Solving Very Large or Infinite Problems: Constraint Generation

Optimization - 10725

Carlos Guestrin

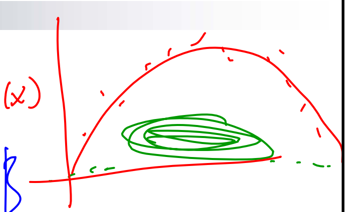
Carnegie Mellon University

February 11th, 2008

Weighted Least-Squares

- Least-squares regression problem:
  - Basis functions: $f_1(x), f_2(x), \ldots, f_k(x)$
  - Find coefficients: $w_1, \ldots, w_k$

$$\min_w \sum_j \left(t_j - \sum_i w_i f_i(x_j)\right)^2$$



- Some points are more important than others:
  - Weighted least-squares: $\alpha_j \leftarrow$ weight of point j

$$\min_w \sum_j \alpha_j \left(t_j - \sum_i w_i f_i(x_j)\right)^2$$

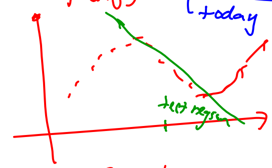
if care more about j then α_j is larger ...

Robust Least Squares*

→ train dist ← $p(x)$
→ test dist ← $q(x)$

Robust LS
→ robust to outliers
not today
→ robust to test set
today

- Weighted least squares:
- Test set distribution may be different from training set!
  - Must reweigh according to likelihood ratio: $\alpha_j = \frac{q(x_j)}{p(x_j)}$
- But what is the test set distribution???
- Don't want to commit!
  - Pick worst case weights!
  - Robust LS:



$$\max_{\alpha} \min_{w} \sum_j \alpha_j \left(t_j - \sum_i w_i f_i(x_j)\right)^2$$

$$\alpha \geq 0 \qquad \sum_j \alpha_j = 1$$

* many other optimization problems are called robust least-squares… :)

Optimization of Robust LS

$w' f_j = \sum_i w_i f_i(x_j)$

- Robust LS problem: $\max_{\alpha} \min_{w} \sum_j \alpha_j (t_j - w' f_j)^2$ $\quad \alpha \geq 0 \quad \sum_j \alpha_j = 1$
- For each set of weights, must solve weighted least squares:

give you $\hat{\alpha}$, $\quad w_{\hat{\alpha}} \leftarrow \arg\min_{w} \sum_j \hat{\alpha}_j (t_j - w' f_j)^2$

closed form, matrix ops.

- How do we find worst case weights?
  - Option B : guess weights, solve least squares, tweak weights,…

hope for the best...

Equivalent optimization problem

- Robust LS: $\max_{\alpha} \min_{w} \sum_j \alpha_j (t_j - w' f_j)^2$

  $\alpha \geq 0 \quad \sum_j \alpha_j = 1$

- Pushing min **w** into constraint: $\max_{\alpha, \varepsilon} \varepsilon$

  $\boxed{\varepsilon = \min_w \sum_j \alpha_j (t_j - w' f_j)^2}$

  $\alpha \geq 0 \quad \sum_j \alpha_j = 1$

- Non-linear constraint, give up!

Minimum over **w** as infinite constraints

- Non-linear min constraint: $\max_{\alpha, \varepsilon} \varepsilon$

  $\varepsilon = \min_w \sum_j \alpha_j (t_j - w' f_j)^2$

  $\sum_j \alpha_j = 1 \quad \alpha \geq 0$

  how many w? $\mathbb{R}^n$

- Infinite constraint set:

  $\varepsilon = \min_w \sum_j \alpha_j (t_j - w' f_j)^2 \Rightarrow \varepsilon \leq \sum_j \alpha_j (t_j - w' f_j)^2 \ \forall w$

  linear wrt α !! :)

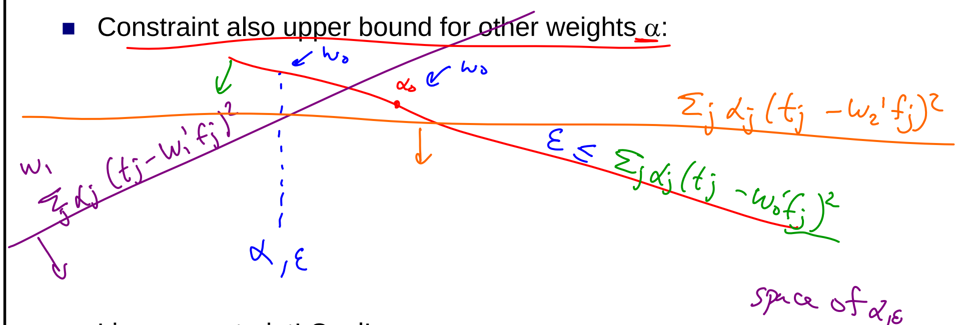
- Great! Had a non-linear constraint, now all I have are infinite constraints, for each alpha!

Constraints for one alpha, help with other alphas

- Suppose you have α_0^0, and introduce a constraint for some coefficients $\mathbf{w}_0$:

$$\varepsilon \leq \sum_j \alpha_j^0 \left(t_j - w_0' f_j\right)^2$$

- Constraint also upper bound for other weights α:



- Linear constraint! Cool!

A geometric view

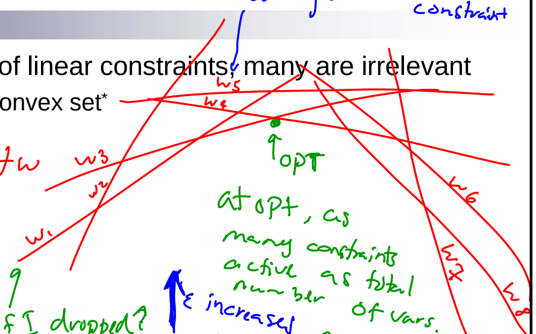
- We have an infinite number of linear constraints, many are irrelevant
  - Set of constraints forms a convex set*

always irrelevant constraint

$\max_{\alpha,\varepsilon} \; \varepsilon$

$\varepsilon \leq \sum_j \alpha_j (t_j - w' f_j)^2 \quad \forall w$

$\sum_j \alpha_j = 1 \quad \alpha \geq 0$



at opt, as many constraints active as total number of vars.

numvars = n+1

if I dropped? nothing happens!!

ε increases

at n+1 have non zero lagrange multipliers

- Linear program with one constraint per w
  - Still infinite…

→ if I could find the "optimal" n+1 constraints then → same solution with an LP with finite constraints

* There is better machinery to understand this, but more when we talk about convexity

Suppose we use a subset of the constraints

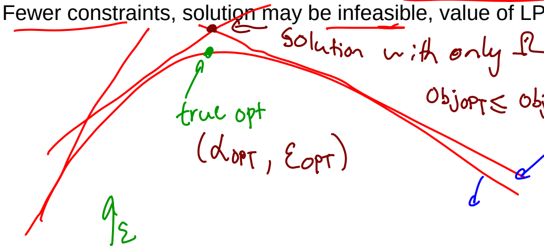
- What if we use a finite number of constraints $|\Omega| < \infty$ $w \in \mathbb{R}^n$ $\Omega \subset \mathbb{R}^n$
  - Set of constraints at a finite set of coefficients Ω

$$\max \varepsilon$$
$$\varepsilon \leq \sum_j \alpha_j (t_j - w'f_j)^2 \quad w \in \Omega$$
$$\alpha \geq 0 \quad \sum_j \alpha_j = 1$$

- Can solve with any LP solver!
- But, solution with subset of constraints may not be a solution to original problem
  - Fewer constraints, solution may be infeasible, value of LP too high… $(\alpha_\Omega, \varepsilon_\Omega)$

Solution with only Ω → infeasible wrt original problem

Ω:



$Obj_{OPT} \leq Obj_\Omega \Rightarrow \varepsilon_{OPT} \leq \varepsilon_\Omega$

infinite set of constraints

Active constraints

- Original LP with infinite constraints:

$$\max_{\alpha,\varepsilon} \varepsilon$$
$$\varepsilon \leq \sum_j \alpha_j (t_j - w'f_j)^2 \quad w \in \Omega$$
$$\alpha \geq 0 \quad \sum_j \alpha_j = 1$$

- How many variables? $n+1$
- How many active constraints at optimal solution? at most $n+1$

non zero lagrange multipliers

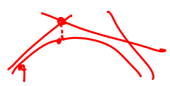
- So, if we knew set of active constraints at optimal solution Ω^*
  - Could discard all other constraints

finite set

$$\max_{\alpha,\varepsilon} \varepsilon \quad \varepsilon \leq \sum_j \alpha_j (t_j - w'f_j)^2 \quad w \in \Omega^* \quad \equiv \quad \max_\alpha \min_w \sum_j \alpha_j (t_j - w'f_j)^2$$
$$\alpha \geq 0 \quad \sum_j \alpha_j = 1 \qquad\qquad \alpha \geq 0 \quad \sum_j \alpha_j = 1$$

Active Constraints at Optimal Point

- Original problem: $\max_{\alpha,\varepsilon} \varepsilon$ $\quad \varepsilon \le \sum_j \alpha_j (t_j - w' f_j)^2 \quad w \in \Omega$ $\quad \alpha \ge 0 \quad \sum_j \alpha_j = 1$
- If we knew set of active constraints at optimal solution Ω^*
  - Could discard all other constraints
  - Solution will be feasible with respect to original problem $\Rightarrow$ $\varepsilon = \min_w \sum_j \alpha_j (t_j - w' f_j)^2$

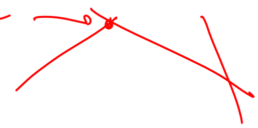


- Consider some set of constraints Ω:
  - Too few, infeasible solution: $(\varepsilon_\Omega, \alpha_\Omega) \Rightarrow \varepsilon_\Omega > \varepsilon_{OPT}$
  - Just right, feasible solution: $(\varepsilon_\Omega, \alpha_\Omega) \Rightarrow \varepsilon_\Omega = \varepsilon_{OPT}$

Constraint Generation

this weighted LS will get w that is tangent to true surface at $\alpha = \alpha_\Omega$

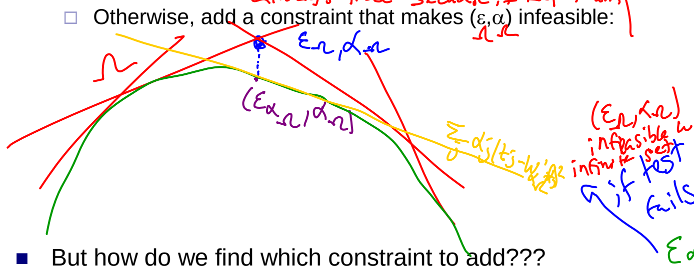
- Start with some finite set of constraints Ω
  - Solve LP, obtain α_Ω, ε_Ω



must check if this constraint holds

- Check is $(\varepsilon_\Omega,\alpha_\Omega)$ is feasible for infinite constraints: $\varepsilon_\Omega \overset{?}{=} \min_w \sum_j \alpha_{\Omega j} (t_j - w' f_j)^2$
  - If feasible, done! $\alpha_\Omega \ge 0 \quad \sum_j \alpha_{\Omega j} = 1$ always true because I kept them
  - Otherwise, add a constraint that makes $(\varepsilon_\Omega,\alpha_\Omega)$ infeasible:

solve weighted least squares with weight α_Ω $\Downarrow$ obtain opt for these weights: $(w_{\alpha_\Omega}, \varepsilon_{\alpha_\Omega})$ $\quad \varepsilon_{\alpha_\Omega} \overset{?}{=} \varepsilon_\Omega$



- But how do we find which constraint to add???

Separation Oracle for Robust LS

- Original problem: $\max_{\alpha, \varepsilon} \varepsilon$ $\quad \varepsilon \le \sum_j \alpha_j (t_j - w' f_j) \;\; \forall w$ $\quad \alpha \ge 0 \quad \sum_j \alpha_j = 1$
- Is $(\varepsilon_\Omega, \alpha_\Omega)$ feasible?
  - infeasibility ➔ ε_Ω too high for this particular α_Ω
- What's the smallest possible ε?

$$\varepsilon = \min_w \sum_j \alpha_{\Omega j} (t_j - w' f_j)^2$$

solution of weighted least squares with weights α_Ω

- Standard weighted LS!
  - If result is ε_Ω then we are done!
  - Otherwise found a violated constraint ← $\varepsilon \le \sum_j \alpha_j (t_j - w_{\alpha_\Omega}' f_j)^2$

Constraint Generation: I is large The General Case

- Given an LP with (possibly infinitely) many constraints: $\max c'x$ $\quad a_i' x \le b_i \;\; \forall i \in I$
- Start with some subset of the constraints $\Omega \subset I$
- Solve LP to find a solution with new subset of the constraints: $\max c'x$ $\quad a_i' x \le b_i \;\; \forall i \in \Omega$ (solution $\hat{x}$)
- Separation oracle:
  - If $\hat{x}$ is feasible: $\Rightarrow a_i \hat{x} \le b_i \;\; \forall i \in I$, done!!
  - If x is infeasible: $\Rightarrow \exists i \in I$, such that $a_i' \hat{x} > b_i$

general definition of a separation oracle: Return some violated i; "best" oracle: most violated constraint: $\max_i a_i' \hat{x} - b_i$

iterate

- Add violated constraint to set $\Omega \leftarrow \Omega \cup \{i\}$
- (It is also possible to remove (some or all) inactive constraints, in addition to adding violated constraints)
  - Makes LP solver step faster
  - But requires more outer loop iterations
  - Trade-off is application specific

Are we there yet?

- When do we stop?
- Solve with infinite set of constraints:
  - Obtain (ε_{OPT},α_{OPT})

$\max \varepsilon$
$\varepsilon \leq \sum_j \alpha_j (t_j - w' f_j)^2 \quad \forall w$
$\alpha \geq 0 \quad \sum_j \alpha_j = 1$

- Solve with constraints Ω
  - Obtain (ε,α) Ω
- Optimizing subset of constraints, same objective
  - $\varepsilon_{opt} \leq \varepsilon_\Omega$
- If we get any feasible point with infinite constraints $(\varepsilon_f, \alpha_f)$
  - E.g., $(\varepsilon_{\alpha_\Omega}, \alpha_\Omega) \leftarrow$ obtain with weighted least squares $\quad \varepsilon_f \leq \varepsilon_{OPT}$
- Bound on how far we are from optimal solution:

$\varepsilon_{\alpha_\Omega} \leq \varepsilon_{OPT} \leq \varepsilon_\Omega$,

stop when $\varepsilon_\Omega - \varepsilon_{\alpha_\Omega}$ is small
return $(\varepsilon_{\alpha_\Omega}, \alpha_\Omega)$

Bound on optimal solution - General case

- Problem with many constraints:

$\max c'x$
$a_i' x \leq b_i \quad \forall i \in I$

- Some relaxation:
  - E.g., only subset of constraints

$\Omega \subset I$
$\hookrightarrow x_\Omega$

- If you can obtain some feasible point for the original problem:

problem specific issue $\quad x_f$

- Bound on the optimal solution: $c' x_f \leq c' x_{OPT} \leq c' x_\Omega$

Practicalities of Constraint Generation

- Constraint generation converges in a finite number of iterations if the original set is finite
  - Can't guarantee fast rate, similar to simplex algorithm
  - Infinite case: will get arbitrarily close, but not necessarily to the optimum
- Idea of using relaxations to obtain bounds is very useful in general
  - E.g., useful in duality (more later in the semester)
- Separation oracle:
  - Must find some violated constraint
  - If we find most violated constraint, usually faster
  - Also very useful for proving that LPs can be solved in polytime (ellipsoid algorithm, more later)
- Constraint generation is extremely useful in practice
  - Often, e.g., robust LS, we have a poly-time separation oracle, even if there are exponentially or infinitely many constraints
  - Even if polynomially many constraints, a fast oracle can make constraint generation faster than using a standard solver
- Constraint generation can be useful for solving general convex problems, not just LP
- Remember: most LP solvers allow you to start from previous solution
  - (the one found with fewer constraints)
  - Make sure you do this, otherwise approach will be much much much slower