Cutting Plane Training of Structural SVM

Seth Neel

University of Pennsylvania

sethneel@wharton.upenn.edu

September 28, 2017

Overview

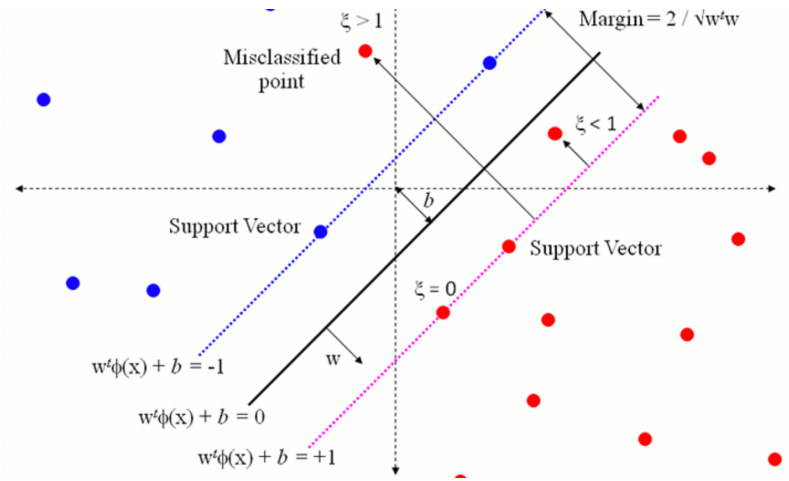
- Structural SVMs have shown promise for building accurate structural prediction models in natural language processing and other domains
- Current training algorithms (while polynomial time) are expensive on large datasets; superlinear
- Paper proposes a new cutting plane training method that runs in linear time
- Number of iterations until convergence is independent of the number of training examples
- Empirical work shows that it is much faster than conventional cutting plane methods

SVM

Binary classification (primal):

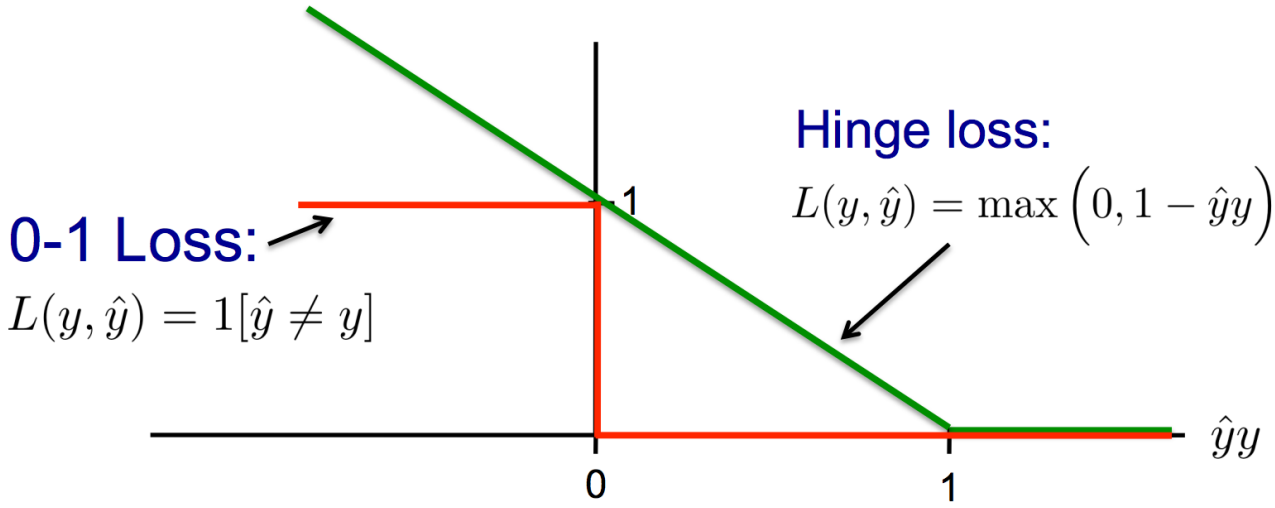
$$\min_{w,\epsilon_i \geq 0} \frac{1}{2}||w||^2 + \frac{C}{n}\sum_i \epsilon_i$$

$$s.t.\ \forall i \in \{1, \ldots n\} : y_i(w^T x_i) \geq 1 - \epsilon_i$$



Hinge Loss

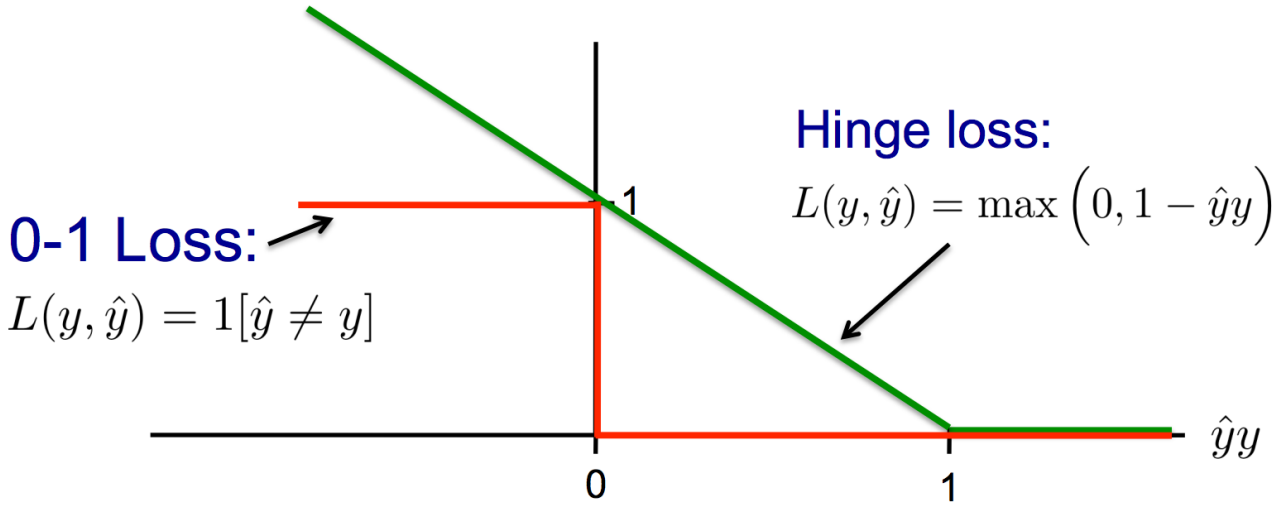
Hinge loss is a convex upper bound for the $0-1$ loss.



The SVM can be represented as unconstrained minimization of the regularized hinge loss.

Hinge Loss

Hinge loss is a convex upper bound for the $0-1$ loss.



The SVM can be represented as unconstrained minimization of the regularized hinge loss.

$$\min ||w||^2 + C \sum_j \max\left(0, 1 - \left(w \cdot x_j + b\right) y_j\right)$$

Multiclass SVM

- Recall: multiclass margin is defined as the score difference between the highest scoring label and the second highest

$$\min \frac{1}{2} \sum_{i=1}^{k} ||w_k||^2 + C \sum_i \epsilon_i$$

$$s.t. \ \forall i, (x_i, y_i) \in D, k \neq y_i : w_{y_i}^t x_i - w_k^t x_i \geq 1 - \epsilon_i, \epsilon_i \geq 0$$

Structured Prediction

- Learn a function $f : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{Y}$ is a space of multivariate and structured outputs
- for a given x predict $f_w(x) = \text{argmax}_{y \in \mathcal{Y}} w^T \Psi(x, y)$
- intuitively Ψ measures the compatibility of y and x

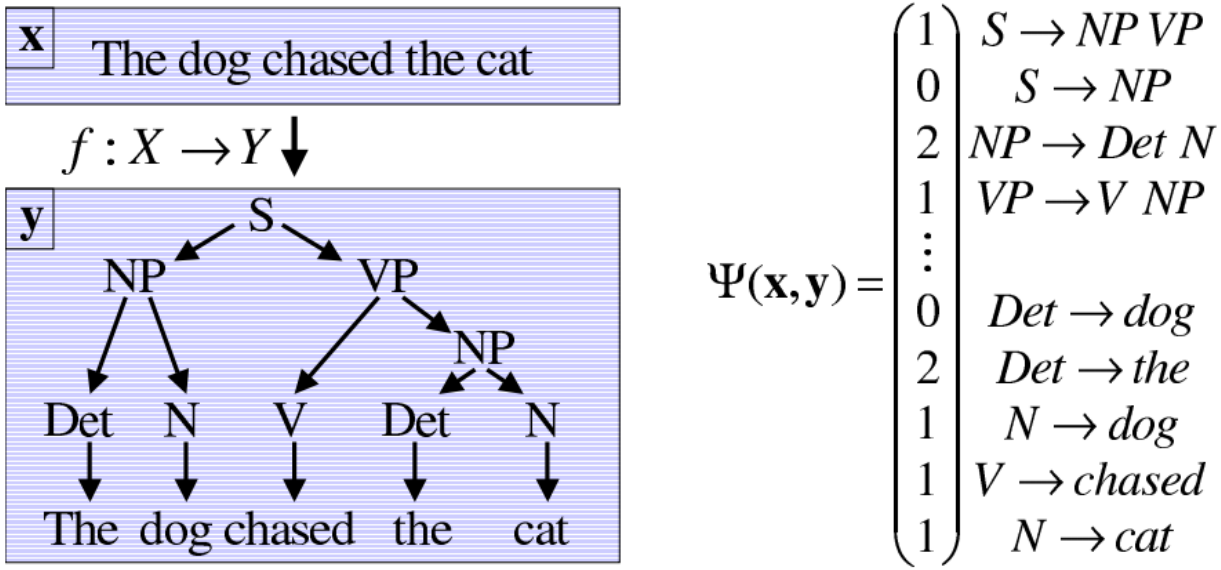


Figure 1: Illustration of natural language parsing model.

Structured SVM

- Assume that $f_w(x, y) = w^t \Psi(x, y)$
- We generalize the SVM optimization problem to train w
- The true loss function is $\Delta(y, h_w(x)) = \mathbf{1}\{y \neq h_w(x)\}$
- hinge loss provides a convex upper bound to the true loss function:

Structured SVM

- Assume that $f_w(x, y) = w^t \Psi(x, y)$
- We generalize the SVM optimization problem to train w
- The true loss function is $\Delta(y, h_w(x)) = \mathbf{1}\{y \neq h_w(x)\}$
- hinge loss provides a convex upper bound to the true loss function:
  - margin-rescaling:
    $\Delta_{MR}(h_w(x), y) = \max_{y' \in \mathcal{Y}} \{\Delta(y, y') - w^t \Psi(x, y) + w^t \Psi(x, y')\}$
  - slack-rescaling:
    $\Delta_{SR}(h_w(x), y) = \max_{y' \in \mathcal{Y}} \{\Delta(y, y') * (1 - w^t \Psi(x, y) + w^t \Psi(x, y'))\}$
  - Conceptually similar; we focus on MR.

n-slack structural svm optimization (MR)

$$\min_{w,\epsilon \geq 0} \frac{1}{2}||w||^2 + \frac{C}{n}\sum_i \epsilon_i$$

$$s.t.\ \forall y' \in \mathcal{Y}, i \in [n] : w^t[\Psi(x_i, y_i) - \Psi(x_i, y')] \geq \Delta(y_i, y') - \epsilon_i$$

Note that if Ψ is the $|\mathcal{Y}| * p$ dimensional embedding we recover the multiclass formulation, where p is the dimension of x_i.

Past Approaches

- Note the n-slack optimization problem has $|\mathcal{Y}| * n$ constraints, not obviously efficiently solvable
- In fact it is: a greedily constructed cutting plane model requires only $O(n/\epsilon^2)$ constraints.
- other methods: exponentiated gradient methods, Taskar reformulation as a QP, stochastic gradient methods.

Cutting Plane Algorithm 1

Algorithm 1 for training Structural SVMs (with margin-rescaling) via the n-Slack Formulation (OP2).

Input: $S = ((x_1, y_1), \ldots, (x_n, y_n))$, C, ε
$\mathcal{W}_i \leftarrow \emptyset$, $\xi_i \leftarrow 0$ for all $i = 1, \ldots, n$
repeat
  for i=1,...,n **do**
    $\hat{y} \leftarrow \operatorname{argmax}_{\hat{y} \in \mathcal{Y}} \{\Delta(y_i, \hat{y}) - \boldsymbol{w}^T [\Psi(x_i, y_i) - \Psi(x_i, \hat{y})]\}$
    if $\Delta(y_i, \hat{y}) - \boldsymbol{w}^T [\Psi(x_i, y_i) - \Psi(x_i, \hat{y})] > \xi_i + \varepsilon$ **then**
      $\mathcal{W}_i \leftarrow \mathcal{W}_i \cup \{\hat{y}\}$
      $(\boldsymbol{w}, \boldsymbol{\xi}) \leftarrow \operatorname{argmin}_{\boldsymbol{w}, \boldsymbol{\xi} \geq 0} \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} + \frac{C}{n} \sum_{i=1}^{n} \xi_i$

$$\text{s.t. } \forall \bar{y}_1 \in \mathcal{W}_1 : \boldsymbol{w}^T [\Psi(x_1, y_1) - \Psi(x_1, \bar{y}_1)] \geq \Delta(y_1, \bar{y}_1) - \xi_1$$
$$\vdots$$
$$\forall \bar{y}_n \in \mathcal{W}_n : \boldsymbol{w}^T [\Psi(x_n, y_n) - \Psi(x_n, \bar{y}_n)] \geq \Delta(y_n, \bar{y}_n) - \xi_n$$

    end if
  end for
until no $\mathcal{W}_i$ has changed during iteration
return(w,ξ)

Cutting Plane Algorithm 1

$$5: \qquad \hat{y} \leftarrow \text{argmax}_{\hat{y} \in \mathcal{Y}} \{\Delta(y_i, \hat{y}) - \boldsymbol{w}^T [\Psi(x_i, y_i) - \Psi(x_i, \hat{y})]\}$$

- This algorithm is efficient assuming existence of a separation oracle calculating the most violated constraint [line 5]
- note for natural choices of Δ this is the assumption that we can efficiently solve the inference problem

Contribution (this paper): Faster Training!

Using a simple reformulation of the optimization problem:

1. constant iteration complexity $\implies$ linear runtime in n
2. several orders of magnitude improvement in runtime (worst case analysis)
3. empirical study shows speedup in practice

1-slack formulation

Theorem

The n-slack formulation is equivalent to the optimization:

$$\min_{w,\epsilon \geq 0} \frac{1}{2}||w||^2 + C\mathcal{E}$$

$$s.t.\ \forall (y_1', y_2', \ldots y_n') \in \mathcal{Y}^n :$$

$$\frac{1}{n} w^t \sum_{i=1}^{n} [\Psi(x_i, y_i) - \Psi(x_i, y_i')] \geq \frac{1}{n} \sum_i \Delta(y_i, y_i') - \mathcal{E}$$

Note now we only have 1 slack variable shared among all constraints, but we actually have exponentially more constraints

What is the point?

- New reformulation has constraint that binds on linear combination of data points $\implies$ support vectors are linear combinations of data points
- This flexibility gives a much **sparser** set of non-zero dual variables, which translates into a **smaller cutting plane model** (constant)
- High-level: exponentially more constraints, but only a constant number of them *matter*

1-slack: proof

It suffices to show that for any fixed w, the smallest feasible $\mathcal{E}$ in the 1-slack formulation and smallest feasible $\frac{1}{n}\sum_i \epsilon_i$ in the n-slack formulation are equal.

1-slack: proof

It suffices to show that for any fixed w, the smallest feasible $\mathcal{E}$ in the 1-slack formulation and smallest feasible $\frac{1}{n}\sum_i \epsilon_i$ in the n-slack formulation are equal.

Proof.

In n-slack, for a fixed w, for each i, each constraint gives $\epsilon_i \geq \Delta(y', y_i) - w^t[\Psi(x_i, y_i) - \Psi(x_i, y')]$, which shows that the smallest feasible is

$$\epsilon_i = \max_{y'}\{\Delta(y', y_i) - w^t[\Psi(x_i, y_i) - \Psi(x_i, y')]\}$$

In 1-slack the smallest feasible $\mathcal{E}$ is similarly at:

$$\max_{y_1', y_2', \ldots, y_n'}\{\frac{1}{n}\sum_i(\Delta(y_i, y_i') - w^t[\Psi(x_i, y_i) - \Psi(x_i, y')])\}$$

1-slack: proof ctd

Proof.

But this function separates over each y_i' and so it is equal to: $\frac{1}{n}\sum_i \max_{y_i'}(\Delta(y_i, y_i') - w^t[\Psi(x_i, y_i) - \Psi(x_i, y')])$ which as defined in the 1-slack formulation is simply $\frac{1}{n}\epsilon_i$. □

The Main Algorithm

- Similar to the previous cutting plane algorithm, only adds 1 constraint in each iteration.
- What is different is that only a constant number of constraints are sufficient to find an ϵ-approximate solution.

Algorithm 3 for training Structural SVMs (with margin-rescaling) via the 1-Slack Formulation (OP4).

Input: $S = ((x_1, y_1), \ldots, (x_n, y_n))$, C, ε
$\mathcal{W} \leftarrow \emptyset$
repeat
  $(\boldsymbol{w}, \xi) \leftarrow \operatorname{argmin}_{\boldsymbol{w}, \xi \geq 0} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\xi$
    s.t. $\forall(\bar{y}_1, \ldots, \bar{y}_n) \in \mathcal{W}$: $\frac{1}{n}\boldsymbol{w}^T \sum_{i=1}^{n} [\Psi(x_i, y_i) - \Psi(x_i, \bar{y}_i)] \geq \frac{1}{n} \sum_{i=1}^{n} \Delta(y_i, \bar{y}_i) - \xi$
  for i=1,...,n **do**
    $\hat{y}_i \leftarrow \operatorname{argmax}_{\hat{y} \in \mathcal{Y}} \{\Delta(y_i, \hat{y}) + \boldsymbol{w}^T \Psi(x_i, \hat{y})\}$
  end for
  $\mathcal{W} \leftarrow \mathcal{W} \cup \{(\hat{y}_1, \ldots, \hat{y}_n)\}$
until $\frac{1}{n}\sum_{i=1}^{n} \Delta(y_i, \hat{y}_i) - \frac{1}{n}\boldsymbol{w}^T \sum_{i=1}^{n} [\Psi(x_i, y_i) - \Psi(x_i, \hat{y}_i)] \leq \xi + \varepsilon$
return$(\boldsymbol{w}, \xi)$

Correctness

Obvious; when (if) the algorithm terminates:

1. The objective is optimized over a strictly smaller set of constraints, and hence has a smaller value
2. If the algorithm terminates the solution is approximately feasible

Iteration Complexity

- Let $\Delta = \max_{i,y'} \Delta(y_i, y')$. Let $R = \max_{i,y'} ||\Psi(x_i, y_i) - \Psi(x_i, y')||$.
- Then the iteration complexity is $\frac{16R^2C}{\epsilon} + \log_2(\frac{\Delta}{4R^2C})$

Iteration Complexity: Outline

$$\min_{w,\epsilon \geq 0} \frac{1}{2}||w||^2 + C\mathcal{E}$$

$$s.t.\ \forall (y_1', y_2', \ldots y_n') \in \mathcal{Y}^n : \frac{1}{n} w^t \sum_{i=1}^{n} [\Psi(x_i, y_i) - \Psi(x_i, y_i')] \geq \frac{1}{n} \sum_i \Delta(y_i, y_i') - \mathcal{E}$$

- The pair $(w, \mathcal{E}) = (0, \Delta)$ is feasible in the above
- Thus the optimal value is $\leq C\Delta$.
- This means that the dual program has optimal value $\leq C\Delta$
- We show each iteration increases the dual objective by some constant amount, forcing constant iteration complexity.

Classic SVM Dual

Binary classification (primal):

$$\min_{w,\epsilon_i \geq 0} \frac{1}{2}||w||^2 + \frac{C}{n}\sum_i \epsilon_i$$

$$s.t.\ \forall i \in \{1, \ldots n\} : y_i(w^T x_i) \geq 1 - \epsilon_i$$

Classic SVM Dual

Binary classification (primal):

$$\min_{w,\epsilon_i \geq 0} \frac{1}{2}||w||^2 + \frac{C}{n}\sum_i \epsilon_i$$

$$s.t.\ \forall i \in \{1, \ldots n\} : y_i(w^T x_i) \geq 1 - \epsilon_i$$

Binary classification (dual):

$$\min_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{n}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^t x_j$$

$$s.t.\ \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\forall i \in \{1, \ldots n\} : 0 \leq \alpha_i \leq \frac{C}{n}$$

The Dual Program

The dual program is easy to compute by forming the Lagrangian and taking derivatives. First we define:

$$\Delta(\bar{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}\Delta(y_i,\bar{y_i})$$

$$H_{MR}(\bar{\mathbf{y}},\bar{\mathbf{y}}') = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\Big[\Psi(x_i,y_i)^T\Psi(x_j,y_j) - \Psi(x_i,y_i)^T\Psi(x_j,\bar{y}'_j) - \Psi(x_i,\bar{y_i})^T\Psi(x_j,y_j) + \Psi(x_i,\bar{y_i})^T\Psi(x_j,\bar{y}'_j)\Big]$$

Optimization Problem 6 (1-Slack Structural SVM with Margin-Rescaling (dual))

$$\max_{\boldsymbol{\alpha}\geq 0} \; D(\boldsymbol{\alpha}) = \sum_{\bar{\mathbf{y}}\in\mathcal{Y}^n}\Delta(\bar{\mathbf{y}})\alpha_{\bar{\mathbf{y}}} - \frac{1}{2}\sum_{\bar{\mathbf{y}}\in\mathcal{Y}^n}\sum_{\bar{\mathbf{y}}'\in\mathcal{Y}^n}\alpha_{\bar{\mathbf{y}}}\alpha_{\bar{\mathbf{y}}'}H_{MR}(\bar{\mathbf{y}},\bar{\mathbf{y}}')$$

$$s.t. \quad \sum_{\bar{\mathbf{y}}\in\mathcal{Y}^n}\alpha_{\bar{\mathbf{y}}} = C$$

Line Search Lemma Proof

Note that the dual objective is a QP of the form $\Theta(\alpha) = h^t\alpha - \frac{1}{2}\alpha^t H\alpha$ where $h = \{\Delta(\bar{y})\}_{\bar{y}\in\mathcal{Y}^n}$, $H_{a,b} = \alpha_a\alpha_b H_{MR}(y_a, y_b)$

Lemma 2. *For any unconstrained quadratic program*

$$\max_{\boldsymbol{\alpha}\in\Re^n}\{\Theta(\boldsymbol{\alpha})\} < \infty, \quad \Theta(\boldsymbol{\alpha}) = \boldsymbol{h}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T H\boldsymbol{\alpha} \tag{32}$$

with positive semi-definite H, and derivative $\partial\Theta(\boldsymbol{\alpha}) = \boldsymbol{h} - H\boldsymbol{\alpha}$, a line search starting at $\boldsymbol{\alpha}$ along an ascent direction $\boldsymbol{\eta}$ with maximum step-size $C > 0$ improves the objective by at least

$$\max_{0\le\beta\le C}\{\Theta(\boldsymbol{\alpha}+\beta\boldsymbol{\eta})\} - \Theta(\boldsymbol{\alpha}) \ge \frac{1}{2}\min\left\{C, \frac{\nabla\Theta(\boldsymbol{\alpha})^T\boldsymbol{\eta}}{\boldsymbol{\eta}^T H\boldsymbol{\eta}}\right\}\nabla\Theta(\boldsymbol{\alpha})^T\boldsymbol{\eta}. \tag{33}$$

Proof of Line Search Lemma.

Proof.

Direct calculation yields:

$$\Theta(\alpha + \beta\eta) - \Theta(\alpha) = \beta[\nabla\Theta(\alpha)^T\eta - \frac{1}{2}\beta\eta^T H\eta]$$

Setting the derivative with respect to β equal to 0 gives $\beta = \frac{\nabla\theta(\alpha)^T\eta}{\eta^T H\eta}$.
Substituting this value of β gives

$$\max_{\beta} \Theta(\beta\eta + \alpha) - \Theta(\alpha) = \frac{1}{2}\frac{(\nabla\Theta(\alpha)^T\eta)^2}{\eta^T H\eta}$$

This is the maximum value, unless the value of β at the optimum is $> C$, in which case the optimum occurs at $\beta = C$, by concavity in β.
Substituting in $\beta = C$ gives the desired result. □

Iteration Complexity (Sketch)

Algorithm 3 for training Structural SVMs (with margin-rescaling) via the 1-Slack Formulation (OP4).

1: Input: $S = ((x_1, y_1), \ldots, (x_n, y_n))$, C, ε
2: $\mathcal{W} \leftarrow \emptyset$
3: **repeat**
4: $(\boldsymbol{w}, \xi) \leftarrow \operatorname{argmin}_{\boldsymbol{w}, \xi \geq 0} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\xi$
$$\text{s.t. } \forall (\bar{y}_1, \ldots, \bar{y}_n) \in \mathcal{W} : \frac{1}{n}\boldsymbol{w}^T \sum_{i=1}^{n} [\Psi(x_i, y_i) - \Psi(x_i, \bar{y}_i)] \geq \frac{1}{n} \sum_{i=1}^{n} \Delta(y_i, \bar{y}_i) - \xi$$
5: **for** i=1,...,n **do**
6: $\hat{y}_i \leftarrow \operatorname{argmax}_{\hat{y} \in \mathcal{Y}} \{\Delta(y_i, \hat{y}) + \boldsymbol{w}^T \Psi(x_i, \hat{y})\}$
7: **end for**
8: $\mathcal{W} \leftarrow \mathcal{W} \cup \{(\hat{y}_1, \ldots, \hat{y}_n)\}$
9: **until** $\frac{1}{n}\sum_{i=1}^{n} \Delta(y_i, \hat{y}_i) - \frac{1}{n}\boldsymbol{w}^T \sum_{i=1}^{n} [\Psi(x_i, y_i) - \Psi(x_i, \hat{y}_i)] \leq \xi + \varepsilon$
10: return$(\boldsymbol{w}, \xi)$

In step 8 we add a constraint.

This adds a variable to the dual program: $\alpha_{\hat{y}}$. Note can set $\alpha_{\hat{y}} = 0$, and so it only increases the objective value.

Optimization Problem 6 (1-SLACK STRUCTURAL SVM WITH MARGIN-RESCALING (DUAL))

$$\max_{\boldsymbol{\alpha} \geq 0} \; D(\boldsymbol{\alpha}) = \sum_{\bar{\mathbf{y}} \in \mathcal{Y}^n} \Delta(\bar{\mathbf{y}}) \alpha_{\bar{\mathbf{y}}} - \frac{1}{2} \sum_{\bar{\mathbf{y}} \in \mathcal{Y}^n} \sum_{\bar{\mathbf{y}}' \in \mathcal{Y}^n} \alpha_{\bar{\mathbf{y}}} \alpha_{\bar{\mathbf{y}}'} H_{MR}(\bar{\mathbf{y}}, \bar{\mathbf{y}}')$$

$$s.t. \quad \sum_{\bar{\mathbf{y}} \in \mathcal{Y}^n} \alpha_{\bar{\mathbf{y}}} = C$$

- Adding a new constraint $\hat{y}$ to the model adds a new parameter to the dual problem; increases its objective value

- Adding a new constraint $\hat{y}$ to the model adds a new parameter to the dual problem; increases its objective value
- We pick an $\eta : \eta_{\hat{y}} = 1, \eta'_y = -\frac{1}{C}\alpha_{y'}$ such that $\alpha + \beta\eta$ is always dual-feasible (since $\eta^T \mathbf{1} = 0$)

- Adding a new constraint $\hat{y}$ to the model adds a new parameter to the dual problem; increases its objective value
- We pick an $\eta : \eta_{\hat{y}} = 1, \eta'_y = -\frac{1}{C}\alpha_{y'}$ such that $\alpha + \beta\eta$ is always dual-feasible (since $\eta^T \mathbf{1} = 0$)
- The increase in objective value is lower bounded by the increase in a line search along η (since we are only searching a feasible region of the dual)

- Adding a new constraint $\hat{y}$ to the model adds a new parameter to the dual problem; increases its objective value
- We pick an $\eta : \eta_{\hat{y}} = 1, \eta'_y = -\frac{1}{C}\alpha_{y'}$ such that $\alpha + \beta\eta$ is always dual-feasible (since $\eta^T \mathbf{1} = 0$)
- The increase in objective value is lower bounded by the increase in a line search along η (since we are only searching a feasible region of the dual)
- Line search lemma (with some algebra) lets us lower bound this increase as a function of ϵ, C, R.

Time Complexity

Summary

- $O(n)$ calls to the separation oracle
- $O(n)$ computation time per iteration
- Constant number of iterations
- Each QP is of constant size, and is hence solved in constant time.
- For non-linear kernel it is $O(n^2)$

Experiments

- Applied to binary classification, multi-class classification, linear chain HMMs, and CFG Grammar learning
- Two Questions:
  1. Does the 1-slack algorithm achieve significant speedups in practice?
  2. Is the $w*$ value as good a solution?

Speed: 1-slack vs. n-slack

			CPU-Time		# Sep. Oracle		# Support Vec.	
	n	N	1-slack	n-slack	1-slack	n-slack	1-slack	n-slack
MultiC	522,911	378	1.05	1180.56	4,183,288	10,981,131	98	334,524
HMM	35,531	18,573,781	0.90	177.00	1,314,647	4,476,906	139	83,126
CFG	9,780	154,655	2.90	8.52	224,940	479,220	70	12,890

Slower speedup with slower separation oracle - still significant gains

Speed: 1-slack vs. SVM light on binary classification

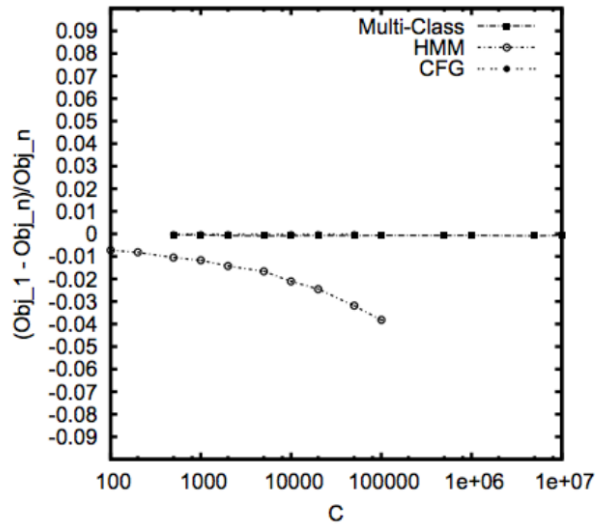
Table 2 Training CPU time (in seconds) for five binary classification problems comparing the 1-slack algorithm (without caching) with SVM-light. n is the number of training examples, N is the number of features, and s is the fraction of non-zero elements of the feature vectors. The SVM-light results are quoted from (Joachims, 2006), the 1-slack results are re-run with the latest version of SVM-struct using the same experiment setup as in (Joachims, 2006).

				CPU-Time		# Support Vec.	
	n	N	s	1-slack	SVM-light	1-slack	SVM-light
Reuters CCAT	804,414	47,236	0.16%	58.0	20,075.5	8	230388
Reuters C11	804,414	47,236	0.16%	71.3	5,187.4	6	60748
ArXiv Astro-ph	62,369	99,757	0.08%	4.4	80.1	9	11318
Covertype 1	522,911	54	22.22%	53.4	25,514.3	27	279092
KDD04 Physics	150,000	78	38.42%	9.2	1,040.2	13	99123

Faster even on binary classification than state-of-the-art SVM training algorithms

Accuracy vs. *n*-slack

Cutting-Plane Training of Structural SVMs 31



Task	Measure	1-slack	*n*-slack
MultiC	Accuracy	72.33	72.35
HMM	Token Accuracy	96.71	96.69
CFG	Bracket F_1	70.22	70.09

- Very similar for all tasks except HMM where *n*-slack outperforms
- This is because the duality gap $C\epsilon$ is significant part of the objective value since the data was almost linearly separable.
- Generalization performance for HMM was still comparable.

References

Joachims, Finley, Yu (2009)

Cutting Plane Training of Structural SVMs