Lecture #23: Optimality Conditions

Last time we talked about a class of optimization problems:

"equality constrained quadratic programming"

we reasoned about their optimal solutions via closed form expressions, e.g.

$$(LS) \quad \min \|x\|^2 \quad \text{s.t. } Ax = b$$

$$\Rightarrow \quad x^* = A^T(AA^T)^{-1}b$$

Key Point: For more complex problems we won't always have closed-form solutions!

Instead we will use iterative methods to find an optima

<u>Main Question</u>: How will we know when to stop?

i.e. what properties hold at optimal x^*?

Before moving on to harder problems, let's think about <u>optimality conditions</u> for QP

First we need to understand how to take the derrivative in matrix-vector notation

<u>def</u>: the <u>gradient</u> of function $f(x_1, x_2, \dots x_d)$ is a d-dimensional vector

$$\nabla f = \begin{bmatrix} \frac{d}{dx_1} f(x_1, \dots x_d) \\ \frac{d}{dx_2} f(x_1 \dots x_d) \\ \vdots \end{bmatrix}$$

Let's revisit the example from last time:

$$f(x,y) = \underbrace{[x \ y]}_{z^T} \underbrace{\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{z}$$

Q: So what is the gradient?

$$\frac{\partial}{\partial x} f(x,y) = \frac{\partial}{\partial x}\left(2x^2 - 2xy + 2y^2\right) = 4x - 2y$$

Similarly $\frac{\partial}{\partial y} f(x,y) = 4y - 2x$

We can actually express the answer in matrix-vector notation

$$\nabla f \underset{= \; 2 \; A \; z}{=} 2 \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4x-2y \\ 4y-2x \end{bmatrix}$$

This is true more generally!

<u>Fact:</u> Let $f(\underset{\substack{\uparrow \\ \text{vector}}}{z}) = z^T A z$ then $\nabla f(z) = 2Az$

This should look sort of familiar from univariate calculus:

$$f(\underset{\substack{\uparrow \\ \text{scalar}}}{z}) = az^2 \Rightarrow \frac{d}{dz} f(z) = 2az$$

Let's do another important example

Fact: Let $f(z) = z^T b$. Then $\nabla f(z) = b$

Poll: Let $f(z) = b^T z$. What is $\nabla f(z)$?

(a) b^T (b) b

Now that we understand gradients, let's return to QP

$$\min_x \frac{x^T P x}{2} + q^T x = f(x)$$

$$\text{s.t. } Ax = b$$

Now the set of feasible points, defined as

$$\{x \mid Ax = b\}$$

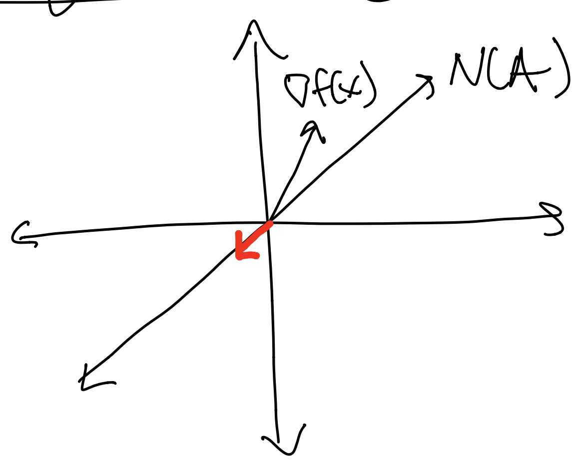
is a plane (not necessarily thru origin)

And the gradient is the direction of largest increase

$$f(x+\delta) \approx f(x) + \delta^T \nabla f(x)$$

Two things can happen:

<u>Case #1</u>: The gradient is <u>not</u> orthogonal to the directions you can move while maintaining feasibility



<u>Q2</u>: Is the current solution x optimal?

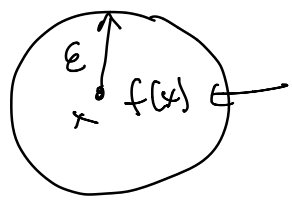
No, because you can move a small amount in N(A) and decrease the objective

Q3: Why did I say a <u>small</u> amount?

Because $f(x) + \delta^T \nabla f(x)$ is only a good approximation when δ is small

<u>Case #2</u>: The gradient is orthogonal to $N(A)$

<u>Lemma</u>: If $\nabla f(x) \perp N(A)$ then x is <u>locally optimal</u>



at least as small objective as anything else in ball

Let's figure out what this means for QPs specifically

$$\nabla f(x) = Px + q \perp N(A)$$

This is the same thing as

$$Px + q \in N(A)^{\perp}$$

But recall $C(A^T) = N(A)^{\perp}$, thus

<u>Lemma:</u> For equality constrained QP, a feasible point x (i.e. $Ax=b$) is <u>locally optimal</u> iff

$$Px + q \in C(A^T)$$

or alternatively if there is ν s.t.

$$Px + q = A^T\nu$$

We can actually incorporate feasibility into the condition too:

$$x \text{ is feasible and locally optimal} \iff \exists \nu \text{ s.t. } \begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \nu \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$$

Actually things are even nicer:

<u>Lemma:</u> For equality constrained QP with PSD P and a feasible point x

$$x \text{ is locally optimal} \iff x \text{ is globally optimal}$$

Let's see why this is true

Let x = feasible and locally optimal

Let $x+z$ = feasible

Now let's compare their objective values

$$f(x+z) = \frac{1}{2}(x+z)^T P (x+z) + (x+z)^T q$$

$$= \frac{1}{2} z^T P z + \underbrace{z^T P x + z^T q}_{\text{I claim this zero}} + f(x)$$

Let's see why: From the local optimality condition we have $\exists \nu$

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \nu \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$$

$$\Rightarrow \quad Px + A^T \nu = -q$$

$$\Rightarrow \quad Px = -q - A^T \nu$$

$$\Rightarrow \quad z^T P x = -z^T q \quad -\underbrace{\cancel{z^T A^T \nu}}_{\text{same as } \nu^T A z} \xrightarrow{} 0 \quad z \in N(A)$$

$$\Rightarrow \quad z^T P x + z^T q = 0$$

Now putting it all together

$$f(x+z) = \underbrace{\tfrac{1}{2} z^T P z}_{\geq 0 \text{ b/c } P \text{ is PSD}} + f(x)$$

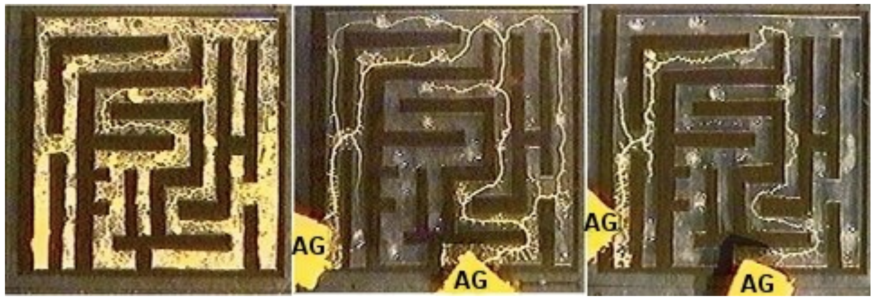
Thus x is globally optimal (though there could be others)

Now let's return to an earlier thread:

Nature can solve interesting optimization problems like LS

Anything else?

Slime mold can solve shortest path



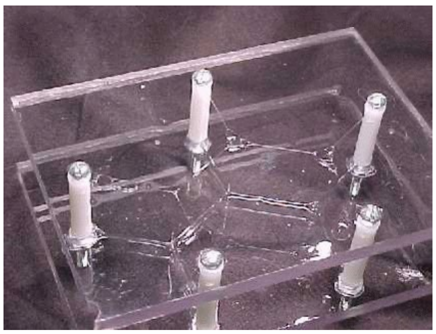
First it coats the entire maze, then settles on most efficient route to food source.

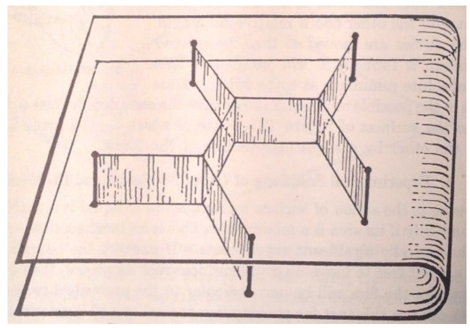
What next? Soduku?

<u>Another example:</u> Foams / Bubbles

= smallest surface area of given volume

What if we give it some landmarks to attach to?

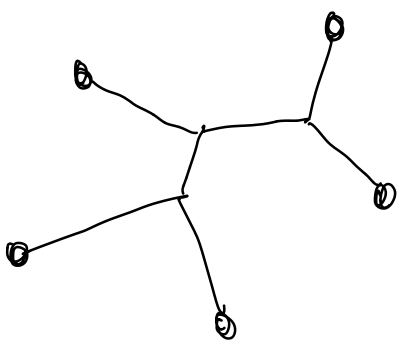




This looks like the Euclidean, 2-d steiner tree problem:

Given a set of nodes, connect them using minimum total length

e.g.



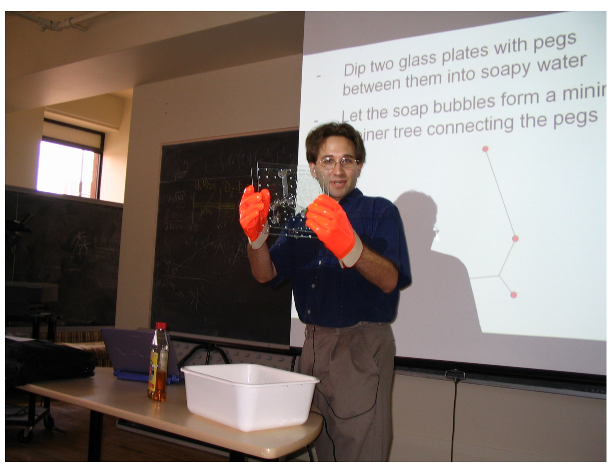
Natural Conjecture: Nature finds the optima

The trouble is this problem is "hard",
like "concave minimization"

Hmm. If my laptop can't always find the best solution, but soap does, should I switch CPUs?

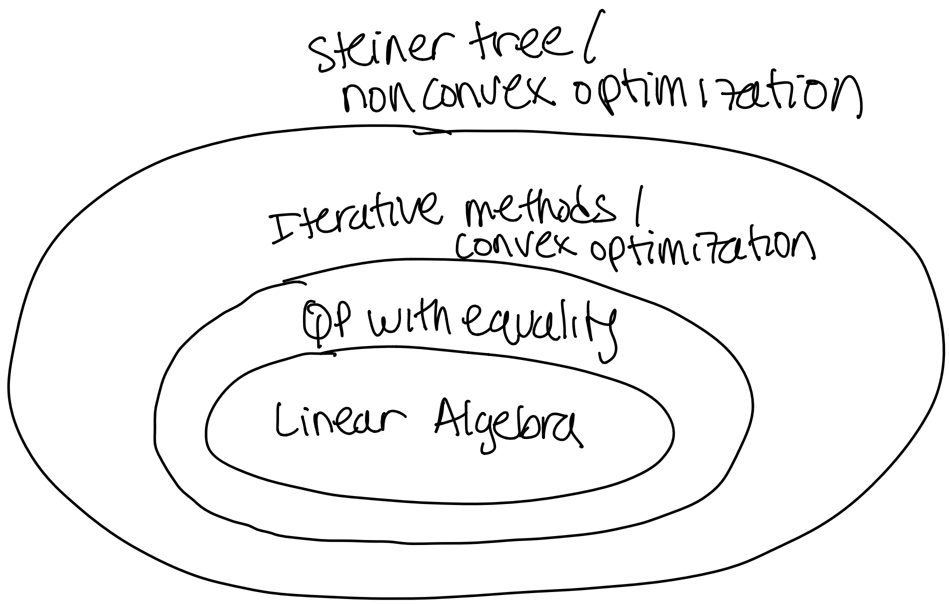
(Blog Post)

Well no, because it doesn't find a global optima



Scott Aaronson, ca. 2007

Let's put this all together:



Next time: QPs with inequalities, and applications