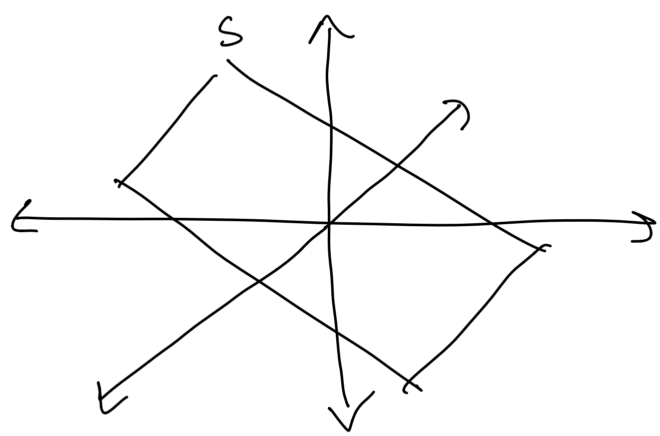
Lecture #24: QP with Inequality Constraints

Recall the <u>feasibility region</u> is the set of points we are optimizing over

e.g. $S = \{x \mid Ax = b\}$



What makes such regions particularly simple:

<u>Q:</u> when I start at a <u>feasible</u> x (i.e. $Ax=b$) what directions z can I move in while maintaining feasibility? $x+z \in S$

Last time we crucially used that the set of directions you can move in is $N(A)$

i.e. Suppose $x \in S$. Then

$$x + z \in S \iff z \in N(A)$$

Key Point: The directions you can move in does not depend on x

This will no longer be true when we allow more complex types of constraints

First a key definition:

def: Given two d-dimensional vectors x and y, say $x \leq y$ if

$$x_1 \leq y_1, \; x_2 \leq y_2 \; \ldots \; x_d \leq y_d$$

i.e. if x is component-wise less than or equal to y

Some Examples

(1) $\begin{bmatrix} 1 \\ 3 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 4 \end{bmatrix}$

(2) $\begin{bmatrix} 8 \\ 0 \\ -2 \end{bmatrix} \leq \begin{bmatrix} 9 \\ 0 \\ 0 \end{bmatrix}$

Q2: But what about $x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $y = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$?

Is $x \leq y$? Is $y \leq x$?

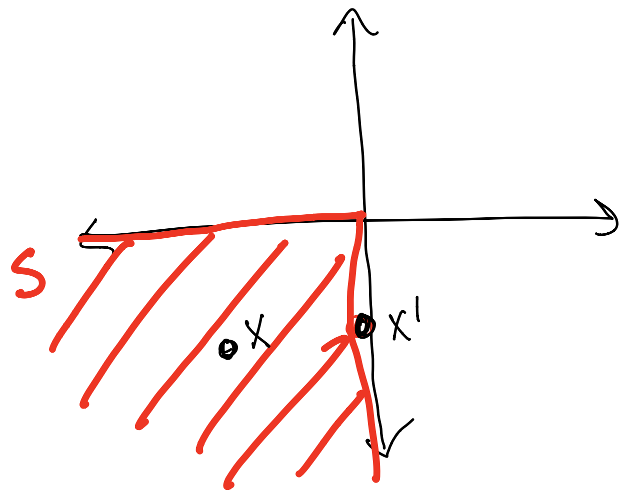
Actually, neither is true!

This is different than comparing 1-d vectors — i.e. scalars. For higher-d, not every pair of vectors can be compared one way or another

We will use this notation to define feasible regions

e.g. Consider $S = \{ x \mid x \leq \underset{\text{vector of 0s}}{0} \}$

Q3: What does S look like?



Notice: which directions you can move in (and still stay in S) indeed depends on where you start

Ultimately the feasible regions we will be interested in will look like

$$Ax \le b$$

meaning you take all x's so that the vector Ax is component-wise less than or equal to b

Poll Suppose $S = \{x \mid Ax \le b\}$ and

$$S' = \left\{x \;\middle|\; \begin{bmatrix} A \\ c^T \end{bmatrix} x \le \begin{bmatrix} b \\ b' \end{bmatrix}\right\}$$

which of the following is always true?

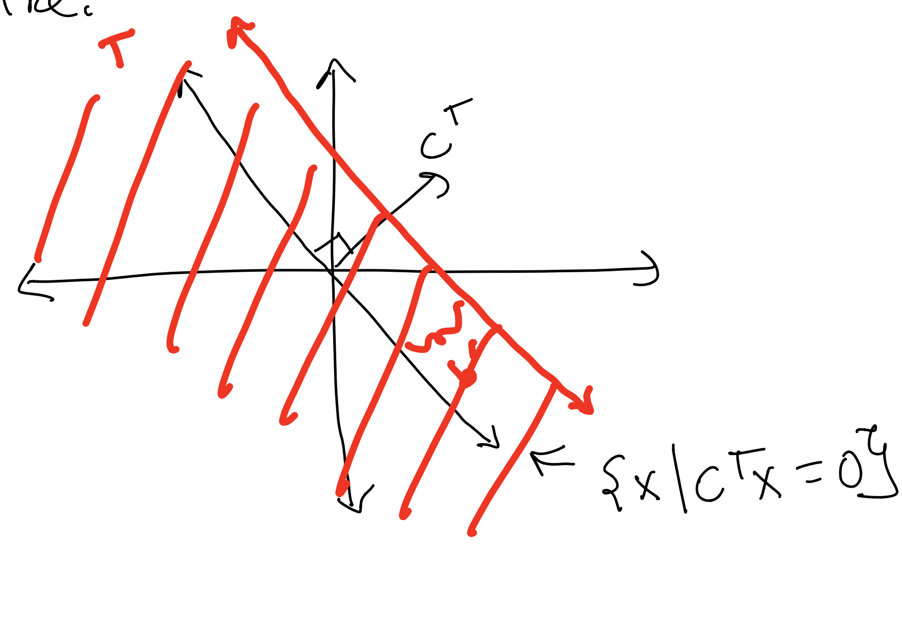
(a) $S' \subseteq S$ (b) $S' = S$ (c) $S \subseteq S'$ (d) none

This leads in to a helpful way to think about things:

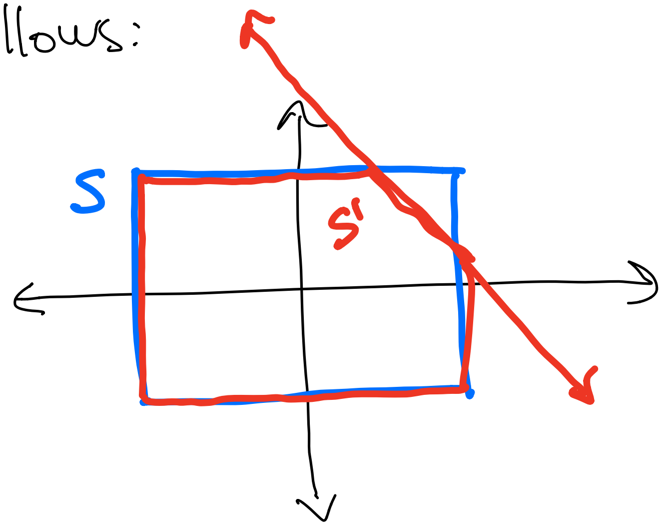
Q4: what does the set

$$T = \{x \mid c^T x \le b'\}$$

look like?



Now we can visualize how to get S' from S as follows:



So each row of the system

$$\underset{\substack{\uparrow \\ d\text{-dimensional}}}{Ax} \le b$$

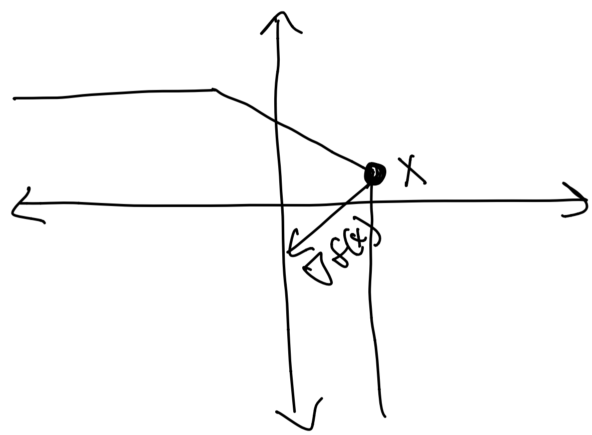
just successively cuts off some of the feasible region, starting from $\mathbb{R}^d$

Now we are ready to talk about QP with inequalities:

$$\min_x \frac{x^T P x}{2} + q^T x \quad \text{s.t. } Ax \le b$$

Optimality conditions are now more complicated

We won't formally cover them, but I'll give you some geometric intuition:

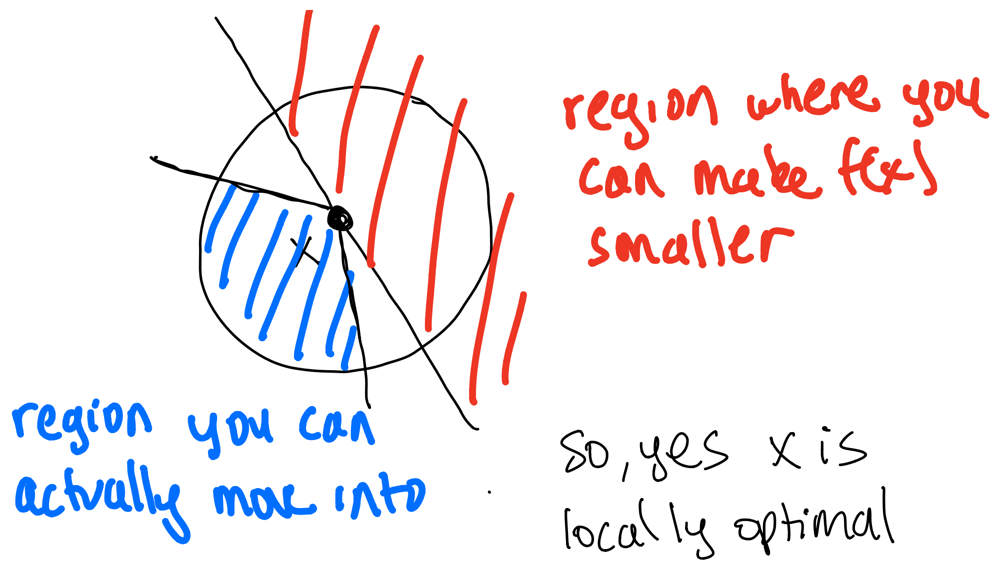


Q5: Is the point x optimal?

Recall, there are two notions of optimality:

(1) local optimality

Let's plot it



so, yes x is locally optimal

(2) Global optimality

Just like for QP with equalities:

local optimality ≡ global optimality

Later we will talk about general methods for finding an optimal solution

For now just think of QP with inequalities as a rich class of problems that is still solvable

Let's understand what we can express:

Key Example: Support vector machines

Imagine we have a collection of emails, each represented as a vector:

$$\begin{array}{c} w_1 \\ w_2 \\ \vdots \\ w_d \end{array} \begin{bmatrix} \\ \\ \boxed{} \\ \\ \end{bmatrix}$$

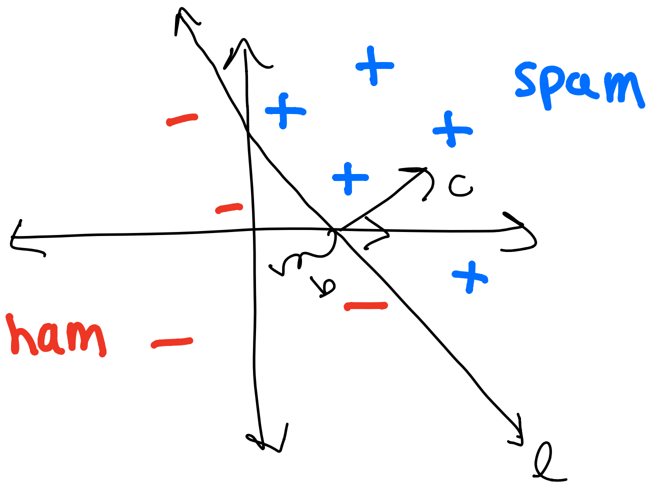
how many times does word w_i appear?

Goal: Classify emails as spam vs. ham, from labelled examples

Q6: What words might make you think an email is spam?

Prince

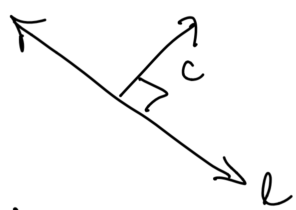
Geometrically what this might look like is the following:



what is a simple and accurate decision rule?

Q7: How do you describe it algebraically?

same as before. Let c be the vector $\perp$ to l



Then our rule is:

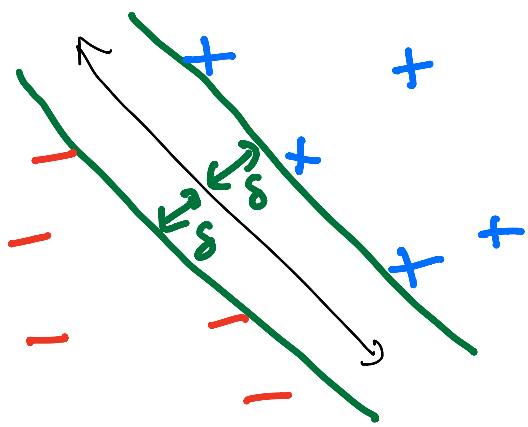
(1) If $c^T x + b > 0$, output $+$

(2) Else, output $-$

<u>Main Question</u>: How do we find this rule <u>automatically</u> (e.g. if you can't visualize the data)?

For now, we'll be happy to turn it into a QP

In fact what we really want to do is maximize the margin



Here we are thinking of c as a unit vector, but alternatively we can ask to solve:

$$\min_{c,b} \|c\|^2 \text{ s.t. for all examples } x \text{ with label } y \in \{\pm 1\} \text{ we have:}$$

$$y(c^T x + b) \geq 1$$

This is now a QP with inequalities

Recall, when we talked about optimality conditions, it was:

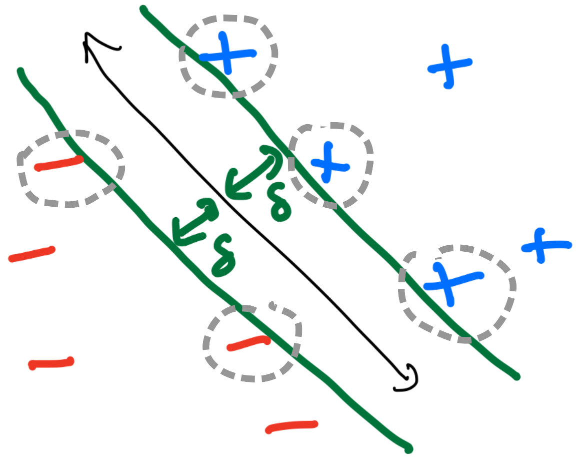
rows of $Ax \leq b$

that prevented us from doing better — i.e. we got stuck

In SVMs we have:

rows of constraints $\Longleftrightarrow$ labelled examples

Q8: what prevents us from increasing the margin?



These are called supporting vectors/ examples

Thus you can use:

optimality conditions $\Rightarrow$ interpretability

i.e. what examples caused me to learn the rule that I did?