Constrained optimization:

- Two kinds of problem - equality constraints
  - neq constraints.

These are quite different, in important ways.

\[ g(x) = 0 \]
\[ \min f \text{ subject to } g = 0 \]
\[ \nabla f \text{ is normal to the surface } g = 0 \]

(because otherwise, we could move along \( g = 0 \) in a way that reduces \( f \)).
Inequality Picture:

- at (a), constraints are irrelevant; locally, problem involves min f(x) w/o constraints
- at (b), \( h_1 \) applies, but no others, so locally problem looks like min f(x) s.t. \( h_1(x) = 0 \)
- at (c), \( h_1, h_2, \ldots \) 

\[
\min f(x) \quad \text{s.t.} \quad h_1(x) = 0 \\
\quad h_2(x) = 0
\]

but: one step may mean picture changes!
Equality constraints:

- Simple picture: In 3D, one constraint $g(x) = 0$
- Minimize $f(x)$ subject to $g(x) = 0$
- Answer occurs at points where $\nabla f$ is normal to $\{g(x) = 0\}$
- Normal of implicit surface $g(x) = 0$ is $\nabla g$

$\nabla f = \lambda \nabla g$

some unknown constant
What if there are many constraints in $N \cdot \Omega$?

$g_1(x) = 0, \ g_2(x) = 0, \ etc.$

$\nabla f$ is normal

$\nabla f \in \text{span} \{ \nabla g_1, \nabla g_2, \nabla g_3, \ldots \}$

$\nabla f = \lambda_1 \nabla g_1 + \lambda_2 \nabla g_2 + \ldots$

equivalently, write $\begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$

$J_g = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}, & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1}, & \frac{\partial g_2}{\partial x_2} \\ \vdots & \vdots \end{bmatrix}$
then
\[ \nabla f = \lambda^T Jg. \]
This justifies writing the Lagrangian
\[ L = f - \lambda^T g. \]

at minimum:
\[ \nabla_\lambda L = \nabla f - \lambda^T Jg = 0. \]
\[ \nabla_x L = -g = 0. \]

These conditions must be true at a minimum.
I will deal with pages 6-10 later - DAF

V. Important special cases for constrained optimization

\[ \max \quad x^T A x \quad \quad \quad \text{st.} \quad x^T x = 1 \]

Lagrangian

\[ x^T A x - \lambda (x^T x - 1) \]

\[ A x = \lambda x \]

\[ \text{eigenvalue problem} \]
\begin{align*}
\max \quad & x^T A x \\ \\
\text{Lagrangian} \quad & x^T A x - \lambda (x^T B x - 1) \\ \\
\therefore \quad & A x - \lambda B x = 0
\end{align*}

Notice: \text{NOT the same as} \\
\begin{align*}
B^{-1} A x - \lambda x &= 0 \\
\text{because} \quad B^{-1} \text{ may not exist}
\end{align*}

\text{Any good Numerical linear Alg package can do these.}
\[ \min \frac{1}{2} x^T x \quad \text{s.t.} \quad Ax = b \]

(i.e. closest point on linear subspace to the origin)

Lagrangian:

\[
\begin{align*}
&\min \frac{1}{2} x^T x - \lambda^T (Ax - b) \\
\text{subject to} &\quad x - A^T \lambda = 0
\end{align*}
\]

So

\[ AA^T \lambda = b \]

Alg

solve linear system \( x \) for \( \lambda \), then sub for \( x \) in \( (\beta) \).
\[ \begin{align*}
\min & \quad \frac{x^T Ax}{2} + b^T x \\
\text{s.t.} & \quad Cx = d
\end{align*} \]

Lagrangian:
\[ x^T Ax + b^T x - \lambda^T (Cx - d) \]

\[ \nabla x J : \quad Ax + b - C^T \lambda = 0 \]

\[ \nabla \lambda J : \quad Cx - d = 0 \]

\[ \begin{bmatrix} A & -C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -b \\ d \end{bmatrix} \]

Solve this.
Notice how useful it has been to know the Lagrange multipliers.

Algorithms for other cases

Eliminating constraints

- Sometimes, we can parametrize the constraint set and move on that.
  - Not usually a good idea

Eg. \[ \text{min} \ x^2 + (y-10)^2 \]

subject to
\[ y - \sin x = 0 \]
Notice the rich supply of local min.
we could rewrite as
\[ \min x^2 + (\sin x - 10)^2 \]
\ without constraints
then try to min this

Notice when we do this, we are
confining our steps to the
constraint space:

- Problem - don't see large scale
structure of objective

- Equivalent:
  - take step \textit{Tangent} to
  \textit{constraint space}
  - project back.

(\textit{i.e. make up local parametrization})
For example:

\[
\min \quad x^T Ax + b^T x
\]

s.t. \quad \phi(x) = 0

\text{vector function.}

Now consider a step \( \Delta x \)

\[
\phi(x + \Delta x) \approx \phi(x) + J_\phi \cdot \Delta x.
\]

so we could try:

\text{Step:}

1) \quad \min \quad x^T Ax + b^T x

\min \quad (x_k + \Delta x)^T A (x_k + \Delta x) + b^T (x_k + \Delta x)

s.t. \quad J_\phi \cdot \Delta x = 0

2) correct by finding \( x_{k+1} \)

s.t. \quad \phi(x_{k+1}) = 0, \quad \text{Start search at} \quad x_k + \Delta x.
Again, not usually a great plan, because we may have a hard time taking big steps

**Quadratic penalty method**

\[
\begin{align*}
\min & \quad f(x) \\
\text{st} & \quad g(x) = 0
\end{align*}
\]

- approach by \( \min f(x) + \frac{c}{2} g^T g \)
- if \( c \) is big, this forces \( g^T g \) to be small
- advantage: we could take steps off the constraint space
- now it's unconstrained.
Disadvantages (big)

1) big \( c \) ⇒ some big terms in Hessian

\[ H = H_f + c \left[ \nabla J_g J_g^T + \ldots \right] \]

(\textit{so we should see terms that look like } \sum_k \frac{\partial g_k^2}{\partial x_i} \text{ on Hess diag})

2) at soln, \( g \) isn't zero

\[ \nabla f + c g J_g = 0 \]

\( \nabla f \) won't be zero, in general, so \( g \) can't be!
The method of multipliers
(also, augmented Lagrangian)
method.

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

form: Augmented Lagrangian

\[
A(x; \lambda) = f(x) - \lambda^T g(x) + \frac{c}{2} (g^T g)
\]

Now, assume we have an estimate \( \lambda^k \) of the hM's

Minimize \( A(x; \lambda^k) \) to get \( x^{(k)} \)
at $x^k$ we have

$$\nabla f(x^k) - \lambda^k g_j^T J_g + c g^T J_g = 0.$$  

Now, pattern match to conditions

$$\nabla x L = 0$$

$$\nabla x L = \nabla f - \lambda^T J_g$$

This suggests

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

Notice: we could have a solution if $g = 0$.!
ALM:

start w x₀, λ₀, c₀

min A(x, λ) = f(x) - λᵀg(x) + \frac{c₀}{2} gᵀg(x)

to get xₖ

λₖ₊₁ = λₖ - \frac{cₖ}{2} gᵀ(xₖ)

cₖ₊₁ = cₖ + \text{often 2}

Q: How do we know if it's converged?
A: In ALM, usually nothing to do - we don't reject steps - but issue for future.

Q: do we have Hessian probs?
A: No, because λ consts help.

(formally, there is some band on the c required to get exact soln.)
First glimpse of duality:

\[ \text{we have } \min f(x) \text{ st } g(x) = 0 \]
\[ L = f(x) - \lambda^T g(x) = L(x, \lambda) \]

We have solution when
\[ \nabla_x L = 0 \]
\[ \nabla_{\lambda} L = 0 \]
so solution is at a critical point of \( L \).

→ What kind of C.P.?

- fix \( \lambda = \lambda^* \), then \( L(x, \lambda) \) is (locally) at a min

- but for fixed \( x = x^* \), \( L(x, \lambda) \) is linear

- at \( \frac{x^c}{\text{solu}}, \frac{\partial L}{\partial \lambda} = 0 \)
i.e. think about

\[ H = \text{Hessian of } \mathcal{L} \text{ w.r.t. } x \text{ and } \lambda \]

at \( x^*, \lambda^* \), there are some divs 
(the x divs) \( S_x \) s.t.

\[ S_x^T H S_x \geq 0 \]

AND some divs (\( \lambda \) divs)

\[ \forall x^* \text{ s.t. } S_x^T H S_x = 0 \]

So \( x^*, \lambda^* \) must be a saddle point

\[ \cup \rightarrow x\text{-space} \quad \cap \rightarrow \lambda\text{-space} \]
This means we could think about

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

Notice:

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

This is fairly easy; consider a constraint, then

\[ L = f(x) + \lambda g(x) \]

For \( q(x) \) to be finite, we must have \( \lambda g(x) \) bounded below, if \( \lambda g(x) \) bound is greater than zero, no feasible point, so it's less than zero, but then \( \inf_{x} f(x) + \lambda g(x) \) is less than zero, \( \lambda \) and we're done.

Multiple dimensions follow.
This is powerful because we could consider

$$\max_{\lambda} q(\lambda) \leq f(x^*) .$$

If we have $q, \lambda^k, x^k,$ and $q(x^k) - f(x^k)$ is small,

$$f(x^k) - f(x^*)$$

is also small.

This could help us track progress.

Simple duals:

1. $-\frac{x^T A x}{2}$
2. $\min \frac{T x}{2}$
3. $x^T x - x^T (A x + b)$

$s.t. A x + b = 0$
Now \( \inf_{x} L(x, \lambda) \) occurs when

\[
x - A^T \lambda = 0
\]

so \( q(\lambda) = \lambda^T A A^T \lambda \)

\[
q(\lambda) = -\lambda^T A A^T \lambda - b^T \lambda \over 2
\]

(subs. into \( L \))

(it's not always this easy)

Notice \( \max_{\lambda} q(\lambda) \) occurs when

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

(i.e. at \( SDU \)).
Interesting example

**Problem:** find a PDF that has a fixed set of expectations (i.e., \( \sum_p E_p(f_i) = m_i \) known number)

while maximizing entropy.

→ useful modelling idea, we observe good estimates of some expectations in data, and want model to respect these. But we know nothing else, so max entropy.

so \[ \max -\int p \log p \, dx \]

st. \[ \int p \, dx = 1 \]
\[ \int p \cdot f_i \, dx = m_i \]

**Variational problem, with constraint.**
\[ \mathcal{L}(p) = -\int p \log p \, dx \]
\[ - \phi \int p \, dx - 1 \]
\[ - \sum_i \lambda_i \left[ \int p f_i \, dx - m_i \right] \]

# we want to form 2 gradients
\[ \nabla_x \mathcal{L} \text{ is easy} \]
\[ \nabla_p \mathcal{L} \text{ follows the case we saw earlier.} \]

(i.e. at \( p^* \), \( \left[ \frac{d}{d\varepsilon} \mathcal{L}(p^* + \varepsilon \varphi) \right]_{\varepsilon=0} = 0 \) \text{ for any } \varphi.)

\[ \left[ \frac{d}{d\varepsilon} \mathcal{L}(p^* + \varepsilon \varphi) \right]_{\varepsilon=0} = \int \varphi \left[ -\log p^* - 1 - \lambda_0 - \sum_i \lambda_i f_i \right] \, dx \]

this must be zero for any \( \varphi \),
so
\[ p^* \propto e^{-\lambda_0 - \sum_i \lambda_i f_i(x)} \]
This class of model used to be called a maximum entropy model.

Fitting:

(Old way)

1. adjust $\lambda_i$ so that

$$\int p^* f_i \, dx = m_i$$

2. and $\lambda_0$ so that

$$\int p^* \, dx = 1$$

$\frac{1}{N} \sum_i f(x_j)$ - an estimate from data of the expectation.
But imagine we have a model of the form

\[ p^*(x) = e^{-\lambda_0 - \sum_i \lambda_i f_i(x)} \]

and we fit with Max likelihood

We must solve

\[ \max \sum_j \log p^*(x_j) \]

(st. \( \int p^*(x) dx = 1 \))

(problem in \( \lambda_0, \lambda_i \))

Now \( \int p^*(x) dx = 1 = \int e^{-\lambda_0 - \sum_i \lambda_i f_i(x)} \) \( dx \)

\[ = e^{-\lambda_0} \int e^{-\sum_i \lambda_i f_i(x)} \) \( dx \]

So \( \lambda_0 = \log \left[ \int e^{-\sum_i \lambda_i f_i(x)} dx \right] = \log Z(\lambda, i) \)
So we must solve:

\[
\max_{\lambda_i} \sum_j \left[ -\log Z - \sum_i \lambda_i f_i(x_j) \right] = Q(\lambda)
\]

\[
\frac{\partial Q}{\partial \lambda_k} = \sum_j \left[ -\frac{1}{Z} \frac{\partial Z}{\partial \lambda_k} - \frac{Z f_k(x_j)}{Z} \right]
\]

\[
Z = e^{\lambda_0} = \int e^{-\sum_i \lambda_i f_i(x)} \, dx
\]

\[
\frac{\partial Z}{\partial \lambda_k} = -\int e^{-\sum_i \lambda_i f_i(x)} \, f_k(x) \, dx
\]

So \[-\frac{1}{Z} \frac{\partial Z}{\partial \lambda_k} = \int e^{-\lambda_0} \cdot e^{-\sum_i \lambda_i f_i(x)} \cdot f_k(x) \, dx\]

So we must solve:

\[
\int p^* \cdot f_k(x) \, dx = \frac{1}{N} \sum_j f_k(x_j)
\]

Exact expectations \quad \uparrow \quad \text{empirical expectations}