Efficient Discriminative Training Method for Structured Predictions

Huizhen Yu[1] Dimitri P. Bertsekas[2] Juho Rousu[1]

[1]Department of Computer Science
University of Helsinki

[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

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

$$\theta = \{\theta_i, i \in \mathcal{I}\}, \quad \theta_i : \ln p(X = \cdot \mid pa_X), \text{ for some variable } X$$

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

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 θ 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

$$\theta = \{\theta_i, i \in \mathcal{I}\}, \quad \theta_i : \ln p(X = \cdot \mid pa_X), \text{ for some variable } X$$

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

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 θ 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

$$\theta = \{\theta_i, i \in \mathcal{I}\}, \quad \theta_i : \ln p(X = \cdot \mid pa_X), \text{ for some variable } X$$

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

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 θ 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 $k \in \mathcal{K}$,

- $\mathcal{S}_k$: the space of all possible prediction outcomes
- (s^*, o): values of hidden and non-hidden variables, resp.

Introduce margin constraints: $\forall k \in \mathcal{K}, \; \forall s \in \mathcal{S}_k,$

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

ϵ_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

$$\sum_{i \in \mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in \mathcal{S}_k, \; k \in \mathcal{K}$$

Formulation of Discriminative Training Problem

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

- $\mathcal{S}_k$: the space of all possible prediction outcomes
- (s^*, o): values of hidden and non-hidden variables, resp.

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

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

ϵ_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

$$\sum_{i \in \mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in \mathcal{S}_k,\ k \in \mathcal{K}$$

Formulation of Discriminative Training Problem

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

- $\mathcal{S}_k$: the space of all possible prediction outcomes
- (s^*, o): values of hidden and non-hidden variables, resp.

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

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

ϵ_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

$$\sum_{i \in \mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in \mathcal{S}_k,\ k \in \mathcal{K}$$

Primal Problem

Formulate training as solving the convex program:

$$(P) \quad \min_{\theta,\epsilon} \; -\sum_{i\in\mathcal{I}} c_i'\theta_i + \eta \sum_{k\in\mathcal{K}} \epsilon_k$$

$$\text{subj.} \; \sum_{i\in\mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \le \epsilon_k, \;\; \forall s \in \mathcal{S}_k,\; k \in \mathcal{K} \qquad \text{(marg.)}$$

$$\mathbf{1}'e^{\theta_i} = 1 \xrightarrow{\text{relax to}} \quad \mathbf{1}'e^{\theta_i} \le 1, \;\; \forall i \in \mathcal{I}$$

$$\theta_i \le 0, \;\; \forall i \in \mathcal{I}, \;\; \epsilon_k \ge 0, \;\; \forall k \in \mathcal{K}$$

Objective function:

- First term : control degree of deviation from certain given parameters

  $-c_i'\theta_i$ comes from KL-divergence $D(p||q) = -\sum_j p_j \ln q_j - H(p)$

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

  p can be e.g., ML estimate, uniform distribution
- Second term: penalty for margin violation

Primal Problem

Formulate training as solving the convex program:

$$(\mathrm{P}) \qquad \min_{\theta,\epsilon} \; -\sum_{i\in\mathcal{I}} c_i'\theta_i + \eta \sum_{k\in\mathcal{K}} \epsilon_k$$

$$\text{subj.} \sum_{i\in\mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \le \epsilon_k, \quad \forall s \in \mathcal{S}_k,\; k \in \mathcal{K} \qquad \text{(marg.)}$$

$$\mathbf{1}'e^{\theta_i} = 1 \xrightarrow{\text{relax to}} \mathbf{1}'e^{\theta_i} \le 1, \quad \forall i \in \mathcal{I}$$

$$\theta_i \le 0, \quad \forall i \in \mathcal{I}, \quad \epsilon_k \ge 0, \quad \forall k \in \mathcal{K}$$

Objective function:

- First term : control degree of deviation from certain given parameters

  $-c_i'\theta_i$ comes from KL-divergence $D(p\|q) = -\sum_j p_j \ln q_j - H(p)$

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

  p can be e.g., ML estimate, uniform distribution
- Second term: penalty for margin violation

Primal Problem

Formulate training as solving the convex program:

$$(\mathrm{P}) \qquad \min_{\theta,\epsilon} \; -\sum_{i\in\mathcal{I}} c_i'\theta_i + \eta \sum_{k\in\mathcal{K}} \epsilon_k$$

$$\text{subj.} \; \sum_{i\in\mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \le \epsilon_k, \quad \forall s \in \mathcal{S}_k,\; k \in \mathcal{K} \qquad \text{(marg.)}$$

$$\mathbf{1}'e^{\theta_i} = 1 \;\xrightarrow{\text{relax to}}\; \mathbf{1}'e^{\theta_i} \le 1, \quad \forall i \in \mathcal{I}$$

$$\theta_i \le 0, \quad \forall i \in \mathcal{I}, \quad \epsilon_k \ge 0, \quad \forall k \in \mathcal{K}$$

Objective function:

- First term : control degree of deviation from certain given parameters

  $-c_i'\theta_i$ comes from KL-divergence $D(p\|q) = -\sum_j p_j \ln q_j - H(p)$

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

  p can be e.g., ML estimate, uniform distribution
- Second term: penalty for margin violation

Outline

Reparametrization – Dimensionality Reduction

Margin constraints in (P):

$$\sum_{i\in\mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \le \epsilon_k, \quad \forall s\in\mathcal{S}_k,\ k\in\mathcal{K} \qquad \text{(marg.)}$$

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

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

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

$$= \sum_{i\in\mathcal{I}} \underbrace{\left(\sum_{k\in\mathcal{K},s\in\mathcal{S}_k} \beta_k(s)\, a_{i,k}(s)'\right)}_{\stackrel{\text{def}}{=}\ \mu_i}\theta_i + \underbrace{\left(\sum_{k\in\mathcal{K},s\in\mathcal{S}_k} \beta_k(s)\, b_k(s)\right)}_{\stackrel{\text{def}}{=}\ \omega} - \sum_{k\in\mathcal{K},s\in\mathcal{S}_k} \beta_k(s)\,\epsilon_k$$

- *Data-dependent linear transformation* of β
- $\dim(\mu_i) = \dim(\theta_i)$, $\dim(\omega) = 1$

Reparametrization – Dimensionality Reduction

Margin constraints in (P):

$$\sum_{i \in \mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in \mathcal{S}_k, \; k \in \mathcal{K} \qquad \text{(marg.)}$$

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

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

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

$$= \sum_{i \in \mathcal{I}} \underbrace{\left(\sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s)\, a_{i,k}(s)' \right)}_{\stackrel{def}{=} \mu_i} \theta_i + \underbrace{\left(\sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s)\, b_k(s) \right)}_{\stackrel{def}{=} \omega} - \sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s)\, \epsilon_k$$

- *Data-dependent linear transformation* of β
- $\dim(\mu_i) = \dim(\theta_i)$, $\dim(\omega) = 1$

Reparametrization – Dimensionality Reduction

Margin constraints in (P):

$$\sum_{i \in \mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \le \epsilon_k, \quad \forall s \in \mathcal{S}_k,\ k \in \mathcal{K} \qquad \text{(marg.)}$$

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

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

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

$$= \sum_{i \in \mathcal{I}} \underbrace{\left(\sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s)\, a_{i,k}(s)' \right)}_{\stackrel{def}{=} \mu_i} \theta_i + \underbrace{\left(\sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s)\, b_k(s) \right)}_{\stackrel{def}{=} \omega} - \sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s)\, \epsilon_k$$

- *Data-dependent linear transformation* of β
- $\dim(\mu_i) = \dim(\theta_i)$, $\dim(\omega) = 1$

Reparametrization – Dimensionality Reduction

Margin constraints in (P):

$$\sum_{i \in \mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \leq \epsilon_k, \quad \forall s \in \mathcal{S}_k, \; k \in \mathcal{K} \qquad \text{(marg.)}$$

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

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

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

$$= \sum_{i \in \mathcal{I}} \underbrace{\left(\sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s)\, a_{i,k}(s)' \right)}_{\stackrel{\text{def}}{=} \mu_i} \theta_i + \underbrace{\left(\sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s)\, b_k(s) \right)}_{\stackrel{\text{def}}{=} \omega} - \sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s)\, \epsilon_k$$

- *Data-dependent linear transformation* of β
- $\dim(\mu_i) = \dim(\theta_i)$, $\dim(\omega) = 1$

Size-Reduced Dual Problem

With an Implicit Set Constraint

Write the dual problem in terms of (μ, ω) instead of β:

$$\text{(D)} \qquad \max_{\mu,\omega,\lambda} \; \omega - \sum_{i \in \mathcal{I}} \lambda_i + \sum_{i \in \mathcal{I}} q_i(\mu_i, \lambda_i)$$

$$\text{subj. } \lambda \geq 0, \quad (\mu, \omega) \in \mathcal{D}$$

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

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

- $\mathcal{D}$: an implicit set constraint determined by reparametrization

$$\mathcal{D} = \Big\{ (\mu, \omega) \,\Big|\, \mu_i = \sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s) a_{i,k}(s), \;\; \omega = \sum_{k \in \mathcal{K}, s \in \mathcal{S}_k} \beta_k(s) b_k(s),$$
$$\beta_k \geq 0, \;\; \mathbf{1}'\beta_k \leq \eta, \; \forall k \in \mathcal{K} \Big\}$$

- Dim. of dual function $=$ Dim. of primal variables $+|\mathcal{I}| + 1$
- Size of (D) "independent" of $|\mathcal{S}_k|$ and $|\mathcal{K}|$
- $\mathcal{D}$ 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 (μ, ω) instead of β:

$$\text{(D)} \qquad \max_{\mu,\omega,\lambda} \ \omega - \sum_{i\in\mathcal{I}} \lambda_i + \sum_{i\in\mathcal{I}} q_i(\mu_i, \lambda_i)$$

$$\text{subj. } \lambda \geq 0, \quad (\mu, \omega) \in \mathcal{D}$$

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

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

- $\mathcal{D}$: an implicit set constraint determined by reparametrization

$$\mathcal{D} = \Big\{ (\mu, \omega) \,\Big|\, \mu_i = \sum_{k\in\mathcal{K}, s\in\mathcal{S}_k} \beta_k(s) a_{i,k}(s), \ \omega = \sum_{k\in\mathcal{K}, s\in\mathcal{S}_k} \beta_k(s) b_k(s),$$
$$\beta_k \geq 0, \ \mathbf{1}'\beta_k \leq \eta, \ \forall k \in \mathcal{K} \Big\}$$

- Dim. of dual function $=$ Dim. of primal variables $+|\mathcal{I}| + 1$
- Size of (D) "independent" of $|\mathcal{S}_k|$ 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 $\max_{z \in \mathcal{Z}} Q(z)$, 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*

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 $\max_{z \in \mathcal{Z}} Q(z)$, 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*

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 $\max_{z \in \mathcal{Z}} Q(z)$, typically solve

$$\max_{z \in \mathcal{Z}} \nabla Q(z^t)'(z - z^t)$$

In our case: “loss-augmented inference” (exact or approximate)

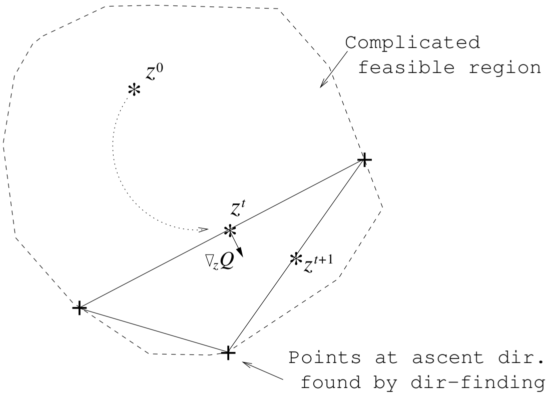
(2) Optimize the function over inner approximations

– *Master problems*

Restricted Simplicial Decomposition (RSD)

RSD (Hearn et al. '87):

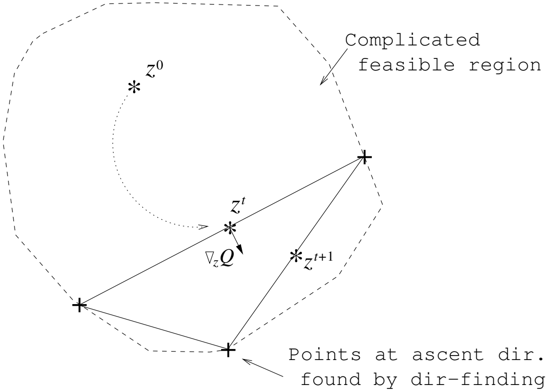
- Set an upper limit to the dimension of the simplex: *complexity of master problems independent of the original problem*
- Apply a projected Newton method (Bertsekas '82) to solve master problems: superlinear convergence, finite convergence for quadratic objective



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



Algorithm: Reparametrization + RSD + $\cdots$

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)

  μ needs to satisfy: $\mu_i \leq c_i, i \in \mathcal{I}$

  Finding a point in $\{(\mu, \omega) \mid \mu_i \leq c_i, i \in \mathcal{I}, \omega \in \Re\} \cap \mathcal{D}$ is costly.

Solution:

- Add a quadratic term $\frac{\gamma_0}{2}\|\theta - \theta^0\|^2$ to (P)
- Moving the center θ^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

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

Algorithm: Reparametrization + RSD + $\cdots$

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)

  μ needs to satisfy: $\mu_i \leq c_i, i \in \mathcal{I}$

  Finding a point in $\{(\mu, \omega) \mid \mu_i \leq c_i, i \in \mathcal{I}, \omega \in \Re\} \cap \mathcal{D}$ is costly.

Solution:

- Add a quadratic term $\frac{\gamma_0}{2}\|\theta - \theta^0\|^2$ to (P)
- Moving the center θ^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

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

Dual Proximal Point Algorithm

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

$$
\begin{aligned}
(\mathrm{P}_n) \qquad & \min_{\theta,\epsilon} \; -\sum_{i\in\mathcal{I}} c_i'\theta_i + \eta \sum_{k\in\mathcal{K}} \epsilon_k + \frac{\gamma_n}{2}\|\theta - \theta^n\|^2 \\
& \text{subj.} \; \sum_{i\in\mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \le \epsilon_k, \; \forall s \in \mathcal{S}_k, \; k \in \mathcal{K} \\
& \qquad \mathbf{1}'e^{\theta_i} \le 1, \;\; \forall i \in \mathcal{I}, \quad \epsilon_k \ge 0, \;\; \forall k \in \mathcal{K}
\end{aligned}
$$

$$
\begin{aligned}
(\mathrm{D}_n) \qquad & \max_{\mu,\omega,\lambda} \; \omega - \sum_{i\in\mathcal{I}} \lambda_i + \sum_{i\in\mathcal{I}} q_i^n(\mu_i,\lambda_i) \\
& \text{subj.} \; \lambda \ge 0, \;\; (\mu,\omega) \in \mathcal{D}
\end{aligned}
$$

where $q_i^n(\mu_i,\lambda_i) = \min_{\theta_i\in\Re^{d_i}} \left[(\mu_i - c_i)'\theta_i + \lambda_i \mathbf{1}'e^{\theta_i} + \frac{\gamma_n}{2}\|\theta_i - \theta_i^n\|^2 \right].$

- Can efficiently evaluate q_i^n (Newton's method, global quadratic convergence) and its 1st and 2nd order derivatives
- $\mathcal{D}$ does not depend on θ^n

Dual Proximal Point Algorithm

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

$$(\mathrm{P}_n) \qquad \min_{\theta,\epsilon} \; -\sum_{i\in\mathcal{I}} c_i'\theta_i + \eta \sum_{k\in\mathcal{K}} \epsilon_k + \frac{\gamma_n}{2}\|\theta - \theta^n\|^2$$

$$\text{subj.} \; \sum_{i\in\mathcal{I}} a_{i,k}(s)'\theta_i + b_k(s) \le \epsilon_k, \; \forall s \in \mathcal{S}_k, \; k \in \mathcal{K}$$

$$\mathbf{1}'e^{\theta_i} \le 1, \;\; \forall i \in \mathcal{I}, \quad \epsilon_k \ge 0, \;\; \forall k \in \mathcal{K}$$

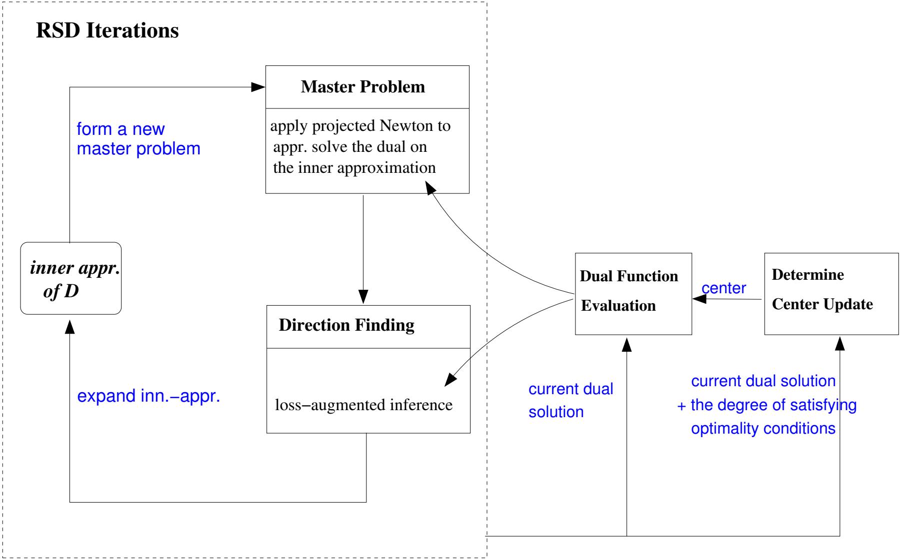
$$(\mathrm{D}_n) \qquad \max_{\mu,\omega,\lambda} \; \omega - \sum_{i\in\mathcal{I}} \lambda_i + \sum_{i\in\mathcal{I}} q_i^n(\mu_i,\lambda_i)$$

$$\text{subj.} \; \lambda \ge 0, \;\; (\mu,\omega) \in \mathcal{D}$$

$$\text{where} \quad q_i^n(\mu_i,\lambda_i) = \min_{\theta_i \in \Re^{d_i}} \left[(\mu_i - c_i)'\theta_i + \lambda_i \mathbf{1}'e^{\theta_i} + \frac{\gamma_n}{2}\|\theta_i - \theta_i^n\|^2 \right].$$

- Can efficiently evaluate q_i^n (Newton's method, global quadratic convergence) and its 1st and 2nd order derivatives
- $\mathcal{D}$ does not depend on θ^n

Algorithm Chart from Dual Viewpoint



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 $\mathcal{L}$:

$$\sum_{i \in \mathcal{I}} \Big(\sum_{j=1}^{m} \underbrace{\sum_{k \in \mathcal{K}_j, s \in \mathcal{S}_k} \beta_k(s)\, a_{i,k}(s)'}_{\stackrel{def}{=} \mu^j} \Big) \theta_i + \Big(\sum_{j=1}^{m} \underbrace{\sum_{k \in \mathcal{K}_j, s \in \mathcal{S}_k} \beta_k(s)\, b_k(s)}_{\stackrel{def}{=} \omega^j} \Big)$$

- 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 $\mathcal{K}_j$, $m = |\mathcal{K}|$

Algorithm Variants with Same Idea

Further remarks on reparametrization:

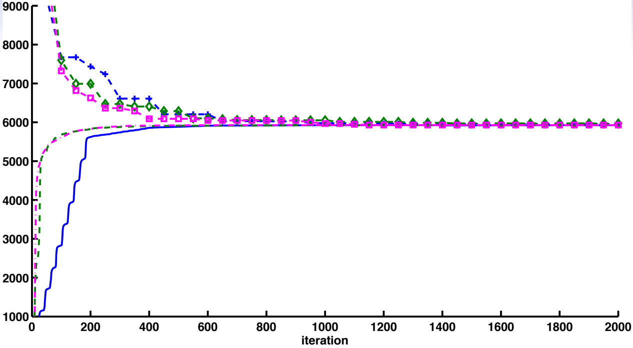
- Arbitrary and varying working sets can also be handled in the first reparametrization (μ, ω): use the inner approximation view
- For different margin violation penalties: e.g., quadratic or loss-rescaled slacks (Tsochantaridis et al. '05); $\mathcal{D}$ may be unbounded, but the same algorithm can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

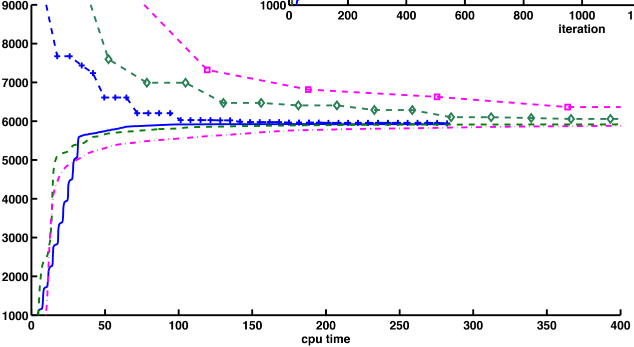
Note:

- 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
$\dim(\theta) = 180$, $|\mathcal{I}| = 21$





Batch size $\times m$:

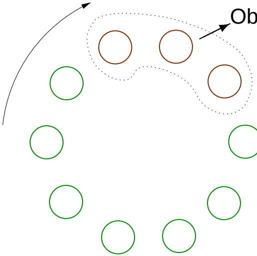
- B 100×10
- G 500×2
- M 1000×1

Outline

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

Test loss:
DT: 82.2 ± 13.5 per seq.
ML: 101.6 ± 14.0 per seq.

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

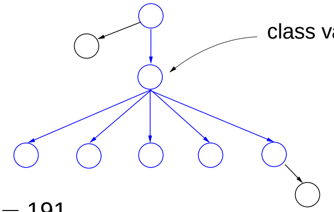
Comparison of the dimensionalities of dual variables:

- $|\mathcal{I}| = 21$, $\dim(\theta) = 180$, $\dim(\beta) = 1000 \times 10^{50}$
- 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

$$\max_s \ln p(s \mid o\,;\theta_{ML}) - \ln p(s^* \mid o\,;\theta_{ML}) \le \delta, \quad \delta \ge 0: \text{ selection level}$$

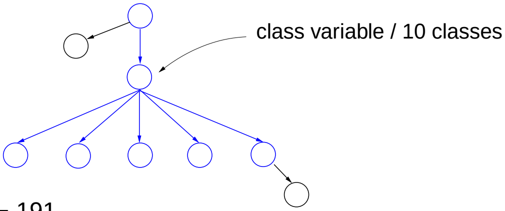
- Reason: avoid difficult instances

  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

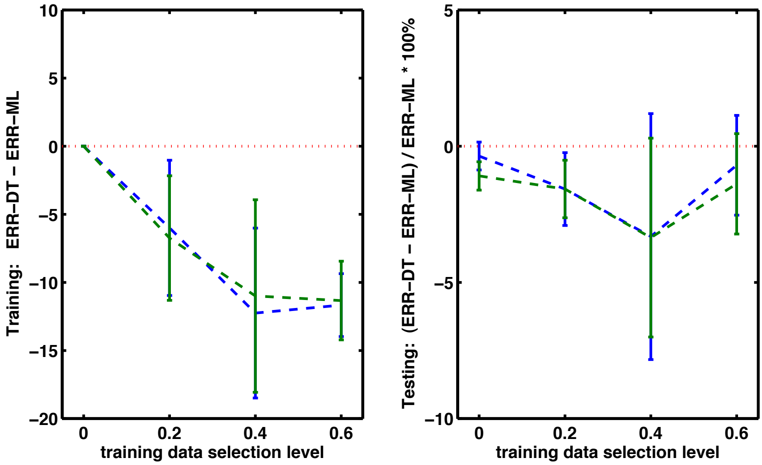
- Select instances (s^*, o) such that

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

- Reason: avoid difficult instances

  alternative to further selection: set loss differently for each instance in training

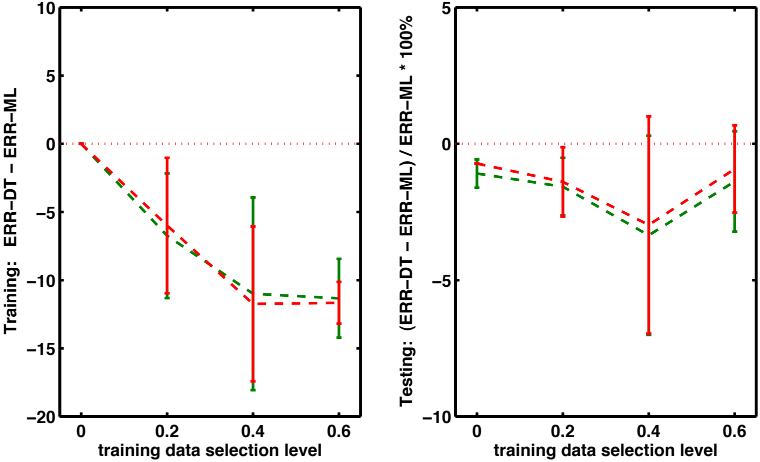
II: Yeast Dataset – a Case Study on Modeling



B c_i = ML weighted by $\gamma_i > 0$
$\|\theta^* - \theta_{ML}\|$: 0.18 ± 0.10

G c_i = uniform
$\|\theta^* - \theta_{ML}\|$: 4.18 ± 0.03

II: Yeast Dataset – a Case Study on Modeling



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

R $c_i = 0$, use $\|\theta\|^2$ as regularizer
$\|\theta^* - \theta_{ML}\| : 4.82 \pm 0.04$

Outline

Overview and Problem Formulation

Algorithm

Preliminary Experiments

Summary

Discussion

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