ORIE 6326: Convex Optimization

Branch and Bound Methods

Professor Udell

Operations Research and Information Engineering
Cornell

May 15, 2017

Nonconvex optimization

want to solve **nonconvex** problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & f_i(x) \leq 0 \\ & Ax = b \\ & x \in \mathcal{X} \end{array}$$

- $f : \mathbf{R}^n \to \mathbf{R}$ possibly not convex
- $f_i : \mathbf{R}^n \to \mathbf{R}$ possibly not convex
- $\mathcal{X} \subseteq \mathbf{R}^n$ possibly not convex set

Nonconvex optimization

two approaches to nonconvex optimization:

1. **local optimization**: use algo from convex optimization
   - find feasible point and objective value at that point
   - objective value at feasible point is **upper bound** on optimal value
   - usually, use gradient or quasi-Newton method, or variant
2. **relaxation**: relax problem to convex problem
   - finds **lower bound** on optimal value, but not a feasible point
   - relax using duality, or by replacing f by convex lower bound and $\mathcal{X}$ by **conv** $\mathcal{X}$
   - can be hard to explicitly compute convex lower bound

Relaxed problem

nonconvex problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & f_i(x) \leq 0 \\ & Ax = b \\ & x \in \mathcal{X} \end{array}$$

- $f : \mathbf{R}^n \to \mathbf{R}$ possibly not convex
- $f_i : \mathbf{R}^n \to \mathbf{R}$ possibly not convex
- $\mathcal{X} \subseteq \mathbf{R}^n$ possibly not convex set

relaxed problem

$$\begin{array}{ll} \text{minimize} & \hat{f}(x) \\ \text{subject to} & \hat{f}_i(x) \leq 0 \\ & Ax = b \\ & x \in \mathbf{conv}\, \mathcal{X} \end{array}$$

- $\hat{f}(x)$, $\hat{f}_i(x)$ convex

Relaxed problem

relaxed problem

$$\begin{array}{ll} \text{minimize} & \hat{f}(x) \\ \text{subject to} & \hat{f}_i(x) \leq 0 \\ & Ax = b \\ & x \in \mathbf{conv}\, \mathcal{X} \end{array}$$

properties:

- relaxed problem is convex
- $\mathcal{X} \subseteq \mathbf{conv}\, \mathcal{X}$
- for all i and all $x \in \mathbf{R}^n$, $\hat{f}_i(x) \leq f_i(x)$
- so feasible set of relaxed problem contains feasible set of nonconvex problem
- for all $x \in \mathbf{R}^n$, $\hat{f}(x) \leq f(x)$
- so optimal value of relaxed problem $\leq$ optimal value of nonconvex problem

Branch and bound method

branch and bound method recursively computes both the upper and lower bound to find **global** optimum

basic idea:

- partition feasible set into convex sets, and find lower/upper bounds for each
- form global lower and upper bounds; quit if close enough
- else, refine partition and repeat

properties:

- nonheuristic
  - maintains provable lower and upper bounds on global objective value
  - terminates with certificate proving ϵ-suboptimality
- often slow; exponential worst case performance
- but (with luck) can (sometimes) work well

Plan for today

two examples of branch and bound method:

- sigmoidal programming
- integer programming

(examples chosen so we can solve relaxed problem easily)

Outline

Sigmoidal programming

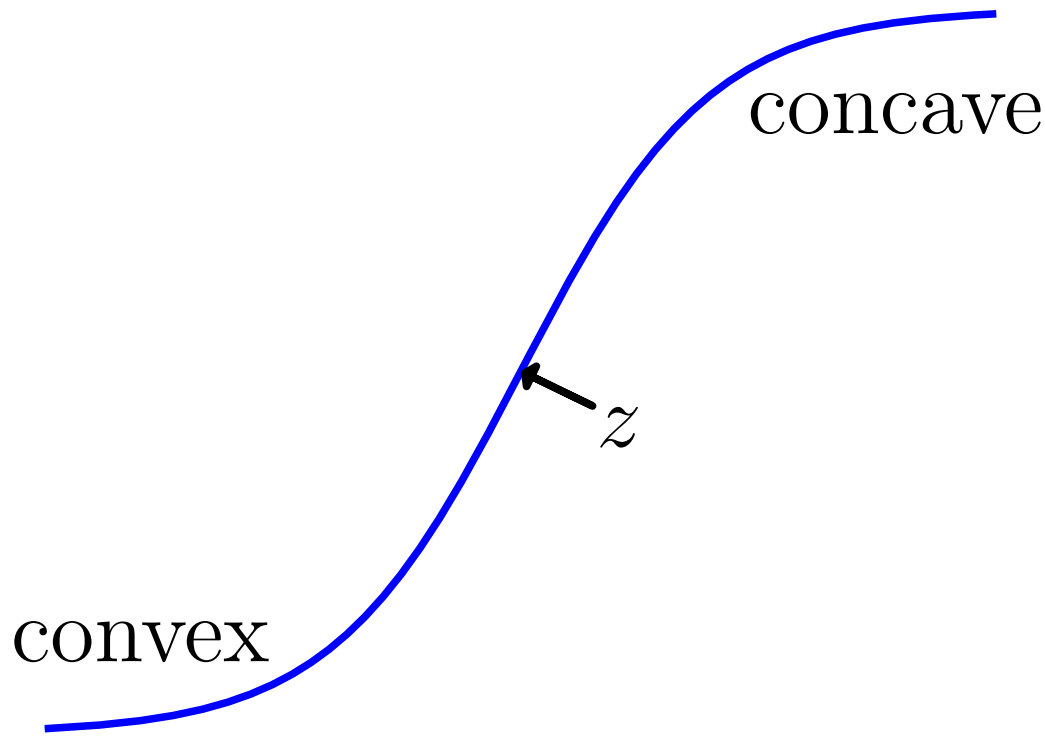
Define the **sigmoidal programming** problem

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} & x \in \mathcal{C} \end{array}$$

- f_i are **sigmoidal** functions
- $\mathcal{C}$ is a **convex** set of constraints

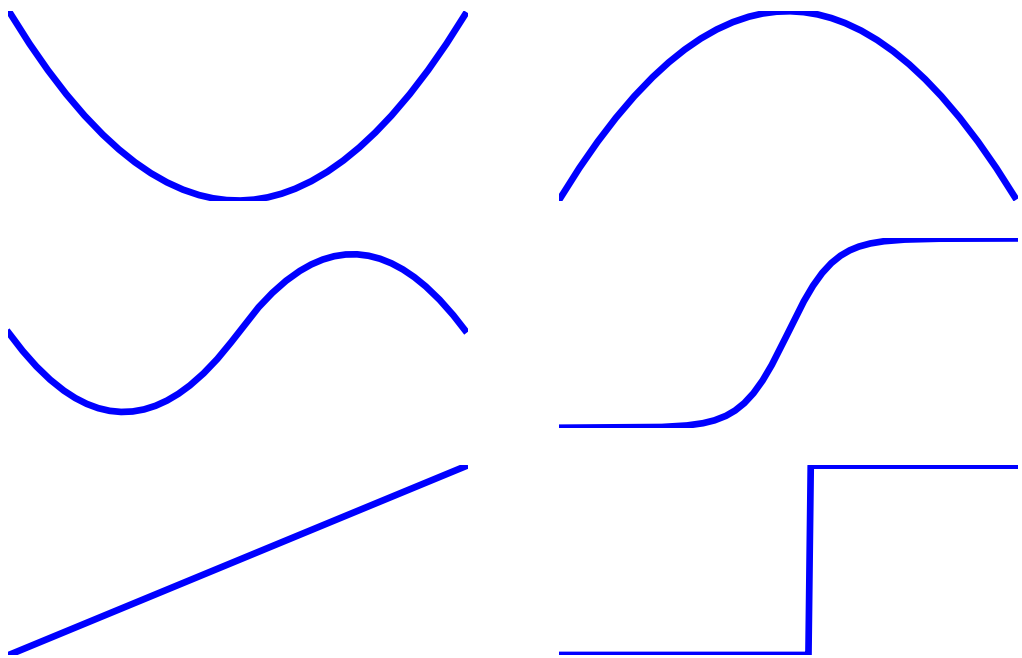
Sigmoidal functions

A continuous function $f : [l, u] \to \mathbf{R}$ is called **sigmoidal** if it is either convex, concave, or convex for $x \leq z \in \mathbf{R}$ and concave for $x \geq z$



Examples of sigmoidal functions

- logistic function $\mathrm{logistic}(x) = 1/(1 + \exp(x))$
- profit function $f(x) = (v - x)\mathrm{logistic}(\alpha x + \beta)$
- admittance function (aproximation to step function)
- CDF of any quasiconcave PDF



Outline

Bid optimization

- Each i corresponds to an auction
- v_i is the value of auction i
- $p_i(b_i)$ is probability of winning auction i with bid b_i
- To maximize winnings (in expectation), choose b by solving

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} v_i p_i(b_i) \\ \text{subject to} & b \in \mathcal{C} \end{array}$$

- $\mathcal{C}$ represents constraints on our bidding portfolio

Convex constraints for auctions

- Minimum and maximum bids: $l \leq b \leq u$
- Budget constraint: $\sum_{i=1}^{n} b_i \leq B$
- Sector constraint: $\sum_{i \in S} b_i \leq B_S$
- Diversification constraint: $\forall |S| > k$,

$$\sum_{i \in S} b_i \geq \epsilon \sum_{i=1}^{n} b_i$$

- And more (intersections are ok!)

The politician's problem

- Each i corresponds to a **constituency** (*e.g.* state, demographic, ideological group)
- p_i is # votes in constituency i (*e.g.* electoral, popular, etc)
- Politician chooses actions y
- Constituency i prefers actions w_i
- $f_i(w_i^T y)$ is probability of winning constituency i
- To win the most votes (in expectation), choose y by solving

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} p_i f_i(w_i^T y) \\ \text{subject to} & y \in \mathcal{C} \end{array}$$

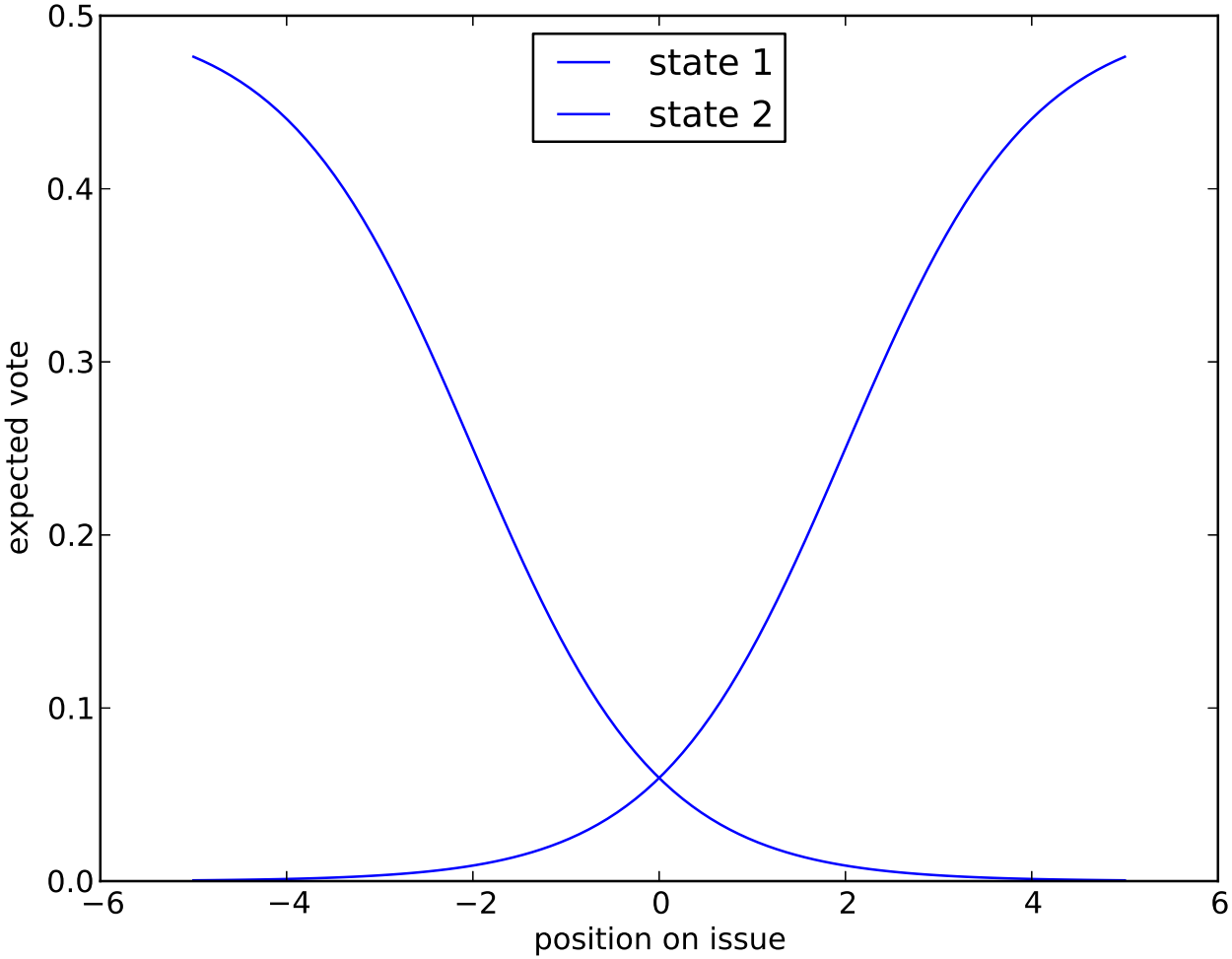
- $\mathcal{C}$ represents constraints on what actions we are willing or able to take

Convex constraints for politicians

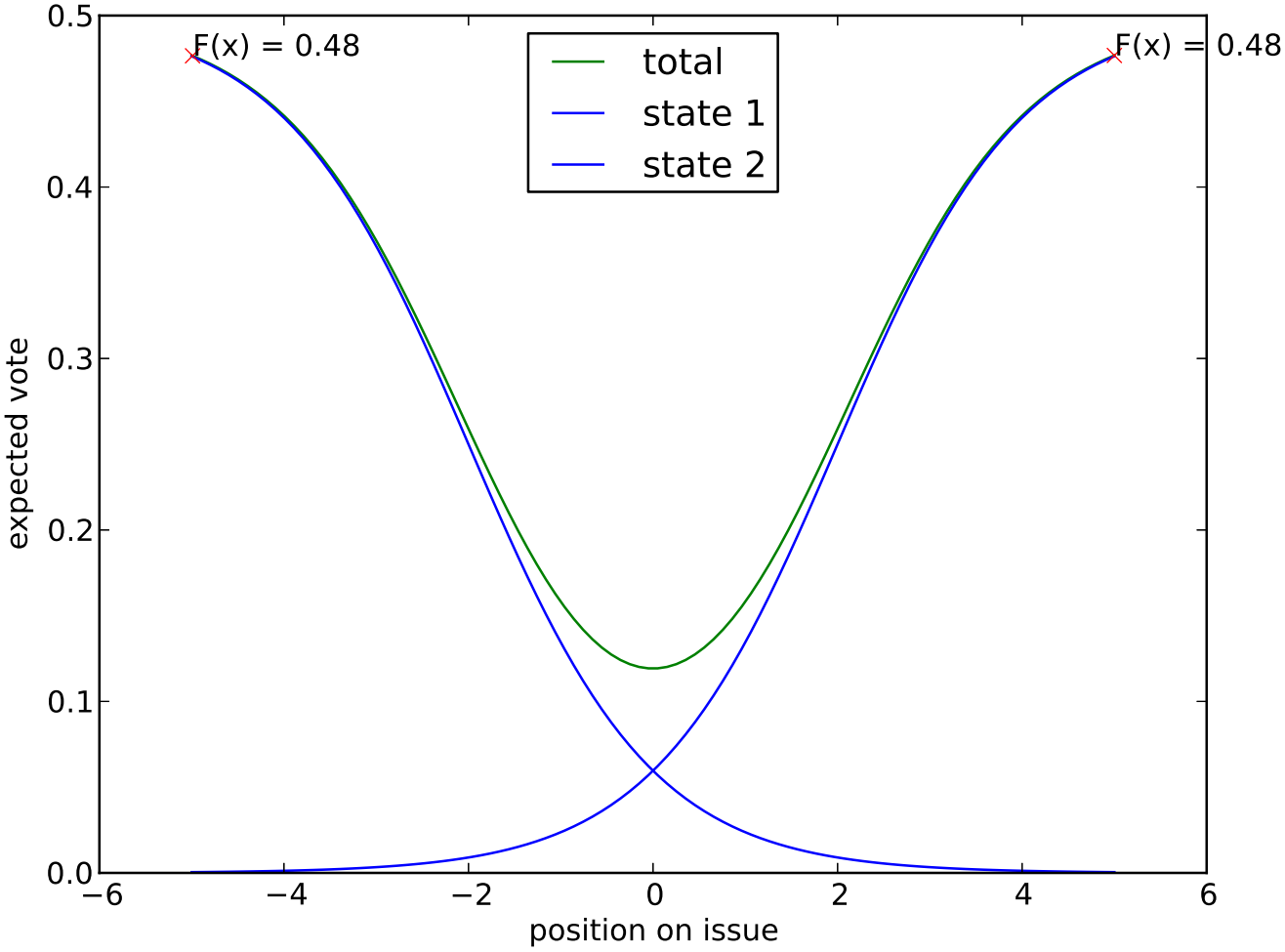
- min, max position: $l \leq y \leq u$
- max (political) budget: $\sum_{i=1}^{n} |y_i| \leq B$
- max hours in day: $\sum_{i=1}^{n} y_i \leq B$
- don't annoy any constituency too much: $w_i^T y \geq -\gamma$

Outline

Multiple peaks



Which way to go?



Sigmoidal programming is NP hard

Reduction from integer linear programming:

$$\begin{array}{ll} \text{find} & x \\ \text{subject to} & Ax = b \\ & x \in \{0,1\}^n \end{array}$$

Cast as sigmoidal programming:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^n g(x_i) \\ \text{subject to} & Ax = b \\ & 0 \leq x_i \leq 1 \quad i = 1, \ldots, n \end{array}$$

where $g : [0,1] \to \mathbf{R}$ is sigmoidal, and $g(z) = 0 \iff z \in \{0,1\}$

Optimal value of sigmoidal programming problem is $0 \iff$ there is an integral solution to $Ax = b$

(Also NP-hard to approximate, using reduction from MAXCUT)

Outline

Branch and bound

Idea of **branch-and-bound** method

- We can't search every **point** in the space, but we'd be willing to search a lot of **boxes**
- Need a method that won't overlook the global maximum

Branch and bound

So we:

- Partition space into smaller regions $Q \in \mathcal{Q}$
- Compute upper and lower bounds $U(Q)$ and $L(Q)$ on optimal function value

$$f^\star(Q) = \max_{x \in Q} \sum_{i=1}^{n} f_i(x_i)$$

in region Q:

$$L(Q) \leq f^\star(Q) \leq U(Q)$$

- Repeat until we zoom in on global max:

$$\max_{Q \in \mathcal{Q}} L(Q) \leq f^\star \leq \max_{Q \in \mathcal{Q}} U(Q).$$

Ingredients for branch and bound success

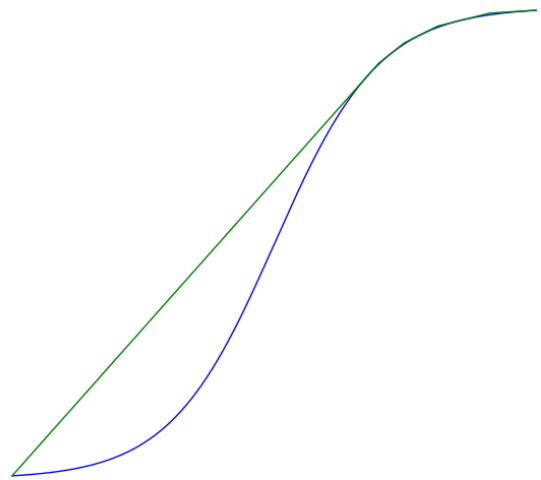
We need methods to

- Easily compute upper and lower bounds $U(Q)$ and $L(Q)$
- Choose the next region to split
- Choose how to split it

so that the bounds become provably tight around the true solution reasonably quickly.

Ingredients for sigmoidal programming

- Easily compute upper and lower bounds $U(Q)$ and $L(Q)$ using **concave envelope** of the functions f_i.
- Choose the region with highest upper bound as the next region to split.
- Split it at the solution to the previous problem along the coordinate with the highest error.



Bound f

Suppose we have functions $\hat{f}_i$ such that

- $f_i(x_i) \leq \hat{f}_i(x_i)$ for every $x \in Q$, and
- $\hat{f}_i$ are all concave.

Then $\sum_{i=1}^n \hat{f}_i(x_i)$ is also concave, and so it's easy to solve

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^n \hat{f}_i(x_i) \\ \text{subject to} & x \in \mathcal{Q}. \end{array}$$

Let $\hat{x}^\star$ be the solution to this relaxed problem.

Bound $f^\star(Q)$

Then we have an upper and lower bound on $f^\star(Q)$!

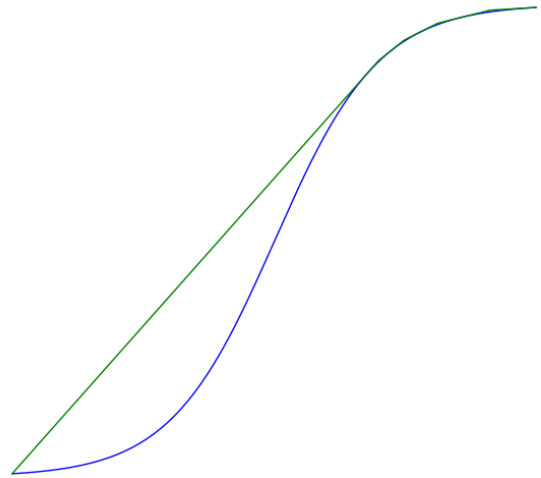
$$
\begin{aligned}
f_i(x_i) &\leq \hat{f}_i(x_i) \quad \forall x, \quad i = 1, \ldots, n \\
\sum_{i=1}^n f_i(x_i) &\leq \sum_{i=1}^n \hat{f}_i(x_i) \quad \forall x \\
\max_{x \in Q} \sum_{i=1}^n f_i(x_i) &\leq \max_{x \in Q} \sum_{i=1}^n \hat{f}_i(x_i) \\
f^\star(Q) &\leq \max_{x \in Q} \sum_{i=1}^n \hat{f}_i(x_i)
\end{aligned}
$$

Then

$$
\underbrace{\sum_{i=1}^n f_i(\hat{x}_i^\star)}_{L(Q)} \leq f^\star(Q) \leq \underbrace{\sum_{i=1}^n \hat{f}_i(\hat{x}_i^\star)}_{U(Q)}
$$

Concave envelope

The tightest concave approximation to f is obtained by choosing $\hat{f}_i$ to be the **concave envelope** of the function f.



$\hat{f}_i = -(-f)^{**}$ is the (negative) **bi-conjugate** of $-f$, where

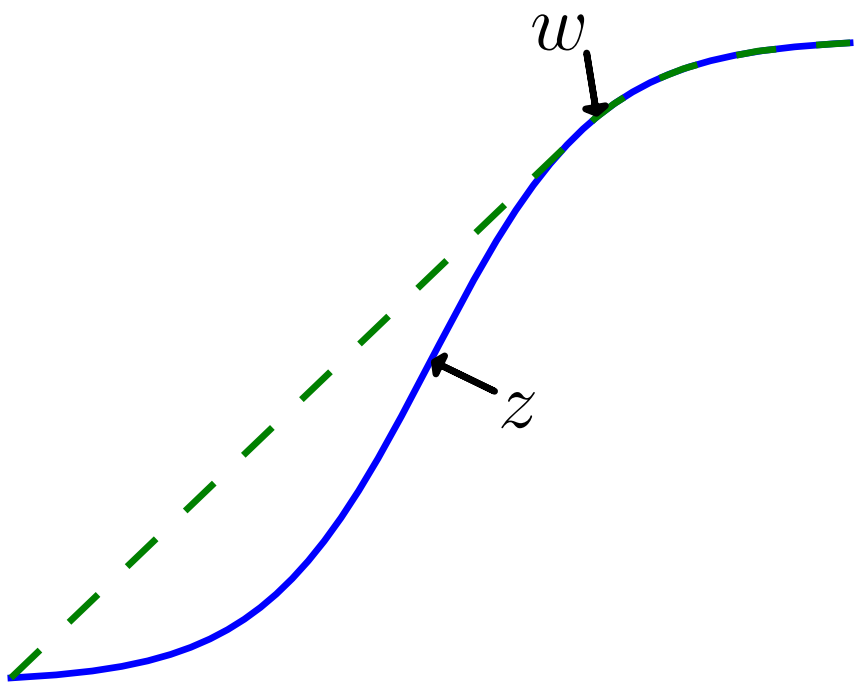
$$f^\star(y) = \sup_{x \in I}(f(x) - yx)$$

is the **conjugate** of f.

Concave envelope

The concave envelope can be computed by first locating the point w such that $f(w) = f(a) + f'(w)(w - a)$. Then the concave envelope $\hat{f}$ of f can be written piecewise as

$$\hat{f}(x) = \left\{ \begin{array}{cc} f(a) + f'(w)(x - a) & a \leq x \leq w \\ f(x) & w \leq x \leq b \end{array} \right\}.$$



Branch

Compute tighter approximation by splitting rectangles wisely

e.g.,

- optimism: split rectangle Q with highest upper bound
- greed: split along coordinate i with greatest error
- hope: split at previous solution x_i^*

(better heuristics for these choices can dramatically improve performance)

Convergence

The number of concave subproblems that must be solved to achieve accuracy ϵ is bounded by

$$\prod_{i=1}^{n} \left(\left\lfloor \frac{(h_i(z_i) - h_i(l_i))\,(z_i - l_i)}{\epsilon/n} \right\rfloor + 1 \right),$$

where

- $Q_{\text{init}} = (l_1, u_1) \times \cdots \times (l_n, u_n)$
- $f_i(x) = \int_{l_i}^{x} h_i(t)\, dt,\ i = 1, \ldots, n$
- $z_i = \arg\max_{[l_i, u_i]} h_i(x),\ i = 1, \ldots, n$

Prune

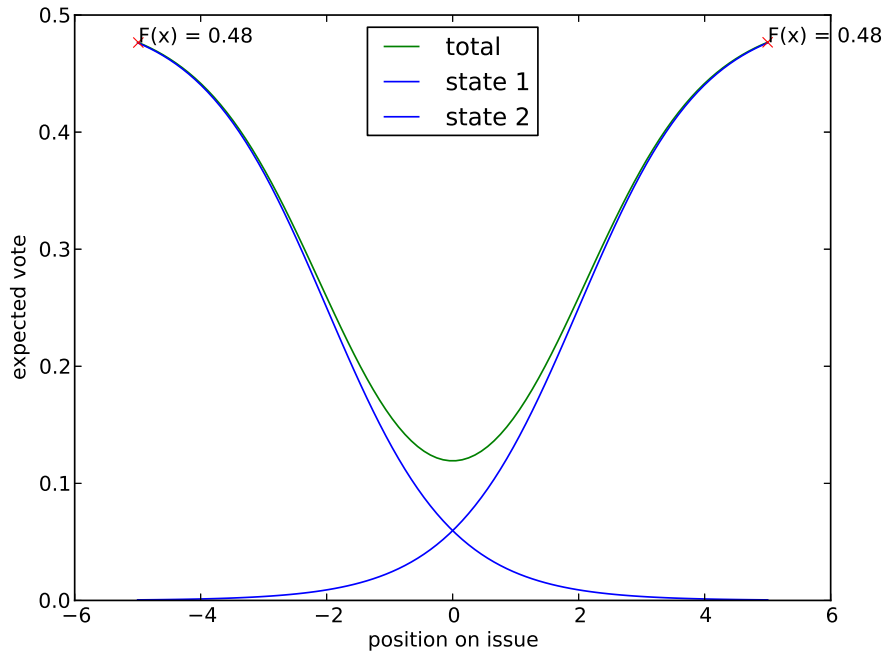
$$\max_{Q \in \mathcal{Q}} L(Q) \leq f^\star \leq \max_{Q \in \mathcal{Q}} U(Q)$$

- Q is **active** if $\max_{Q \in \mathcal{Q}} L(Q) \leq U(Q)$
- otherwise the solution cannot lie in Q

Outline

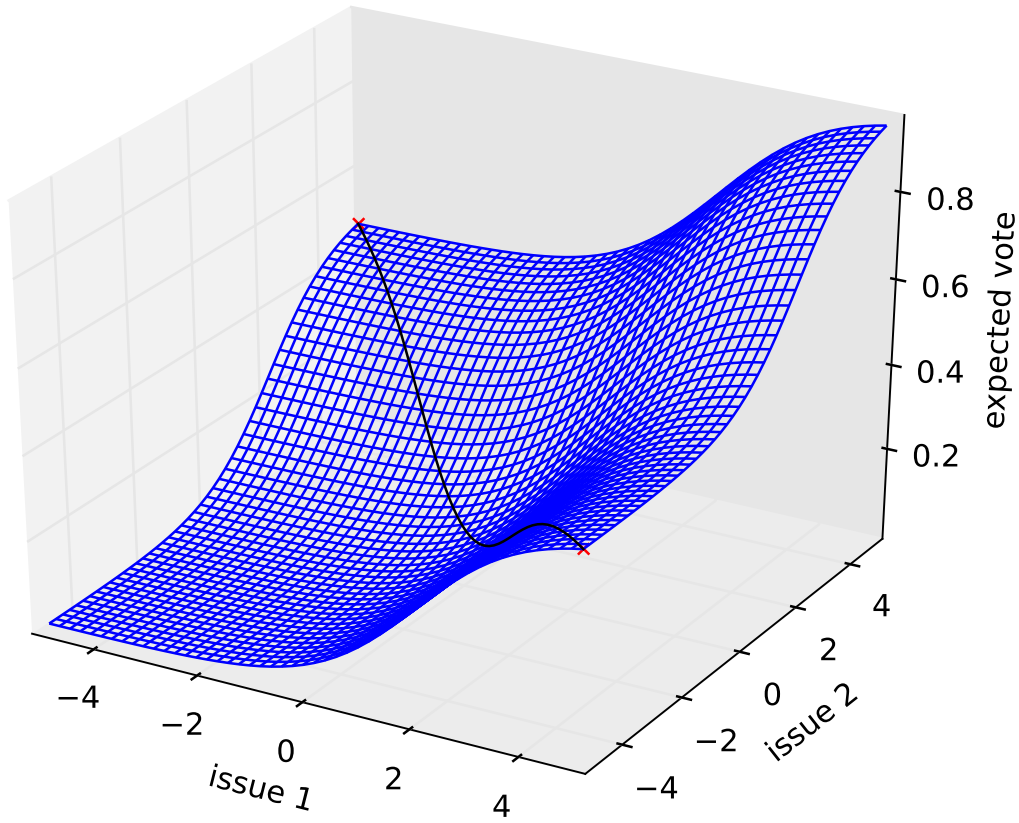
Example: opposing interests

$$\begin{array}{ll} \text{maximize} & \sum_{i=1,2} \text{logistic}(x_i - 2) \\ \text{subject to} & \sum_{i=1,2} x_i = 0 \end{array}$$



Example: opposing interests

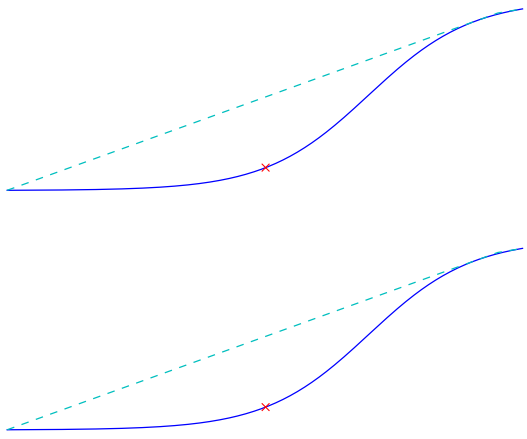
Black line is feasible set.



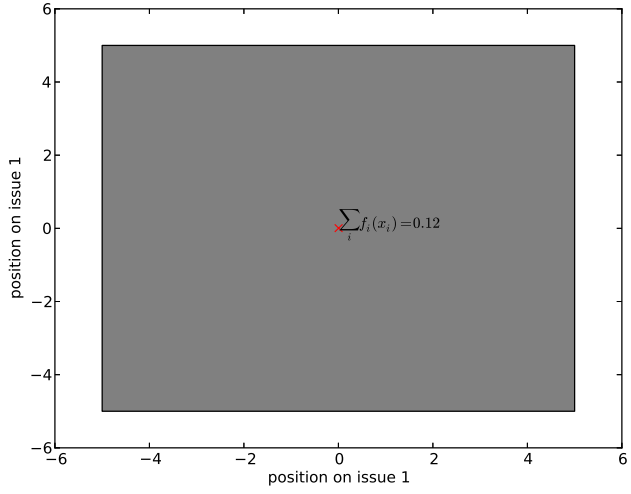
Example: opposing interests

1st iteration:

Best solution so far



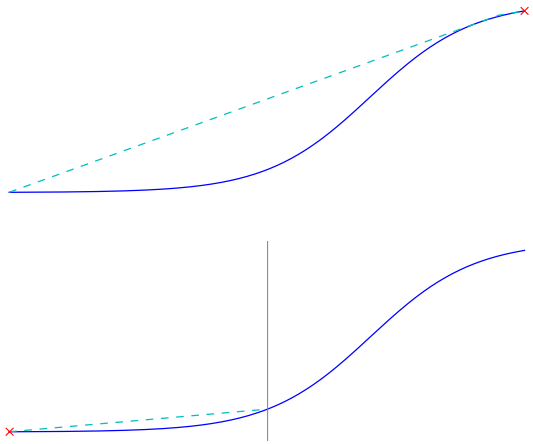
Active nodes



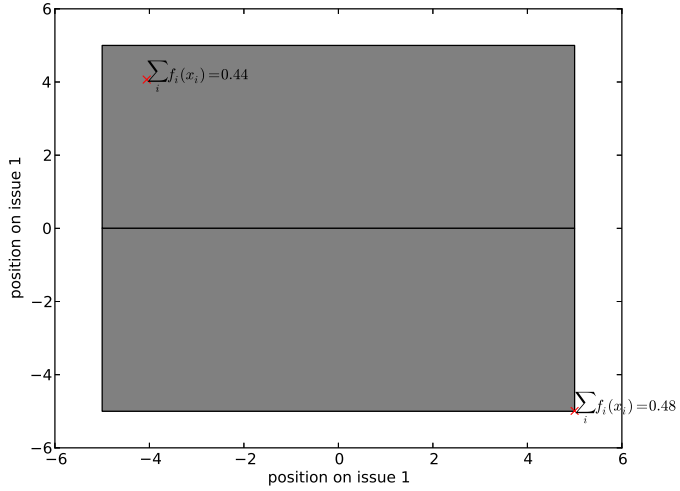
Example: opposing interests

2nd iteration:

Best solution so far



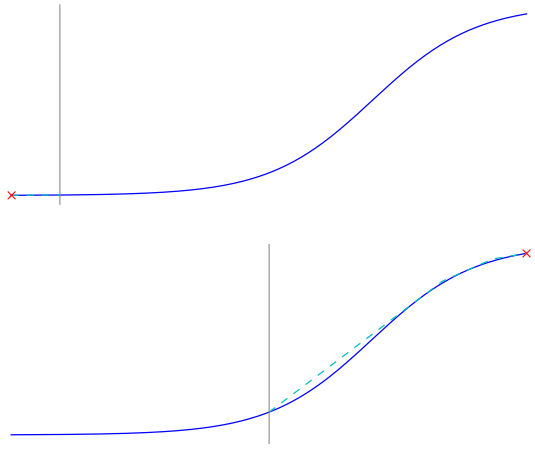
Active nodes



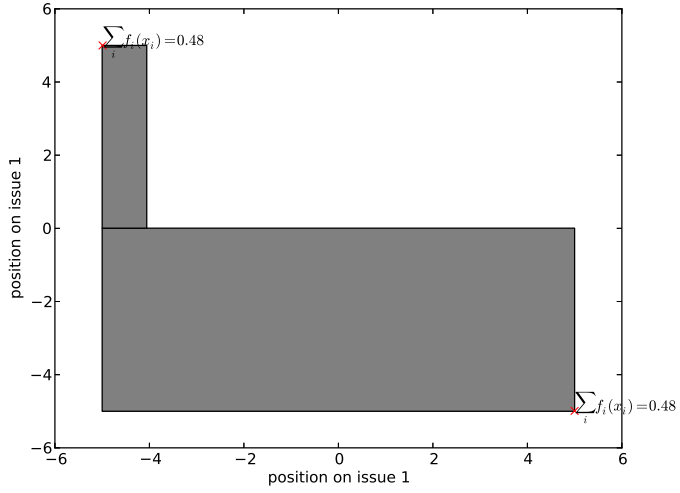
Example: opposing interests

3rd iteration:

Best solution so far

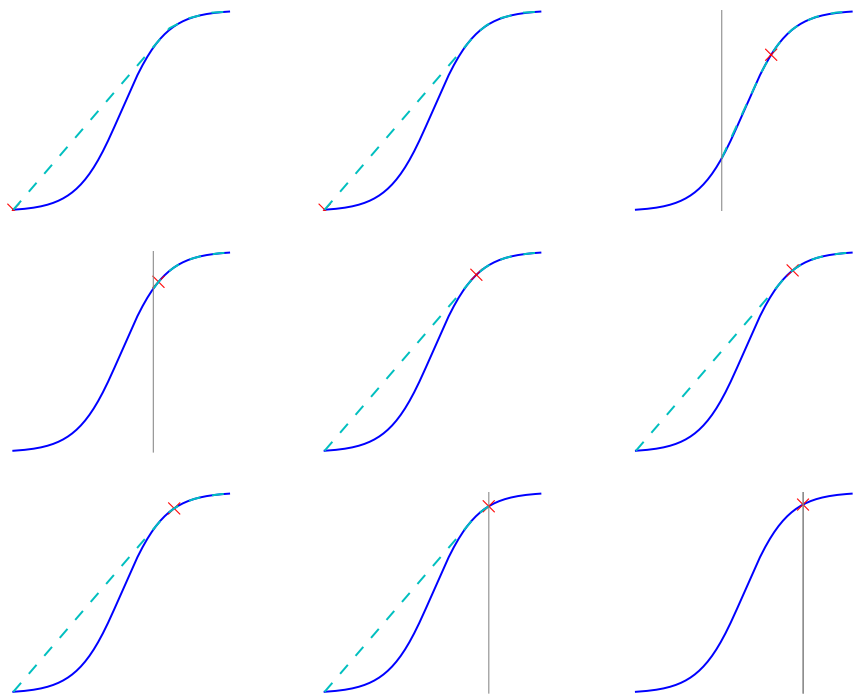


Active nodes



Bid optimization

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} v_i \text{logistic}(b_i - \beta_i) \\ \text{subject to} & \sum_{i=1}^{n} b_i \leq B \\ & b \geq 0 \end{array}$$



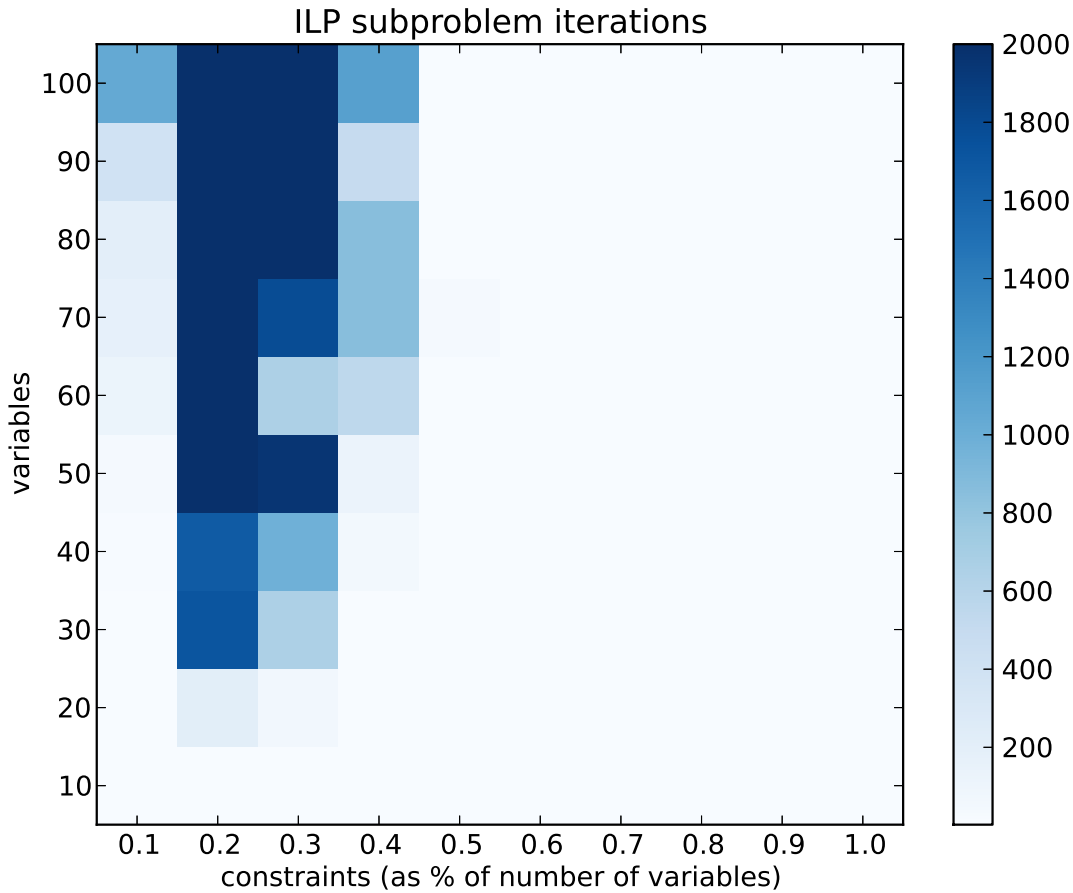
Bid optimization

Simulate for $v_i \sim \mathcal{U}(0,1)$, $B = 1/5 \sum_{i=1}^{n} v_i, \beta_i = 1$. To solve to accuracy $.00001 \cdot n$ takes very few steps! (1 step = 1 LP solve)

n	steps
10	2
20	4
30	4
40	6
50	7
60	7
70	6
80	6
90	7
100	6

Integer linear programming

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} |x_i - 1/2| \\ \text{subject to} \quad & Ax = b \\ & 0 \leq x_i \leq 1 \quad i = 1, \ldots, n \end{aligned}$$



Outline

Mixed Boolean-convex problem

$$\begin{array}{ll} \text{minimize} & f_0(x,z) \\ \text{subject to} & f_i(x,z) \leq 0, \quad i = 1,\ldots,m \\ & z_j \in \{0,1\}, \quad j = 1,\ldots,n \end{array}$$

- $x \in \mathbf{R}^p$ is called **continuous variable**
- $z \in \{0,1\}^n$ is called **Boolean variable**
- $f_0,\ldots,f_n$ are convex in x and z
- optimal value denoted $p^\star$
- for each fixed $z \in \{0,1\}^n$, reduced problem (with variable x) is convex

Solution methods

- **brute force**: solve convex problem for each of the 2^n possible values of $z \in \{0,1\}^n$
  - possible for $n \leq 15$ or so, but not $n \geq 20$
- **branch and bound**
  - in worst case, we end up solving all 2^n convex problems
  - hope that branch and bound will actually work much better

Lower bound via convex relaxation

convex relaxation

$$\begin{array}{ll} \text{minimize} & f_0(x,z) \\ \text{subject to} & f_i(x,z) \leq 0, \quad i = 1,\ldots,m \\ & 0 \leq z_j \leq 1, \quad j = 1,\ldots,n \end{array}$$

- convex with (continuous) variables x and z, so easily solved
- optimal value (denoted L_1) is lower bound on $p^\star$, optimal value of original problem
- L_1 can be $+\infty$ (which implies original problem infeasible)

Upper bounds

can find an upper bound (denoted U_1) on $p^\star$ several ways

- simplest method: round each relaxed Boolean variable $z_i^\star$ to 0 or 1
- more sophisticated method: round each Boolean variable, then solve the resulting convex problem in x
- randomized method:
  - generate random $z_i \in \{0, 1\}$, with $\mathbf{prob}(z_i = 1) = z_i^\star$
  - (optionally, solve for x again)
  - take best result out of some number of samples
- upper bound can be $+\infty$
  (if method failed to produce a feasible point)
- if $U_1 - L_1 \leq \epsilon$ we can quit

Branching

pick any index k, and form two subproblems.

first problem:

$$\begin{array}{ll} \text{minimize} & f_0(x,z) \\ \text{subject to} & f_i(x,z) \leq 0, \quad i = 1,\ldots,m \\ & z_j \in \{0,1\}, \quad j = 1,\ldots,n \\ & z_k = 0 \end{array}$$

second problem:

$$\begin{array}{ll} \text{minimize} & f_0(x,z) \\ \text{subject to} & f_i(x,z) \leq 0, \quad i = 1,\ldots,m \\ & z_j \in \{0,1\}, \quad j = 1,\ldots,n \\ & z_k = 1 \end{array}$$

- each has $n-1$ Boolean variables
- optimal value of original problem is the smaller of the optimal values of the two subproblems
- convex relaxations of subproblems give upper/lower bounds

New bounds from subproblems

- let $\tilde{L}, \tilde{U}$ be lower, upper bounds for $z_k = 0$
- let $\bar{L}, \bar{U}$ be lower, upper bounds for $z_k = 1$
- $\min\{\tilde{L}, \bar{L}\} \geq L_1$
- can assume w.l.o.g. that $\min\{\tilde{U}, \bar{U}\} \leq U_1$
- thus, we have new bounds on $p^\star$:

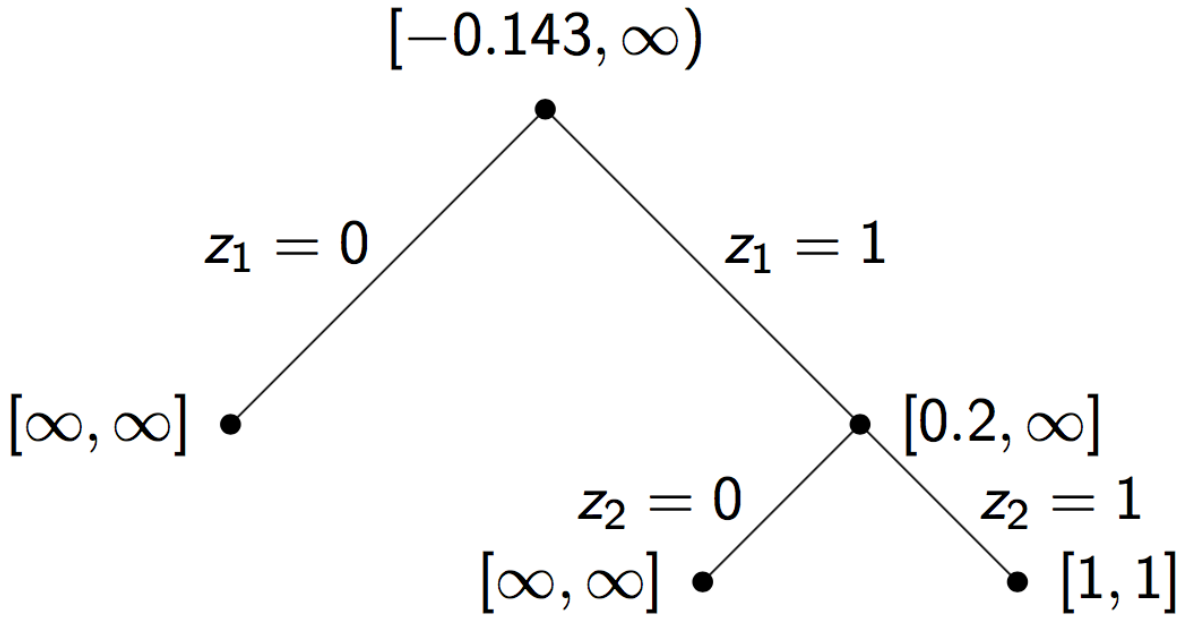
$$L_2 = \min\{\tilde{L}, \bar{L}\} \leq p^\star \leq U_2 = \min\{\tilde{U}, \bar{U}\}$$

Branch and bound algorithm

- continue to form binary tree by splitting, relaxing, calculating bounds on subproblems
- convergence proof is trivial: cannot go more than 2^n steps before $U = L$
- can prune nodes with L excceding current U_k
- common strategy is to pick a node with smallest L
- can pick variable to split several ways
  - 'least ambivalent': choose k for which $z^\star = 0$ or 1, with largest Lagrange multiplier
  - 'most ambivalent': choose k for which $|z_k^\star - 1/2|$ is minimum

Small example

nodes show lower and upper bounds for three-variable Boolean LP



Practical notes

- integer linear programming is **much** easier than integer convex programming

Practical notes

- integer linear programming is **much** easier than integer convex programming
- don't use an interior point method as a subproblem solver for a branch and bound method

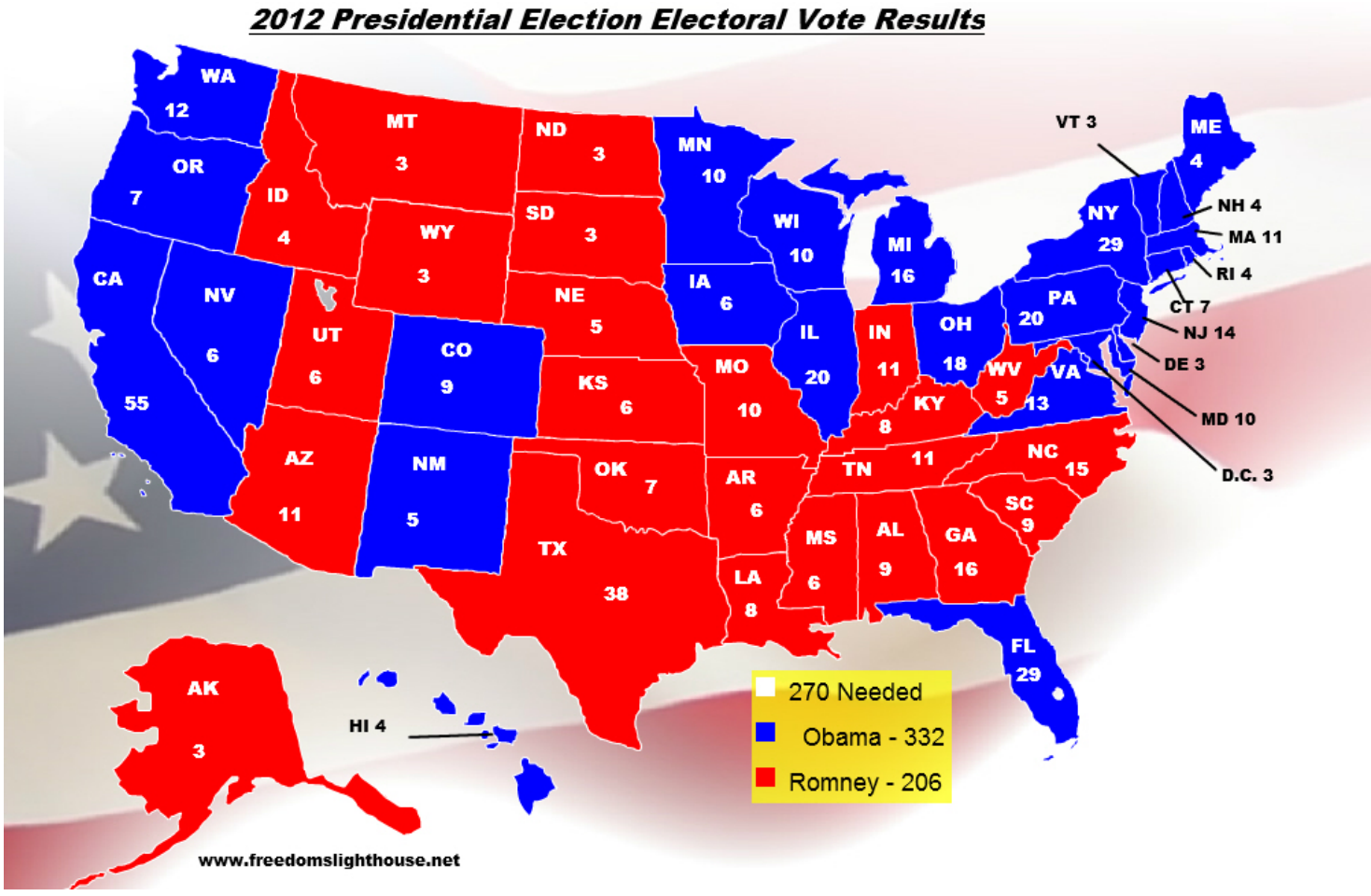
Practical notes

- integer linear programming is **much** easier than integer convex programming
- don't use an interior point method as a subproblem solver for a branch and bound method
- for ILPs, use a dedicated solver; they have branching and pruning heuristics that can reduce solve times significantly

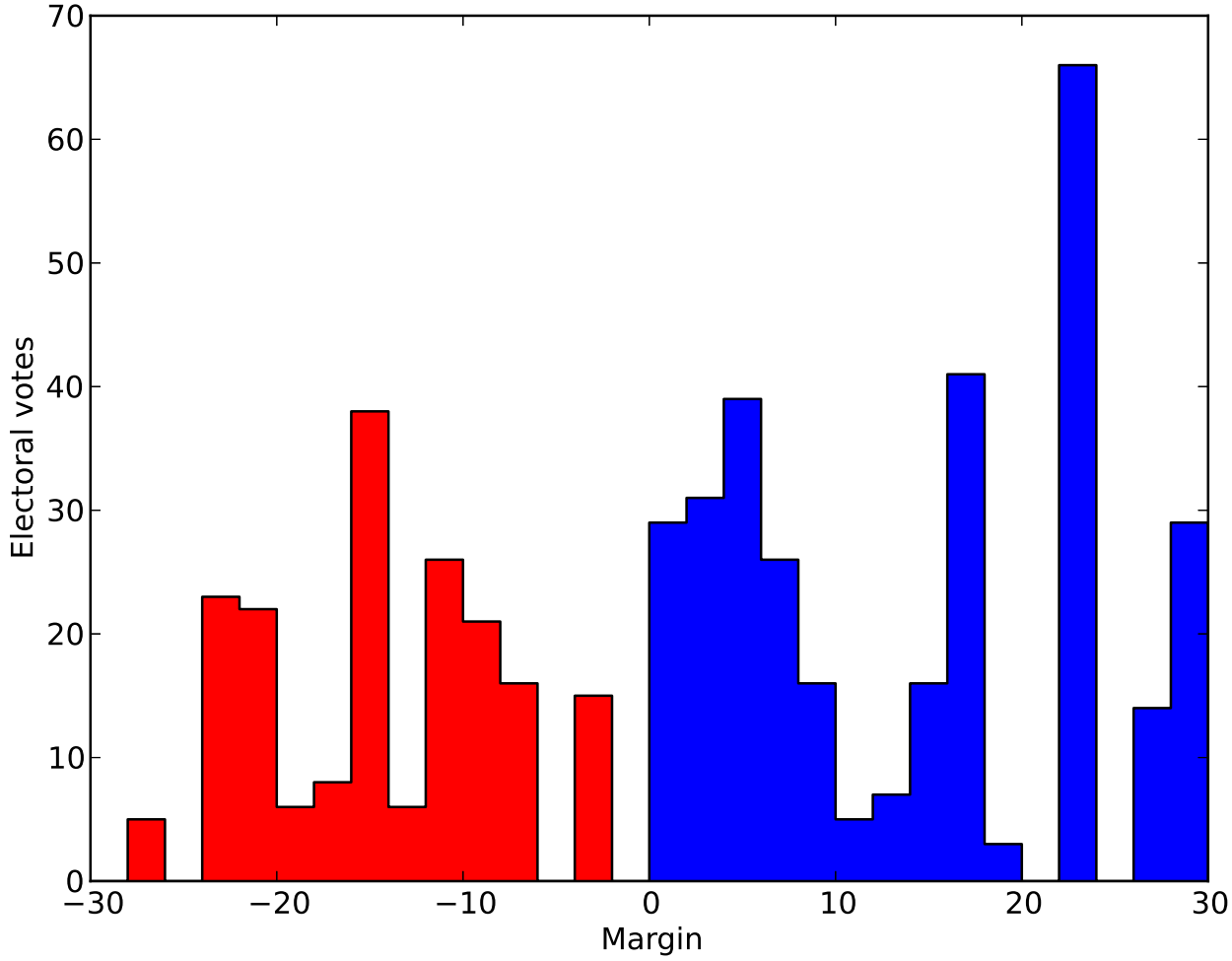
Practical notes

- integer linear programming is **much** easier than integer convex programming
- don't use an interior point method as a subproblem solver for a branch and bound method
- for ILPs, use a dedicated solver; they have branching and pruning heuristics that can reduce solve times significantly
- even for non-ILPs, often the best approach is to approximate it as (a sequence of) ILPs

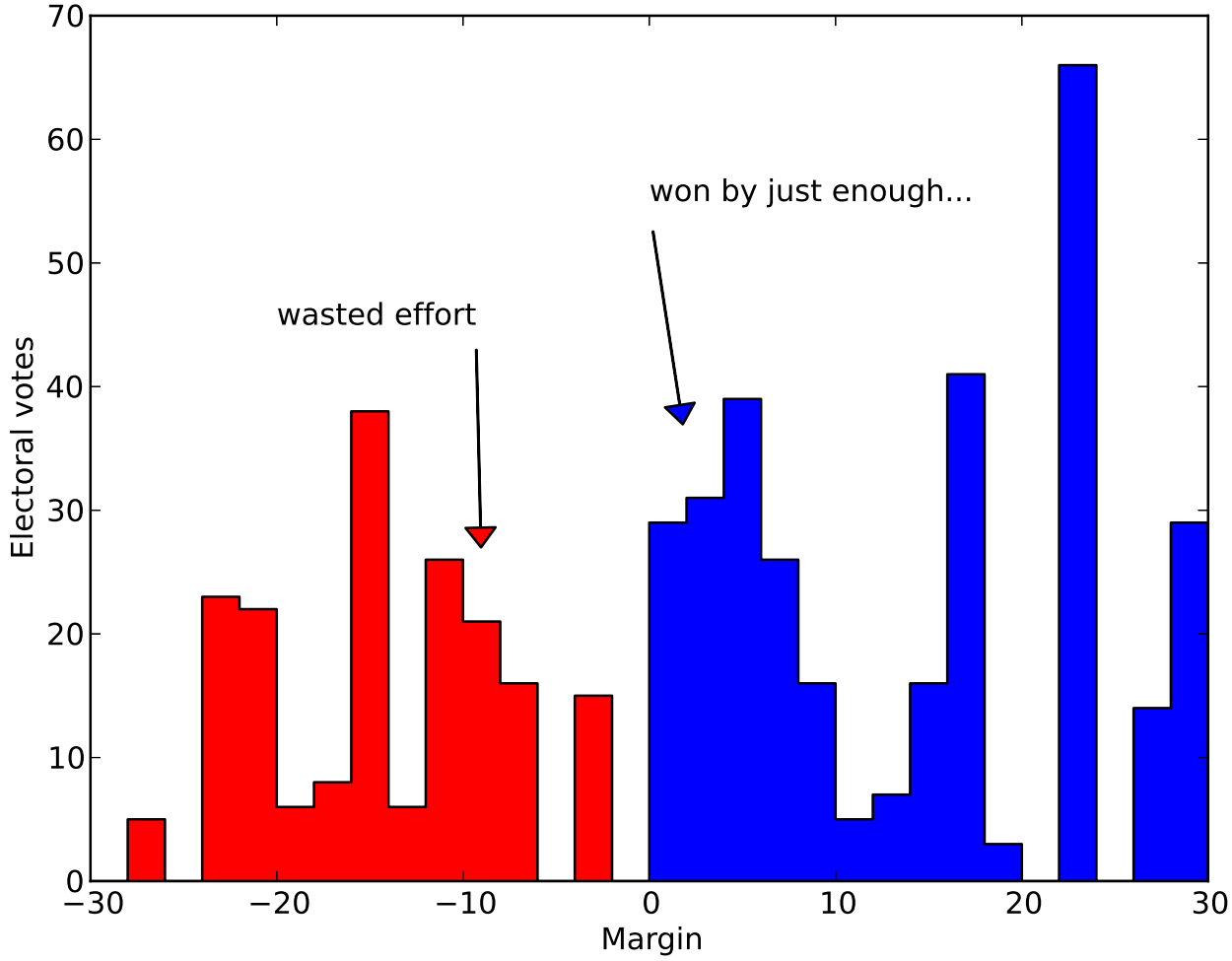
Motivation


2012 Presidential Election Electoral Vote Results
WA 12
OR 7
CA 55
NV 6
ID 4
MT 3
WY 3
UT 6
AZ 11
CO 9
NM 5
ND 3
SD 3
NE 5
KS 6
OK 7
TX 38
MN 10
IA 6
MO 10
AR 6
LA 8
WI 10
IL 20
MS 6
AL 9
MI 16
IN 11
KY 8
TN 11
GA 16
FL 29
SC 9
NC 15
OH 18
WV 5
VA 13
PA 20
NY 29
VT 3
ME 4
NH 4
MA 11
RI 4
CT 7
NJ 14
DE 3
MD 10
D.C. 3
AK 3
HI 4
270 Needed
Obama - 332
Romney - 206
www.freedomslighthouse.net

Optimization on the Obama campaign



Optimization on the Obama campaign



Political positioning: data

- Data from 2008 American National Election Survey (ANES)
- Respondents r rate candidates c as having positions $y^{rc} \in [1,7]^m$ on m issues.
- Respondents say how happy they'd be if the candidate won $h^{rc} \in [1,7]$.
- We suppose a respondent would vote for a candidate c if $h^{rc} > h^{rc'}$ for any other candidate c'. If so, $v^{rc} = 1$ and otherwise $v^{rc} = 0$.
- For each candidate c and state i we construct a model to predict the likelihood of a respondent $r \in S_i$ in state i voting for candidate c as a function of the candidate's perceived positions y^{rc}.

Political positioning: model

For each candidate c and state i we construct a model to predict the likelihood of a respondent $r \in S_i$ in state i voting for candidate c as a function of the candidate's perceived positions y^{rc}:

$$\text{minimize} \quad \sum_{r \in S_i} l(y^{rc}, v^{rc}; w_i),$$

where $l(y^{rc}, v^{rc}; w_i)$ is the penalty for predicting $\text{logistic}(w_i^T y^{rc})$ rather than v^{rc}, *i.e.*

$$l(y^{rc}, v^{rc}; w_i) = \text{logistic}(w_i^T y^{rc})^{v^{rc}} (1 - \text{logistic}(w_i^T y^{rc}))^{1-v^{rc}}.$$

Note: only 34 states, some with only 14 respondents ...

Political positioning: electoral vote

- Suppose each state i has v_i votes, which they allocate entirely to the winner of the popular vote.
- y denotes the positions the politician takes on the issues.
- Then using our model,the politician's pandering to state i is given by $x_i = w_i^T y$, and the expected number of votes from state i is

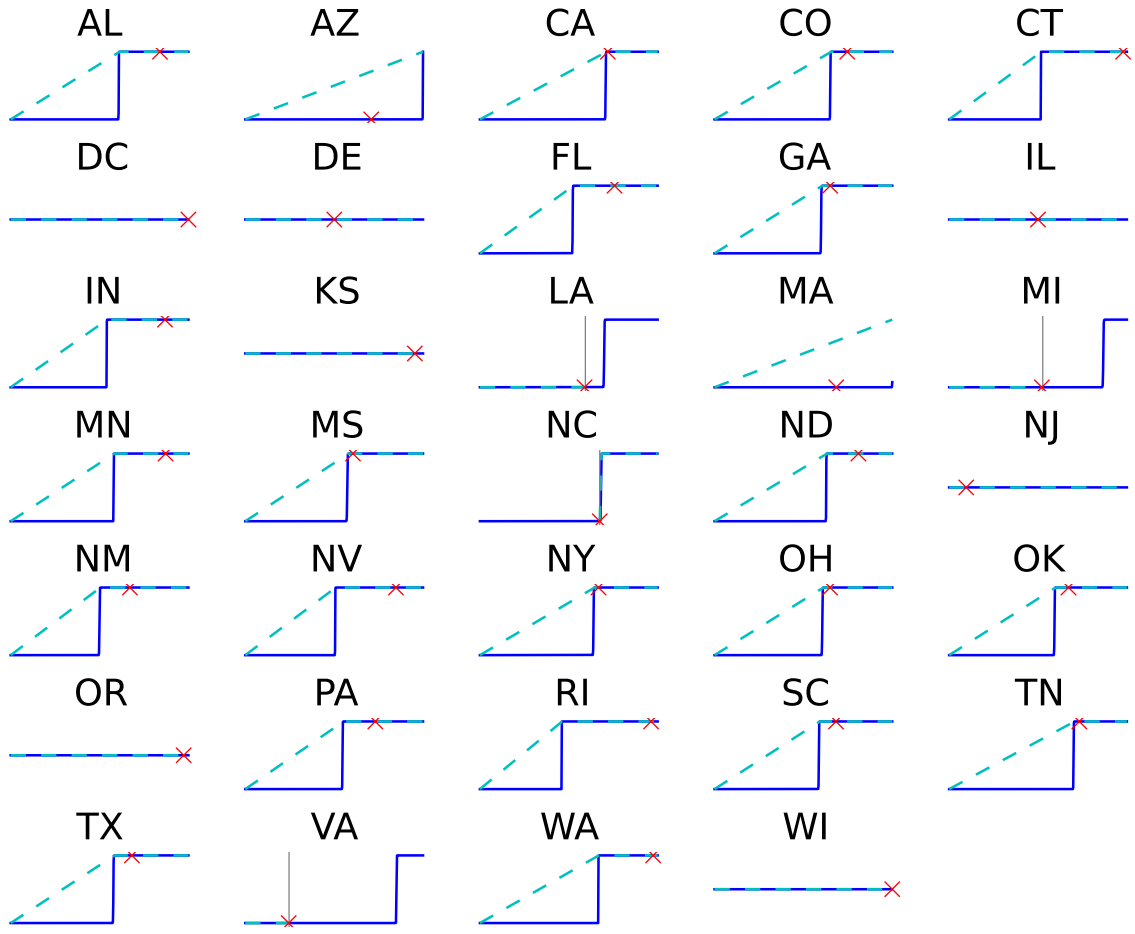
$$v_i \mathbf{1}(\text{logistic}(x_i) > .5),$$

  where $\mathbf{1}(x)$ is 1 if x is true and 0 otherwise.
- Hence the politician will win the most votes if y is chosen by solving

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} v_i \mathbf{1}(\text{logistic}(x_i) > .5) \\ \text{subject to} & x_i = w_i^T y \quad \forall i \\ & 1 \leq y \leq 7. \end{array}$$

Optimal solutions to the politician's problem

Obama's optimal pandering:



Optimal positions for Obama

Issue	Optimal position	Previous position
Spending and Services	1.26	5.30
Defense spending	1.27	3.69
Liberal conservative	1.00	3.29
Govt assistance to blacks	1.00	3.12

Issue	1	7
Spending and services	provide many fewer services	provide many more services
Defense spending	decrease defense spending	increase defense spending
Liberal conservative	liberal	conservative
Assistance to blacks	govt should help blacks	blacks should help themselves