Nonconvex Global Optimization for Latent Variable Models

Matthew R. Gormley and Jason Eisner

August 5, 2013

ACL

JOHNS HOPKINS UNIVERSITY

THE PROBLEM: NONCONVEXITY

The Viterbi Objective

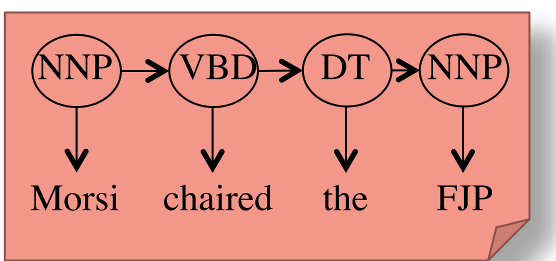
$$\arg\max_{\vec{\theta},\vec{x}} \; \vec{\theta} \cdot \vec{x}$$

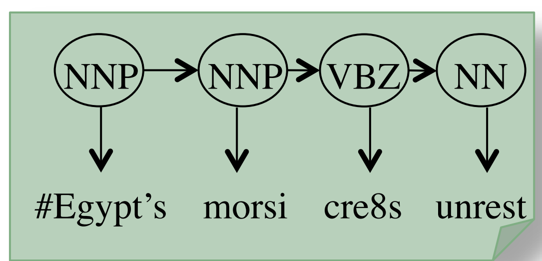
- Find the model parameters $\vec{\theta}$ that best explain the features $\vec{x}$ of the data
- Features are unknown because we don't see the latent structure

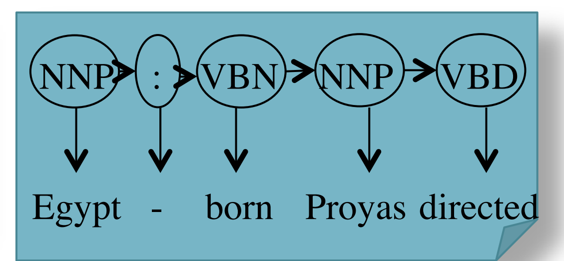
The Viterbi Objective

$$\arg\max_{\vec{\theta},\vec{x}} \ \vec{\theta} \cdot \vec{x}$$

Example: Unsupervised POS Tagging



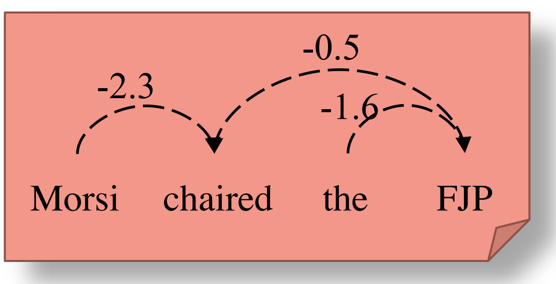


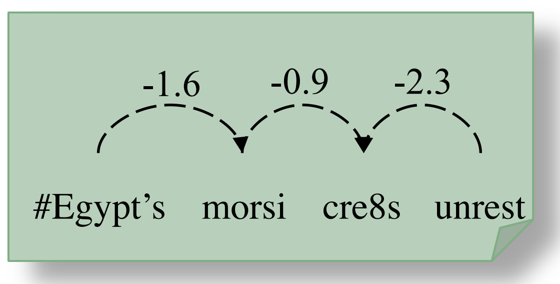


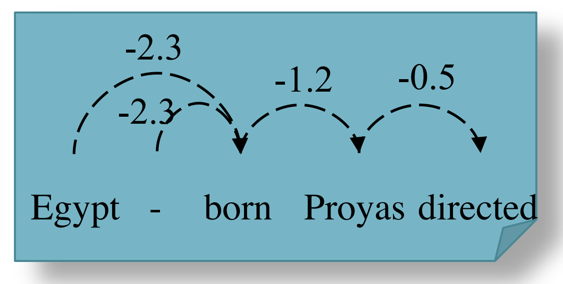
The Viterbi Objective

$$\arg\max_{\vec{\theta},\vec{x}} \; \vec{\theta} \cdot \vec{x}$$

Example: Unsupervised Dependency Parsing



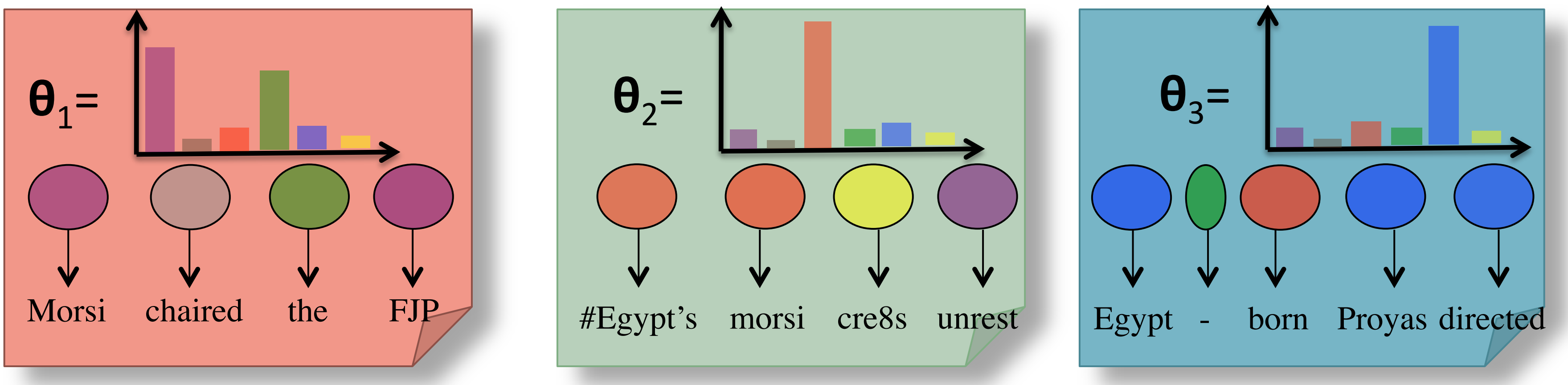




The Viterbi Objective

$$\arg\max_{\vec{\theta},\vec{x}} \; \vec{\theta} \cdot \vec{x}$$

Example: Topic Modeling



Viterbi Objective as a Quadratic Program

Variables:

θ_m	Log-probability for feature m
x_m	Corpus-wide feature count for m

Viterbi EM objective in log space.

$$\arg\max_{\vec{\theta},\vec{x}} \ \vec{\theta} \cdot \vec{x}$$

Viterbi Objective as a Quadratic Program

Variables:

θ_m	Log-probability for feature m
x_m	Corpus-wide feature count for m

Viterbi EM objective in log space.

$$\max \sum_m \theta_m x_m$$

Viterbi Objective as a Quadratic Program

Variables:

θ_m	Log-probability for feature m
x_m	Corpus-wide feature count for m

Indices and constants:

m	Feature / model parameter 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.

$$\max \sum_m \theta_m x_m$$

Sum-to-one constraints on model parameters.

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

Parameters must be log-probabilities.

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

Viterbi Objective as a Quadratic Program

Variables:

θ_m	Log-probability for feature m
x_m	Corpus-wide feature count for m

Indices and constants:

m	Feature / model parameter 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.

$$\max \sum_m \theta_m x_m$$

Sum-to-one constraints on model parameters.

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

Parameters must be log-probabilities.

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

Model constraints.

$$A\vec{x} \leq b$$

Feature counts must be integers.

$$x_m \in \mathbb{Z}, \forall m \in \mathcal{I}$$

Viterbi Objective as a Quadratic Program

- Properties
  - Nonconvex
  - NP Hard to solve (Cohen & Smith, 2010)
  - Differs from the soft EM objective which marginalizes over $\vec{x}$
- Spitkovsky et al. (2009) show hard (Viterbi) EM sometimes outperforms soft EM.

$$
\begin{aligned}
\max \; & \sum_m \theta_m x_m \\
\text{s.t.} \; & \sum_{m \in \mathcal{M}_c} \exp(\theta_m) = 1, \; \forall c \\
& \theta_m \leq 0, \forall m \\
& A\vec{x} \leq b \\
& x_m \in \mathbb{Z}, \; \forall m \in \mathcal{I}
\end{aligned}
$$

Viterbi EM

E-step:

$$\arg\max_{\vec{x}} \; \vec{\theta} \cdot \vec{x}$$

M-step:

$$\arg\max_{\vec{\theta}} \; \vec{\theta} \cdot \vec{x}$$

Viterbi EM

E-step:

$$\arg\max_{\vec{x}} \ \vec{\theta} \cdot \vec{x}$$

(Viterbi decoding)

M-step:

$$\arg\max_{\vec{\theta}} \ \vec{\theta} \cdot \vec{x}$$

(Supervised learning)

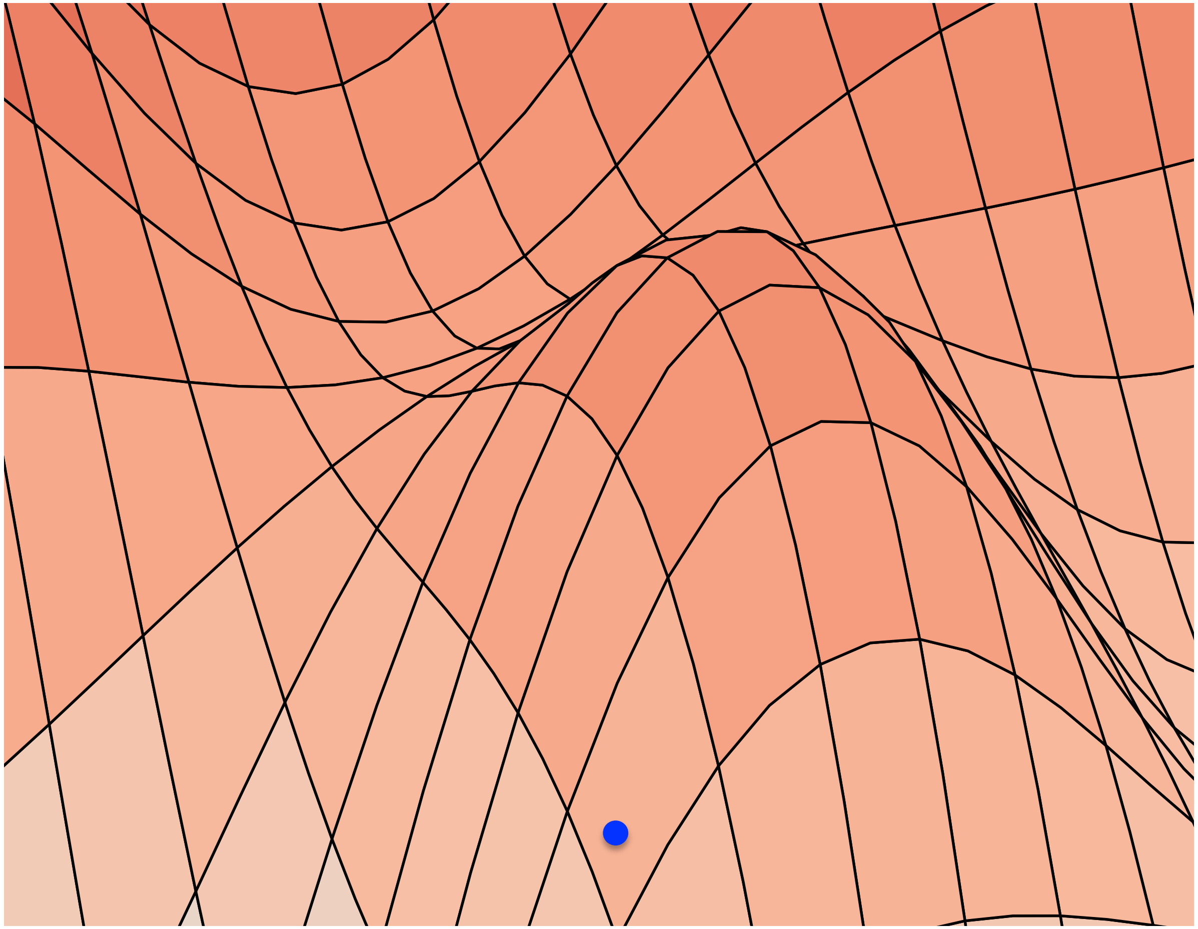
Viterbi EM

E-step:

$$\arg\max_{\vec{x}} \; \vec{\theta} \cdot \vec{x}$$

M-step:

$$\arg\max_{\vec{\theta}} \; \vec{\theta} \cdot \vec{x}$$



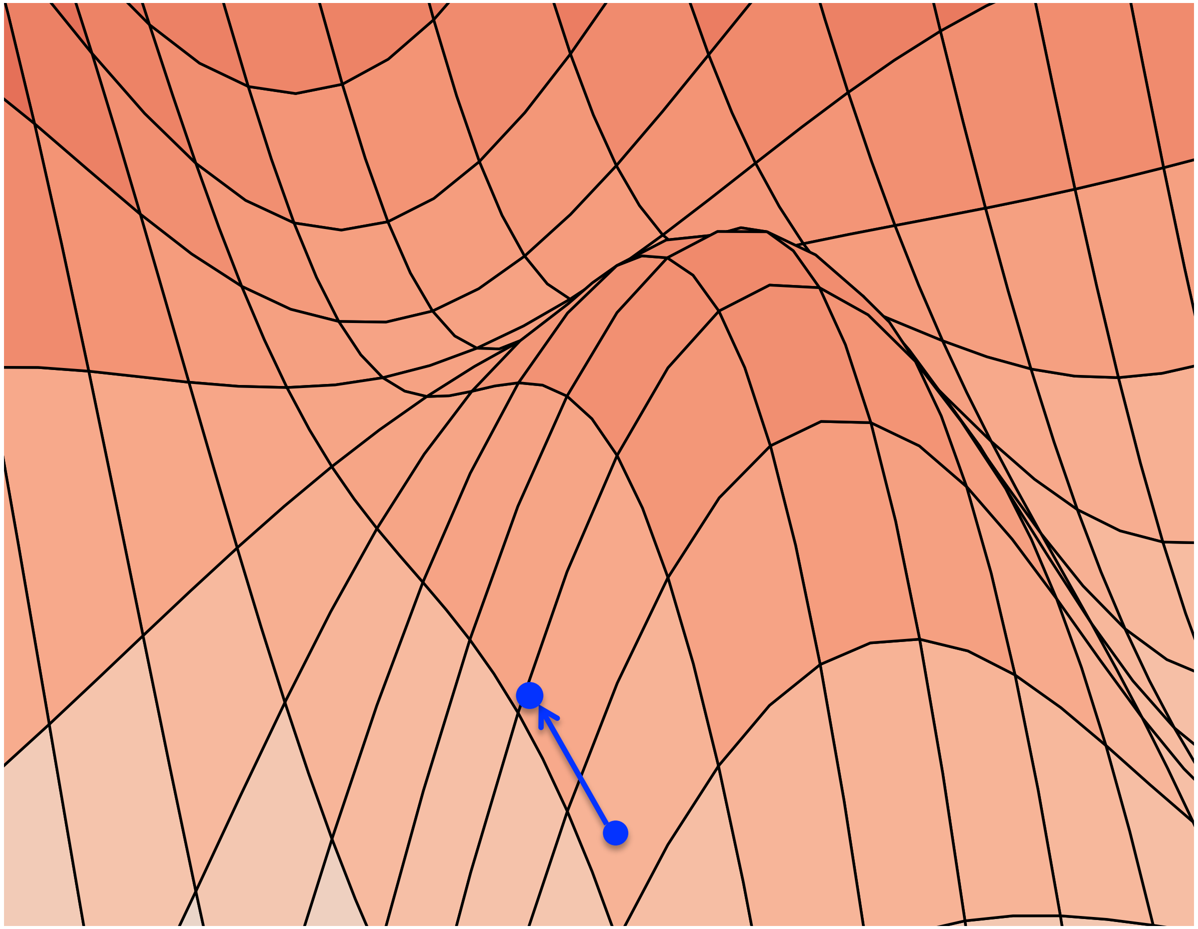
Viterbi EM

E-step:

$$\arg\max_{\vec{x}} \ \vec{\theta} \cdot \vec{x}$$

M-step:

$$\arg\max_{\vec{\theta}} \ \vec{\theta} \cdot \vec{x}$$



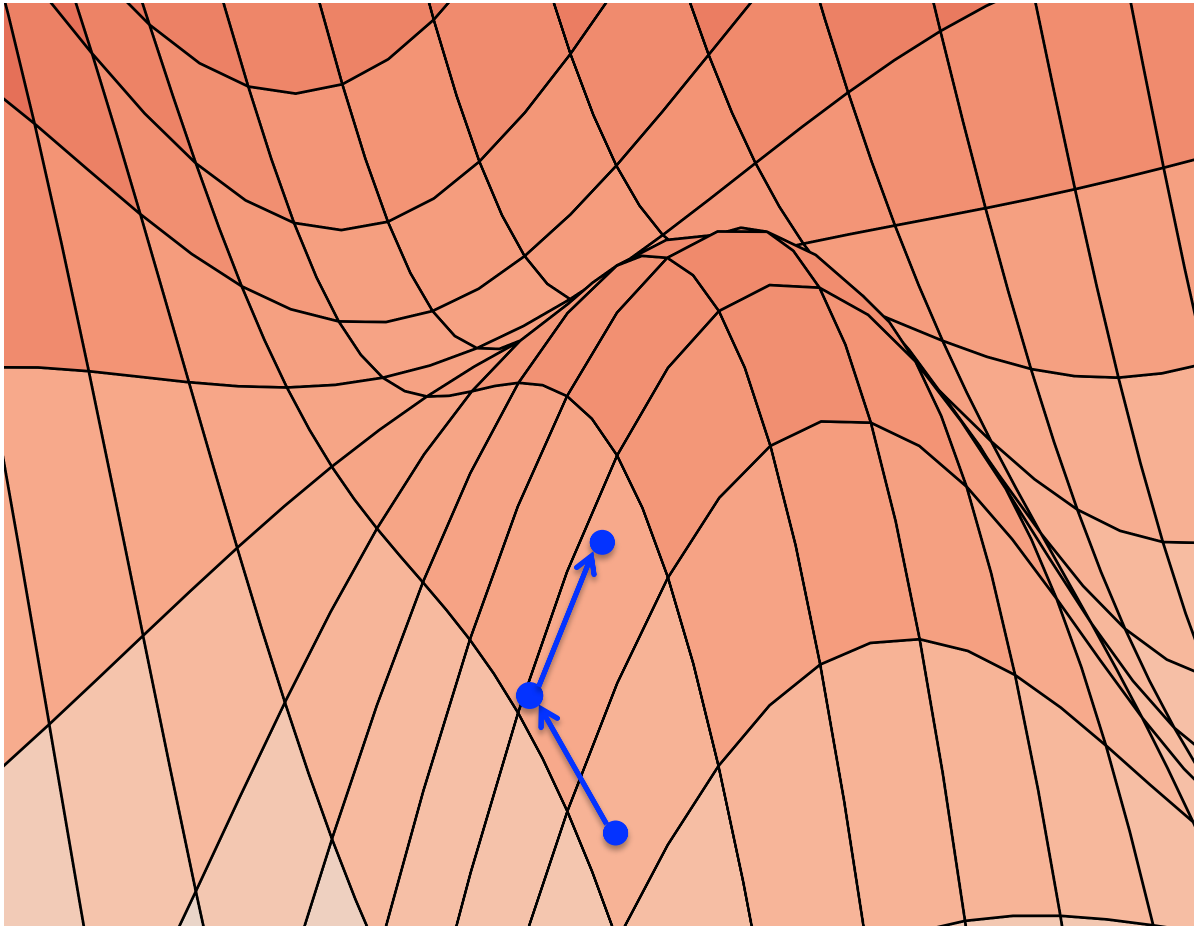
Viterbi EM

E-step:

$$\arg\max_{\vec{x}} \ \vec{\theta} \cdot \vec{x}$$

M-step:

$$\arg\max_{\vec{\theta}} \ \vec{\theta} \cdot \vec{x}$$



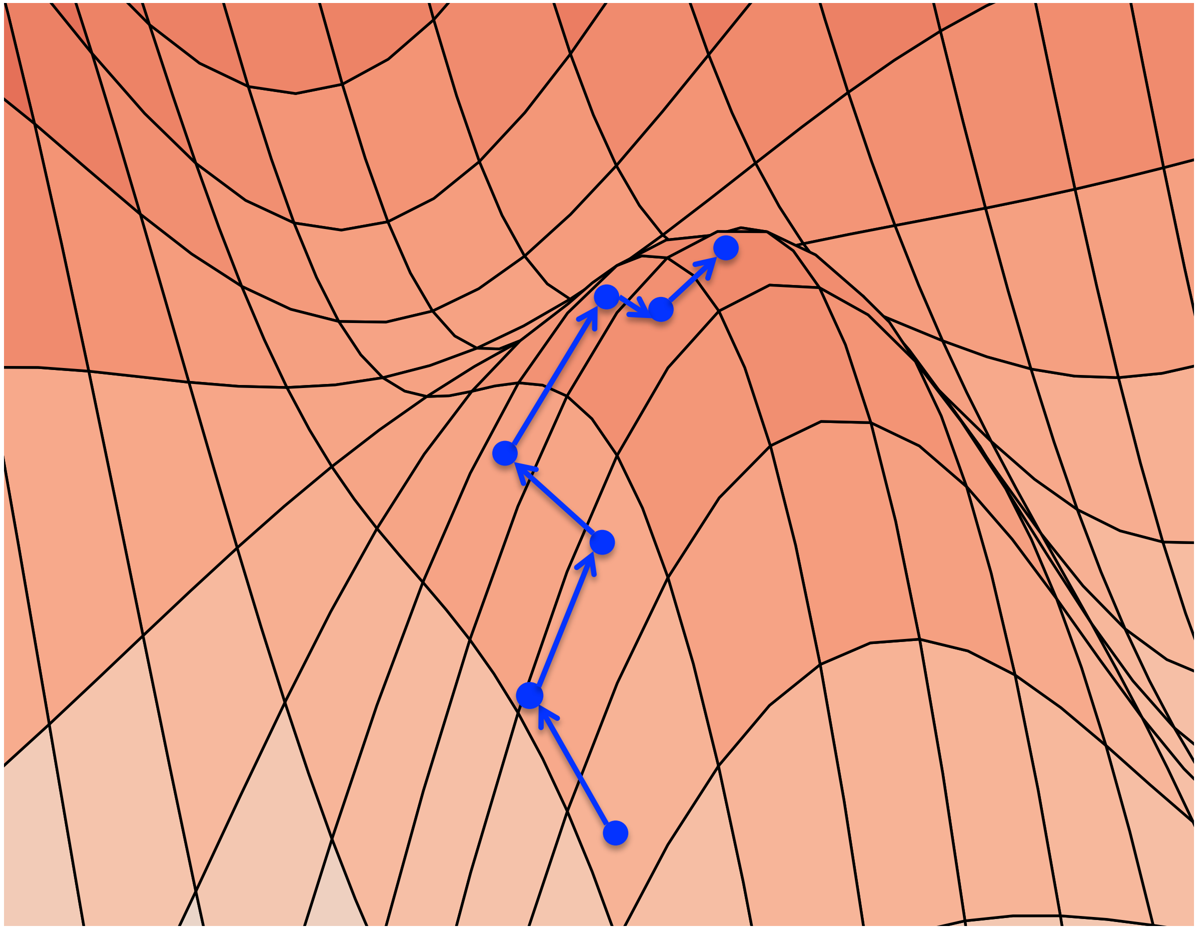
Viterbi EM

E-step:

$$\arg\max_{\vec{x}} \ \vec{\theta} \cdot \vec{x}$$

M-step:

$$\arg\max_{\vec{\theta}} \ \vec{\theta} \cdot \vec{x}$$



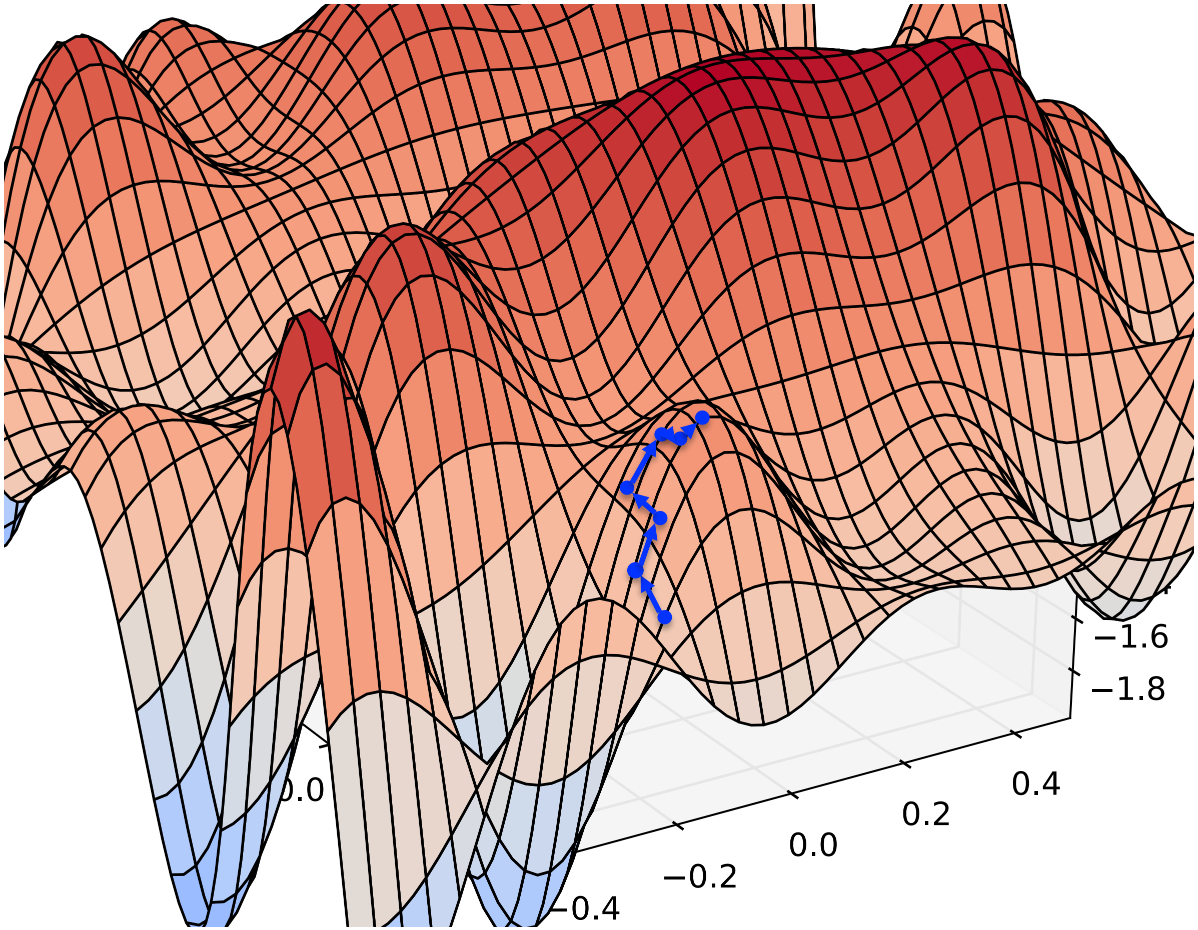
Viterbi EM

E-step:

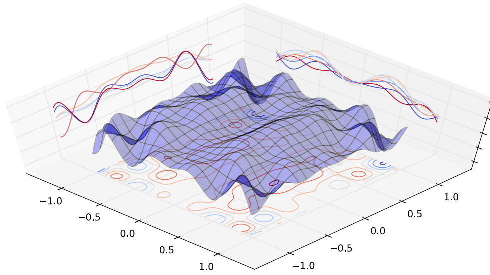
$$\arg\max_{\vec{x}} \ \vec{\theta} \cdot \vec{x}$$

M-step:

$$\arg\max_{\vec{\theta}} \ \vec{\theta} \cdot \vec{x}$$



Our Goals



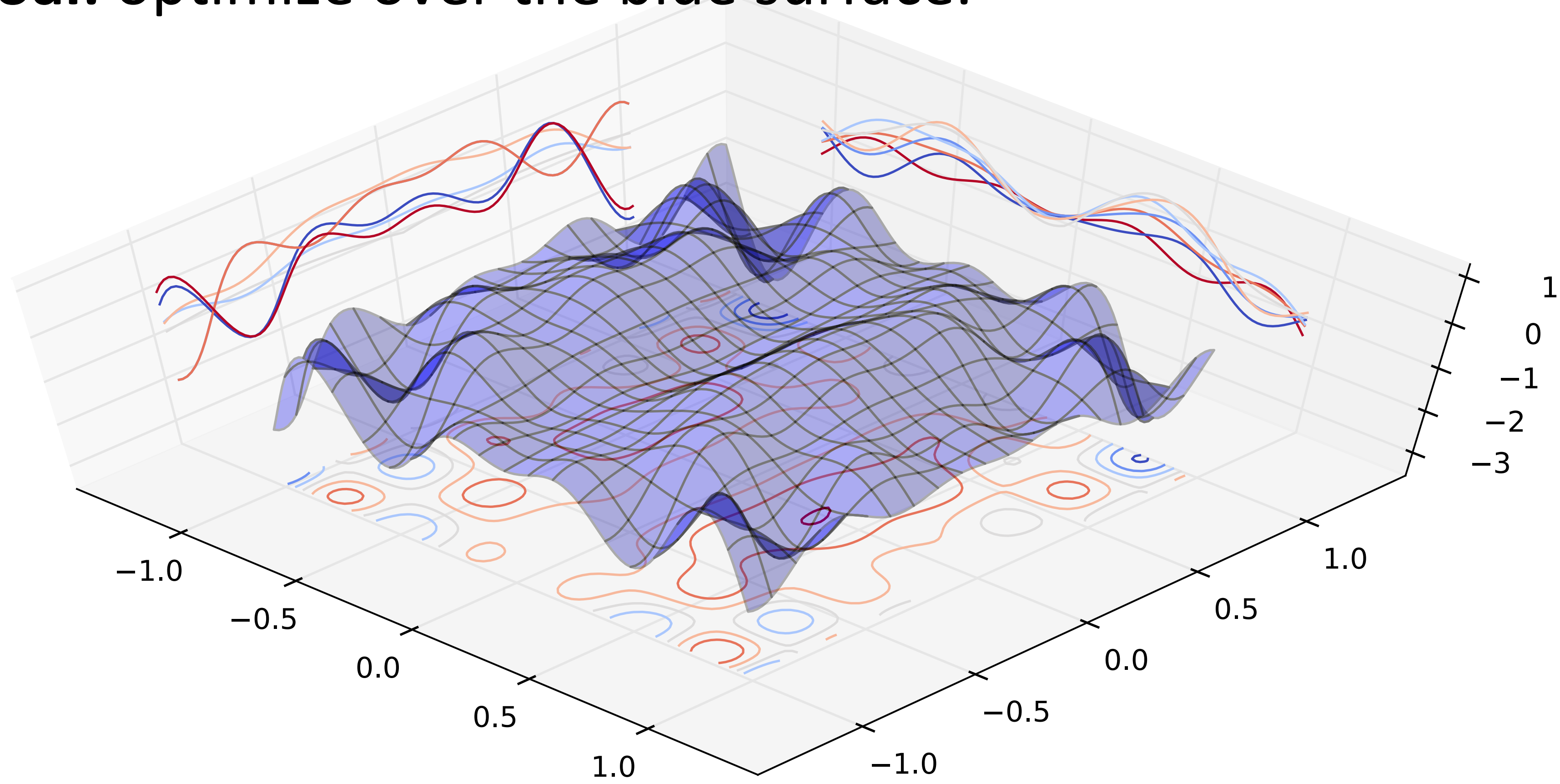
- **Learn more** about these commonplace nonconvex likelihood objectives
- Go **beyond local-search**.
- Develop a search method capable of finding a **provably ε-optimal solution**.

Overview

OUR APPROACH: GLOBAL SEARCH

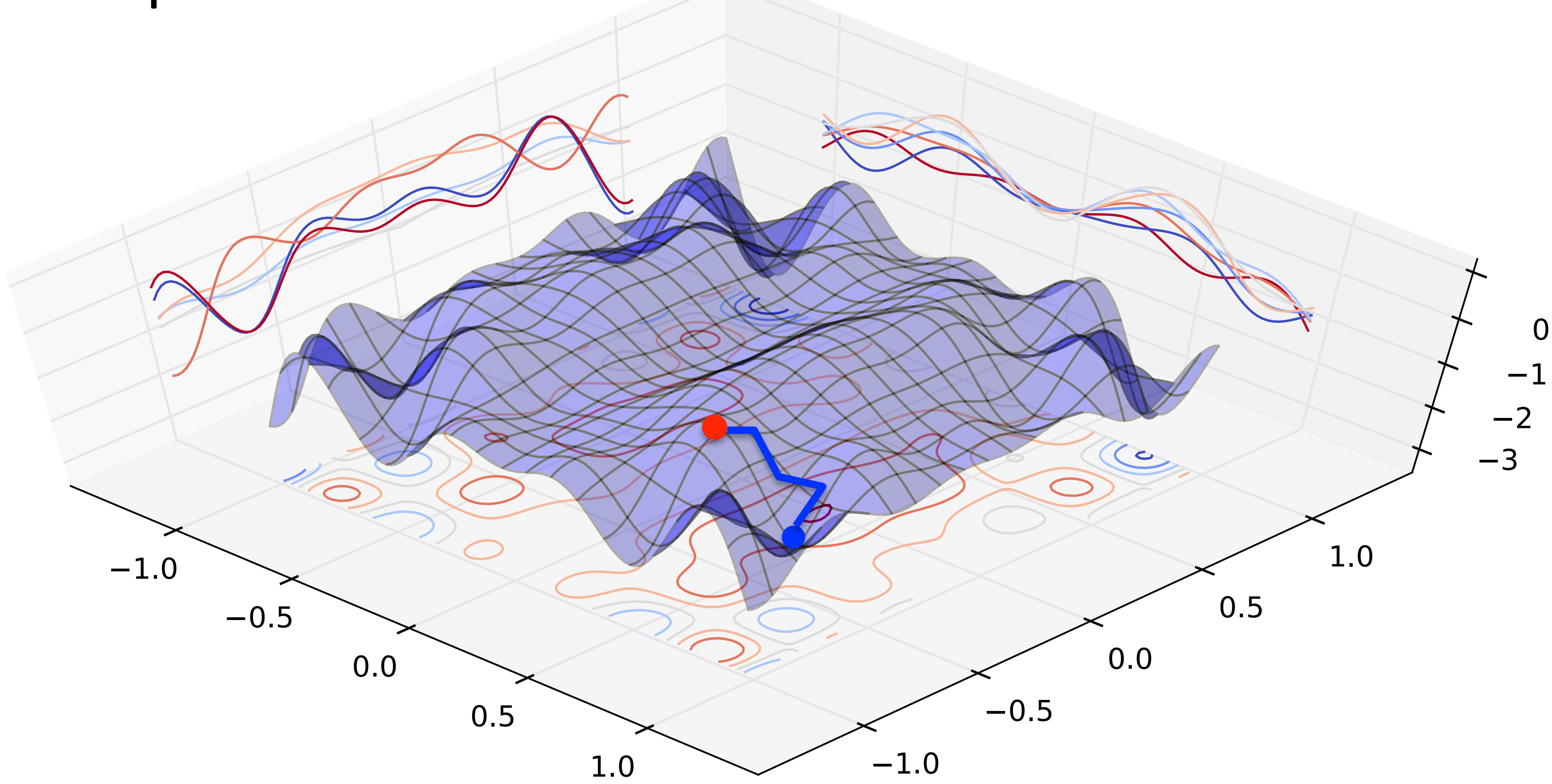
Background: Nonconvex Global Optimization

Goal: optimize over the blue surface.



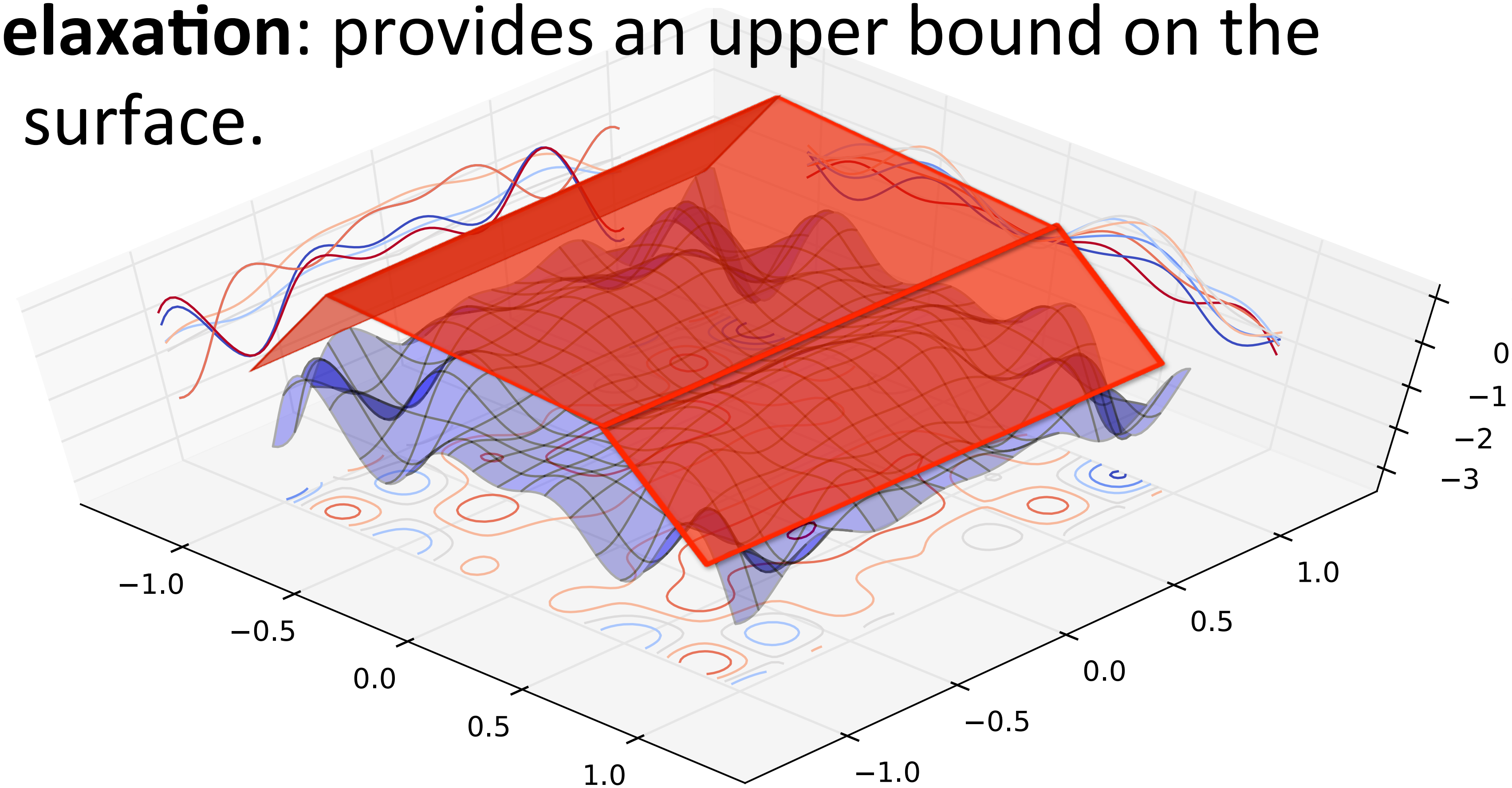
Background: Nonconvex Global Optimization

Goal: optimize over the blue surface.



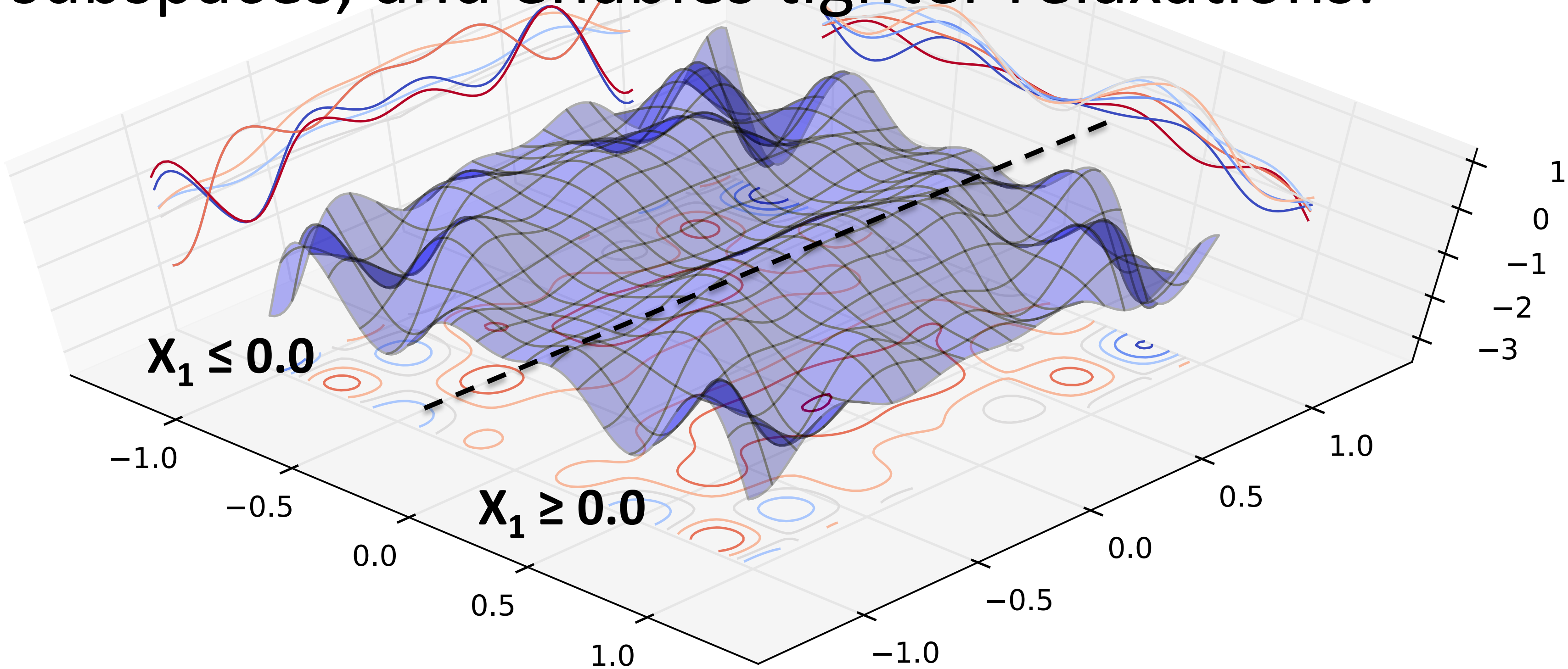
Background: Nonconvex Global Optimization

Relaxation: provides an upper bound on the surface.



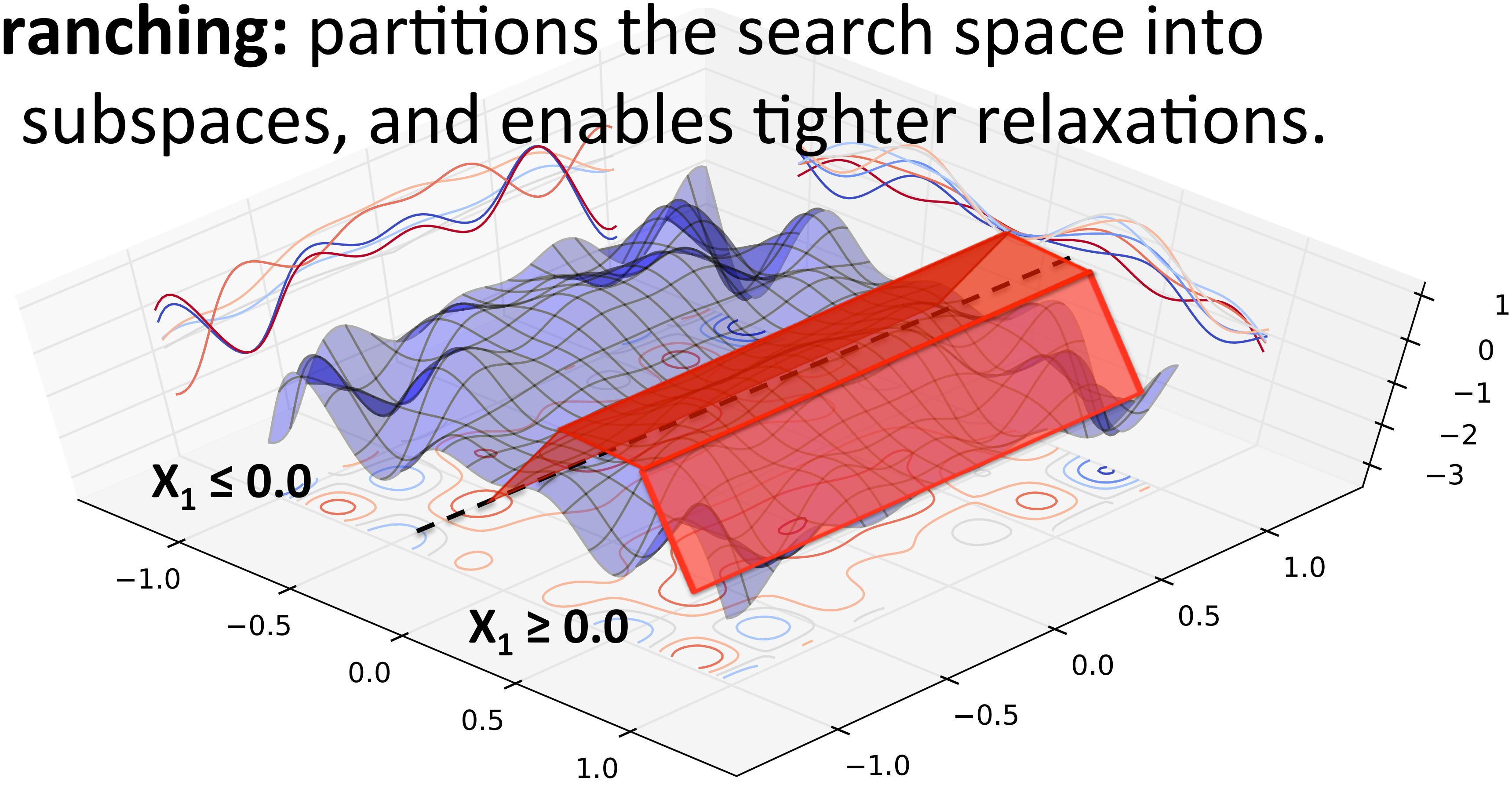
Background: Nonconvex Global Optimization

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



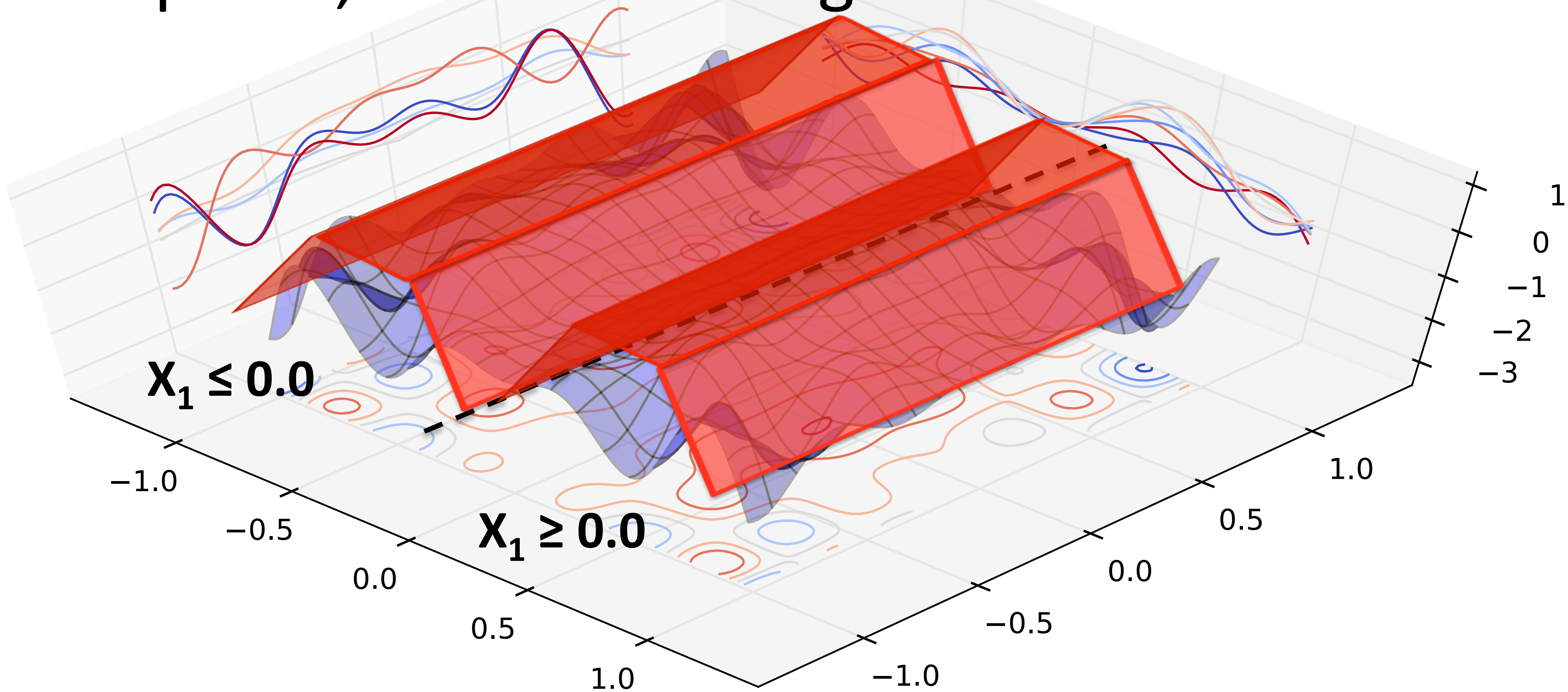
Background: Nonconvex Global Optimization

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



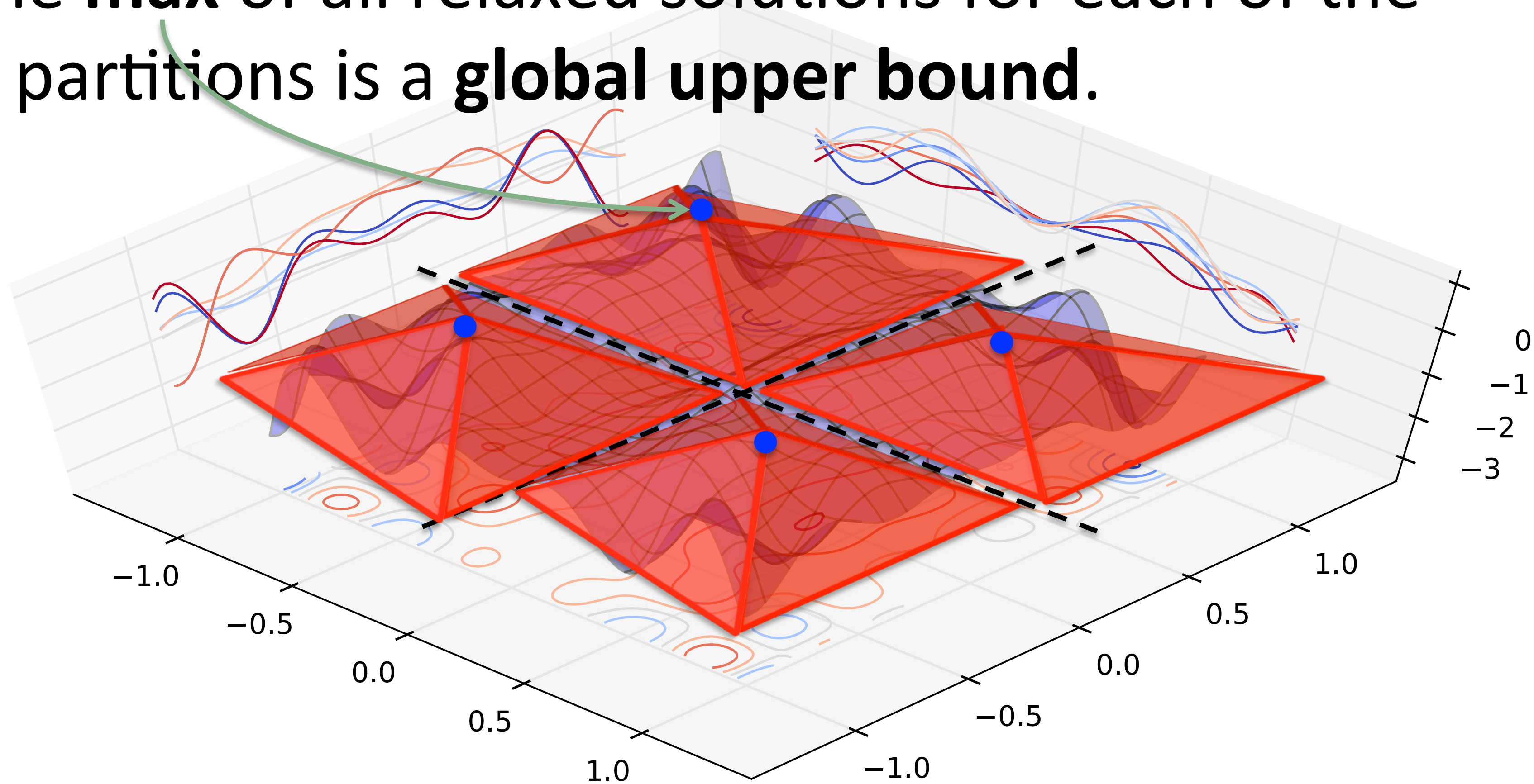
Background: Nonconvex Global Optimization

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



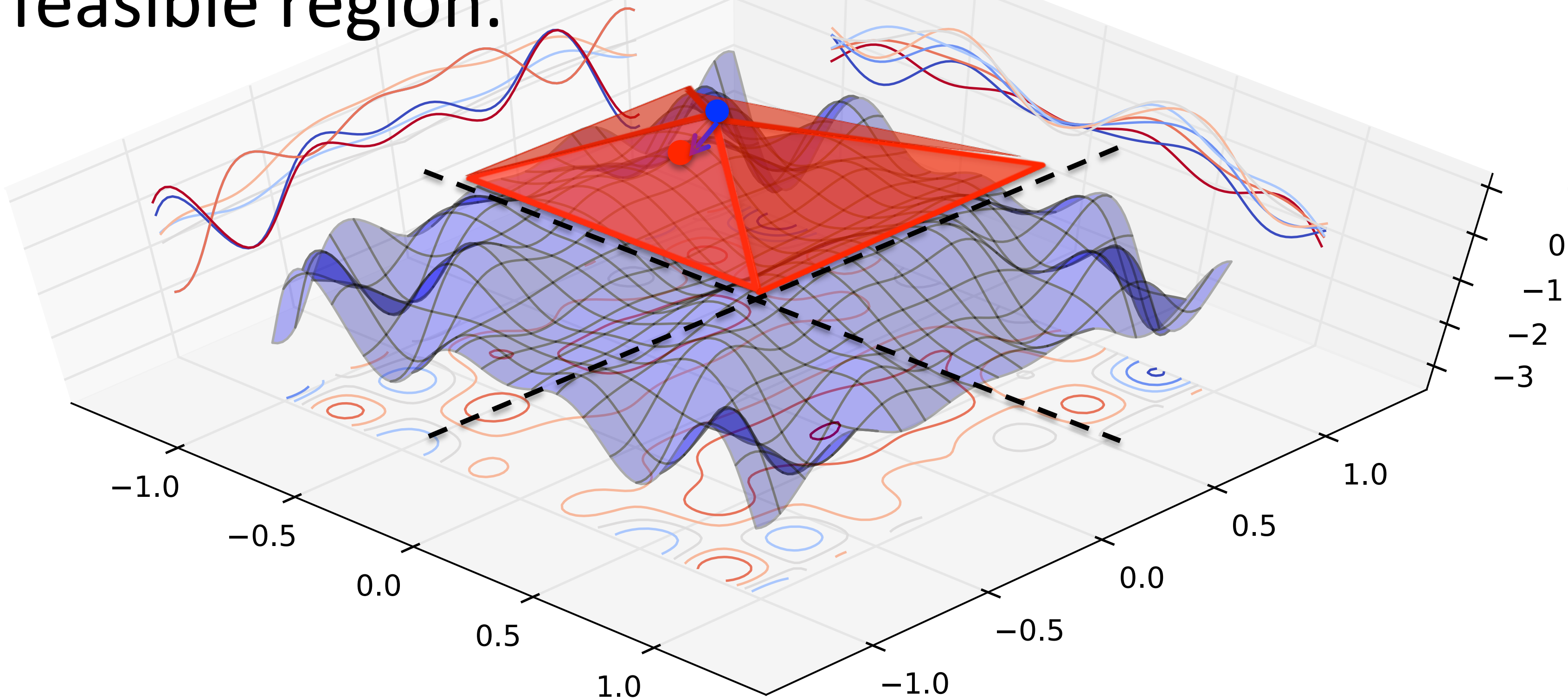
Background: Nonconvex Global Optimization

The **max** of all relaxed solutions for each of the partitions is a **global upper bound**.



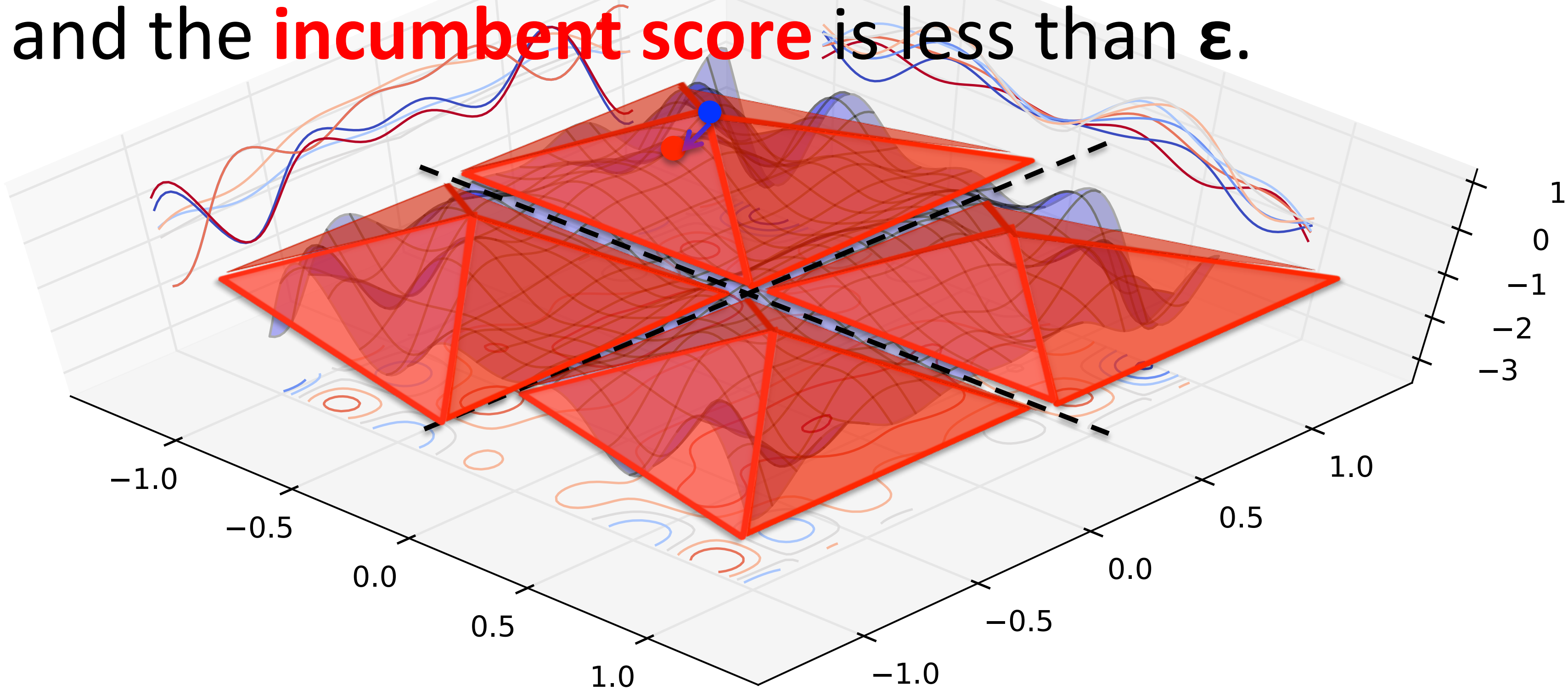
Background: Nonconvex Global Optimization

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



Background: Nonconvex Global Optimization

The **incumbent** is **ε-optimal** if the relative difference between the **global upper bound** and the **incumbent score** is less than **ε**.



How much should we subdivide?

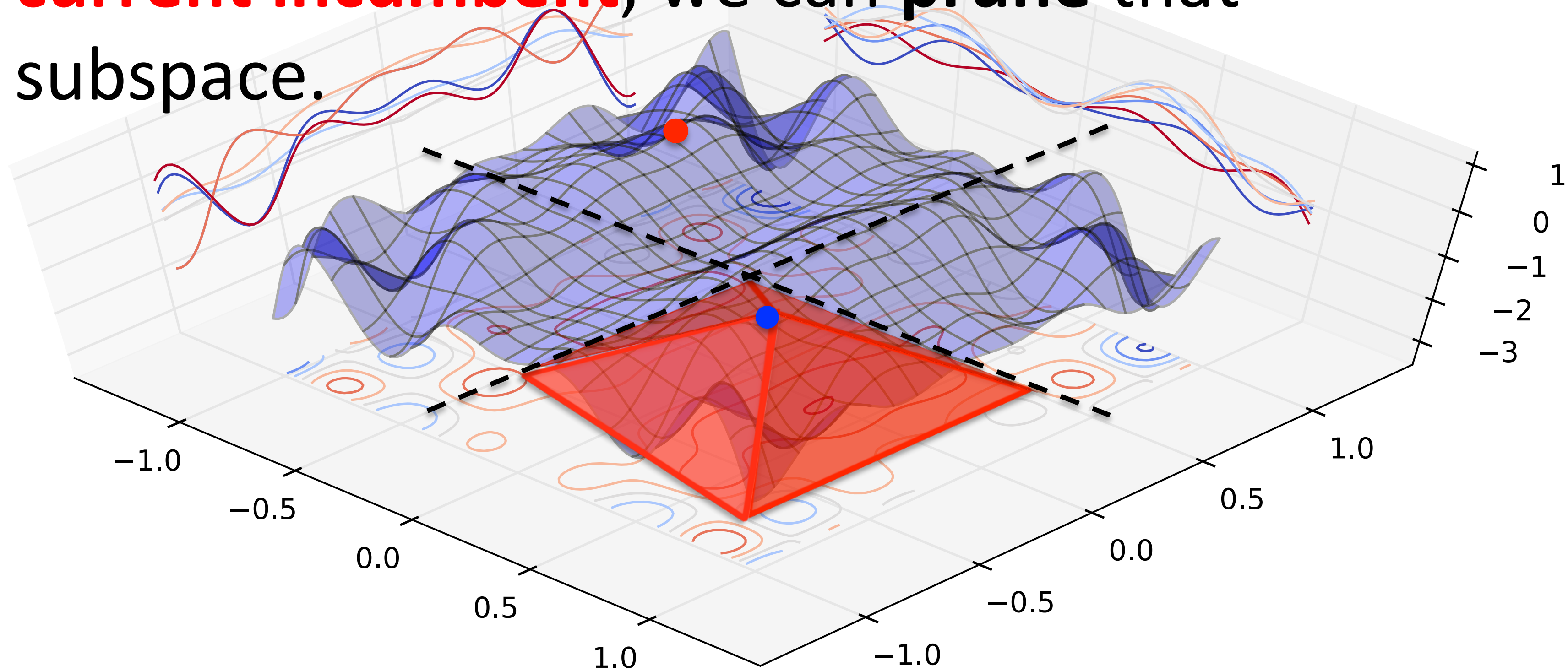
How much should we subdivide?

BRANCH-AND-BOUND

- Method for **recursively subdividing** the search space
- **Subspace order** can be determined heuristically (e.g. best-first search with depth-first plunging)
- **Prunes** subspaces that can't yield better solutions

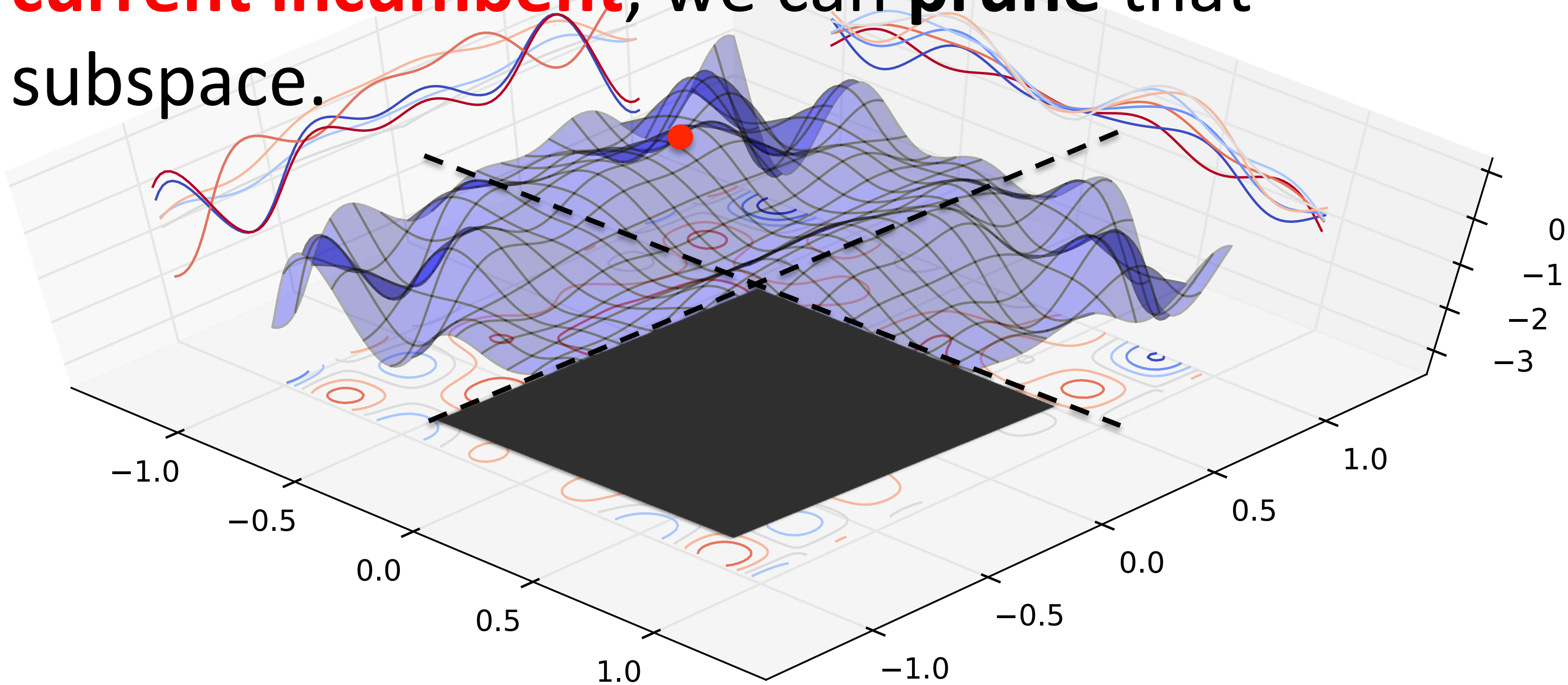
Background: Nonconvex Global Optimization

If the **subspace upper bound** is worse than the **current incumbent**, we can **prune** that subspace.



Background: Nonconvex Global Optimization

If the **subspace upper bound** is worse than the **current incumbent**, we can **prune** that subspace.



Limitations:
Branch-and-Bound for the Viterbi Objective

- The Viterbi Objective
  - Nonconvex
  - NP Hard to solve (Cohen & Smith, 2010)
- Branch-and-bound
  - Kind of tricky to get it right…
  - Curse of dimensionality kicks in quickly
    - Nonconvex quadratic optimization by LP-based branch-and-bound usually fails with more than 80 variables (Burer and Vandenbussche, 2009)
    - Our smallest (toy) problems have hundreds of variables

- Preview of Experiments
  - We solve 5 sentences, but on 200 sentences, we couldn't run to completion
  - Our (hybrid) global search framework incorporates local search
  - This hybrid approach sometimes finds higher likelihood (and higher accuracy) solutions than pure local search

BRANCH-AND-BOUND INGREDIENTS

Mathematical Program

Relaxation

Projection

(Branch-and-Bound Search Heuristics)

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
- Solve the relaxation with the Simplex Algorithm

Relaxing the Sum-to-one Constraints

Variables:

θ_m	Log-probability for feature m
x_m	Corpus-wide feature count for m

Indices and constants:

m	Feature / model parameter 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.

$$\max \sum_m \theta_m x_m$$

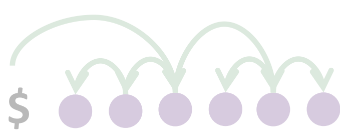
Sum-to-one constraints on model parameters.

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

Parameters must be log-probabilities.

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

Tree constraints.



$$A\vec{x} \leq b$$

Feature counts must be integers.

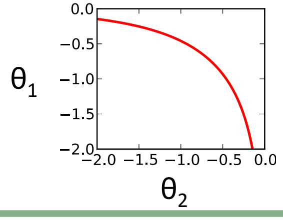
$$x_m \in \mathbb{Z}, \; \forall m \in \mathcal{I}$$

Relaxing the Sum-to-one Constraints

Example plots of two parameter case:

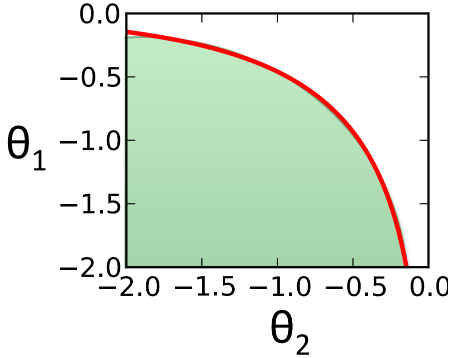
1. Original nonlinear constraint:

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



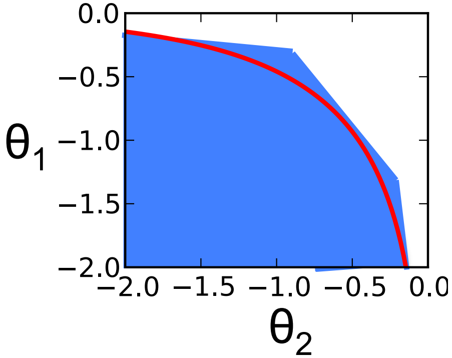
2. Nonlinear relaxation:

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



3. Linear relaxation:

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



“Relaxing” the Objective

Variables:

θ_m	Log-probability for feature m
x_m	Corpus-wide feature count for m

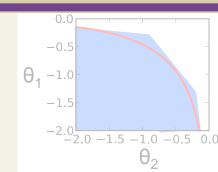
Indices and constants:

m	Feature / model parameter 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.

$$\max \sum_m \theta_m x_m$$

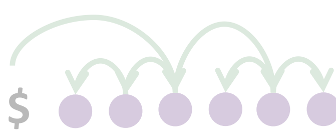
Sum-to-one constraints on model parameters.

s.t. 

$, \forall c$

Parameters must be log-probabilities.

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

Tree constraints. 

$$A\vec{x} \leq b$$

Feature counts must be integers.

$$x_m \in \mathbb{Z}, \forall m \in \mathcal{I}$$

Definitions

$$\vec{x} = \begin{bmatrix} \vec{f} \\ \vec{e} \end{bmatrix}$$

←Corpus-wide feature counts

←Token-specific variables

"Relaxing" the Objective

Original nonconvex **quadratic** objective:

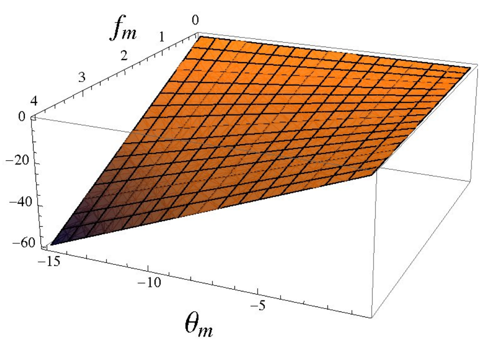
$$\max \sum_m \theta_m f_m$$

Each B&B subspace specifies bounds:

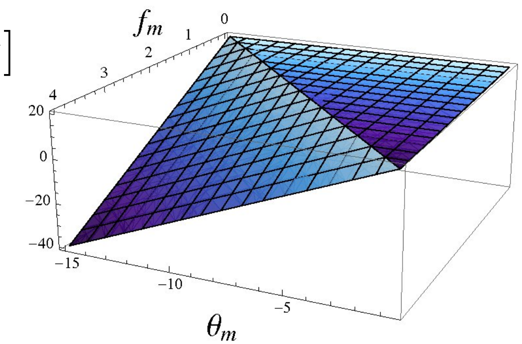
$$\theta_m^{\min} < \theta_m < \theta_m^{\max}, \forall m$$
$$f_m^{\min} \leq f_m \leq f_m^{\max}, \forall m$$

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:

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

Linear Relaxation of Viterbi QP

Variables:

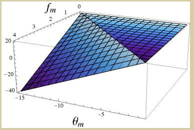
θ_m	Log-probability for feature m
f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

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

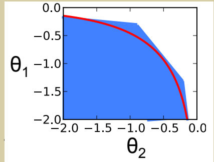
Viterbi EM objective in log space.

$$\max \sum_m$$



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

s.t.



$$, \forall c$$

Model constraints.

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

Each B&B subspace specifies bounds.

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

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

TIGHTENING THE RELAXATION

- Branching
- Reformulation Linearization Technique

Branching

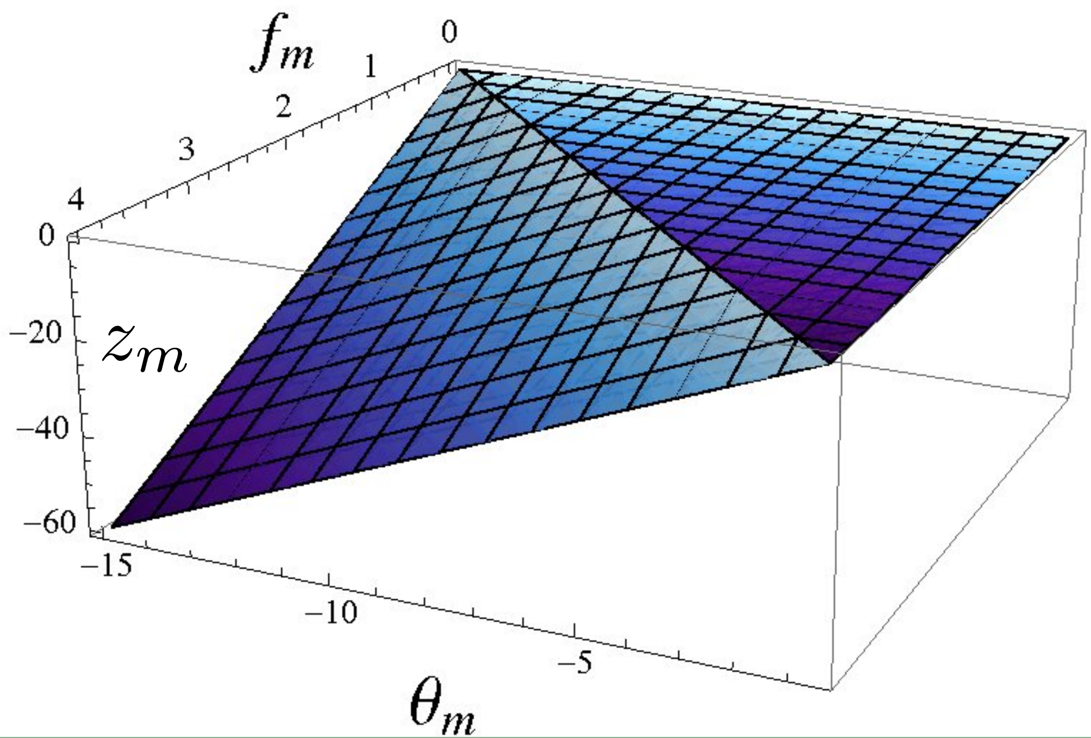
Concave Envelope
(e.g. McCormick (1976))

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**.



Branching

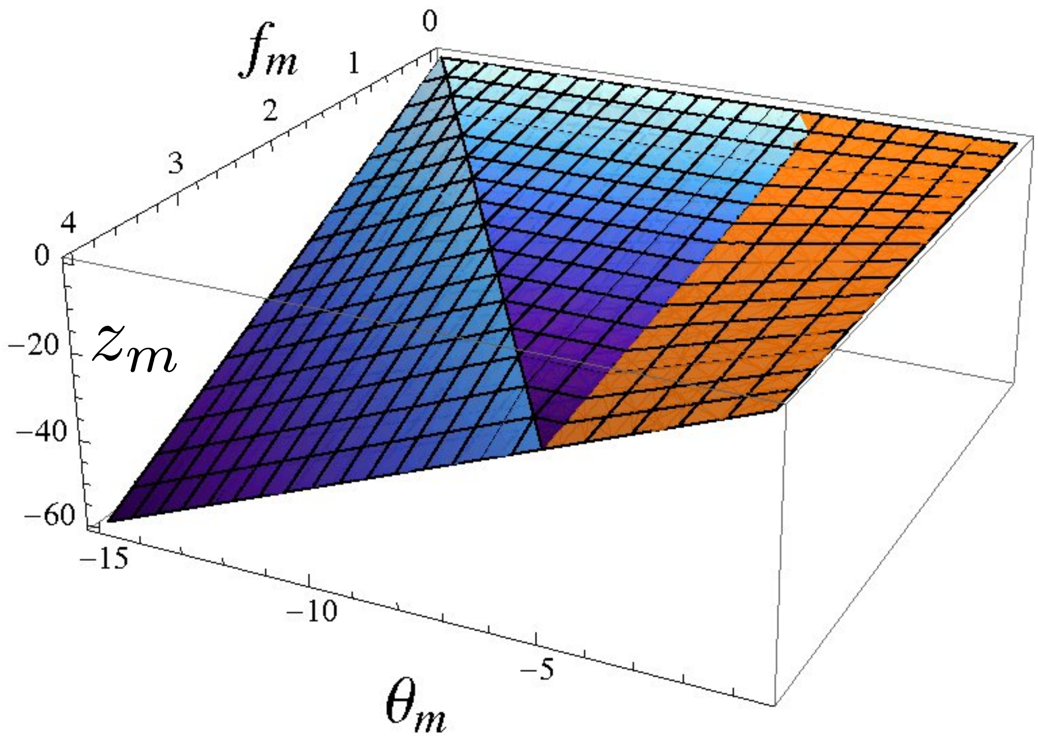
Concave Envelope
(e.g. McCormick (1976))

Each B&B subspace specifies bounds:

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

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

Tightness of relaxation improves with **branching**.



Branching

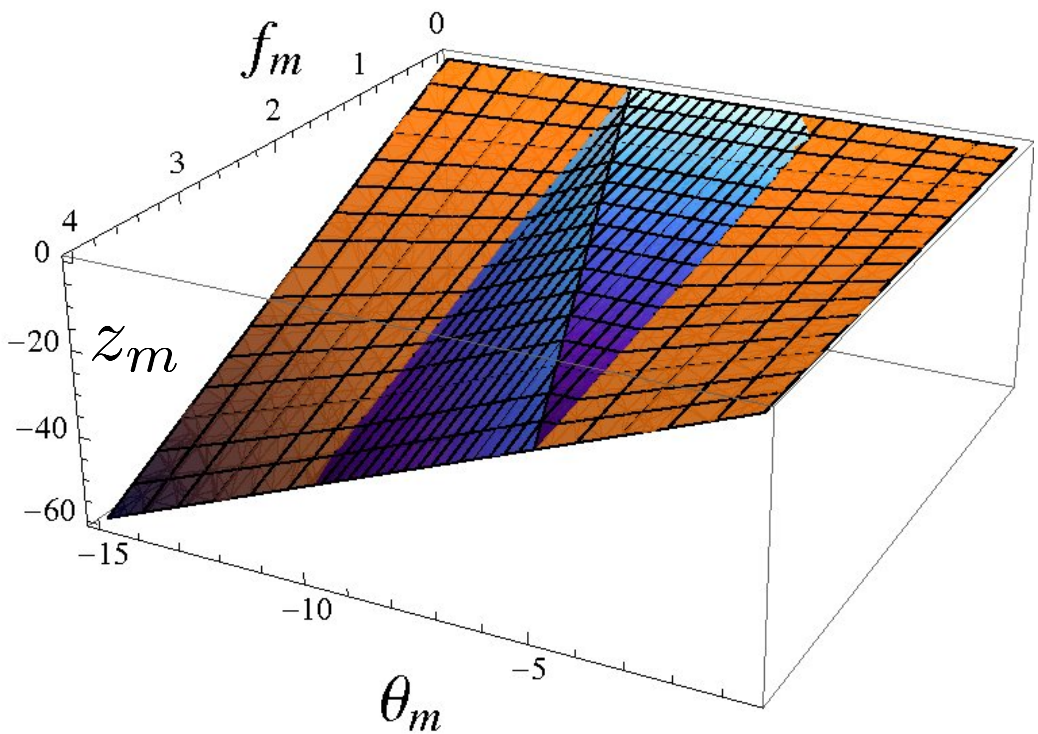
Concave Envelope
(e.g. McCormick (1976))

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

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)

- 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:

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

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:

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

RLT LP:

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

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

PROJECTIONS AND CONSTRAINTS

Viterbi Objective as a Quadratic Program

Variables:

θ_m	Log-probability for feature m
x_m	Corpus-wide feature count for m

Indices and constants:

m	Feature / model parameter 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.

$$\max \sum_m \theta_m x_m$$

Sum-to-one constraints on model parameters.

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

Parameters must be log-probabilities.

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

Model constraints.

$$A\vec{x} \leq b$$

Feature counts must be integers.

$$x_m \in \mathbb{Z}, \forall m \in \mathcal{I}$$

Grammar Induction as a Quadratic Program

Variables:

θ_m	Log-probability for feature m
x_m	Corpus-wide feature count for m

Indices and constants:

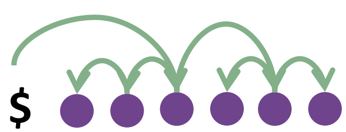
m	Feature / model parameter index
c	Conditional distribution index
$\mathcal{M}_c$	c^{th} Set of feature indices that sum to 1.0

Viterbi EM objective in log space.

Sum-to-one constraints on model parameters.

Parameters must be log-probabilities.

Tree constraints.



Feature counts must be integers.

$$\max \sum_m \theta_m x_m$$

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

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

$$A\vec{x} \leq b$$

$$x_m \in \mathbb{Z}, \forall m \in \mathcal{I}$$

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)

$$\sum_{j=1}^{N_s} \phi_{s0j} = N_s, \ \forall j \tag{21}$$

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

$$\phi_{sij} \leq N_s e_{sij}, \ \forall i, j \tag{23}$$

$$e_{sij} \in \{0, 1\}, \ \forall i, j \tag{24}$$

- Spanning tree is **projective**

Projectivity (Martins et al., 2009)

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

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

DMV root/child feature counts

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

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

DMV decision feature counts

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

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

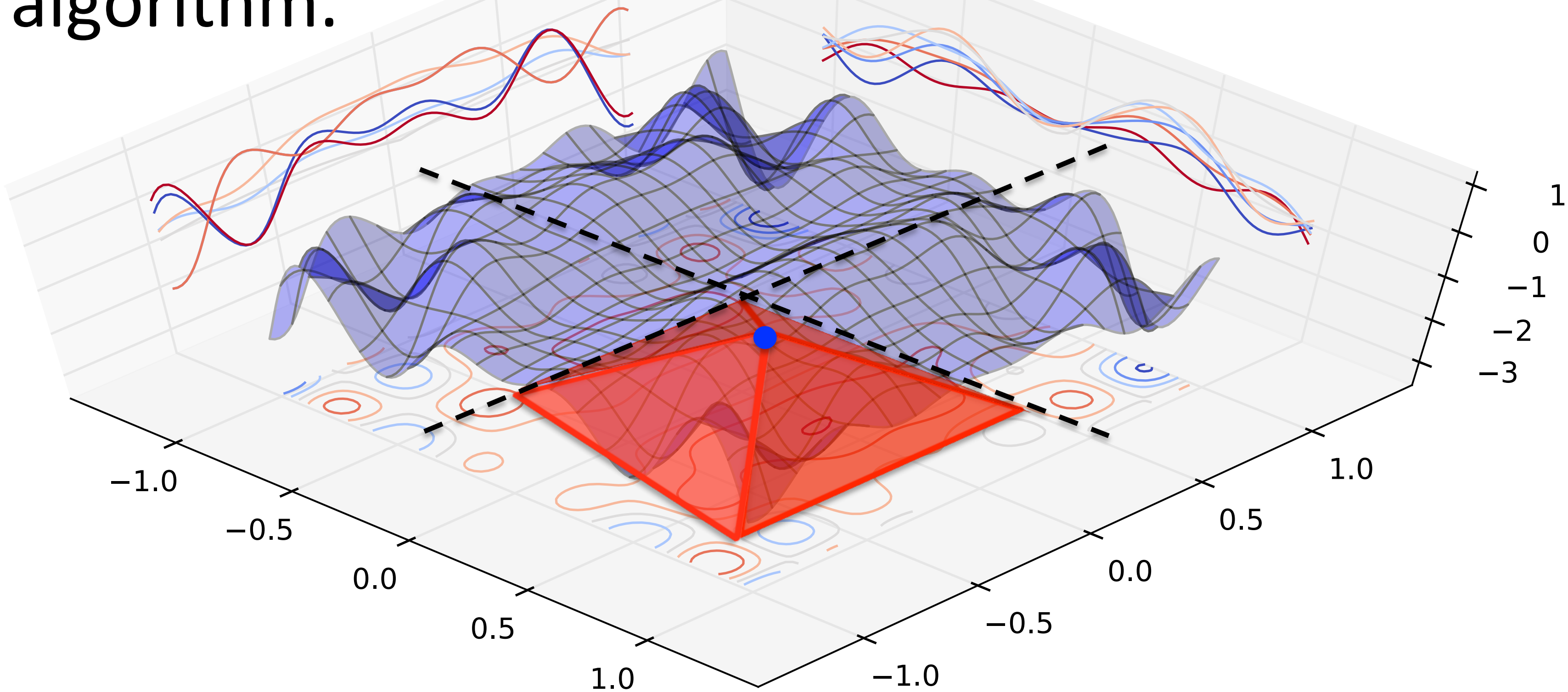
$$f^{(s,i)}_{\text{dec.L.0},t,\texttt{stop}} = 1 - f^{(s,i)}_{\text{dec.L.0},t,\texttt{cont}} \tag{30}$$

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

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

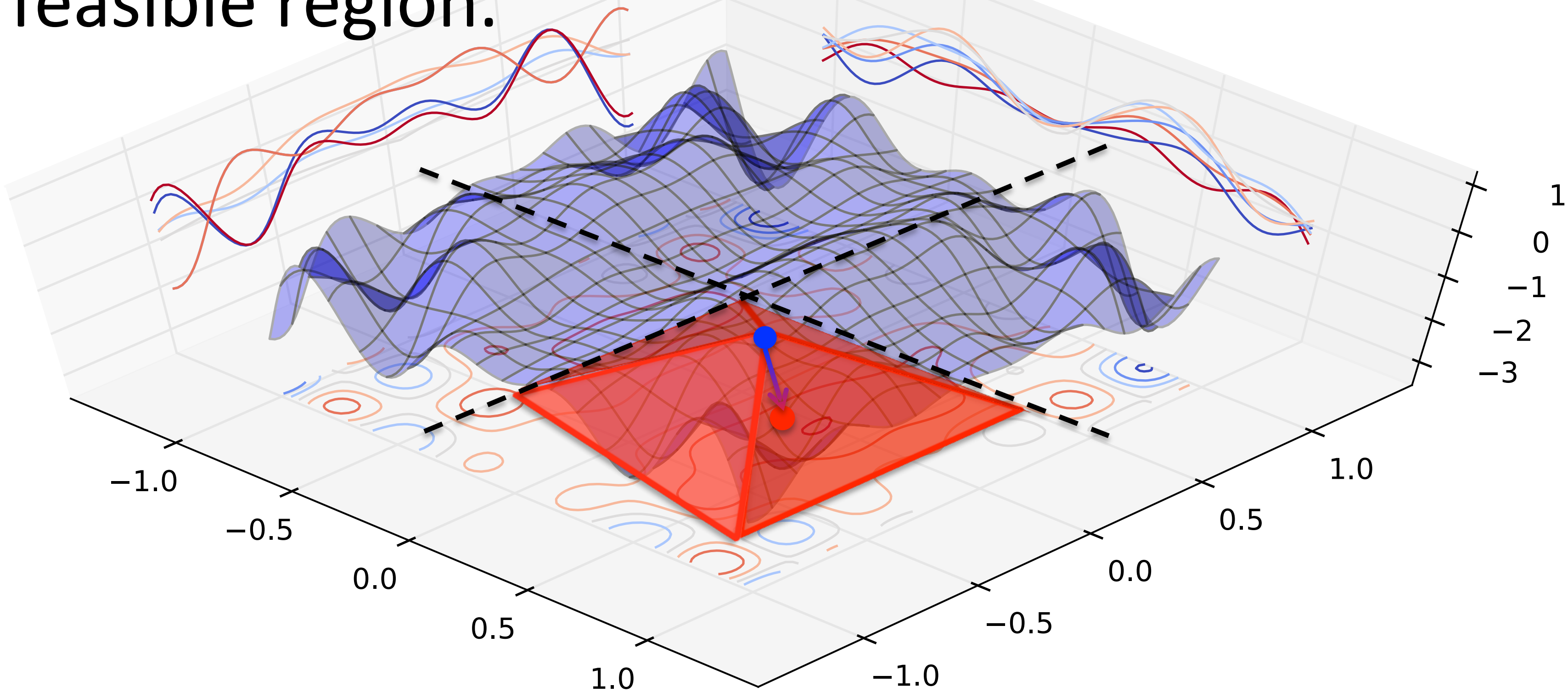
Background: Nonconvex Global Optimization

We solve the **relaxation** using the Simplex algorithm.



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 (Martins et al., 2009).

Linguistic Constraints

- 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).

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

Root $\rightarrow$ Auxiliary	Noun $\rightarrow$ Adjective
Root $\rightarrow$ Verb	Noun $\rightarrow$ Article
Verb $\rightarrow$ Noun	Noun $\rightarrow$ Noun
Verb $\rightarrow$ Pronoun	Noun $\rightarrow$ Numeral
Verb $\rightarrow$ Adverb	Preposition $\rightarrow$ Noun
Verb $\rightarrow$ Verb	Adjective $\rightarrow$ Adverb
Auxiliary $\rightarrow$ Verb	

Relaxed Viterbi EM with Linguistic Constraints

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

Variables:

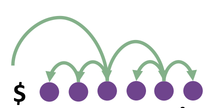
f_m	Corpus-wide feature count for m
e_{sij}	Indicator of an arc from i to j in tree s

Indices and constants:

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

Linear Viterbi objective.

$$\max \sum_m \theta_m f_m$$



Tree constraints.

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

Integer feature counts.

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

Linguistic constraints.

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

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
- ILP Dependency Parsing
  - Supervised approaches
    - Riedel and Clarke (2006)
    - Martins et al. (2009)
    - Riedel et al. (2012)
  - Inspired our unsupervised formulation
- 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
- Branch-and-bound
  - Chapellle et al. (2007) applied branch-and-bound to semi-supervised SVM training, with a relaxation derived from the dual.

EXPERIMENTS

Experimental Setup: Datasets

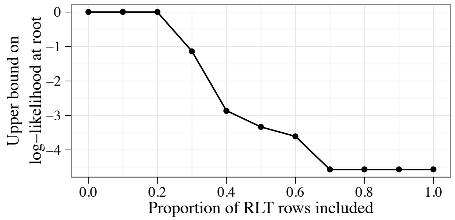
- Task: Unsupervised Dependency Parsing
- Toy Synthetic Data:
  - Generated from a synthetic DMV over three POS tags (Verb, Noun, Adjective)
  - Parameters chosen to favor short sentences with English word order
- Real Data:
  - 200 random sentences of no more than 10 tokens from the WSJ portion of the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

Experimental Setup

- Search Methods:
  - Branch-and-bound with various RLT relaxations
  - Viterbi EM with random restarts
- We consider each search method with/ without linguistic constraints

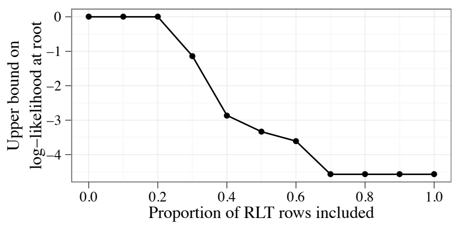
Experiments: Takeaways

- RLT produces **very tight relaxations**.
- Size of **RLT relaxation grows quadratically** with the length of the corpus.

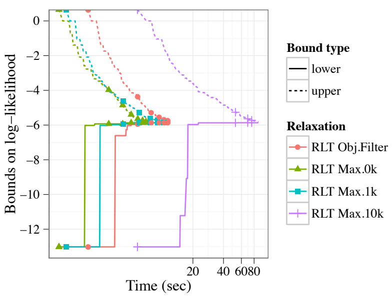


Experiments: Takeaways

- RLT produces **very tight relaxations**.
- Size of **RLT relaxation grows quadratically** with the length of the corpus.

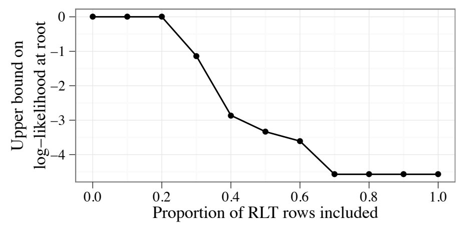


- **Random samples** and random projections can help!
- Toy problem (5 synthetic sentences) **solved to completion** (ε = 0.1).
- Tradeoff between relaxation size and speed.

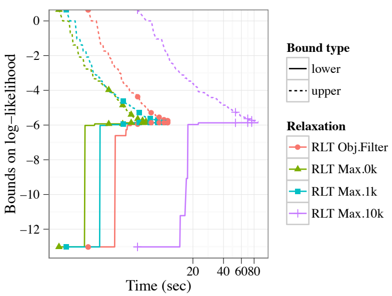


Experiments: Takeaways

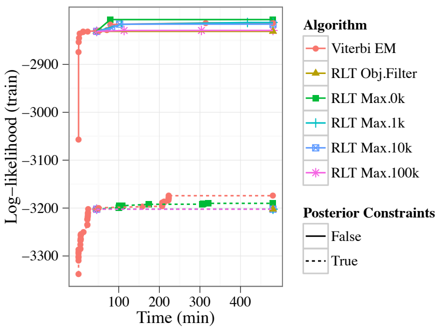
- RLT produces **very tight relaxations**.
- Size of **RLT relaxation grows quadratically** with the length of the corpus.



- **Random samples** and random projections can help!
- Toy problem (5 synthetic sentences) **solved to completion** ($\varepsilon = 0.1$).
- Tradeoff between relaxation size and speed.

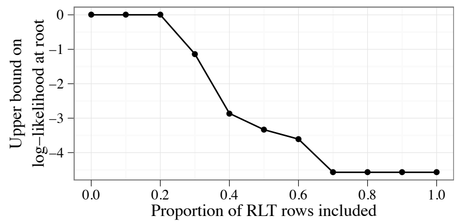


- On 200 WSJ sentences, global search **sometimes finds higher likelihood solutions** than local search in the same amount of time.

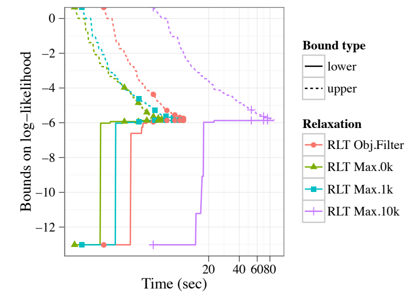


Experiments: Takeaways

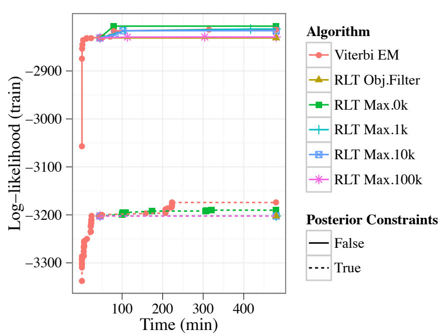
- RLT produces **very tight relaxations**.
- Size of **RLT relaxation grows quadratically** with the length of the corpus.



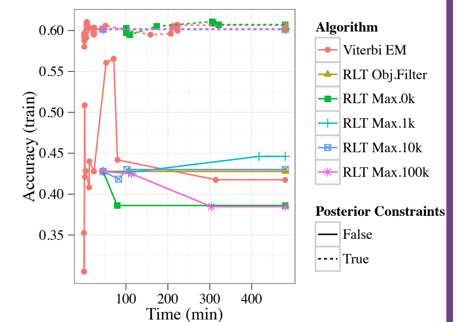
- **Random samples** and random projections can help!
- Toy problem (5 synthetic sentences) **solved to completion** (ε = 0.1).
- Tradeoff between relaxation size and speed.



- On 200 WSJ sentences, global search **sometimes finds higher likelihood solutions** than local search in the same amount of time.



- With linguistic constraints, global search **sometimes finds higher accuracy solutions** than local search in the same amount of time.



Summary

Contributions

- Formulation of the Viterbi objective as a **mathematical program**.
- **Global optimization framework** for nonconvex likelihood function of a latent variable model.
- Novel **posterior constrained Viterbi EM** baseline.
- Applied to grammar induction.

Future Work

- Development of tighter relaxations
  - Lagrangian relaxation / Dantzig-Wolfe decomposition
  - Semidefinite relaxations
- Better B&B search heuristics
- Apply to soft EM objective

Thank you!

Questions?

Additional slides available here: http://www.cs.jhu.edu/~mrg/