Nonconvex Global Optimization for Grammar Induction

Matthew R. Gormley

CLSP Seminar

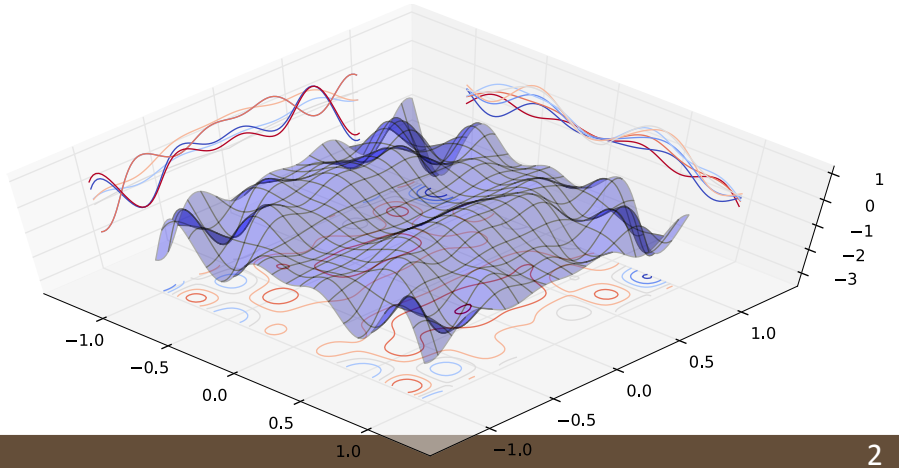
04/02/13

Joint work with Jason Eisner

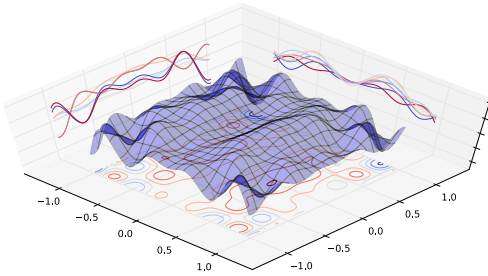
The Problem

For many models in NLP/ML/Speech, we optimize **nonconvex likelihood objectives**.

- We **know little** about these nonconvex surfaces.
- Our usual search methods suffer from problems with **local optima**.
- Even if we find the global optimum, we can't **prove** it.



Our Goals



- **Learn more** about these commonplace nonconvex likelihood objectives
- Go **beyond local-search**.
- Develop a search method capable of finding a **provably ε-optimal solution**.

Overview

I. The Problem: Nonconvexity
II. Example Task: Grammar Induction
III. Background
IV. Search Methods
    A. Nonconvex Global Optimization
    B. Relaxed Viterbi EM with Posterior Constraints
V. Experiments

Motivating Example

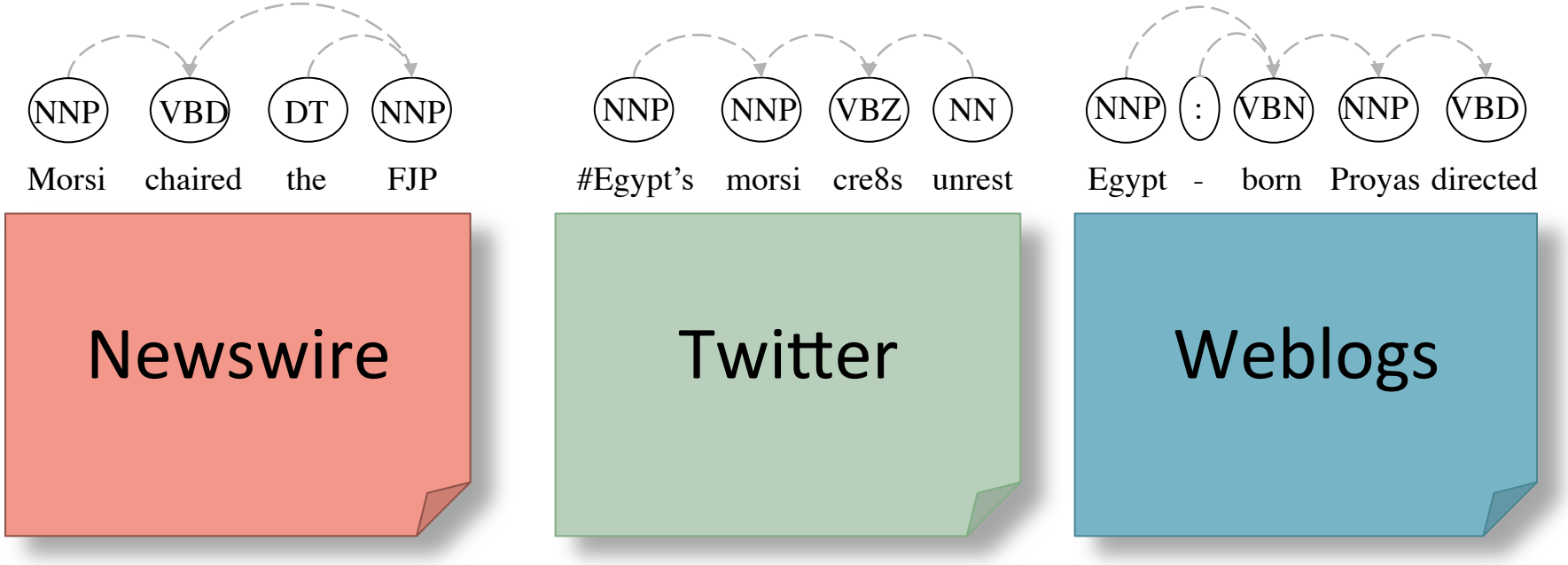
- Suppose you wanted to do unsupervised dependency parsing.
- Your data inventory:
  (just sentences)

Motivating Example

- Suppose you wanted to do unsupervised dependency parsing.
- Your data inventory:
  (just sentences and maybe POS tags)

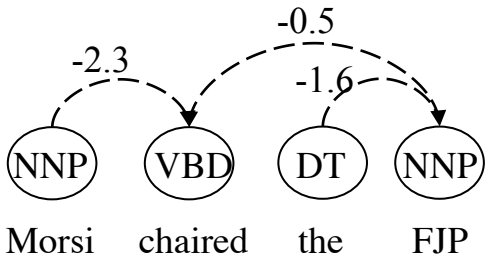
Motivating Example

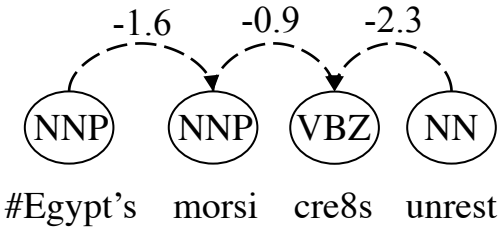
- Goal: Recover **unobserved** syntax.
- Our Focus:
  - Parameter estimation in the presence of nonconvexity.
  - Posterior constrained inference in these types of models.

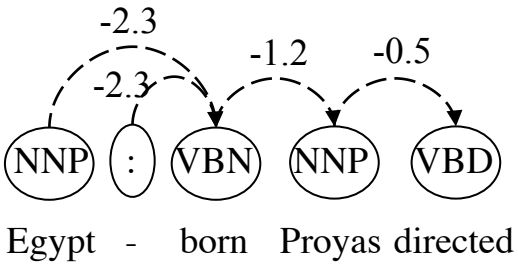


Motivating Example

- Spitkovsky et al. (2009) show hard (Viterbi) EM outperforms soft EM.
- Viterbi EM objective function: $\max \sum_m \theta_m f_m$
- NP Hard to solve







Newswire

Twitter

Weblogs

Background: Local Search for Grammar Induction

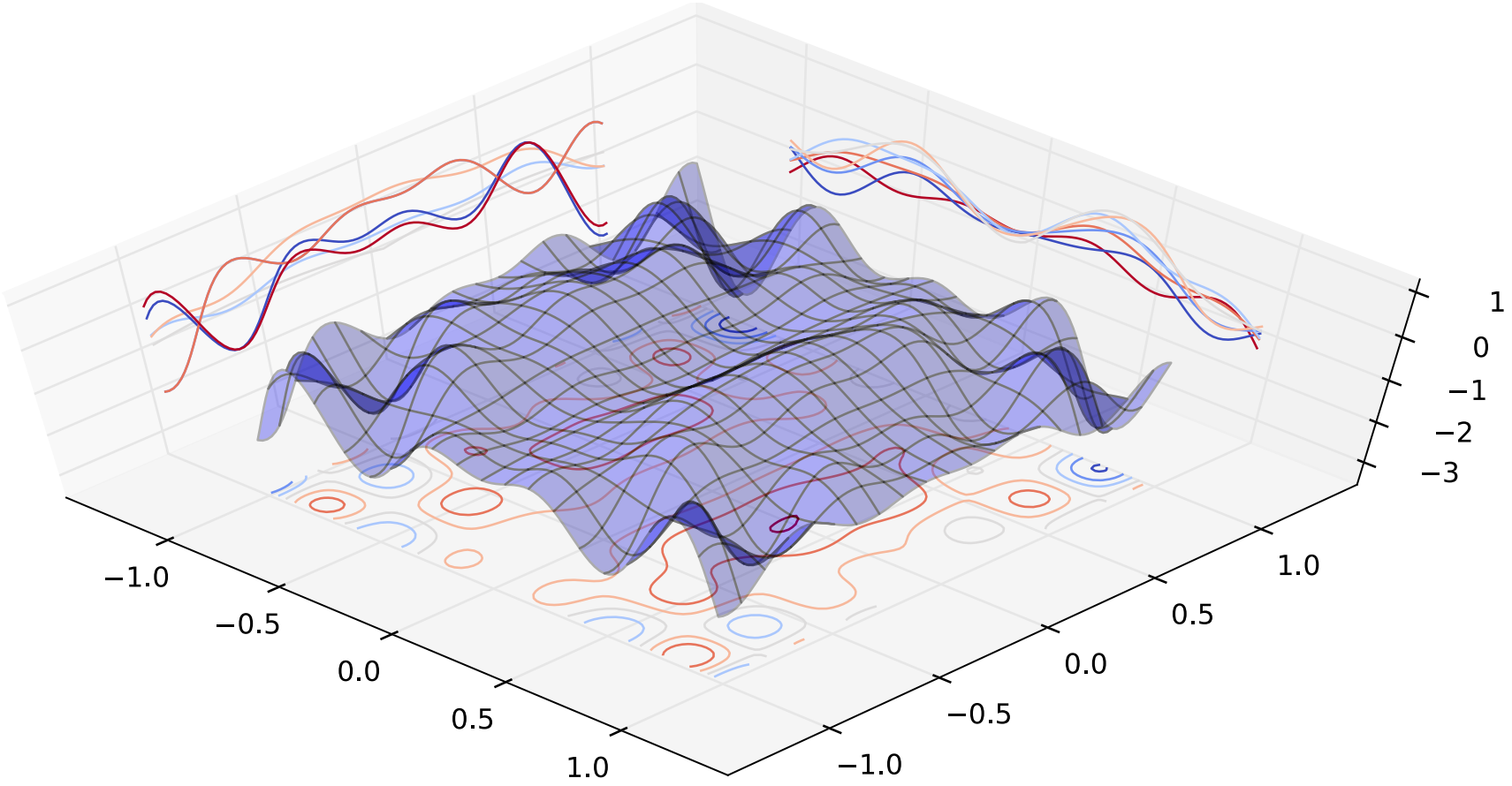
- Many **parameter estimation techniques** have been tried for grammar induction:

soft EM	(Klein and Manning, 2004; Spitkovsky et al., 2010)
contrastive estimation	(Smith and Eisner, 2006; Smith, 2006)
hard (Viterbi) EM	(Spitkovsky et al., 2010b)
variational EM	(Cohen et al., 2009; Cohen and Smith, 2009; Naseem et al., 2010)

- These are all **local search** techniques used to optimize a nonconvex objective.
- **Tricks**: random restarts, heuristic initialization of model parameters, annealing, posterior constraints.

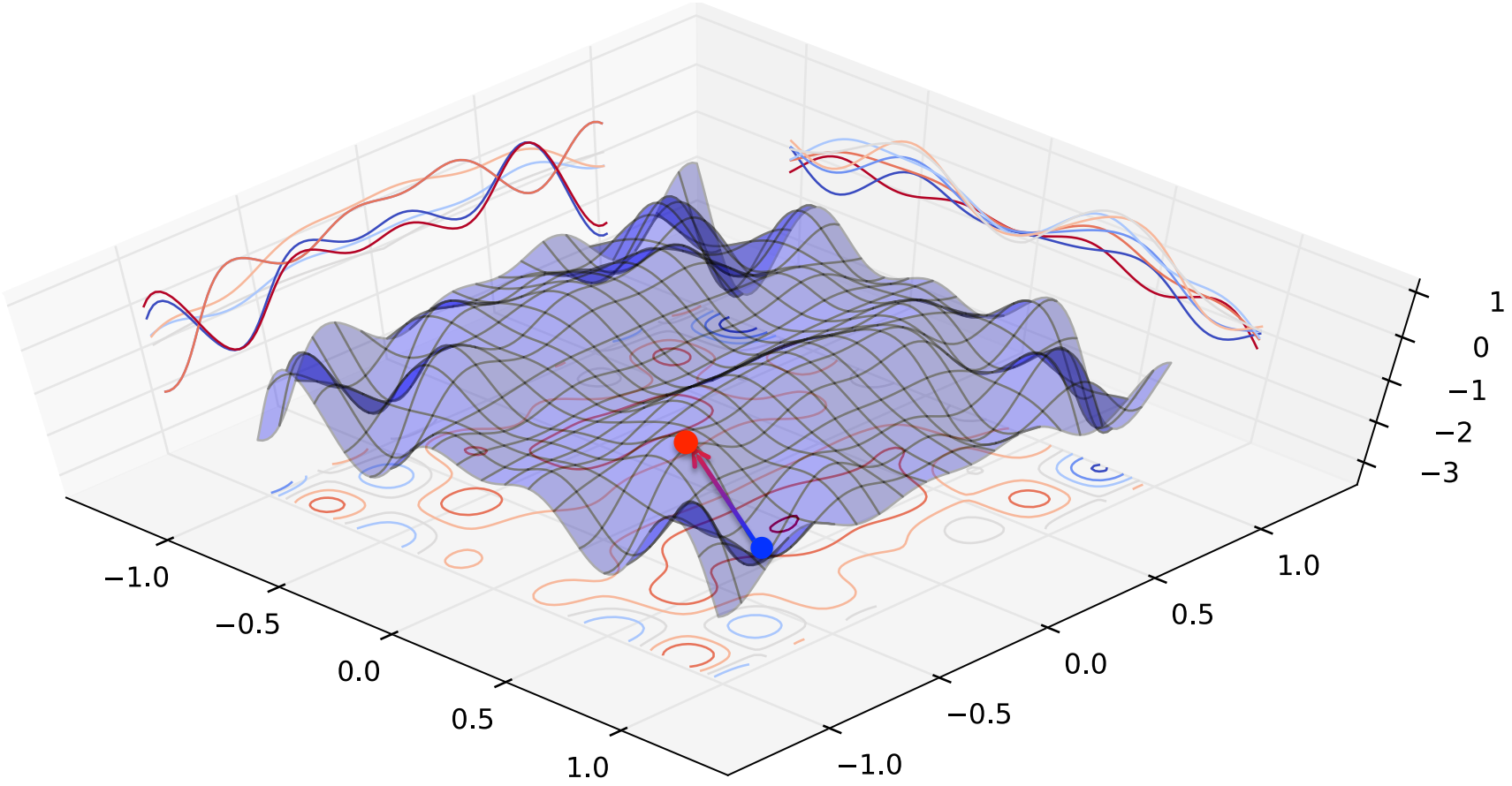
Background: Local Search for Grammar Induction

The problem of local optima.



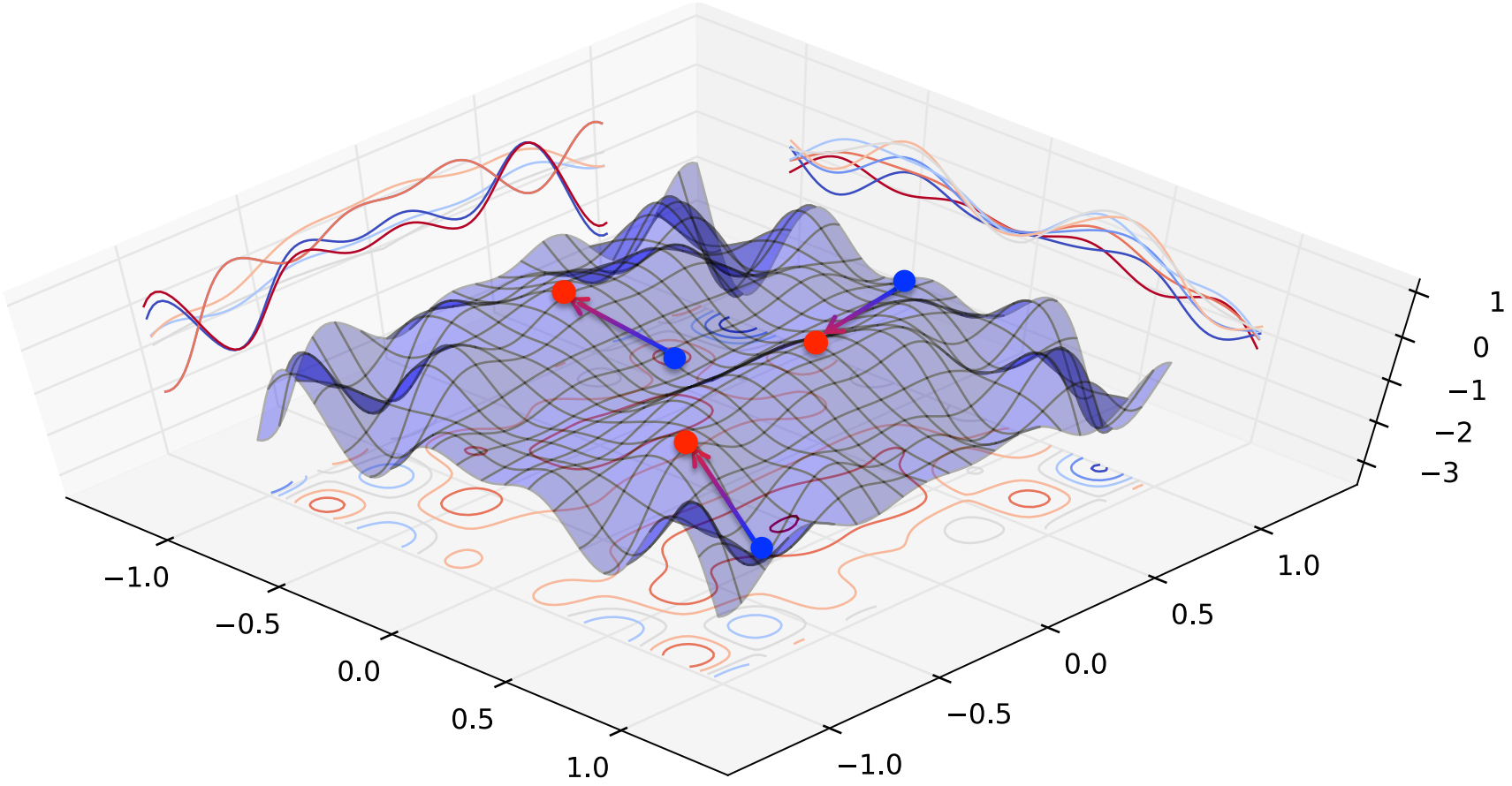
Background: Local Search for Grammar Induction

The problem of local optima.



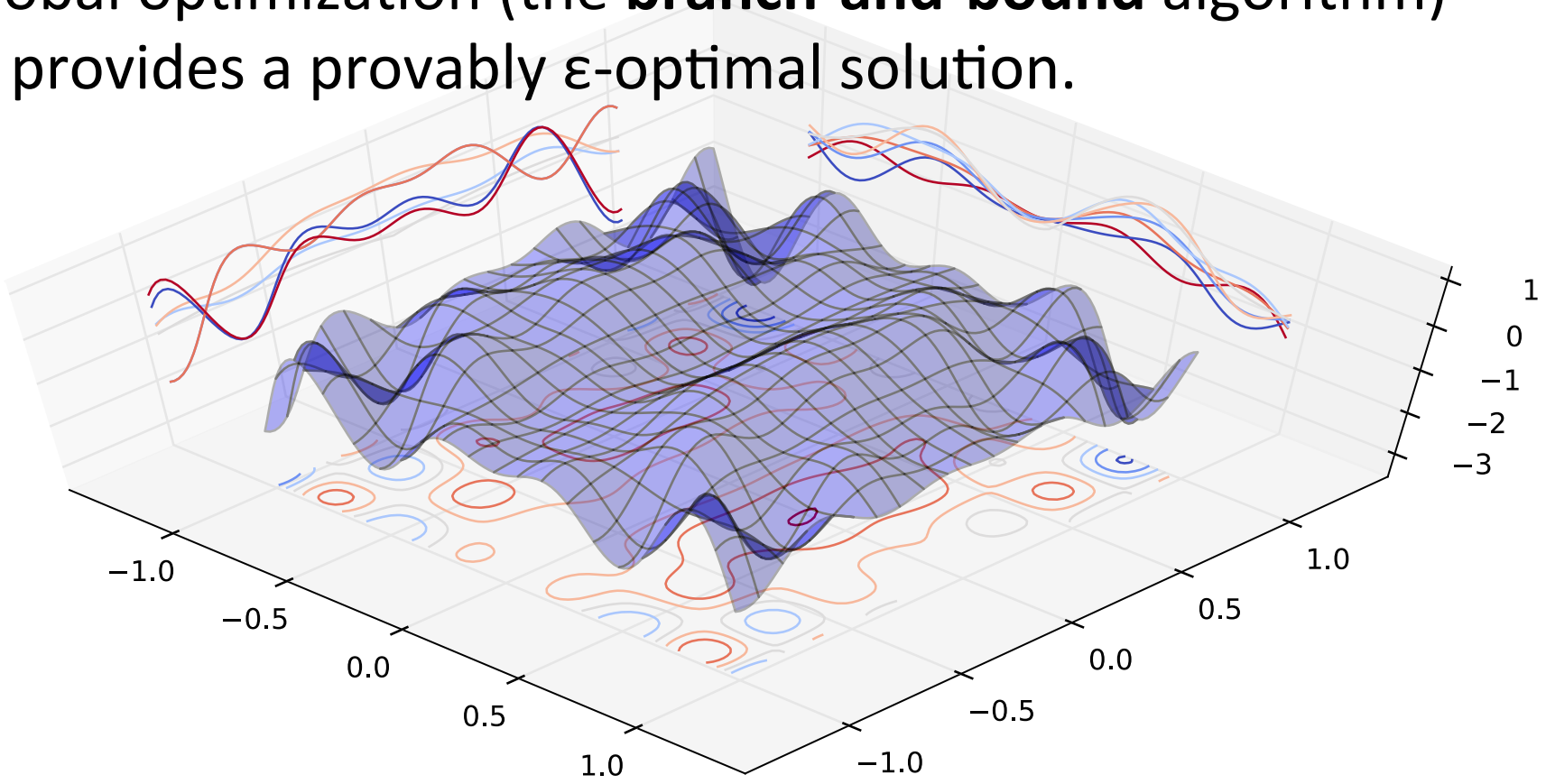
Background: Local Search for Grammar Induction

The problem of local optima.



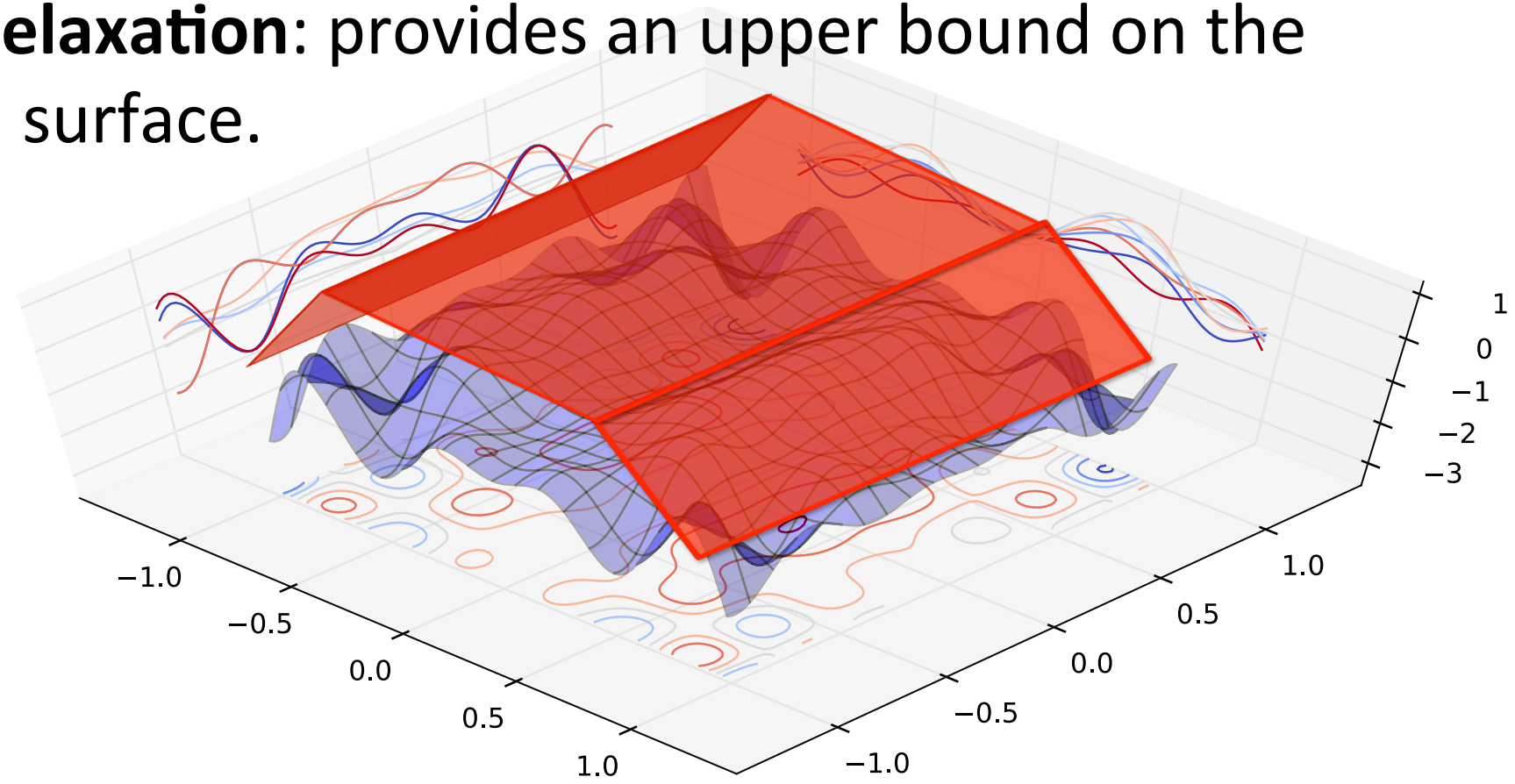
Background: Nonconvex Global Optimization

Global optimization (the **branch-and-bound** algorithm) provides a provably ε-optimal solution.



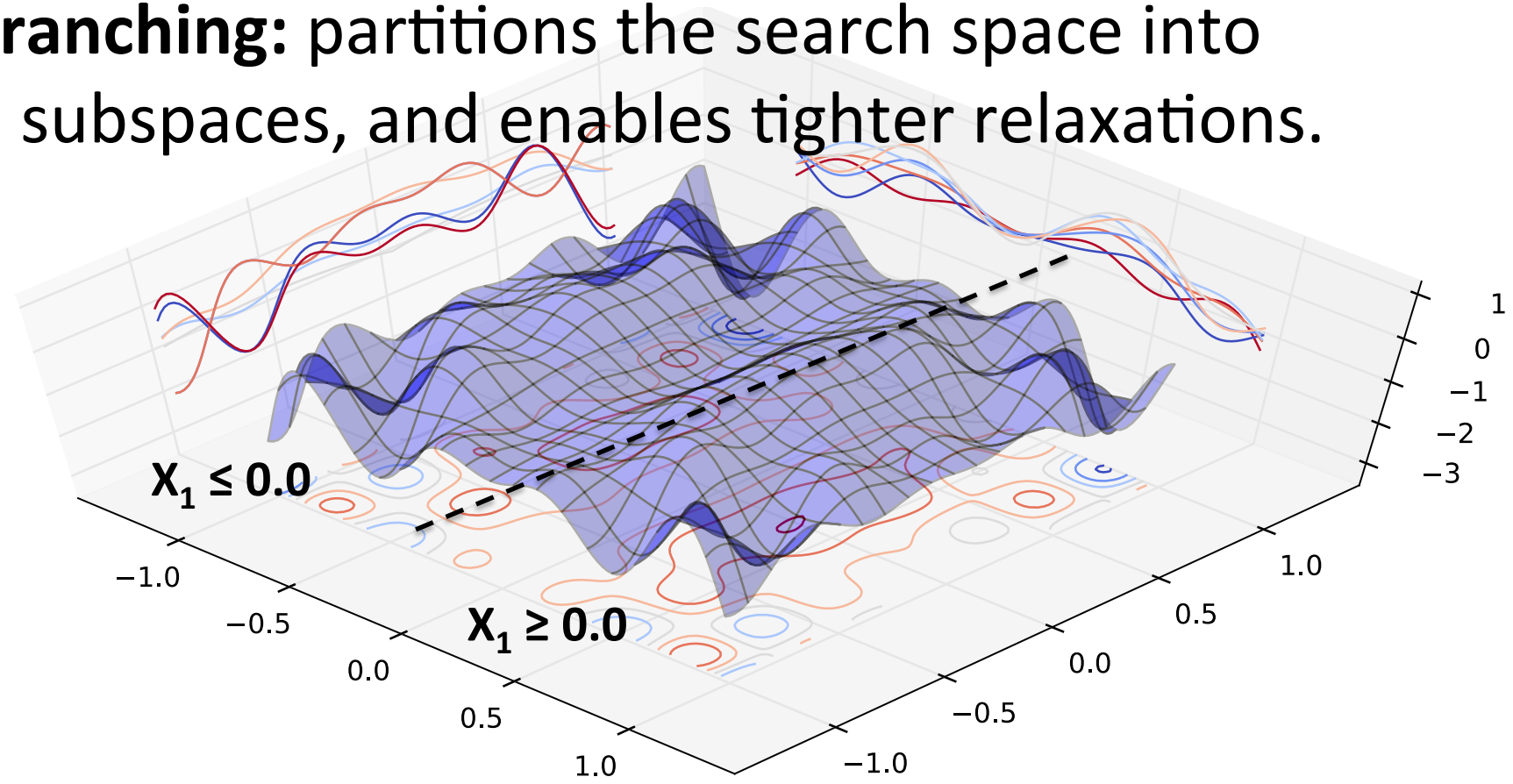
Background: Nonconvex Global Optimization

Relaxation: provides an upper bound on the surface.



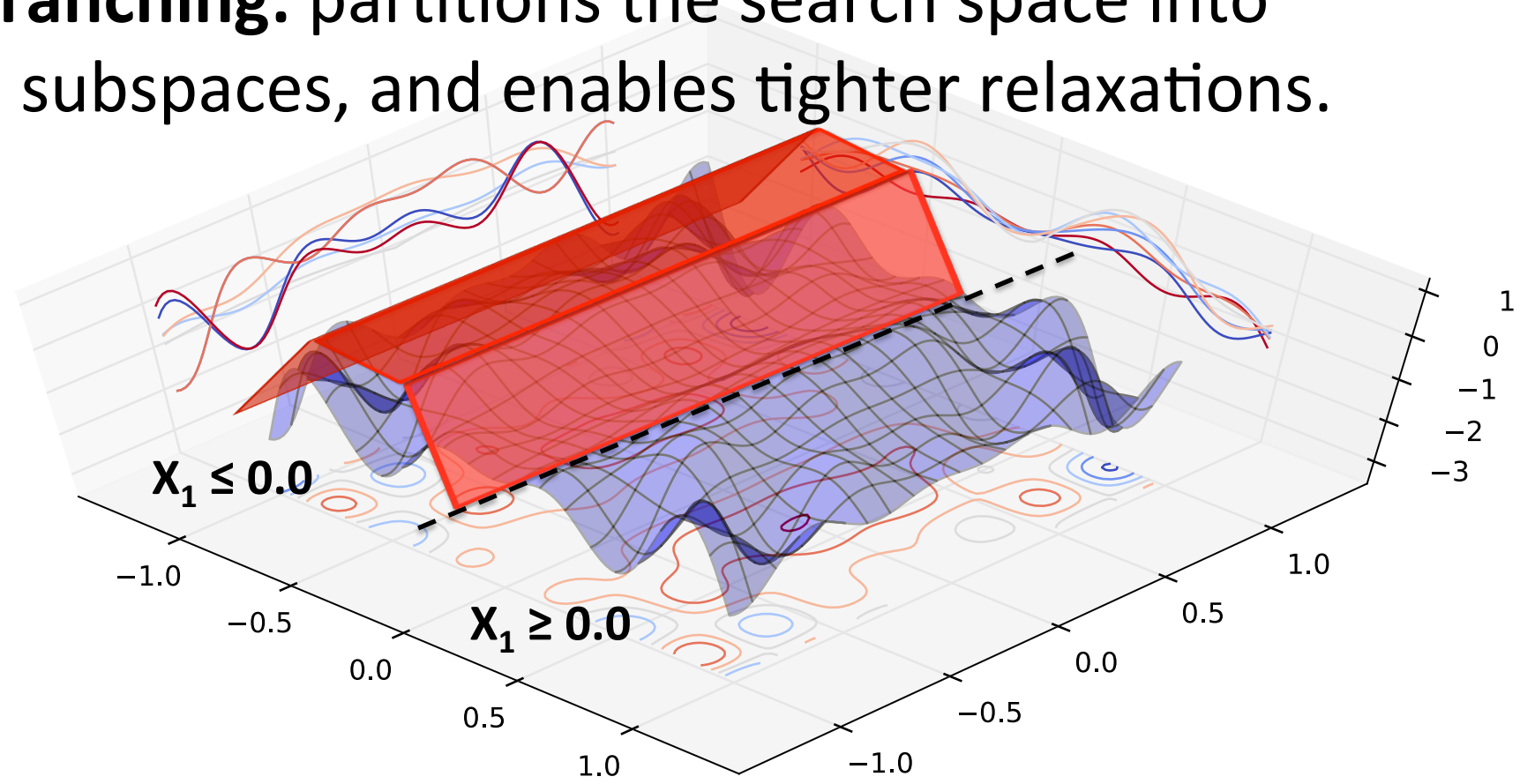
Background: Nonconvex Global Optimization

Branching: partitions the search space into subspaces, and enables tighter relaxations.



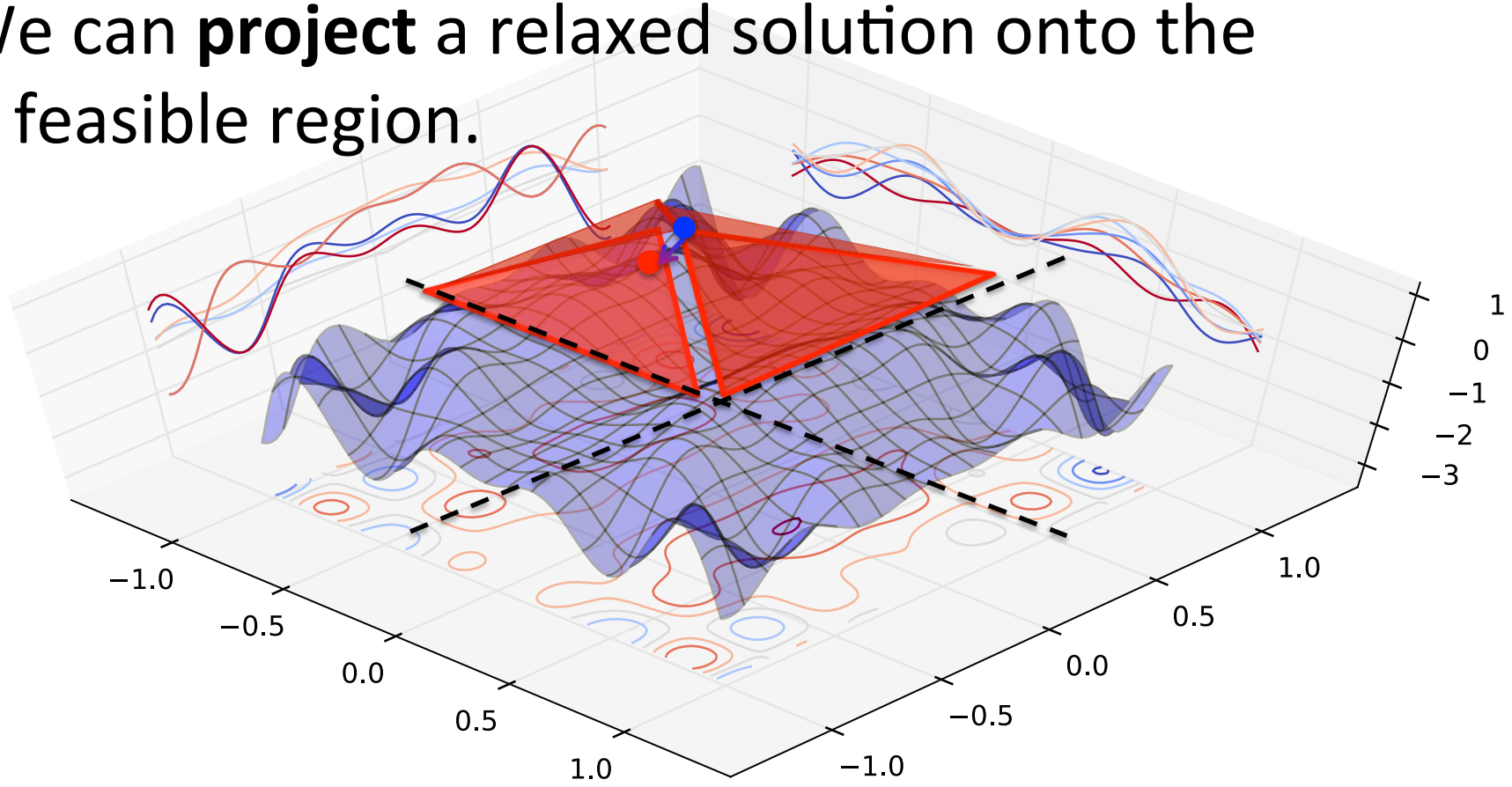
Background: Nonconvex Global Optimization

Branching: partitions the search space into subspaces, and enables tighter relaxations.



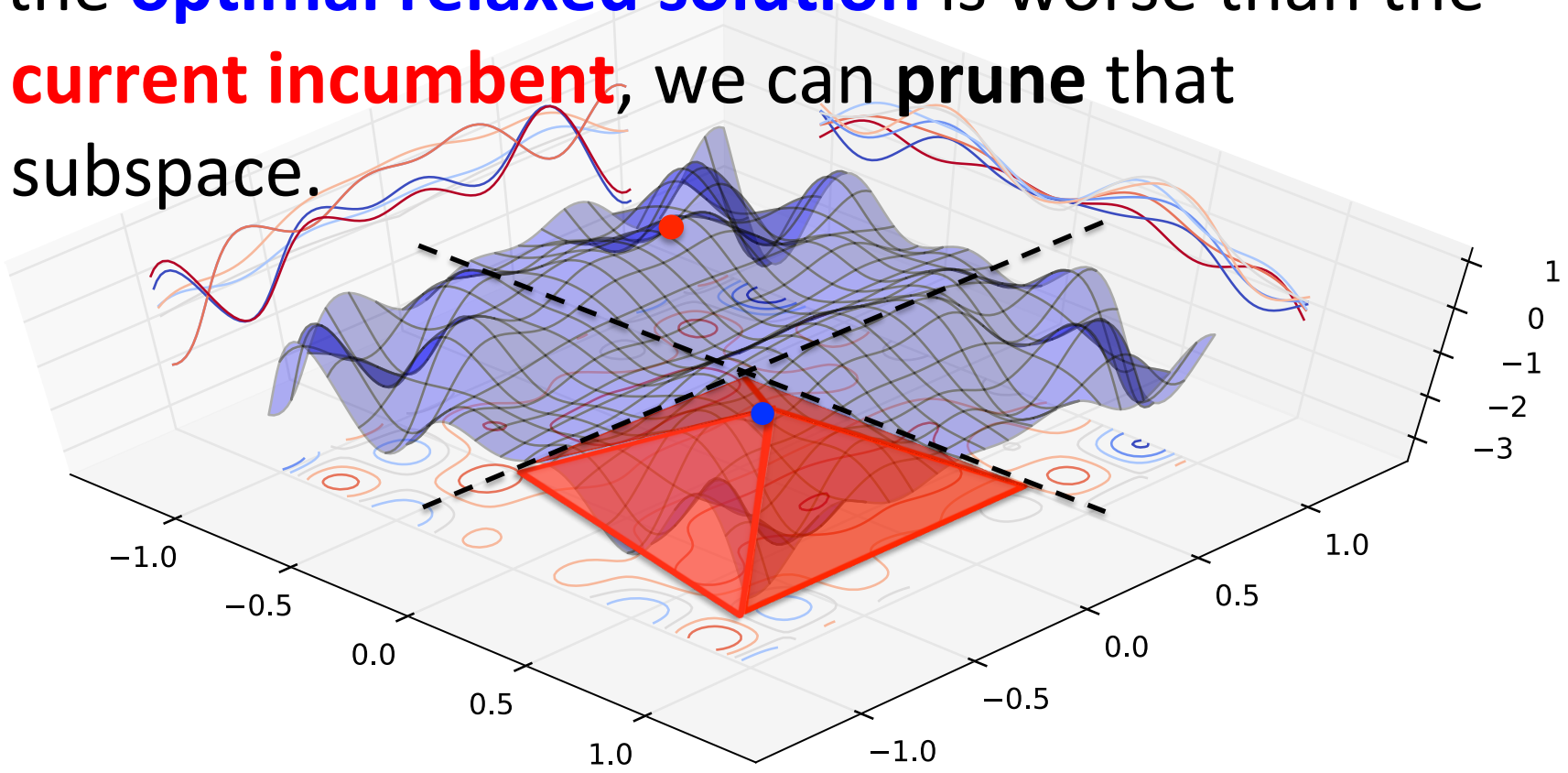
Background: Nonconvex Global Optimization

We can **project** a relaxed solution onto the feasible region.



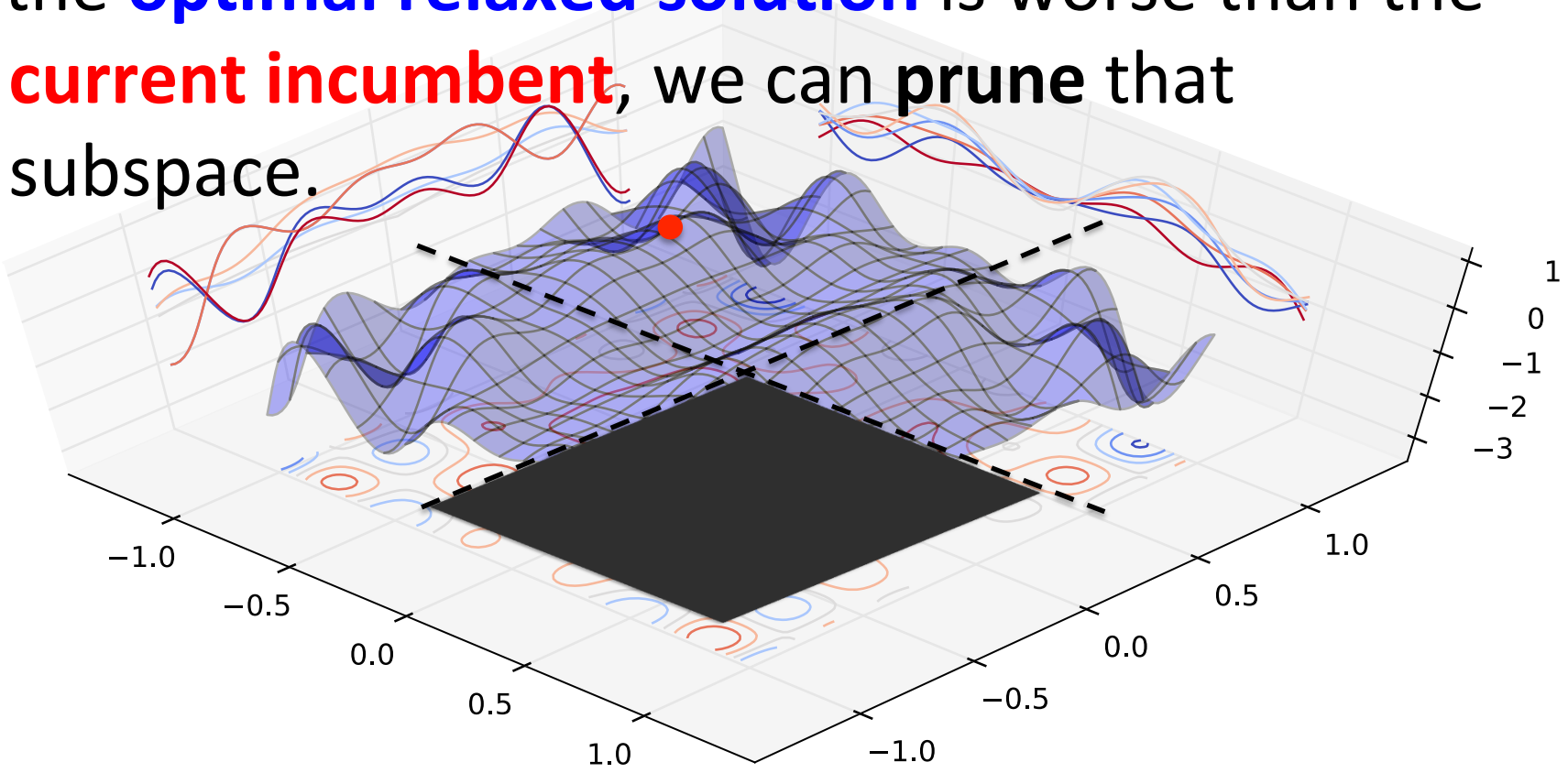
Background: Nonconvex Global Optimization

If the **optimal relaxed solution** is worse than the **current incumbent**, we can **prune** that subspace.



Background: Nonconvex Global Optimization

If the **optimal relaxed solution** is worse than the **current incumbent**, we can **prune** that subspace.



Overview

I. The Problem: Nonconvexity
II. Example Task: Grammar Induction
III. Background
IV. Search Methods
    A. Nonconvex Global Optimization
        i. Grammar Induction as a Mathematical Program
        ii. Relaxations
        iii. Projections
    B. Posterior Constraints
    C. Relaxed Viterbi EM
V. Experiments

Grammar Induction as a Mathematical Program

- Relaxations are typically linear programs (LP).

convex

$$\text{LP:} \quad \max c^T x \quad \text{s.t. } Ax \leq b$$

nonconvex

$$\text{QP:} \quad \max x^T Q x \quad \text{s.t. } Ax \leq b$$

$$\text{NLQP:} \quad \max x^T Q x \quad \text{s.t. } Ax \leq b, \;\; f(x) \leq e$$

- We will:
  1. Define grammar induction as an NLQP.
  2. Relax it to an LP, which can be solved by Simplex.

Grammar Induction as a Mathematical Program

Variables:

θ_m	Log-probability for feature m
f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

Indices and constants:

m	Feature / model parameter index
s	Sentence index
c	Conditional distribution index
$\mathcal{M}_c$	c^{th} Set of feature indices that sum to 1.0

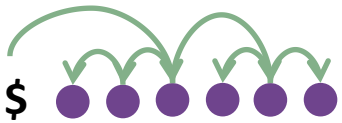
Viterbi EM objective in log space.

$$\max \sum_m \theta_m f_m$$

Sum-to-one constraints on model parameters.

$$\text{s.t.} \sum_{m \in \mathcal{M}_c} \exp(\theta_m) = 1, \forall c$$

Tree constraints.



$$A \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{e} \end{bmatrix} \leq b$$

Parameters must be log-probabilities.

$$\theta_m \leq 0, \forall m$$

Feature counts must be integers.

$$f_m, e_{sij} \in \mathbb{Z}, \forall m, s, i, j$$

Dependency Tree Constraints

$$A \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{e} \end{bmatrix} \leq b$$

- Edges form a **spanning tree**

Single-commodity flow (Magnanti & Wolsey, 1994)

$$\sum_{j=1}^{N_s} \phi_{s0j} = N_s, \forall j \quad (21)$$

$$\sum_{i=0}^{N_s} \phi_{sij} - \sum_{k=1}^{N_s} \phi_{sjk} = 1, \forall j \quad (22)$$

$$\phi_{sij} \leq N_s e_{sij}, \forall i, j \quad (23)$$

$$e_{sij} \in \{0, 1\}, \forall i, j \quad (24)$$

- Spanning tree is **projective**

Projectivity (Martins et al., 2009)

$$\sum_{(k,l) \in \mathcal{X}_{ij}} e_{skl} \leq N_s(1 - e_{sij}) \quad (25)$$

- Valid **feature counts** for the Dependency Model with Valence (DMV)

DMV root/child feature counts

$$f_{\text{root},t} = \sum_{s=1}^{N_s} \sum_{j \in \mathcal{W}_{st}} e_{s0j}, \forall t \quad (26)$$

$$f_{\text{child.L},t,t'} = \sum_{s=1}^{N_s} \sum_{j<i} \delta \begin{bmatrix} i \in \mathcal{W}_{st} \wedge \\ j \in \mathcal{W}_{st'} \end{bmatrix} e_{sij}, \forall t, t' \quad (27)$$

DMV decision feature counts

$$n_{s,i,l} = \sum_{j=1}^{i-1} e_{sij} \quad (28)$$

$$n_{s,i,l}/N_s \leq f^{(s,i)}_{\text{dec.L.0},t,\texttt{cont}} \leq 1 \quad (29)$$

$$f^{(s,i)}_{\text{dec.L.0},t,\texttt{stop}} = 1 - f^{(s,i)}_{\text{dec.L.0},t,\texttt{cont}} \quad (30)$$

$$f^{(s,i)}_{\text{dec.L.}\geq 1,t,\texttt{stop}} = f^{(s,i)}_{\text{dec.L.0},t,\texttt{cont}} \quad (31)$$

$$f^{(s,i)}_{\text{dec.L.}\geq 1,t,\texttt{cont}} = n_{s,i,l} - f^{(s,i)}_{\text{dec.L.0},t,\texttt{cont}} \quad (32)$$

Relaxations

- Three separate steps:
  1. Relax the **nonlinear** sum-to-one constraints
  2. Relax the **integer** constraints
  3. "Relax" the **quadratic** objective
- Resulting relaxation will be an LP

Relaxing the Sum-to-one Constraints

Variables:

θ_m	Log-probability for feature m
f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

Indices and constants:

m	Feature / model parameter index
s	Sentence index
c	Conditional distribution index
$\mathcal{M}_c$	c^{th} Set of feature indices that sum to 1.0

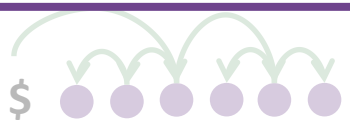
Viterbi EM objective in log space.

$$\max \sum_m \theta_m f_m$$

Sum-to-one constraints on model parameters.

$$\text{s.t.} \sum_{m \in \mathcal{M}_c} \exp(\theta_m) = 1, \forall c$$

Tree constraints.



$$A \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{e} \end{bmatrix} \leq b$$

Parameters must be log-probabilities.

$$\theta_m \leq 0, \forall m$$

Feature counts must be integers.

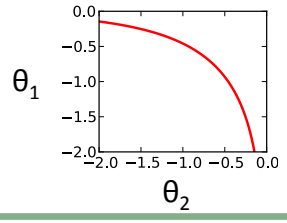
$$f_m, e_{sij} \in \mathbb{Z}, \forall m, s, i, j$$

Relaxing the Sum-to-one Constraints

Example plots of two parameter case:

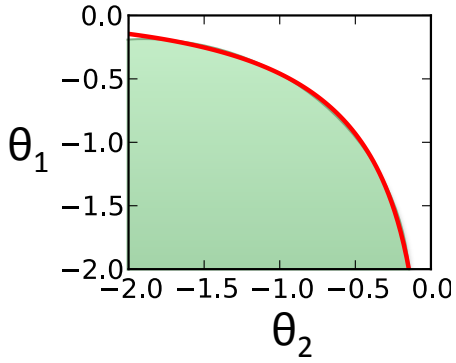
1. Original nonlinear constraint:

$$\sum_{m \in \mathcal{M}_c} \exp(\theta_m) = 1$$



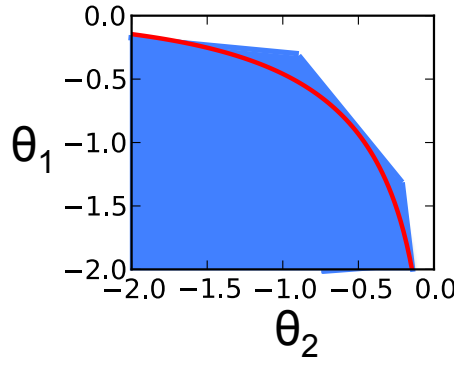
2. Nonlinear relaxation:

$$\sum_{m \in \mathcal{M}_c} \exp(\theta_m) \leq 1$$



3. Linear relaxation:

$$\sum_{m \in \mathcal{M}_c} \left(\theta_m + 1 - \hat{\theta}_{c,m}^{(i)}\right) \exp\left(\hat{\theta}_{c,m}^{(i)}\right) \leq 1$$



Relaxing the Integer Constraints

Variables:

θ_m	Log-probability for feature m
f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

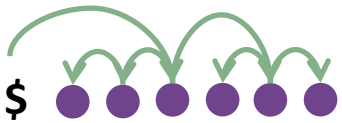
Indices and constants:

m	Feature / model parameter index
s	Sentence index
c	Conditional distribution index
$\mathcal{M}_c$	c^{th} Set of feature indices that sum to 1.0

Viterbi EM objective in log space.

Relaxed linear sum-to-one constraints on model parameters.

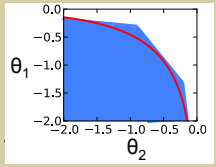
Tree constraints.



Parameters must be log-probabilities.

Feature counts must be integers.

$$\max \sum_m \theta_m f_m$$

$$\text{s.t.} \quad \text{  } \quad , \forall c$$

$$A \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{e} \end{bmatrix} \leq b$$

$$\theta_m \leq 0, \forall m$$

$$f_m, e_{sij} \in \mathbb{Z}, \forall m, s, i, j$$

Relaxing the Integer Constraints

Variables:

θ_m	Log-probability for feature m
f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

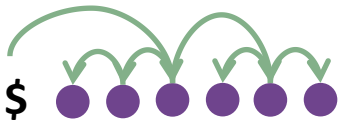
Indices and constants:

m	Feature / model parameter index
s	Sentence index
c	Conditional distribution index
$\mathcal{M}_c$	c^{th} Set of feature indices that sum to 1.0

Viterbi EM objective in log space.

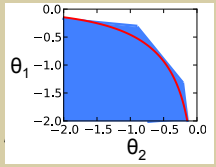
Relaxed linear sum-to-one constraints on model parameters.

Tree constraints.



Each B&B subspace specifies bounds.

$$\max \sum_m \theta_m f_m$$

$$\text{s.t.} \quad \text{  } \quad , \forall c$$

$$A \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{e} \end{bmatrix} \leq b$$

$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$

$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

“Relaxing” the Objective

Variables:

θ_m	Log-probability for feature m
f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

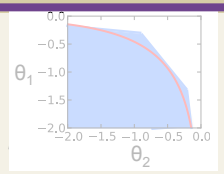
Indices and constants:

m	Feature / model parameter index
s	Sentence index
c	Conditional distribution index
$\mathcal{M}_c$	c^{th} Set of feature indices that sum to 1.0

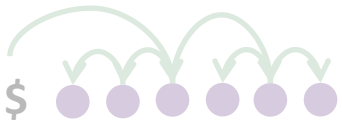
Viterbi EM objective in log space.

$$\max \sum_m \theta_m f_m$$

Relaxed linear sum-to-one constraints on model parameters.

$$\text{s.t.} \quad \text{  } \quad , \forall c$$

Tree constraints.



$$A \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{e} \end{bmatrix} \leq b$$

Each B&B subspace specifies bounds.

$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$

$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

“Relaxing” the Objective

Original **quadratic** objective:	Each B&B subspace specifies bounds:
$\max \sum_m \theta_m f_m$	$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$ $f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$

- We explore two relaxations:
  1. Concave envelope (e.g. McCormick (1976))
  2. Reformulation Linearization Technique (RLT) (Sherali & Adams, 1990)

“Relaxing” the Objective

Original nonconvex **quadratic** objective:

$$\max \sum_m \theta_m f_m$$

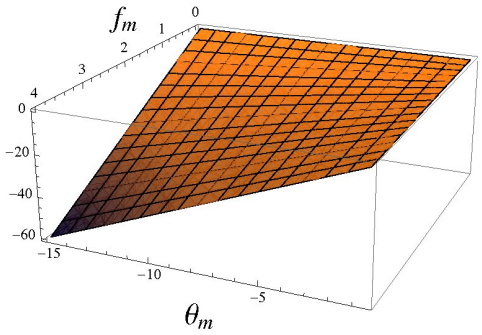
Each B&B subspace specifies bounds:

$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$

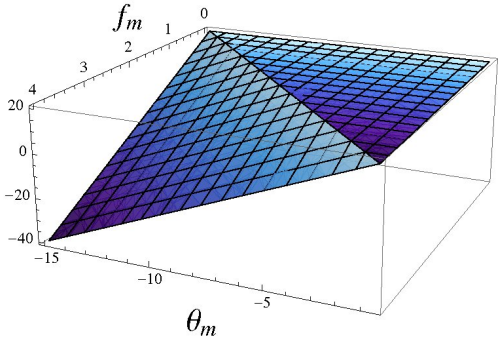
$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

Concave Envelope (e.g. McCormick (1976))

$$\theta_m f_m \leq \min[f_m^{\max}\theta_m + \theta_m^{\min} f_m - \theta_m^{\min} f_m^{\max},$$
$$f_m^{\min}\theta_m + \theta_m^{\max} f_m - \theta_m^{\max} f_m^{\min}]$$



Example plots for a single quadratic term.



Relaxed convex **linear** objective:

$$\max \sum_m z_m$$
$$\text{s.t. } z_m \leq f_m^{\max}\theta_m + \theta_m^{\min} f_m - \theta_m^{\min} f_m^{\max}$$
$$z_m \leq f_m^{\min}\theta_m + \theta_m^{\max} f_m - \theta_m^{\max} f_m^{\min}$$

“Relaxing” the Objective

Concave Envelope
(e.g. McCormick (1976))

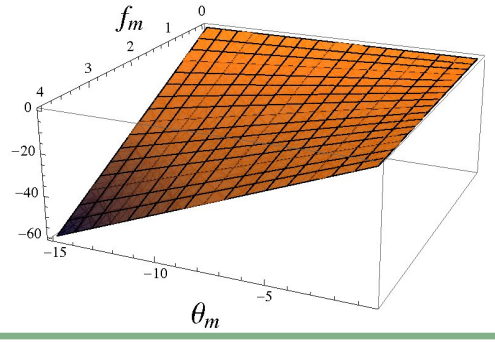
Each B&B subspace specifies bounds:

$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$

$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

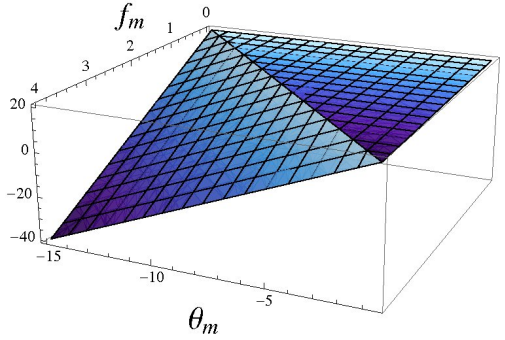
Original nonconvex **quadratic** objective:

$$\max \sum_m$$



Relaxed convex **linear** objective:

$$\max \sum_m$$



“Relaxing” the Objective

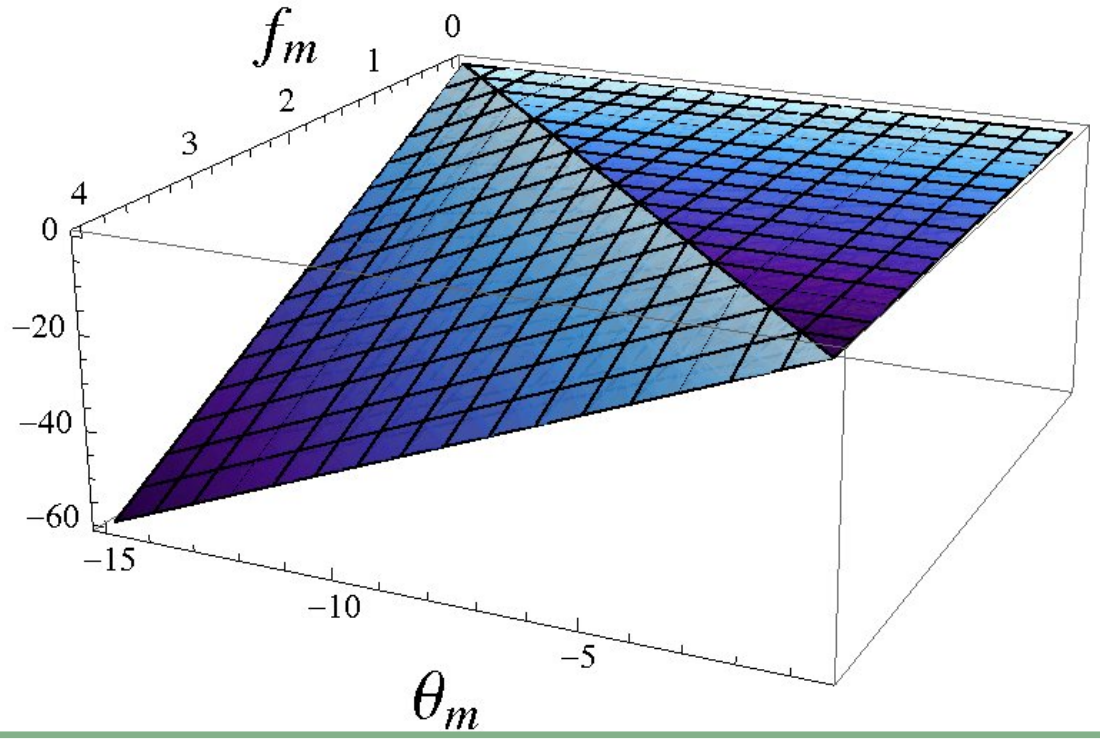
Concave Envelope
(e.g. McCormick (1976))

Each B&B subspace specifies bounds:

$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$

$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

Tightness of relaxation improves with **branching**.



“Relaxing” the Objective

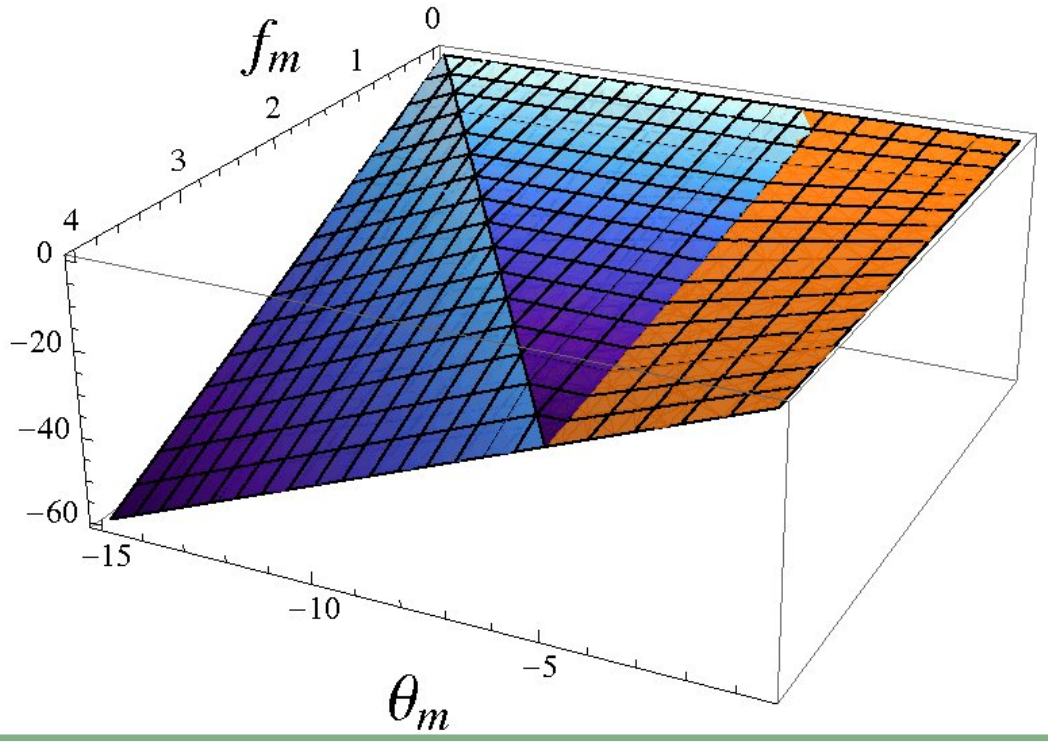
Concave Envelope
(e.g. McCormick (1976))

Each B&B subspace specifies bounds:

$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$

$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

Tightness of relaxation improves with **branching**.



“Relaxing” the Objective

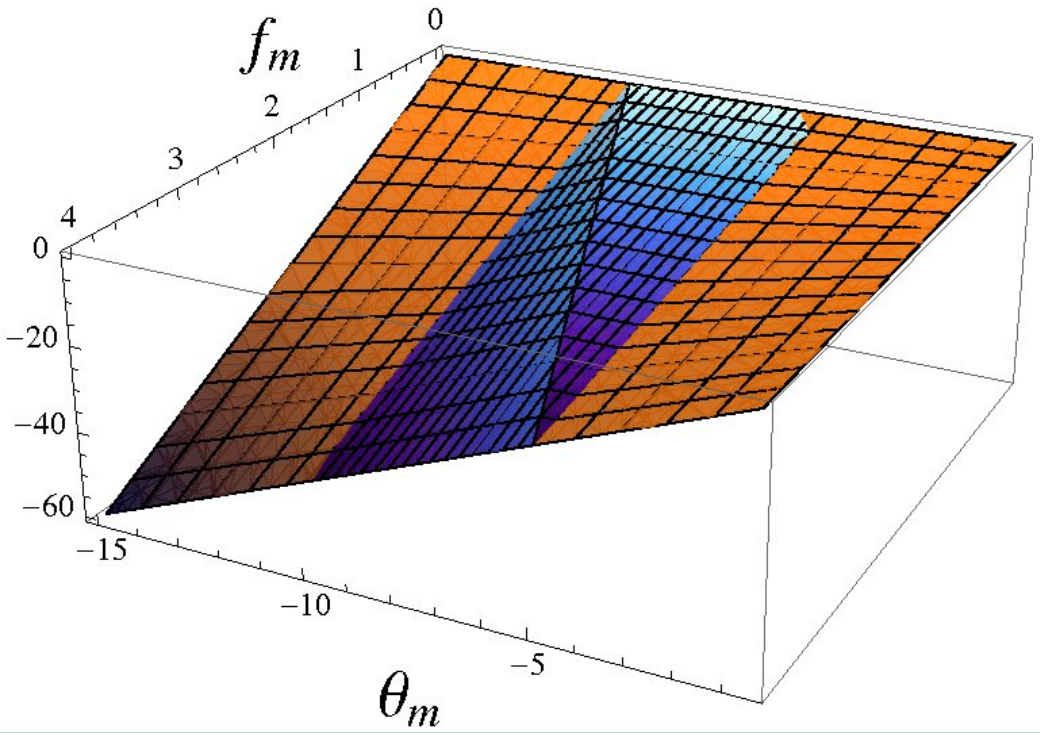
Concave Envelope
(e.g. McCormick (1976))

Each B&B subspace specifies bounds:

$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$

$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

Tightness of relaxation improves with **branching**.



“Relaxing” the Objective

Variables:

θ_m	Log-probability for feature m
f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

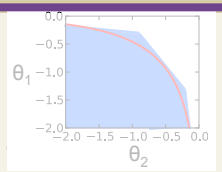
Indices and constants:

m	Feature / model parameter index
s	Sentence index
c	Conditional distribution index
$\mathcal{M}_c$	c^{th} Set of feature indices that sum to 1.0

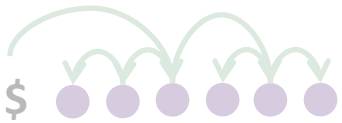
Viterbi EM objective in log space.

$$\max \sum_m \theta_m f_m$$

Relaxed linear sum-to-one constraints on model parameters.

$$\text{s.t.} \quad \text{  } \quad , \forall c$$

Tree constraints.



$$A \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{e} \end{bmatrix} \leq b$$

Each B&B subspace specifies bounds.

$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$

$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

Nonconvex Quadratic Relaxation for Grammar Induction

Variables:

θ_m	Log-probability for feature m
f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

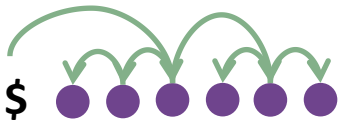
Indices and constants:

m	Feature / model parameter index
s	Sentence index
c	Conditional distribution index
$\mathcal{M}_c$	c^{th} Set of feature indices that sum to 1.0

Viterbi EM objective in log space.

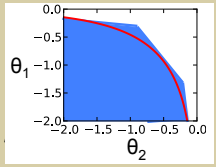
Relaxed linear sum-to-one constraints on model parameters.

Tree constraints.



Each B&B subspace specifies bounds.

$$\max \sum_m \theta_m f_m$$

$$\text{s.t.} \quad \text{  } \quad , \forall c$$

$$A \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{e} \end{bmatrix} \leq b$$

$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$

$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

Nonconvex Quadratic Relaxation for Grammar Induction

Variables:

θ_m	Log-probability for feature m
f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

Indices and constants:

m	Feature / model parameter index
s	Sentence index
c	Conditional distribution index
$\mathcal{M}_c$	c^{th} Set of feature indices that sum to 1.0

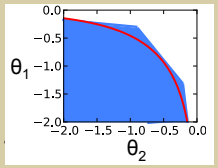
Rewrite as a program with variables x.

$$\max\ x^T Q x$$

$$\text{s.t. } Gx \leq g$$

$$\max \sum_m \theta_m f_m$$

$$\text{s.t. } \quad , \forall c$$



$$A \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{e} \end{bmatrix} \leq b$$

$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$

$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

“Relaxing” the Objective

Reformulation Linearization Technique (RLT)
(Sherali & Adams, 1990)

Original QP: $\max\ x^T Q x$

$\text{s.t. } Gx \leq g$

1. **Rewrite** step:
   - Replace all quadratic terms with auxiliary variables.
   - $w_{ij} \equiv x_i x_j$
2. **Reformulation** step:
   - Add all possible products of the linear constraints.
   - $(g_i - G_i x)(g_j - G_j x) \geq 0$
3. **Linearization** step:
   - Remove quadratic constraints
   - $w_{ij} \equiv x_i x_j$

“Relaxing” the Objective

Reformulation Linearization Technique (RLT)
(Sherali & Adams, 1990)

1. **Rewrite** step:
   - Replace all quadratic terms with auxiliary variables.
   - $w_{ij} \equiv x_i x_j$
2. **Reformulation** step:
   - Add all possible products of the linear constraints.
   - $(g_i - G_i x)(g_j - G_j x) \geq 0$
3. **Linearization** step:
   - Remove quadratic constraints
   - $w_{ij} \equiv x_i x_j$

Original QP:

$$\max\ x^T Q x$$
$$\text{s.t. } Gx \leq g$$

Original QP:

$$\max \sum_{1 \leq i \leq j \leq n} Q_{ij} x_i x_j$$
$$\text{s.t. } (g_i - G_i x) \geq 0,$$
$$\forall 1 \leq i \leq m$$

“Relaxing” the Objective

Reformulation Linearization Technique (RLT)
(Sherali & Adams, 1990)

1. **Rewrite** step:
   - Replace all quadratic terms with auxiliary variables.
   - $w_{ij} \equiv x_i x_j$
2. **Reformulation** step:
   - Add all possible products of the linear constraints.
   - $(g_i - G_i x)(g_j - G_j x) \geq 0$
3. **Linearization** step:
   - Remove quadratic constraints
   - $w_{ij} \equiv x_i x_j$

Original QP:

$$\max\ x^T Q x$$
$$\text{s.t. } Gx \leq g$$

Step 1:

$$\max \sum_{1 \leq i \leq j \leq n} Q_{ij} w_{ij}$$
$$\text{s.t. } (g_i - G_i x) \geq 0,$$
$$\forall 1 \leq i \leq m$$
$$w_{ij} = x_i x_j$$

“Relaxing” the Objective

Reformulation Linearization Technique (RLT)
(Sherali & Adams, 1990)

Original QP:

$$\max\ x^T Q x$$
$$\text{s.t. } Gx \leq g$$

1. **Rewrite** step:
   - Replace all quadratic terms with auxiliary variables.
   - $w_{ij} \equiv x_i x_j$
2. **Reformulation** step:
   - Add all possible products of the linear constraints.
   - $(g_i - G_i x)(g_j - G_j x) \geq 0$
3. **Linearization** step:
   - Remove quadratic constraints
   - $w_{ij} \equiv x_i x_j$

Step 2:

$$\max \sum_{1 \leq i \leq j \leq n} Q_{ij} w_{ij}$$
$$\text{s.t. } (g_i - G_i x)(g_j - G_j x) \geq 0,$$
$$\forall 1 \leq i \leq j \leq m$$
$$w_{ij} = x_i x_j$$

“Relaxing” the Objective

Reformulation Linearization Technique (RLT)
(Sherali & Adams, 1990)

Original QP: $\max\ x^T Q x$

$$\text{s.t. } Gx \leq g$$

1. **Rewrite** step:
   - Replace all quadratic terms with auxiliary variables.
   - $w_{ij} \equiv x_i x_j$
2. **Reformulation** step:
   - Add all possible products of the linear constraints.
   - $(g_i - G_i x)(g_j - G_j x) \geq 0$
3. **Linearization** step:
   - Remove quadratic constraints
   - $w_{ij} \equiv x_i x_j$

Step 2 (expanded):

$$\max \sum_{1 \leq i \leq j \leq n} Q_{ij} w_{ij}$$

$$\text{s.t. } g_i g_j - \sum_{k=1}^{n} g_j G_{ik} x_k - \sum_{k=1}^{n} g_i G_{jk} x_k + \sum_{k=1}^{n} \sum_{l=1}^{n} G_{ik} G_{jl} w_{kl} \geq 0,$$

$$\forall 1 \leq i \leq j \leq m$$

$$w_{ij} = x_i x_j$$

“Relaxing” the Objective

Reformulation Linearization Technique (RLT)
(Sherali & Adams, 1990)

1. **Rewrite** step:
   - Replace all quadratic terms with auxiliary variables.
   - $w_{ij} \equiv x_i x_j$
2. **Reformulation** step:
   - Add all possible products of the linear constraints.
   - $(g_i - G_i x)(g_j - G_j x) \geq 0$
3. **Linearization** step:
   - Remove quadratic constraints
   - $w_{ij} \equiv x_i x_j$

Original QP:

$$\max\ x^T Q x$$
$$\text{s.t. } Gx \leq g$$

Step 3:

$$\max \sum_{1 \leq i \leq j \leq n} Q_{ij} w_{ij}$$

$$\text{s.t. } g_i g_j - \sum_{k=1}^{n} g_j G_{ik} x_k - \sum_{k=1}^{n} g_i G_{jk} x_k + \sum_{k=1}^{n} \sum_{l=1}^{n} G_{ik} G_{jl} w_{kl} \geq 0,$$
$$\forall 1 \leq i \leq j \leq m$$

“Relaxing” the Objective

Reformulation Linearization Technique (RLT)
(Sherali & Adams, 1990)

1. **Rewrite** step:
   - Replace all quadratic terms with auxiliary variables.
   - $w_{ij} \equiv x_i x_j$
2. **Reformulation** step:
   - Add all possible products of the linear constraints.
   - $(g_i - G_i x)(g_j - G_j x) \geq 0$
3. **Linearization** step:
   - Remove quadratic constraints
   - $w_{ij} \equiv x_i x_j$

Original QP:

$$\max\ x^T Q x$$
$$\text{s.t. } Gx \leq g$$

RLT LP:

$$\max \sum_{1 \leq i \leq j \leq n} Q_{ij} w_{ij}$$

$$\text{s.t. } g_i g_j - \sum_{k=1}^{n} g_j G_{ik} x_k - \sum_{k=1}^{n} g_i G_{jk} x_k + \sum_{k=1}^{n} \sum_{l=1}^{n} G_{ik} G_{jl} w_{kl} \geq 0,$$
$$\forall 1 \leq i \leq j \leq m$$

“Relaxing” the Objective

Reformulation Linearization Technique (RLT)
(Sherali & Adams, 1990)

- Theoretical Properties
  - The concave envelope is a formed by a subset of the RLT constraints.
  - The original linear constraints are fully enforced by the resulting RLT constraints (Sherali & Tuncbilek, 1995).
  - The reformulation step can be applied repeatedly to produce polynomial constraints of higher degree.
    When $x \in \{0,1\}^n$, the degree-n RLT constraints will restrict to the convex hull of the feasible region (Sherali & Adams, 1990).
- Trade-off: tightness vs. size

Original QP:

$$\max\ x^T Q x$$
$$\text{s.t. } Gx \leq g$$

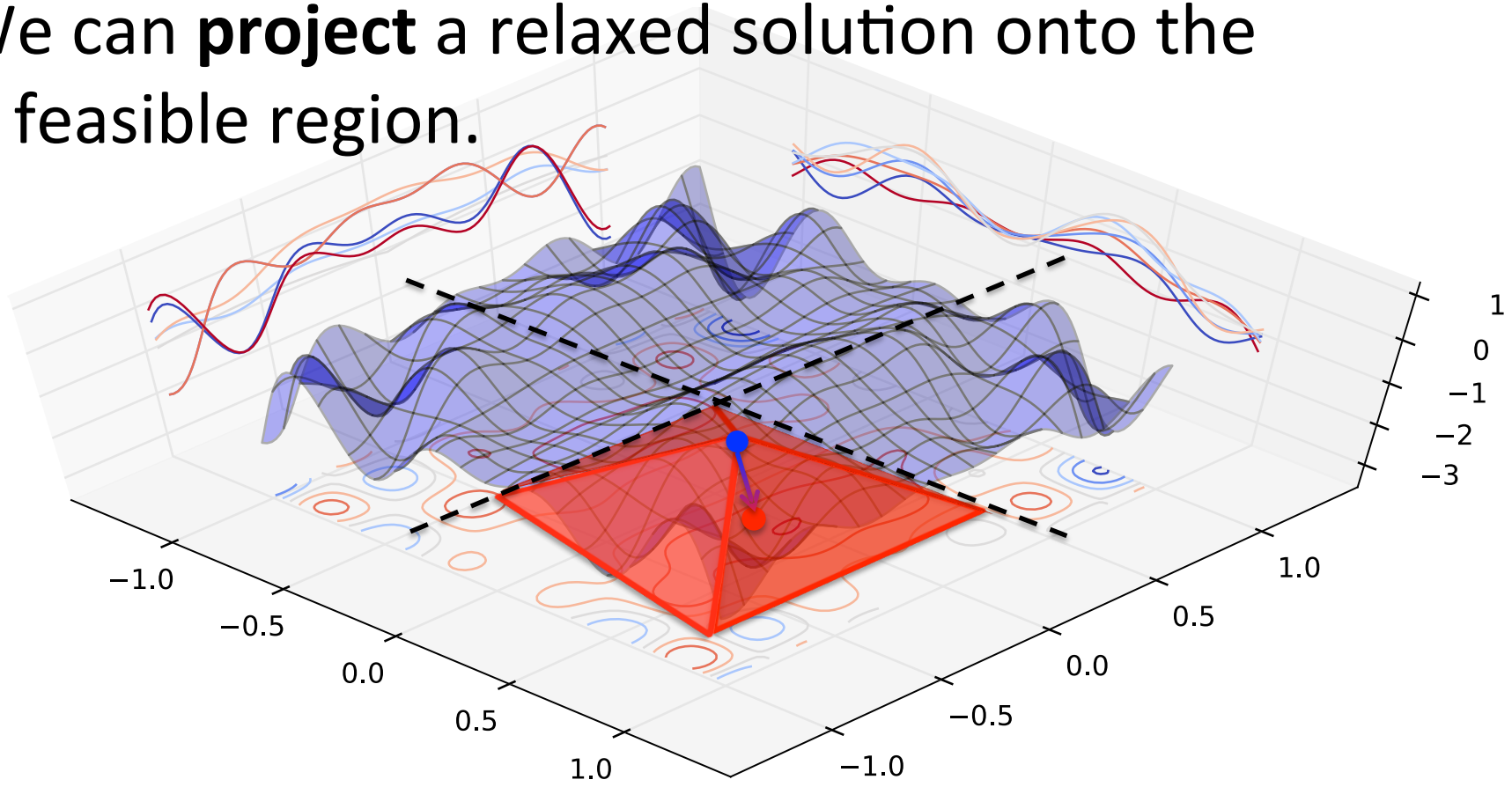
RLT LP:

$$\max \sum_{1 \leq i \leq j \leq n} Q_{ij} w_{ij}$$

$$\text{s.t. } g_i g_j - \sum_{k=1}^{n} g_j G_{ik} x_k - \sum_{k=1}^{n} g_i G_{jk} x_k + \sum_{k=1}^{n} \sum_{l=1}^{n} G_{ik} G_{jl} w_{kl} \geq 0,$$
$$\forall 1 \leq i \leq j \leq m$$

Background: Nonconvex Global Optimization

We can **project** a relaxed solution onto the feasible region.



Projections

- Model parameters
  - *In relaxed solution*: might sum to ≥ 1.0.
  - *Approaches*:
    - Normalize the parameters.
    - Find the point on the simplex that has minimum Euclidean distance (Chen & Ye, 2011)
- Parses
  - *In relaxed solution*: might have fractional edges.
  - *Approach*: Run a dynamic programming parser where the edge weights are given by the relaxed parse.

Overview

I. The Problem: Nonconvexity
II. Example Task: Grammar Induction
III. Background
IV. Search Methods
    A. Nonconvex Global Optimization
        i. Grammar Induction as a Mathematical Program
        ii. Relaxations
        iii. Projections
    B. Posterior Constraints
    C. Relaxed Viterbi EM with Posterior Constraints
V. Experiments

Posterior Constraints

- Posterior constraints allow us to incorporate our **linguistic knowledge** declaratively.
- Examples:
  - Dependencies are **mostly short** (Eisner & Smith, 2010).
  - Arcs do not often cross **punctuation** boundaries (Spitkovsky et al., 2012).
  - Most arc tokens are from the set $\mathcal{E}$ of "shiny" arc types. (Naseem et al., 2010).

Root → Auxiliary	Noun → Adjective
Root → Verb	Noun → Article
Verb → Noun	Noun → Noun
Verb → Pronoun	Noun → Numeral
Verb → Adverb	Preposition → Noun
Verb → Verb	Adjective → Adverb
Auxiliary → Verb	

Posterior Constraints

- In our experiments, we use a variant of the **universal linguistic constraint** of Naseem et al. (2010).
  - **Description:** 80% of arc tokens are from the set $\mathcal{E}$ of "shiny" arc types.

Root → Auxiliary	Noun → Adjective
Root → Verb	Noun → Article
Verb → Noun	Noun → Noun
Verb → Pronoun	Noun → Numeral
Verb → Adverb	Preposition → Noun
Verb → Verb	Adjective → Adverb
Auxiliary → Verb	

  - **Linear Constraint:** $\sum_{m \in \mathcal{E}} f_m \geq 0.8 \left(\sum_{s=1}^{S} N_s \right)$

Relaxed Viterbi EM with Posterior Constraints

- Use the standard **M-step**.
- Modify the **E-step**:
  - Add posterior constraints to the MILP parsing problem.
  - Solve the LP relaxation by removing integer constraints.
  - Project the relaxed solution to the feasible region.

Variables:

f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

Indices and constants:

m	Feature / model parameter index
s	Sentence index
θ_m	Log-probability for feature m

Linear Viterbi objective.

Tree constraints.

Integer feature counts.

Posterior constraints.

$$\max \sum_m \theta_m f_m$$

$$\text{s.t. } A \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{e} \end{bmatrix} \leq b$$

$$f_m, e_{sij} \in \mathbb{Z}, \forall m, s, i, j$$

$$\sum_{m \in \mathcal{E}} f_m \geq 0.8 \left(\sum_{s=1}^{S} N_s \right)$$

Related Work

- Convex objective functions
  - Gimpel and Smith (2012):
    - concave model for unsupervised dependency parsing, using IBM Model 1 to align a sentence with itself
    - initializer for EM
  - Wang et al. (2008):
    - combined unsupervised least squares loss and a supervised large margin loss
    - semi-supervised setting

- Spectral learning
  - Does not maximize the non-convex likelihood function
  - Instead, optimizes a different convex function which gives the same estimate in the infinite data limit
  - Works for HMMs, but **not for trees** if you don't already know the structure
  - Cohen et al. (2012):
    - supervised latent variable PCFGs
  - Luque et al. (2012)
    - supervised hidden-state dependency grammars

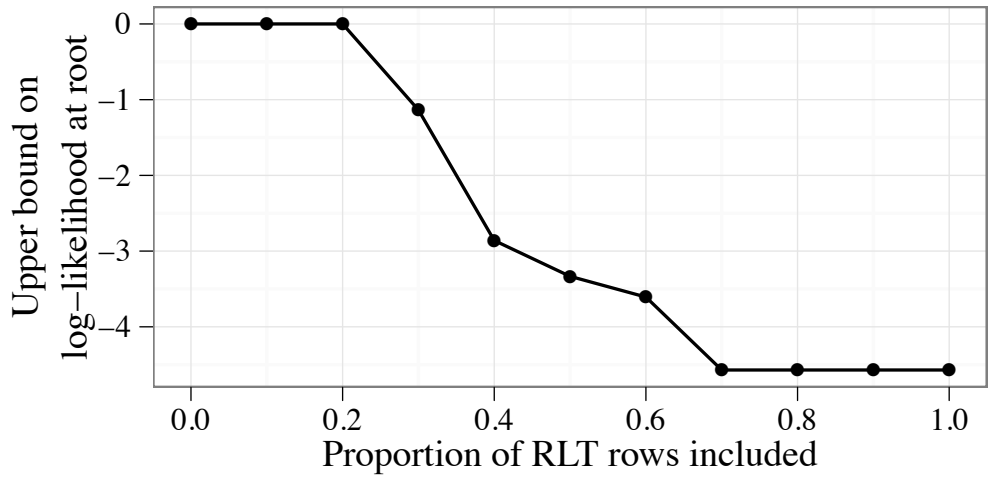
- ILP Dependency Parsing
  - Supervised approaches
    - Riedel and Clarke (2006)
    - Martins et al. (2009)
    - Riedel et al. (2012)
  - Inspired our unsupervised formulation

Experimental Setup: Datasets

- Toy Synthetic Data:
  - Generated from a synthetic DMV over three POS tags (Verb, Noun, Adjective)
  - Parameters chosen to favor short sentences with English word order
- Real Data:
  - 200 random sentences of no more than 10 tokens from the Brown portion of the Penn Treebank
  - Universal set of 12 tags (Petrov et al., 2012) plus a tag for auxiliaries, ignoring punctuation

Synthetic Data Experiments

- **Experiment 1:** How many RLT constraints does the full relaxation contain?
  - 5 synthetic sentences: 320,126 constraints.
  - 20 synthetic sentences: 4,056,498 constraints.
  - Quadratic in the length of the corpus!
- **Experiment 2:** How tight is the RLT relaxation?
  - Data: 5 synthetic sentences.
  - For different random subsets of the RLT constraints, we compute the relaxation's upper bound at the root node.
  - The quality of the root relaxation increases as we approach the full set of RLT constraints.

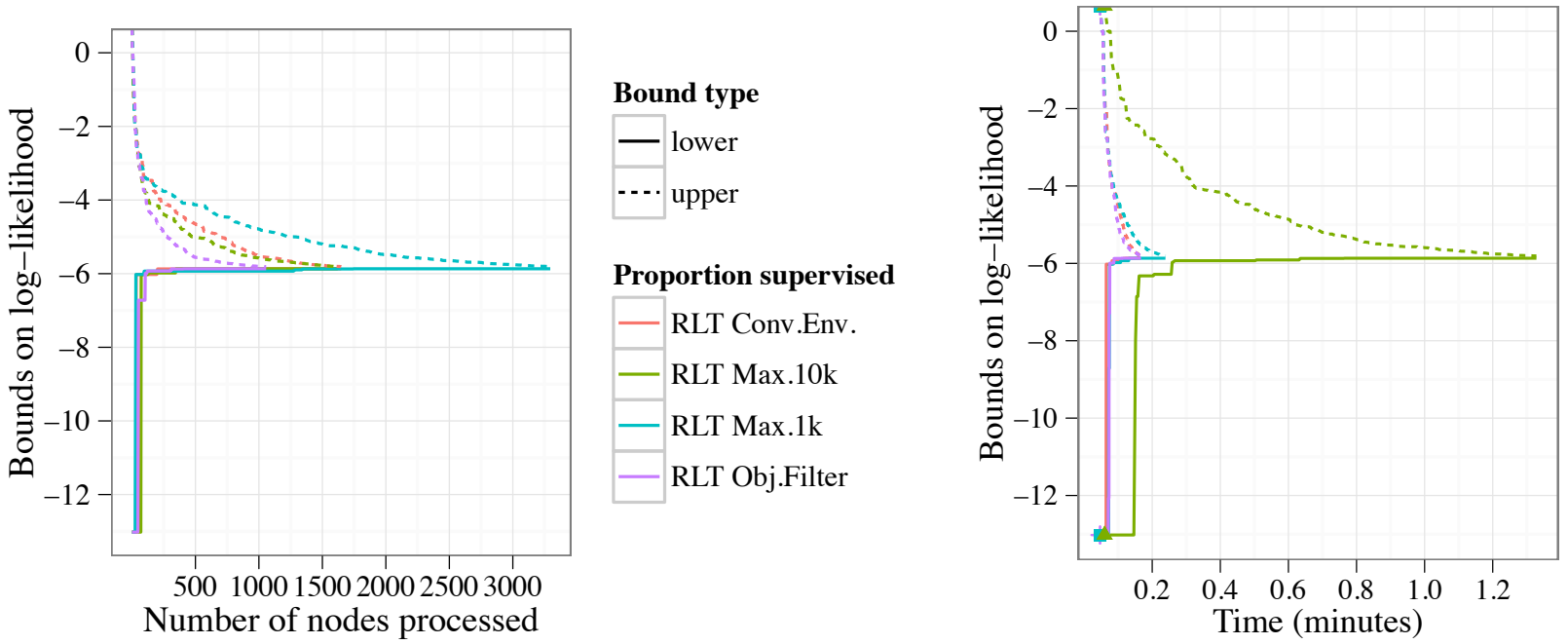


Experimental Setup

- Search Methods:
  - Branch-and-bound with various RLT relaxations
    - **Conv.Env**. only the concave envelope
    - **Obj.Filter** all constraints having a nonzero coefficient for at least one of the RLT variables z_m from the linearized objective
    - **Max.1k** random sample of 1,000 RLT constraints
    - **Max.10k** random sample of 10,000 RLT constraints
    - **Max.100k** random sample of 100,000 RLT constraints
  - Viterbi EM with random restarts
- We consider each search method with/without posterior constraints

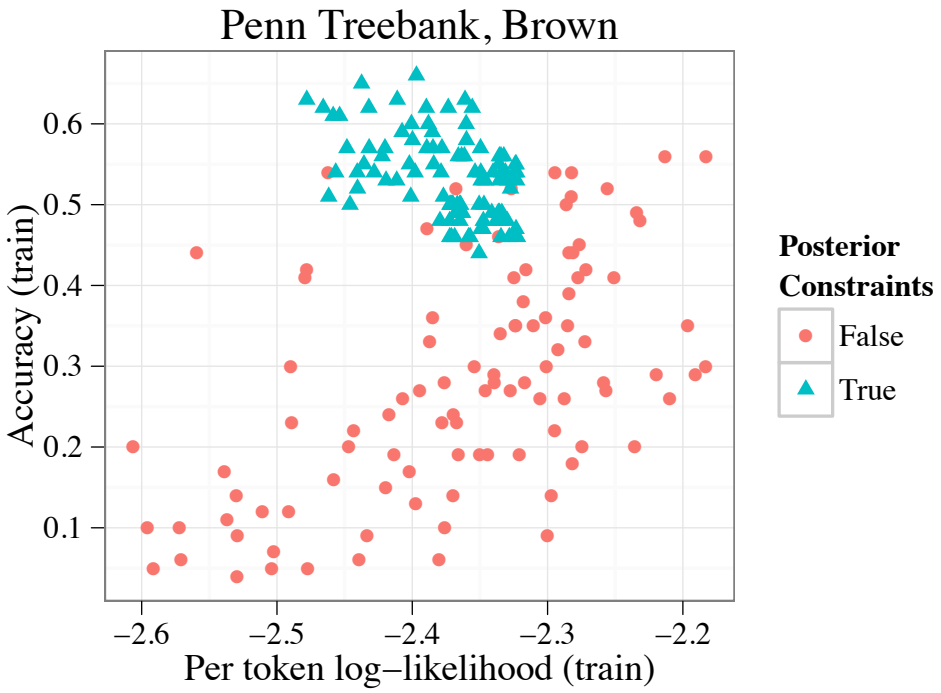
Synthetic Data Experiments

- **Experiment 3:** Comparison of Relaxations for Global Search
  - Data: 5 synthetic sentences
  - Solved to ε-optimality for ε = 0.01
  - More RLT constraints yield tighter relaxations that are slower to solve. This leads to fewer nodes in the branch-and-bound tree, but (potentially) slower solving times.



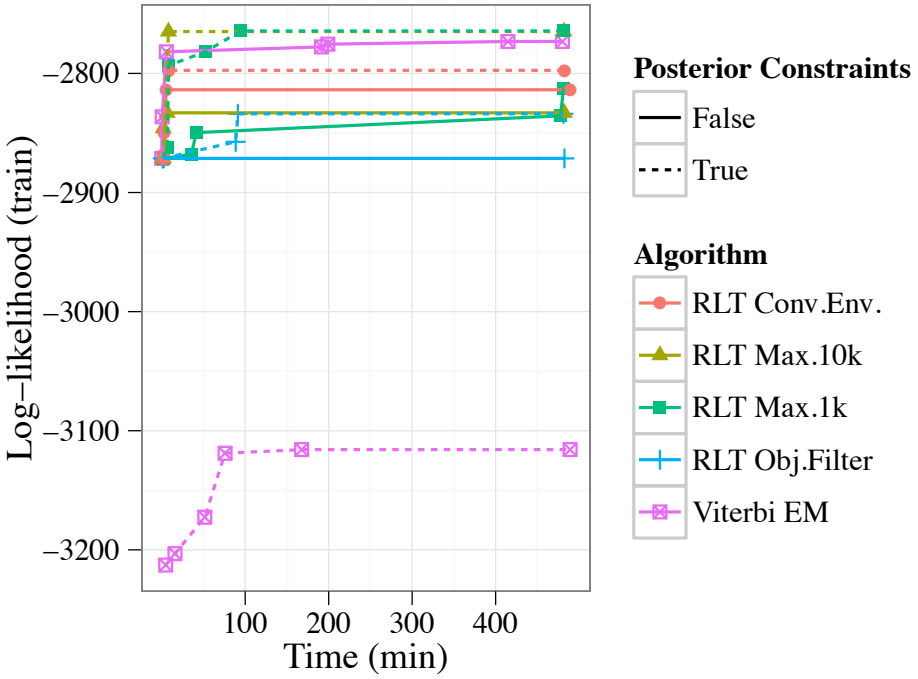
Relaxed Viterbi EM with Posterior Constraints

- **Experiment 4:**
  Comparison of Viterbi EM with and without posterior constraints.
  - 100 random restarts for each setting
  - Data: 200 sentences from Brown
  - Accuracy and log-likelihood are loosely correlated
  - Posterior constraints hurt likelihood, but boost accuracy



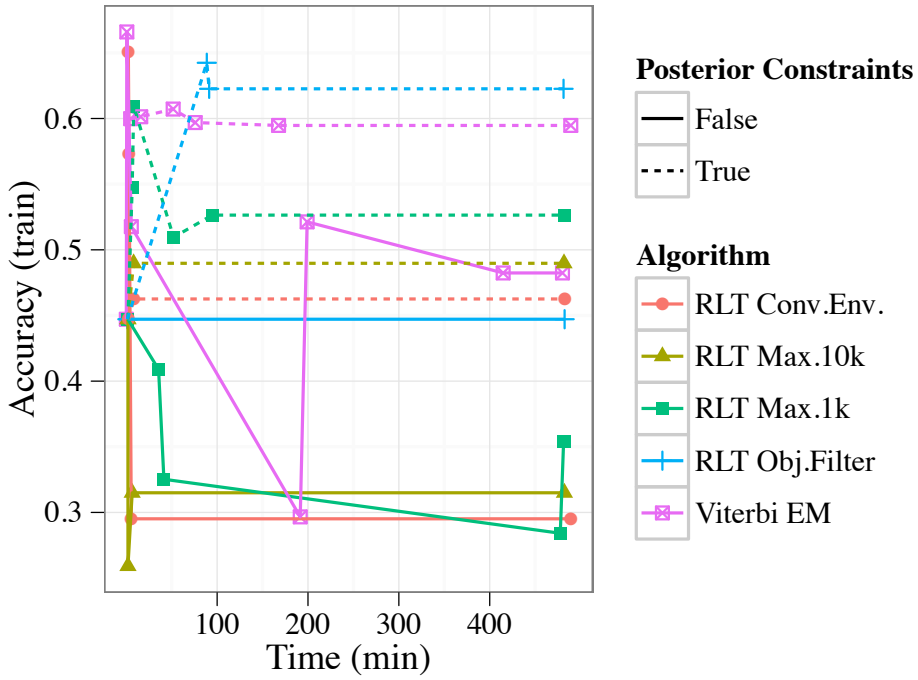
Real Data Experiments

- **Experiment 5:** Compare global search to Viterbi EM with/ without posterior constraints.
  - Data: 200 sentences from Brown.
  - Evaluate the **log-likelihood** of the incumbent solution over time.
  - In our global search method, unlike Viterbi EM, the posterior constraints lead to higher log-likelihoods.



Real Data Experiments

- **Experiment 5:** Compare global search to Viterbi EM with/ without posterior constraints.
  - Data: 200 sentences from Brown.
  - Evaluate the **unlabeled directed dependency accuracy** of the incumbent solution over time.
  - Posterior constraints lead to higher accuracies.



Summary

Contributions

- Formulation of grammar induction as a **mathematical program**.
- **Global optimization framework** for nonconvex likelihood function in any generative model of trees.
- Novel **posterior constrained Viterbi EM** baseline.

Thank you!

Questions?