Generic Structure learning problems

We have something we solve by

$$\hat{y} = \underset{y}{\operatorname{argmax}}\ \Phi(y, x)$$

x: data.

Many examples:

- matching
- inference for HMMs
- etc.

Now given

(y_i^*, x_i^*)

x_i^*: data

y_i^*: best soln

report Φ

This must be hard; but framework

assume $\phi(y, x) = \sum_i w_i \phi_i(y, x)$

↑ "features"

Q: how do we get $\underline{w}$?

have data so

$$w^T \phi(y_i^*, x_i^*) \geq w^T \phi(y, x_i^*)$$

This is NOT JUST 1 ineq.! TRUE for all y!

- observed y_i^* is better than all others.
- Typically, this isn't enough.

  if y "far" from y_i^*

  want $w^T \phi(y_i^*, x_i^*)$ a lot bigger than $w^T \phi(y, x_i^*)$.

so

$$w^T \phi(y_i^*, x_i^*) \geq w^T \phi(y, x_i^*) + \frac{\lambda}{2} \| y - y_i^* \|^2$$

or some norm.

---

true for any y.

This constraint might be violated.

consider

$$\xi_i(y) = \max \left[w^T \phi(y, x_i^*) + \frac{\lambda}{2} \| y - y_i^* \|^2 - w^T \phi(y_i^*, x_i) \, , 0 \right]$$

$= 0$ if constraint satisfied

> 0 otherwise.

We are interested in worst constraint violation

so

$$\xi_i = \max \left[\max_y \left(w^T \phi(y, x_i^*) + \frac{\lambda}{2} \| y - y_i^* \|^2 - w^T \phi(y_i^*, x_i) \right), 0 \right]$$

Now imagine we have many examples

- small w is a good idea
- So

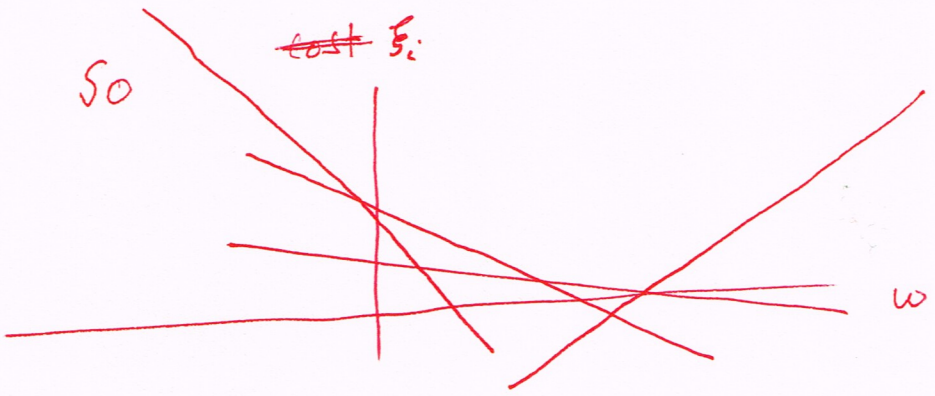
$$\min_{w} \quad \sum_i \xi_i + \frac{\mu}{2} \#w^T w$$

$$\text{st} \quad \xi_i = \max\left[\max_y \left(w^T \phi(y, x_i^*) + \frac{\lambda}{2} \|y - y_i^*\|^2 - w^T \phi(y_i^*, x_i) \right), 0\right]$$

Generically, called structure learning

Startlingly, it's convex

- envelope of convex fns.
- Notice
  - for any particular example
    - if the max doesn't change, its linear
    - and we're minimizing



- Structure learning is NASTY
- Variety of strategies
  - subgradient a/g.
    - w^0 = init
    - obtain subgradient ∂ cost
    - $w^{(n+1)} = w^{(n)} - \eta \partial$
- This will converge but
  - Subgradient is hugely expensive
    - must do inference at every example
- stochastic subgradient
  - obtain stoch est of subgradient (draw sample)
  - do descent
- Notice if there were few examples, you could get clever
  - eg. choose samp using ξ_i etc