Efficient Discriminative Training Method for Structured Predictions

Huizhen Yu\textsuperscript{1}  Dimitri P. Bertsekas\textsuperscript{2}  Juho Rousu\textsuperscript{1}

\textsuperscript{1}Department of Computer Science
University of Helsinki

\textsuperscript{2}Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

MLG08, Helsinki, Finland, Jul. 4-5, 2008
Two Aspects in This Research

New Optimization Approach that can handle very large data sets
- Reparametrization
- Restricted simplicial decomposition
- Proximal point algorithm

Formulation of Discriminative Training of Generative Models
- Max margin
- Control of model deviation
- Similar formulations exist in the literature
Outline

Overview and Problem Formulation

Algorithm

Preliminary Experiments

Summary
Overview

We consider

- Discriminative training (DT) for structured predictions
  - formulation motivated by SVM (e.g., Collins ’02, Altun et al. ’03, Taskar et al. ’04)
  - enforce “margin constraints”
  - result in large scale optimization problems

We present a new dual optimization algorithm:

- Reparametrization for dimensionality reduction
- Applicable to extended DT formulations with additional parameter constraints and non-quadratic objectives

We focus on a particular type of problem:

- Discriminative training for generative models
  - discrete space DAG, log-linear models
  - supervised learning setting
  - an example of the extended DT formulation
Overview

We consider

- Discriminative training (DT) for structured predictions
  - formulation motivated by SVM
    (e.g., Collins '02, Altun et al. '03, Taskar et al. '04)
  - enforce “margin constraints”
  - result in large scale optimization problems

We present a new dual optimization algorithm:

- Reparametrization for dimensionality reduction
- Applicable to extended DT formulations
  with additional parameter constraints and non-quadratic objectives

We focus on a particular type of problem:

- Discriminative training for generative models
  - discrete space DAG, log-linear models
  - supervised learning setting
  - an example of the extended DT formulation
Overview

We consider

- Discriminative training (DT) for structured predictions
  - formulation motivated by SVM (e.g., Collins ’02, Altun et al. ’03, Taskar et al. ’04)
  - enforce “margin constraints”
  - result in large scale optimization problems

We present a new dual optimization algorithm:

- Reparametrization for dimensionality reduction
- Applicable to extended DT formulations with additional parameter constraints and non-quadratic objectives

We focus on a particular type of problem:

- Discriminative training for generative models
  - discrete space DAG, log-linear models
  - supervised learning setting
  - an example of the extended DT formulation
Setting for Supervised Learning

Consider directed graphical models with discrete spaces

- Examples: Bayesian networks (BN), hidden Markov models (HMM)
- Parameters of the model: a set of log of conditional probabilities
  \[ \theta = \{ \theta_i, i \in \mathcal{I} \}, \quad \theta_i : \ln p(X = \cdot | pa_X), \text{ for some variable } X \]

- Parameter constraints: \[ 1' e^{\theta_i} = 1, \ i \in \mathcal{I} \]

For training:

- Fully observed examples, indexed by \( \mathcal{K} \)
- \( \forall k \in \mathcal{K} \), specify prediction variables (considered as hidden) and observation variables (non-hidden)
- Prediction variables may be naturally determined by tasks, or, chosen just for the purpose of training
  e.g., choose different subsets of nodes for different exs. to cover the graph
- Optimize \( \theta \) using the SVM-like DT criteria
  enforce margin constraints

Use of such training: e.g., when prediction accuracy is important, when examples are likely to be dependent
Setting for Supervised Learning

Consider directed graphical models with discrete spaces

- **Examples**: Bayesian networks (BN), hidden Markov models (HMM)
- **Parameters of the model**: a set of log of conditional probabilities
  \[ \theta = \{ \theta_i, i \in \mathcal{I} \}, \quad \theta_i : \ln p(X = \cdot | \text{pa}_X), \text{ for some variable } X \]
- **Parameter constraints**: \( 1^{\prime} \mathbf{e}^{\theta_i} = 1, i \in \mathcal{I} \)

For training:

- Fully observed examples, indexed by \( \mathcal{K} \)
- \( \forall k \in \mathcal{K} \), specify prediction variables (considered as hidden) and observation variables (non-hidden)
- Prediction variables may be naturally determined by tasks, or, chosen just for the purpose of training
  e.g., choose different subsets of nodes for different exs. to cover the graph
- Optimize \( \theta \) using the SVM-like DT criteria
  enforce margin constraints

Use of such training: e.g., when prediction accuracy is important, when examples are likely to be dependent
Setting for Supervised Learning

Consider directed graphical models with discrete spaces

- Examples: Bayesian networks (BN), hidden Markov models (HMM)
- Parameters of the model: a set of log of conditional probabilities

\[ \theta = \{ \theta_i, i \in I \}, \quad \theta_i : \ln p(X = \cdot | \text{pa}_X), \text{ for some variable } X \]

- Parameter constraints: \(1'e^{\theta_i} = 1, i \in I\)

For training:

- Fully observed examples, indexed by \(K\)
- \(\forall k \in K\), specify prediction variables (considered as hidden) and observation variables (non-hidden)
- Prediction variables may be naturally determined by tasks, or, chosen just for the purpose of training
  e.g., choose different subsets of nodes for different exs. to cover the graph
- Optimize \(\theta\) using the SVM-like DT criteria
  enforce margin constraints

Use of such training: e.g., when prediction accuracy is important, when examples are likely to be dependent
Formulation of Discriminative Training Problem

Notation: for each example \( k \in \mathcal{K} \),

- \( S_k \): the space of all possible prediction outcomes
- \((s^*, o)\): values of hidden and non-hidden variables, resp.

Introduce margin constraints: \( \forall k \in \mathcal{K}, \forall s \in S_k \),

\[
\ln p(s, o ; \theta) - \ln p(s^*, o ; \theta) + l_k(s, s^*) \leq \epsilon_k,
\]

\( \epsilon_k \): positive slack variables for the usual non-ideal case; \( l_k \): loss function

- Meaning: ideally, after training, \( p(s \mid o) \) is peaked at \( s^* \)
- Write the linear margin constraints equivalently as

\[
\sum_{i \in \mathcal{I}} a_{i,k}(s)^t \theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in S_k, \quad k \in \mathcal{K}
\]
Formulation of Discriminative Training Problem

Notation: for each example $k \in \mathcal{K}$,

- $S_k$: the space of all possible prediction outcomes
- $(s^*, o)$: values of hidden and non-hidden variables, resp.

Introduce margin constraints: $\forall k \in \mathcal{K}, \forall s \in S_k$,

$$\ln p(s, o; \theta) - \ln p(s^*, o; \theta) + l_k(s, s^*) \leq \epsilon_k,$$

$\epsilon_k$: positive slack variables for the usual non-ideal case; $l_k$: loss function

- Meaning: ideally, after training, $p(s \mid o)$ is peaked at $s^*$
- Write the linear margin constraints equivalently as

$$\sum_{i \in \mathcal{I}} a_{i,k}(s) \theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in S_k, \; k \in \mathcal{K}$$
Formulation of Discriminative Training Problem

Notation: for each example $k \in \mathcal{K}$,

- $S_k$: the space of all possible prediction outcomes
- $(s^*, o)$: values of hidden and non-hidden variables, resp.

Introduce margin constraints: $\forall k \in \mathcal{K}, \forall s \in S_k$,

$$\ln p(s, o; \theta) - \ln p(s^*, o; \theta) + l_k(s, s^*) \leq \epsilon_k,$$

$\epsilon_k$: positive slack variables for the usual non-ideal case; $l_k$: loss function

- Meaning: ideally, after training, $p(s \mid o)$ is peaked at $s^*$
- Write the linear margin constraints equivalently as

$$\sum_{i \in \mathcal{I}} a_{i,k}(s) \theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in S_k, \ k \in \mathcal{K}$$
Primal Problem

Formulate training as solving the convex program:

\[
\begin{align*}
\text{(P)} \quad & \min_{\theta, \epsilon} - \sum_{i \in I} c_i^t \theta_i + \eta \sum_{k \in K} \epsilon_k \\
& \text{subj. } \sum_{i \in I} a_{i,k}(s)^t \theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in S_k, \ k \in K \quad \text{(marg.)}
\end{align*}
\]

\[1^t e^{\theta_i} = 1 \quad \text{relax to} \quad 1^t e^{\theta_i} \leq 1, \quad \forall i \in I\]

\[\theta_i \leq 0, \quad \forall i \in I, \ \epsilon_k \geq 0, \quad \forall k \in K\]

Objective function:

- First term: control degree of deviation from certain given parameters
  
  \(-c_i^t \theta_i\) comes from KL-divergence \(D(p\|q) = -\sum_j p_j \ln q_j - H(p)\)
  
  \(\forall i, \ \ln q : \theta_i, \ c_i \propto p = \text{some fixed distribution}\)
  
  \(p\) can be e.g., ML estimate, uniform distribution

- Second term: penalty for margin violation
Primal Problem

Formulate training as solving the convex program:

\[
\begin{align*}
\text{(P)} \quad & \min_{\theta, \epsilon} - \sum_{i \in I} c_i' \theta_i + \eta \sum_{k \in K} \epsilon_k \\
& \text{subj. } \sum_{i \in I} a_{i,k}(s)' \theta_i + b_k(s) \leq \epsilon_k, \ \forall s \in S_k, \ k \in K \quad \text{(marg.)}
\end{align*}
\]

Objective function:

- First term: control degree of deviation from certain given parameters
  \(-c_i' \theta_i\) comes from KL-divergence \(D(p||q) = -\sum_j p_j \ln q_j - H(p)\)

  \(\forall i, \ \ln q : \theta_i, \ c_i \propto p = \text{some fixed distribution}\)

  \(p\) can be e.g., ML estimate, uniform distribution

- Second term: penalty for margin violation
Primal Problem

Formulate training as solving the convex program:

\[
(P) \min_{\theta, \epsilon} - \sum_{i \in I} c'_i \theta_i + \eta \sum_{k \in K} \epsilon_k \\
\text{subj.} \sum_{i \in I} a_i, k(s)' \theta_i + b_k(s) \leq \epsilon_k, \ \forall s \in S_k, \ k \in K \quad \text{(marg.)}
\]

\[1'e^{\theta_i} = 1 \quad \text{relax to} \quad 1'e^{\theta_i} \leq 1, \ \forall i \in I\]
\[\theta_i \leq 0, \ \forall i \in I, \ \epsilon_k \geq 0, \ \forall k \in K\]

Objective function:

- First term: control degree of deviation from certain given parameters
  
  \[-c'_i \theta_i \text{ comes from KL-divergence } D(p\|q) = -\sum_j p_j \ln q_j - H(p)\]

  \[\forall i, \ \ln q : \theta_i, \quad c_i \propto p \text{ = some fixed distribution}\]

  \[p \text{ can be e.g., ML estimate, uniform distribution}\]

- Second term: penalty for margin violation
Outline

Overview and Problem Formulation

Algorithm

Preliminary Experiments

Summary
Reparametrization – Dimensionality Reduction

Margin constraints in \((P)\):

\[
\sum_{i \in \mathcal{I}} a_{i,k}(s)\theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in S_k, \quad k \in \mathcal{K}
\]  

(marg.)

Corresponding term in the Lagrangian function \(\mathcal{L}\):

with multipliers \(\beta = \{\beta_k(s), k \in \mathcal{K}, s \in S_k\}\),

\[
\sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) \left( \sum_{i \in \mathcal{I}} a_{i,k}(s)^\prime \theta_i + b_k(s) - \epsilon_k \right)
\]

\[
= \sum_{i \in \mathcal{I}} \left( \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) a_{i,k}(s)^\prime \right) \theta_i + \left( \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) b_k(s) \right) - \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) \epsilon_k
\]

\[
\overset{\text{def}}{=} \mu_i
\]

\[
\overset{\text{def}}{=} \omega
\]

- **Data-dependent linear transformation of \(\beta\)**
- \(\dim(\mu_i) = \dim(\theta_i)\), \(\dim(\omega) = 1\)
Reparameterization – Dimensionality Reduction

Margin constraints in (P):

\[ \sum_{i \in I} a_{i,k}(s)' \theta_i + b_k(s) \leq \epsilon_k, \ \forall s \in S_k, \ k \in \mathcal{K} \]  

(marg.)

Corresponding term in the Lagrangian function \( \mathcal{L} \):

with multipliers \( \beta = \{ \beta_k(s), k \in \mathcal{K}, s \in S_k \} \),

\[ \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) \left( \sum_{i \in I} a_{i,k}(s)' \theta_i + b_k(s) - \epsilon_k \right) \]

\[ = \sum_{i \in I} \left( \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) a_{i,k}(s)' \right) \theta_i + \left( \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) b_k(s) \right) - \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) \epsilon_k \]

\[ \overset{\text{def}}{=} \mu_i \]

\[ \overset{\text{def}}{=} \omega \]

- Data-dependent linear transformation of \( \beta \)
- \( \dim(\mu_i) = \dim(\theta_i), \ dim(\omega) = 1 \)
Reparametrization – Dimensionality Reduction

Margin constraints in (P):

\[
\sum_{i \in \mathcal{I}} a_{i,k}(s)' \theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in S_k, \quad k \in \mathcal{K}
\]  

(marg.)

Corresponding term in the Lagrangian function \( \mathcal{L} \):

with multipliers \( \beta = \{ \beta_k(s), k \in \mathcal{K}, s \in S_k \} \),

\[
\sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) \left( \sum_{i \in \mathcal{I}} a_{i,k}(s)' \theta_i + b_k(s) - \epsilon_k \right)
\]

\[
= \sum_{i \in \mathcal{I}} \left( \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) a_{i,k}(s)' \right) \theta_i + \left( \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) b_k(s) \right) - \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) \epsilon_k
\]

\[
\overset{\text{def}}{=} \mu_i \quad \overset{\text{def}}{=} \omega
\]

- Data-dependent linear transformation of \( \beta \)
- \( \dim(\mu_i) = \dim(\theta_i), \dim(\omega) = 1 \)
Reparametrization – Dimensionality Reduction

Margin constraints in (P):

\[ \sum_{i \in \mathcal{I}} a_{i,k}(s)' \theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in S_k, \, k \in \mathcal{K} \quad \text{(marg.)} \]

Corresponding term in the Lagrangian function \( \mathcal{L} \):

with multipliers \( \beta = \{ \beta_k(s), k \in \mathcal{K}, s \in S_k \} \),

\[
\sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) \left( \sum_{i \in \mathcal{I}} a_{i,k}(s)' \theta_i + b_k(s) - \epsilon_k \right)
\]

\[
= \sum_{i \in \mathcal{I}} \left( \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) a_{i,k}(s)' \right) \theta_i + \left( \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) b_k(s) \right) - \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) \epsilon_k
\]

\[ \overset{\text{def}}{=} \mu_i \quad \overset{\text{def}}{=} \omega \]

- Data-dependent linear transformation of \( \beta \)
- \( \dim(\mu_i) = \dim(\theta_i), \, \dim(\omega) = 1 \)
Size-Reduced Dual Problem

With an Implicit Set Constraint

Write the dual problem in terms of \((\mu, \omega)\) instead of \(\beta\):

\[
(D) \quad \max_{\mu, \omega, \lambda} \quad \omega - \sum_{i \in I} \lambda_i + \sum_{i \in I} q_i(\mu_i, \lambda_i)
\]

subj. \(\lambda \geq 0, \quad (\mu, \omega) \in D\)

- \(q_i\) terms: from minimizing \(\mathcal{L}\) w.r.t. primal variables

\[
q_i(\mu_i, \lambda_i) = \min_{\theta_i \leq 0} \left[ (\mu_i - c_i)' \theta_i + \lambda_i 1' e^{\theta_i} \right]
\]

- \(D\): an implicit set constraint determined by reparametrization

\[
D = \left\{ (\mu, \omega) \mid \mu_i = \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) a_{i,k}(s), \quad \omega = \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s) b_k(s), \quad \beta_k \geq 0, \quad 1' \beta_k \leq \eta, \quad \forall k \in \mathcal{K} \right\}
\]

- Dim. of dual function = Dim. of primal variables + \(|I| + 1\)
- Size of \((D)\) “independent” of \(|S_k|\) and \(|\mathcal{K}|\)
- \(D\) can be very complicated; apply feasible direction methods (RSD algorithm)
Size-Reduced Dual Problem
With an Implicit Set Constraint

Write the dual problem in terms of \((\mu, \omega)\) instead of \(\beta\):

\[
(D) \quad \max_{\mu,\omega,\lambda} \quad \omega - \sum_{i \in I} \lambda_i + \sum_{i \in I} q_i(\mu_i, \lambda_i)
\]
subj. \(\lambda \geq 0, \ (\mu, \omega) \in \mathcal{D}\)

- \(q_i\) terms: from minimizing \(\mathcal{L}\) w.r.t. primal variables
  \[q_i(\mu_i, \lambda_i) = \min_{\theta_i \leq 0} \left[ (\mu_i - c_i)'\theta_i + \lambda_i 1'e^\theta_i \right]\]

- \(\mathcal{D}\): an implicit set constraint determined by reparametrization
  \[\mathcal{D} = \left\{(\mu, \omega) \mid \mu_i = \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s)a_{i,k}(s), \ \omega = \sum_{k \in \mathcal{K}, s \in S_k} \beta_k(s)b_k(s), \beta_k \geq 0, \ 1'\beta_k \leq \eta, \forall k \in \mathcal{K}\right\}\]

- Dim. of dual function = Dim. of primal variables + \(|I| + 1\)
- Size of \((D)\) “independent” of \(|S_k|\) and \(|\mathcal{K}|\)
- \(\mathcal{D}\) can be very complicated; apply feasible direction methods (RSD algorithm)
Background: Feasible Direction Methods – Simplicial Decomposition

To deal with an implicit and complicated feasible region:

(1) Make successive inner approximation of the feasible region
   – Direction finding subproblems:
     for $\max_{z \in Z} Q(z)$, typically solve
     $$\max_{z \in Z} \nabla Q(z^t)'(z - z^t)$$
     In our case: “loss-augmented inference” (exact or approximate)

(2) Optimize the function over inner approximations
   – Master problems
Background: Feasible Direction Methods – Simplicial Decomposition

To deal with an implicit and complicated feasible region:

1. Make successive inner approximation of the feasible region
   - Direction finding subproblems:
     for $\max_{z \in Z} Q(z)$, typically solve
     $\max_{z \in Z} \nabla Q(z^t)'(z - z^t)$
   In our case: “loss-augmented inference” (exact or approximate)

2. Optimize the function over inner approximations
   - Master problems
Background: Feasible Direction Methods – Simplicial Decomposition

To deal with an implicit and complicated feasible region:

1. Make successive inner approximation of the feasible region
   - **Direction finding subproblems**: for \( \max_{z \in Z} Q(z) \), typically solve
     \[
     \max_{z \in \mathcal{Z}} \nabla Q(z^t)' (z - z^t)
     \]
     In our case: “loss-augmented inference” (exact or approximate)

2. Optimize the function over inner approximations
   - **Master problems**
Restricted Simplicial Decomposition (RSD)

RSD (Hearn et al. ’87):

- Set an upper limit to the dimension of the simplex: complexity of master problems independent of the original problem
- Apply a projected Newton method (Bertsekas ’82) to solve master problems: superlinear convergence, finite convergence for quadratic objective

Points at ascent dir.

Complicated feasible region found by dir-finding
Restricted Simplicial Decomposition (RSD)

RSD (Hearn et al. ’87):

- Set an upper limit to the dimension of the simplex: *complexity of master problems independent of the original problem*
- Apply a projected Newton method (Bertsekas ’82) to solve master problems: superlinear convergence, finite convergence for quadratic objective

![Diagram of RSD process](image-url)
Algorithm: Reparametrization + RSD + · · ·
Motivation for Applying the Proximal Point Algorithm

Difficulty of applying RSD directly to solve (D):

- The dual function is not everywhere real-valued (unlike the QP case)
  \[ \mu \text{ needs to satisfy: } \mu_i \leq c_i, i \in I \]

Finding a point in \( \{ (\mu, \omega) | \mu_i \leq c_i, i \in I, \omega \in \mathbb{R} \} \cap \mathcal{D} \) is costly.

Solution:

- Add a quadratic term \( \frac{\gamma_0}{2} \| \theta - \theta^0 \|^2 \) to (P)
- Moving the center \( \theta^0 \) in a certain way to approach an optimal solution of (P) – known as the **proximal point algorithm**:

**Exact form:** to solve \( \min_{x \in X} f(x) \), iterate

\[
x^{n+1} = \arg \min_{x \in X} \left[ f(x) + \frac{\gamma_n}{2} \| x - x^n \|^2 \right], \quad \text{with } \gamma_n \geq 0, \sup_n \gamma_n < \infty.
\]
Overview and Problem Formulation

Algorithm

Preliminary Experiments

Summary

Algorithm: Reparametrization + RSD + · · ·

Motivation for Applying the Proximal Point Algorithm

Difficulty of applying RSD directly to solve (D):

- The dual function is not everywhere real-valued (unlike the QP case)
  \[ \mu \text{ needs to satisfy: } \mu_i \leq c_i, i \in \mathcal{I} \]

  Finding a point in \( \{ (\mu, \omega) | \mu_i \leq c_i, i \in \mathcal{I}, \omega \in \mathbb{R} \} \cap \mathcal{D} \) is costly.

Solution:

- Add a quadratic term \( \frac{\gamma_0}{2} \| \theta - \theta^0 \|^2 \) to (P)
- Moving the center \( \theta^0 \) in a certain way to approach an optimal solution of (P) – known as the \textit{proximal point algorithm}:

Exact form: to solve \( \min_{x \in X} f(x) \), iterate

\[
x^{n+1} = \arg\min_{x \in X} \left[ f(x) + \frac{\gamma_n}{2} \| x - x^n \|^2 \right], \quad \text{with } \gamma_n \geq 0, \sup_n \gamma_n < \infty.
\]
Dual Proximal Point Algorithm

We solve a sequence of regularized primal problems by dual optimization with reparametrization and RSD:

\[(P_n) \min_{\theta, \epsilon} - \sum_{i \in I} c_i' \theta_i + \eta \sum_{k \in K} \epsilon_k + \frac{\gamma_n}{2} \|\theta - \theta^n\|^2\]

subj. \(\sum_{i \in I} a_{i,k}(s)' \theta_i + b_k(s) \leq \epsilon_k, \forall s \in S_k, k \in K\)

\(1' e^{\theta_i} \leq 1, \forall i \in I, \epsilon_k \geq 0, \forall k \in K\)

\[(D_n) \max_{\mu, \omega, \lambda} \omega - \sum_{i \in I} \lambda_i + \sum_{i \in I} q^n_i(\mu_i, \lambda_i)\]

subj. \(\lambda \geq 0, (\mu, \omega) \in D\)

where \(q^n_i(\mu_i, \lambda_i) = \min_{\theta_i \in \mathbb{R}^{d_i}} \left[ (\mu_i - c_i)' \theta_i + \lambda_i 1' e^{\theta_i} + \frac{\gamma_n}{2} \|\theta_i - \theta^n_i\|^2 \right].\)

- Can efficiently evaluate \(q^n_i\) (Newton’s method, global quadratic convergence) and its 1st and 2nd order derivatives
- \(D\) does not depend on \(\theta^n\)
Dual Proximal Point Algorithm

We solve a sequence of regularized primal problems by dual optimization with reparametrization and RSD:

\[ \begin{align*}
(P_n) \quad & \min_{\theta, \epsilon} - \sum_{i \in I} c_i^t \theta_i + \eta \sum_{k \in K} \epsilon_k + \frac{\gamma n}{2} \|\theta - \theta^n\|^2 \\
& \text{subj.} \sum_{i \in I} a_{i,k}(s)^t \theta_i + b_k(s) \leq \epsilon_k, \ \forall s \in S_k, \ k \in K \\
& \quad 1^t e^{\theta_i} \leq 1, \ \forall i \in I, \ \epsilon_k \geq 0, \ \forall k \in K
\end{align*} \]

\[ \begin{align*}
(D_n) \quad & \max_{\mu, \omega, \lambda} \omega - \sum_{i \in I} \lambda_i + \sum_{i \in I} q^n_i(\mu_i, \lambda_i) \\
& \text{subj.} \lambda \geq 0, \ (\mu, \omega) \in \mathcal{D}
\end{align*} \]

where \[ q^n_i(\mu_i, \lambda_i) = \min_{\theta_i \in \mathbb{R}^{d_i}} \left[ (\mu_i - c_i)^t \theta_i + \lambda_i 1^t e^{\theta_i} + \frac{\gamma n}{2} \|\theta_i - \theta^n_i\|^2 \right]. \]

- Can efficiently evaluate \( q^n_i \) (Newton’s method, global quadratic convergence) and its 1st and 2nd order derivatives
- \( \mathcal{D} \) does not depend on \( \theta^n \)
Algorithm Chart from Dual Viewpoint

RSD Iterations

form a new master problem

inner appr. of D
expand inn.-appr.

Master Problem
apply projected Newton to appr. solve the dual on the inner approximation

direction finding
loss-augmented inference

Dual Function Evaluation
current dual solution + the degree of satisfying optimality conditions
center

determine center update

Overview and Problem Formulation
Algorithm
Preliminary Experiments
Summary
Algorithm Variants with Same Idea

Alternative reparametrization for working sets:

- Partition training data \( \mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2 \cup \cdots \cup \mathcal{K}_m \)
- Introduce \((\mu^j, \omega^j), j = 1, \ldots, m\) by grouping respective terms in \( \mathcal{L} \):

\[
\sum_{i \in \mathcal{I}} \left( \sum_{j=1}^{m} \sum_{k \in \mathcal{K}_j, s \in \mathcal{S}_k} \beta_k(s) a_{i,k}(s)' \right) \theta_i + \left( \sum_{j=1}^{m} \sum_{k \in \mathcal{K}_j, s \in \mathcal{S}_k} \beta_k(s) b_k(s) \right)
\]

\( \text{def} = \mu^j \quad \text{def} = \omega^j \)

- Dual problem with implicit set constraints \((\mu^j, \omega^j) \in \mathcal{D}_j, j = 1, \ldots, m\) relation with the first reparametrization:

\[
\mu = \sum_{j=1}^{m} \mu^j, \quad \omega = \sum_{j=1}^{m} \omega^j, \quad \mathcal{D} = \mathcal{D}_1 + \mathcal{D}_2 + \cdots + \mathcal{D}_m
\]

- Special case/connection with cutting plane-like methods:
  singleton \( \mathcal{K}_j \), \( m = |\mathcal{K}| \)
Further remarks on reparametrization:

- Arbitrary and varying working sets can also be handled in the first reparametrization \((\mu, \omega)\): use the inner approximation view
- For different margin violation penalties: e.g., quadratic or loss-rescaled slacks (Tsochantaridis et al. ’05); \(D\) may be unbounded, but the same algorithm can be applied.

Note:

- Reparametrization preserves the inference problem structure
- On use of working sets: proper batch size + coordinate ascent trades off the complexity of direction finding subproblems with that of master problems, and achieves overall efficiency.
Algorithm Behavior and Comparisons of Working Set Sizes

Synthetic HMM data:
10 states, 7 observations
1000 sequences/length 50
\[ \text{dim}(\theta) = 180, |\mathcal{I}| = 21 \]

Batch size \( \times m \):
- B \( 100 \times 10 \)
- G \( 500 \times 2 \)
- M \( 1000 \times 1 \)
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview and Problem Formulation</td>
</tr>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td>Preliminary Experiments</td>
</tr>
<tr>
<td>Summary</td>
</tr>
</tbody>
</table>
I: the Synthetic HMM Example

HMM with 10 states and 7 observations:

Dynamics: clockwise, random jump w/ a small probability \( \approx 0.3 \)

Observation: uniform

- Training: 1000 seq. of length 50, \( c_i = \text{uniform} \)
- Test: 100 seq. of length 50, average over 10 runs
  measure loss on MAP state seq. loss: distance on the ring

Test loss:
- DT: \( 82.2 \pm 13.5 \) per seq.
- ML: \( 101.6 \pm 14.0 \) per seq.

Comparison of the dimensionalities of dual variables:

- \( |\mathcal{I}| = 21, \dim(\theta) = 180, \dim(\beta) = 1000 \times 10^{50} \)
- reparametrization w/ \( m \) working sets:
  \( \dim = m \times 181 + 21 \)
- “edge-wise”/“marginal polytope” parametrization:
  \( \dim = 1000 \times 50 \times (10 \times 10) + 21 \)
II: Yeast Dataset – a Case Study on Modeling

UCI Yeast Dataset (discretized)/ multiclass classification

- 9 variables with BN structure (given)
- \(|\mathcal{I}| = 60\) and \(\dim(\theta) = 191\)
- loss: classification error
- 1484 data points: 1115 (80%) for training and 296 (20%) for testing

Further selection from training examples

- Select instances \((s^*, o)\) such that
  \[
  \max_s \ln p(s \mid o; \theta_{ML}) - \ln p(s^* \mid o; \theta_{ML}) \leq \delta, \quad \delta \geq 0 : \text{ selection level}
  \]
- Reason: avoid difficult instances
  alternative to further selection: set loss differently for each instance in training
II: Yeast Dataset – a Case Study on Modeling

UCI Yeast Dataset (discretized)/ multiclass classification

- 9 variables with BN structure (given)

- \(|\mathcal{I}| = 60\) and \(\dim(\theta) = 191\)

- loss: classification error

- 1484 data points: 1115 (80\%) for training and 296 (20\%) for testing

Further selection from training examples

- Select instances \((s^*, o)\) such that

  \[
  \max_s \ln p(s | o ; \theta_{ML}) - \ln p(s^* | o ; \theta_{ML}) \leq \delta, \quad \delta \geq 0 : \text{selection level}
  \]

- Reason: avoid difficult instances

alternative to further selection: set loss differently for each instance in training
II: Yeast Dataset – a Case Study on Modeling

\[ \begin{align*}
B \quad & c_i = \text{ML weighted by } \gamma_i > 0 \\
& \| \theta^* - \theta_{ML} \| : 0.18 \pm 0.10 \\
G \quad & c_i = \text{uniform} \\
& \| \theta^* - \theta_{ML} \| : 4.18 \pm 0.03 
\end{align*} \]
II: Yeast Dataset – a Case Study on Modeling

\[ G \quad c_i = \text{uniform} \]
\[ \| \theta^* - \theta_{ML} \| : 4.18 \pm 0.03 \]

\[ R \quad c_i = 0, \text{ use } \| \theta \|^2 \text{ as regularizer} \]
\[ \| \theta^* - \theta_{ML} \| : 4.82 \pm 0.04 \]
Outline

Overview and Problem Formulation

Algorithm

Preliminary Experiments

Summary
Summary of our algorithm for solving large margin training problems:

- Reparametrization + RSD + proximal point algorithm
- Combine dimensionality reduction, differentiable optimization of feasible direction type, and regularization

For discriminative training of generative models, need to study:

- Tradeoff between faithfulness to the data and discriminative capacity
- Effect of the relaxed sum-of-probabilities constraint
- Combination with structure learning