A Randomized Algorithm For LP Feasibility

from *A Simple Polynomial-time Rescaling Algorithm for Solving Linear Programs* by John Dunagan and Santosh Vempala

Presented by Andrew Menard

September 15, 2003

This presentation is based on: Dunagan, John, and Santosh Vempala. "A Simple Polynomial-Time Rescaling Algorithm for Solving Linear Programs." MIT Mathematics Department, 2003.

Outline

- The Perceptron Algorithm
- Condition Number
- Randomized Rescaling
- The Randomized Rescaling Algorithm

Linear Feasibility

- $Ax \geq 0, x \neq 0$
- Can be used to solve LP, also independently useful
- We want something strictly feasible, so we will be looking for $Ax > 0$

Classic Perceptron Algorithm

- Let $x_0 = 0$
- If x_i satisfies $Ax \geq 0, x \neq 0$, quit
- Otherwise find an unsatisfied constraint $a_j x \leq 0$
- Set $x_{i+1} = x_i + \bar{a}_j$

Runtime of Classic Perceptron

- If there exists a solution, there must be a solution z such that there is a ball of radius 1 around z, and z is the closest point to the origin with that property.
- Thus, $\bar{a_j} z \geq 1$
- Consider the distance from x_{i+1} to z
- $||(z - x_{i+1})|| = ||z|| - 2z(x_i + \bar{a_j}) + ||(x_i + \bar{a_j})||$
- $= (z - x_i)^2 - 2z\bar{a_j} + 2x\bar{a_j} + \bar{a_j}^2$
- But we know $x\bar{a_j} < 0$ by construction, and $z\bar{a_j} > 1$ by construction, so
- $\leq ||(z - x_i)|| - 1$
- Thus the algorithm terminates in no more than z^2 steps, since the norm is always positive.

Condition Numbers

- Thus, when using this algorithm, the difficulty is related to the distance from the origin to the point z.
- We define the condition number ρ of the problem as $\rho = \frac{1}{||z||}$
- Thus the perceptron algorithm terminates in $\frac{1}{\rho^2}$ iterations.
- What if we had a way of modifying a problem to increase ρ, thus decreasing the runtime?
- Idea: find a point near the feasible region, rescale so area near the point (which should include feasible region) expands, while area far from the point contracts.

Rescaling algorithm

- Idea: start with a random unit vector, and move it in the direction of rows of A; if it starts close to feasible region it should stay close to feasible region.
- Let x_0 be a random unit vector
- Repeat at most $1024n^2 \log n$ times:
  - If there exists a row $\bar{a}$ such that $\bar{x}_i\bar{a} < \frac{-1}{32n}$, set $x_{i+1} = x_i - (\bar{a}x_i)\bar{a}$
  - If $x = 0$ restart
- If there still exists a row a with $\bar{a}\bar{x} < \frac{-1}{32n}$, restart

What the heck is this doing?

- Suppose we have the center point z, at a distance 1 with a ball of radius ρ about it in the feasible cone.
- We start with a random vector, which has $zx \geq \frac{1}{\sqrt{n}}$ with probability $\frac{1}{4}$
- As we update x, zx does not decrease:
- $(x - (x\bar{a})\bar{a})z = xz - (x\bar{a})(\bar{a}z) \geq xz$
- Therefore if we started close to z, we stay close.
- Further, because this product cannot decrease, x cannot become zero in this case.
- But the magnitude of x will decrease at each step, so we must terminate:
- $(x - (x\bar{a})\bar{a})^2 = x^2 - (\bar{a}x)^2 \leq x^2(1 - \frac{1}{1024n^2})$
- So after the specified number of iterations, we have a contradiction between the sizes of x^2 and xz, so we must have terminated.

OK, what now?

- So, with probability at least $1/4$ we picked a starting x that terminates with a "good" result; it is no more than $\frac{1}{32n}$ in violation of any constraint, and it is close to z.
- And with probability no more than $3/4$, we have a point that terminates with $x = 0$ or takes too many iterations, in which case we try again, or it ends with a point that is no more than $\frac{1}{32n}$ in violation of any constraint, but might be far from z.
- Now we rescale A; our new problem is $A = A(I + \bar{x}\bar{x}^T)$
- In the good case, ρ increases a lot
- In the bad case, ρ decreases a little.

Results of rescaling

- Prior to the rescaling step, A has condition number ρ and center z
- After, it will have a new condition number of at least ρ'
- Consider the point $z' = z + \alpha(z\bar{x})\bar{x}$
- The radius of a ball around this point in the cone is $\rho' \geq min_i \frac{\bar{a}_i z'}{|z'|}$
- Expand this, and do some algebra.
- We find that in the good case, $\rho' \geq \rho(1 + \frac{1}{4n})$
- And in the bad case, $\rho' \geq \rho(1 - \frac{1}{16n})$
- Observation: If we do this rescaling repeatedly, we expect to see the good case with probability at least $\frac{1}{4}$, so we expect to see ρ growing if we do this repeatedly.

Detailed probabilitites

- Let $X_i = 1$ if we have the good case in iteration i, and 0 otherwise.
- Let $Y_i = \sum_0^i$
- Then clearly $E[Y_i] \geq i/4$
- And further, by the Chernoff bound, $Pr(Y_i < (1-\epsilon)E[Y_i]) \leq e^{-\epsilon^2 E[Y_i]/2}$
- Consider $i = 2048n \log 1/\rho$ and $\epsilon = 1/16$. Then $e^{-\epsilon^2 E[Y_i]/2} \leq e^{-n}$
- Thus, with probability $1 - e^{-n}$, Y_i is within ϵ of its expectation, so $\rho_i \geq \rho(1 + \frac{1}{4n})^{Y_i}(1 - \frac{1}{16n})^{i - Y_i}$
- Expanding all of this mess out, we get that with probability $1 - e^{-n}$, $\rho_i \geq \frac{1}{4n}$
- And once it has grown this large, solving the original problem is easy!

Bringing it all together

- Set $B = I, \sigma = \frac{1}{32n}$
- Perceptron phase
  - Let x be the origin
  - Repeat at most $16n^2$ times: if there exists a row a with $ax \leq 0$, $x = x + \bar{a}$
- If $Ax \geq 0$, output solution Bx
- Improvement phase
  - Let x be a random unit vector
  - Repeat at most $ln(n/\sigma^2)$ times:
    * If there exists a row a with $\bar{a}\bar{x} < -\sigma$, $x = x - (\bar{a}x)\bar{a}$
    * If $x = 0$, restart improvement phase
  - If there still exists a row a with $\bar{a}\bar{x} < -\sigma$, $x = x - (\bar{a}x)\bar{a}$, restart improvement phase
- If $Ax \geq 0$, output solution Bx
- Set $A = A(I + \bar{x}\bar{x}^T)$ and $B = B(I + \bar{x}\bar{x}^T)$; go back to perceptron phase.

Analysis

- We know from the perceptron analysis that if ρ is sufficiently large, the perceptron phase finishes with a feasible solution.
- We know from the probability argument that after $O(n \log 1/\rho)$ times through the improvement phase, ρ will be sufficiently large with high probability (probability at least $1 - e^{-n}$).
- Each pass through the perceptron phase takes $O(n^2)$ iterations, each checking m constraints, each of which takes $O(n)$ time, for a total of $O(n^3 m)$ per pass.
- Each pass through the improvement phase has $O(n^2 \log n)$ iterations; again each one checks m constraints at $O(n)$ time each, for a total of $O(n^3 m \log n)$
- Thus the overall runtime is $O(n^4 m \log n \log 1/\rho)$ with high probability (probability at least $1 - e^{-n}$).
- There is a proof that $\log 1/\rho$ is polynomial in the inputs,so this is a polynomial time algorithm with high probability (probability at least $1 - e^{-n}$).