This means we could think about

\[ q(\lambda) = \inf_{x} L(x, \lambda) \]

Recall: Lagrangian

Dual write \( x^* \) for a global min

\[ \arg\min_{x} f(x) \quad \text{st} \quad g(x) = 0 \]

Then we have

\[ q(\lambda) \leq f(x^*) \]

(This is fairly easy. Think about a single constraint: \( L(x, \lambda) = f(x) + \lambda g(x) \)

Now: either \( q(\lambda) = -\infty \) or it's finite.

- If it's finite, then \( \lambda g(x) \) must be bounded below; if the bound is 0, then \( g(x) = 0 \) or \( \lambda = 0 \).
- If \( g(x) = 0 \), then \( q(\lambda) \) can't be bigger than \( f(x^*) \).
- If \( \lambda = 0 \), same.
If bound is bigger than zero, we can't have a soln.

so $\lambda \geq (\infty)$ so...
Example: \textbf{Duals:}

\[
\min_{x} \frac{x^T x}{2} \quad \text{subject to} \quad Ax + b = 0
\]

\[
\Lambda : \quad x^T x - \lambda^T (Ax + b)
\]

\[
\text{Now:} \quad \inf_{x} f(x) \text{ occurs when} \quad x - \Lambda A^T = 0 \quad (\nabla_x f = 0)
\]

So:

\[
g(\lambda) = -\lambda^T AA^T \lambda - b^T \lambda
\]

Notice: \textbf{Max} \quad g(\lambda) \quad \text{occurs when} \quad

\[
AA^T \lambda - b = 0
\]

(Which we’ve seen before)
Idea:

- Form dual and solve
  \[ \max_{\lambda} q(\lambda) \]

- This isn't usually a good idea, cause we can't get the dual.

Idea:

- Go down in x, then up in x
- We've done something like that already! (AHM!)
- Sometimes called "Dual Ascent"
Recall:

Start: $x^0, \lambda^0, c^0 > 0$

$x^{k+1} = \arg \min_{x} A(x, \lambda^k, c^k) = f(x) - \lambda^{kT} g(x) + \frac{c^{kT}}{2} g(x) g(x)$

$\lambda^{k+1} = \lambda^k - \frac{c^k}{2} g(x^{k+1})$

$c^{k+1} = r c^k$

Notice

$A(x^{k+1}, \lambda^{k+1}, c^{k+1})$

$= A(x^{k+1}, \lambda^k, c^k) + \frac{r c^k}{2} g(x^k) g(x^k)$

and this is $\triangleright$ ne.
Another new idea: Splitting

Example

\[
\begin{align*}
\text{want to min } & \quad \|Ax - b\|_2^2 \\
\text{to tolerable, for very large } A & \quad \text{big } \|A\|
\end{align*}
\]

don't want to do the linear algebra

\[AA\] might be tough to work with

instead, write

\[
A = \begin{bmatrix}
A_1 \\
A_2
\end{bmatrix}, \quad b = \begin{bmatrix}
b_1 \\
b_2
\end{bmatrix}
\]

and consider

\[
\|A_1x_1 - b\|_2^2 + \|A_2x_2 - b_2\|_2^2
\]

\[
st \quad x_1 = x_2
\]

these two problems have solutions in the same place.
Now look at ALM

Aug. Lag. = \| A_1 x_1 - b_1 \|^2 + \| A_2 x_2 - b_2 \|^2
+ \lambda^T (x_1 - x_2)
+ \frac{c}{2} (x_1 - x_2)^T (x_1 - x_2)

We could "split" the step obtaining a minim

so

\begin{align*}
x_1^{k+1} &= \text{argmin}_{x_1} A(x_1, x_2^k, \lambda^k, c^k) \\
x_2^{k+1} &= \text{argmin}_{x_2} A(x_1^k, x_2, \lambda^k, c^k)
\end{align*}

and this is a linear system which is smaller than the \( \| A x - b \|^2 \) system.

\lambda^{k+1} = \lambda^k

{\text{as usual.}
An important working example

- Sparse regression

recall linear regression.

we have a dataset 

\((y_i, x_i)\)

and we wish to predict best estimate of future \(y\) from new \(x\).

one approach:

linear model:

\[ y_{pred}(x) = x_i \theta \]

\(\theta\) parameters

How to choose \(\theta\)?

min least-squares error
\[
Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}
\]

\[
X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix}
\]

and solve:

\[
\min_{\theta} \left[ Y - X \theta \right]^T \left[ Y - X \theta \right]
\]

which is solved by

\[
XX^T \theta = XY
\]

But there are problems with this story

imagine \( d \gg N \) many \( \theta \)!
(Ⅱ) even if \( d \ll N \), we could have 
\[ X^T X \] has small rank, and this must lead to large prediction errors.

(Ⅲ) imagine some of the \( x \)-components are irrelevant — then we want the corresponding \( \theta \)-components to be 0 — otherwise they contribute error.

Practical Q: can we force lots of \( \theta \)-components to be 0?
obvious strategy doesn't work

so solve

\[ \|Y - X\theta\|_2^2 + \frac{\lambda}{2} \theta^T \theta \]

Solve:

\[ (X^TX + \lambda I)\theta = X^TY \]

- Notice if \( \lambda \) is big enough, \( \theta \) has no solution
- Also I, but not convincingly
- We don't get 0's
- \( \ell^2 \) regularization

OR.
Very simple example

\[ y_i = c \]
\[ x_i = (1, \varepsilon_i) \]
a sample from noise, mean = 0, var = \( \sigma^2 \)

then \[ E[X^T X] = \begin{pmatrix} N & 0 \\ 0 & N\sigma^2 \end{pmatrix} \] and \[ E[X^T Y] = \begin{pmatrix} Nc \\ 0 \end{pmatrix} \]

so if we get \( (Nc) \) then \( \Theta = \begin{pmatrix} c \\ 0 \end{pmatrix} \)

But imagine \[ X^T Y = \begin{pmatrix} Nc \\ \varepsilon \end{pmatrix} \]

then \( \Theta = \begin{pmatrix} c \\ \frac{\varepsilon}{N\sigma^2} \end{pmatrix} \)

if we regularize to \( h^2 \) we get

\[ \begin{pmatrix} \frac{c}{1+\alpha} \\ \frac{\varepsilon}{N\sigma^2 + \alpha} \end{pmatrix} \] smaller, but not zero
Why:

- The penalty for $\theta_i$ small, but not 0 is very small.

We would like a small $\theta_i$ to be more expensive.

→ Rather than $\frac{1}{2} \theta^T \theta$

Use $\sum_i |\theta_i| = \|\theta\|_1$, absolute value, $L_1$ norm.
so we could use

$$\min_{\Theta} \| X \Theta - Y \|^2 + \alpha |\Theta|.$$  

(known as a \textit{lasso}).

But how do we find a \textit{min}? 

\textit{Notice:} \quad |\Theta| \text{ isn't differentiable at } \Theta_i = 0 \rangle.

\textit{Notice:} \quad \text{ignoring this will lead to problems } \rightarrow \text{ we won't get zeros!}