbol{\theta}^*) = 0$.

Hence at optimum we might have to compromise by accepting $\nabla f(\boldsymbol{\theta}^*) = \mathbf{v} \neq \mathbf{0}_p$ but in return $g(\boldsymbol{\theta}^*) = 0$.

$$\nabla f(\boldsymbol{\theta}^*) = \begin{bmatrix} \frac{\partial f(\boldsymbol{\theta}^*)}{\partial \theta_1} \\ \frac{\partial f(\boldsymbol{\theta}^*)}{\partial \theta_2} \\ \dots \\ \frac{\partial f(\boldsymbol{\theta}^*)}{\partial \theta_p} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_p \end{bmatrix} = -\lambda \nabla g(\boldsymbol{\theta}^*).$$

Intuitively, when tweaking θ_i^*, for each unit of improvement in f we pay λ in constraint violation.

All that remains is to set the price, λ, that is *sufficient* to make the constraint satisfied.

Optimization of constrained problem

A naive algorithm that solves a constrained problem

$\lambda = 0$
repeat
    $\boldsymbol{\theta} = \mathrm{argmin}_{\boldsymbol{\theta}}\, f(\theta) + \lambda g(\theta)$
    if $g(\theta) > 0$ **then**
        $\lambda = \lambda + \epsilon$
    end if
    if $g(\boldsymbol{\theta}) < 0$ **then**
        $\lambda = \lambda - \epsilon$
    end if
until $|g(\boldsymbol{\theta})| < 10^{-12}$

The algorithm slowly adjusts the λ until the optimal solution satisfies the constraint.

Lagrangian

Given an optimization problem

$$\begin{aligned} \underset{\boldsymbol{\theta} \in \mathbf{R}^p}{\text{minimize}} \quad & f(\boldsymbol{\theta}) \\ \text{subject to} \quad & g_i(\boldsymbol{\theta}) = 0, i = 1, \dots n \\ & h_j(\boldsymbol{\theta}) \leq 0, j = 1, \dots m \end{aligned}$$

Lagrangian is a function

$$L(\boldsymbol{\theta}, \lambda, \mu) = f(\boldsymbol{\theta}) + \sum_{i=1}^{n} \lambda_i g_i(\boldsymbol{\theta}) + \sum_{j=1}^{m} \mu_j h_j(\boldsymbol{\theta})$$

with a requirement that $\mu_j \geq 0$.

The *prices* λ and μ are called dual variables, whereas $\boldsymbol{\theta}$ is a primal variable.

Lagrangian

More on Lagrangian's, primal and dual problem pairs, and more generally convex optimization in coming weeks.

For now, think of them as a means to get an unconstrained objective for a constrained problem.

An alternative for simple constraints

$$\begin{aligned} \underset{\boldsymbol{\theta} \in \mathbf{R}^p}{\text{minimize}} \quad & f(\boldsymbol{\theta}) \\ \text{subject to} \quad & \boldsymbol{\theta}_i \geq 0, i = 1, \ldots n. \end{aligned}$$

We can rewrite by putting $\exp \gamma_i$ in place of θ_i

$$\begin{aligned} \underset{\gamma \in \mathbf{R}^p}{\text{minimize}} \quad & f(\exp \gamma) \\ \text{subject to} \quad & \exp \gamma_i \geq 0, i = 1, \ldots n. \end{aligned}$$

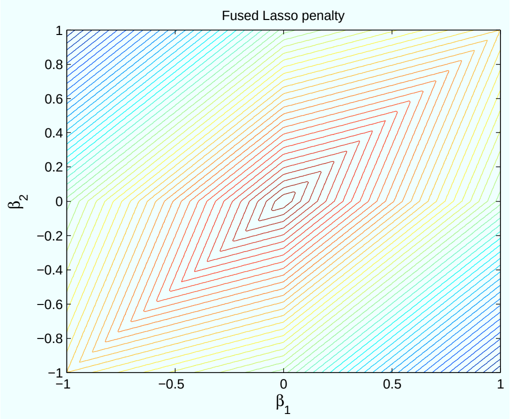
But, $\exp \gamma_i$ is always positive so the constraints are always satisfied and we can drop explicit constraints

$$\underset{\gamma \in \mathbf{R}^p}{\text{minimize}} \quad f(\exp \gamma)$$

which is an unconstrained problem.

Coordinate ascent does not always work

An example of a problem where coordinate ascent gets stuck, fused lasso [2].



$$-|\beta_1| - |\beta_2| - 2|\beta_1 - \beta_2|$$

Optimization of challenging objectives - dual decomposition

Remove coupling across non-smooth objectives by adding more variables.

Challenging

$$\underset{\beta \in \mathbf{R}^2}{\text{minimize}} \quad |\beta_1| + |\beta_2| + \kappa|\beta_1 - \beta_2|$$

Easier after adding an auxiliary variable

$$\begin{aligned} \underset{\beta \in \mathbf{R}^2, \delta \in \mathbf{R}}{\text{minimize}} \quad & |\beta_1| + |\beta_2| + \kappa|\delta| \\ \text{subject to} \quad & \delta = \beta_1 - \beta_2 \end{aligned}$$

Making friends with Lagrangians

An example of a separable objective tied through a constraint

$$\begin{array}{ll} \underset{\mathbf{x},\mathbf{y}}{\text{minimize}} & f(\mathbf{x}) + g(\mathbf{y}) \\ \text{subject to} & \mathbf{x} = \mathbf{y} \end{array}$$

and its Lagrangian

$$L(\mathbf{x}, \mathbf{y}, \lambda) = f(\mathbf{x}) + g(\mathbf{y}) + \lambda'(\mathbf{x} - \mathbf{y})$$

Augmented Lagrangian

$$AL(\mathbf{x}, \mathbf{y}, \lambda) = f(\mathbf{x}) + g(\mathbf{y}) + \sum_i \lambda_i (x_i - y_i) + \rho/2 \sum_i (x_i - y_i)^2$$

Alternating Direction Method of Multipliers[1] blueprint

$$AL(\mathbf{x}, \mathbf{y}, \lambda) = f(\mathbf{x}) + g(\mathbf{y}) + \sum_i \lambda_i (x_i - y_i) + \rho/2 \sum_i (x_i - y_i)^2$$

We will use k in superscript to denote state of variable after k^{th} iteration of algorithm.

$$\begin{aligned}
\mathbf{x}^k &= \operatorname*{argmin}_{\mathbf{x}} AL(\mathbf{x}, \mathbf{y}^{k-1}, \lambda^{k-1}) \\
\mathbf{y}^k &= \operatorname*{argmin}_{\mathbf{y}} AL(\mathbf{x}^k, \mathbf{y}, \lambda^{k-1}) \\
\lambda_i^k &= \lambda_i^{k-1} + \rho(x_i^k - y_i^k), i = 1, \ldots, n
\end{aligned}$$

where $\rho > 0$.

ADMM for fused lasso on 2 variables

The fused lasso reformulated problem

$$\begin{array}{ll} \underset{\beta \in \mathbf{R}^2, \delta \in \mathbf{R}}{\text{minimize}} & |\beta_1| + |\beta_2| + \kappa|\delta| \\ \text{subject to} & \delta - (\beta_1 - \beta_2) = 0 \end{array}$$

and its Augmented Lagrangian

$$\begin{aligned} AL(\beta_1, \beta_2, \delta, \lambda) \quad = \quad & |\beta_1| + |\beta_2| + \kappa|\delta| + \\ & \lambda(\delta - (\beta_1 - \beta_2)) + \rho/2(\delta - (\beta_1 - \beta_2))^2 \end{aligned}$$

Alternating Direction Method of Multipliers for fused lasso on 2 variables

Iterate updates

$$
\begin{aligned}
\beta_1^k &= \operatorname*{argmin}_{\beta_1} AL(\beta_1, \beta_2^{k-1}, \delta^{k-1}, \lambda^{k-1}) \\
\beta_2^k &= \operatorname*{argmin}_{\beta_2} AL(\beta_1^k, \beta_2, \delta^{k-1}, \lambda^{k-1}) \\
\delta^k &= \operatorname*{argmin}_{\delta} AL(\beta_1^k, \beta_2^k, \delta, \lambda^{k-1}) \\
\lambda^k &= \lambda^{k-1} + \rho(\delta^k - (\beta_1^k - \beta_2^k)),
\end{aligned}
$$

where $\rho > 0$.

ADMM for fused lasso on 2 variables

We will pause to notethe update for β_1:

$$\begin{aligned}
\beta_1^k &= \operatorname*{argmin}_{\beta_1} AL(\beta_1, \beta_2^{k-1}, \delta_1^{k-1}, \lambda^{k-1}) \\
&= \operatorname*{argmin}_{\beta_1} |\beta_1| - \lambda^{k-1}\beta_1 + \rho/2(\delta^{k-1} - (\beta_1 - \beta_2^{k-1}))^2
\end{aligned}$$

can be cast into a form that we already know how to solve.

Completing squares

$$\underset{\beta_1}{\operatorname{argmin}} \underbrace{|\beta_1| - \lambda\beta_1 + \rho/2(\delta - (\beta_1 - \beta_2))^2}_{\text{A}}$$

is equal to

$$\underset{\beta_1}{\operatorname{argmin}} \underbrace{|\beta_1| + \rho/2(\beta_1 - (\delta + \beta_2 + \lambda/\rho))^2}_{\text{B}}$$

Why did we bother?

$$\underset{\beta_1}{\operatorname{argmin}} |\beta_1| + \rho/2(\beta_1 - (\delta + \beta_2 + \lambda/\rho))^2$$

Because this is a Lasso problem in a single variable.

Hence we obtain a closed-form update:

$$\beta_1^k = S(\delta^{k-1} + \beta_2^{k-1} + \lambda^{k-1}\rho, 1/\rho)$$

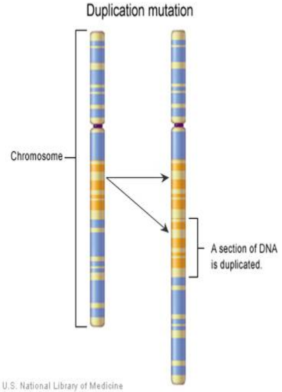
where

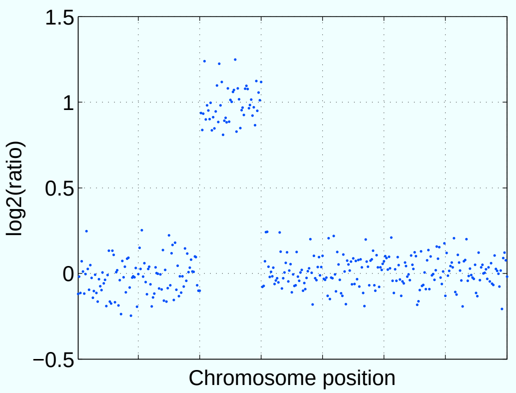
$$S(x, \lambda) \equiv \operatorname{sign}(x) \max(|x| - \lambda, 0)$$

Key steps in optimizing coupled penalties

1. Reformulate problem to remove sharing of variables between non-smooth parts of objective
2. Write down Augmented Lagrangian for your reformulated problem
3. Iterate the ADMM scheme

Application of fused lasso to CNV data





Fused Lasso Signal Approximator

$$\underset{\mathbf{x}}{\text{minimize}} \, \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 + \mu \|\mathbf{D}\mathbf{x}\|_1$$

where $\mathbf{y}$ is the vector we are trying to explain and $\mathbf{D}$ is a matrix that ties different entries in $\mathbf{x}$ together.

For example

$$\mathbf{D} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

gives term

$$\sum_{i=1}^{3} |x_i - x_{i+1}|$$

Demos

Full derivation of ADMM for FLSA

Full derivation is available on the course webpage along with the code for demos.

We did ...

- More notation: norms and inner products
- Constrained optimization and Lagrange multipliers
- Challenges in combining nonsmooth penalties
- Decomposing problems and Alternating Direction Method of Multipliers (ADMM)
- Fused Lasso illustration of ADMM

Solving linear systems

Frequently problem of solving a linear systems of equations

$$\mathbf{Ax} = \mathbf{y}$$

is cast as optimization problem

$$\underset{\mathbf{x}}{\text{minimize}}\ \frac{1}{2}\left\|\mathbf{Ax} - \mathbf{y}\right\|_2^2$$

and its solution can be obtained by setting gradient of objective to zero

$$\nabla \left\|\mathbf{Ax} - \mathbf{y}\right\|_2^2 = (\mathbf{Ax} - \mathbf{y})^T\mathbf{A} = \mathbf{0}$$

giving us normal equations

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}.$$

In MATLAB this is even more succinctly written as

```
x = A\y
```

Solving pairs of linear systems by stacking

Sometime we might run into an optimization problem

$$\underset{\mathbf{x}}{\text{minimize}}\ \frac{1}{2}\left\|\mathbf{A}\mathbf{x}-\mathbf{y}\right\|_2^2+\frac{\rho}{2}\left\|\mathbf{B}\mathbf{x}-\mathbf{z}\right\|_2^2$$

and this is equivalent to solving

$$\underset{\mathbf{x}}{\text{minimize}}\ \frac{1}{2}\left\|\begin{bmatrix}\mathbf{A}\\ \sqrt{\rho}\mathbf{B}\end{bmatrix}\mathbf{x}-\begin{bmatrix}\mathbf{y}\\ \sqrt{\rho}\mathbf{z}\end{bmatrix}\right\|_2^2$$

and in Matlab

```
x = [A;sqrt(rho)*B]\[y;sqrt(rho)*z]
```

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein.
Distributed optimization and statistical learning via the alternating direction method of multipliers.
Foundations and Trends in Machine Learning, 3(1):1–122, 2011.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight.
Sparsity and smoothness via the fused lasso.
Journal of the Royal Statistical Society Series B, 67(1):91–108, 2005.