Introduction to Graphical Models
lecture 9 - Structural Support Vector Machines

Ulf Brefeld
TU Berlin

– support vector machines
– multi-class classification

Recall:

- relation between
  - $P(\mathbf{y}|\mathbf{x})$
  - model $f(\mathbf{x}, \mathbf{y})$
  - log-Viterbi algorithm

- intuition
  - $f(\mathbf{x}, \mathbf{y})$ = how good does $\mathbf{y}$ fits to $\mathbf{x}$
  - log-Viterbi: find top-scoring $\mathbf{y}$

Towards Structured Support Vector Machines

- Add confidence to decision
- Incorporate arbitrary (structured) loss functions
- Impact of ordering resolved by quadratic programming

Confidence Term

- Perceptron:

$$\begin{aligned}
\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle &> \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, A, A]) \rangle \\
\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle &> \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, A, N]) \rangle \\
\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle &> \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, N, A]) \rangle \\
\vdots \quad &> \quad \vdots
\end{aligned}$$

- Now, add a confidence $\bar{\gamma}$:

$$\begin{aligned}
\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, A, A]) \rangle &\geq \bar{\gamma} \\
\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, A, N]) \rangle &\geq \bar{\gamma} \\
\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, N, A]) \rangle &\geq \bar{\gamma} \\
\vdots \qquad\qquad\qquad\qquad\qquad\qquad &\quad \vdots
\end{aligned}$$

Optimization Problem

$$\max_{\bar{\gamma},\mathbf{w}} \quad \frac{\bar{\gamma}}{||\mathbf{w}||}$$

$$\text{s.t.} \quad \forall_{i=1}^{n}, \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} : \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq \bar{\gamma}$$

- We call
  - $\mathbf{w}$ the weight vector
  - $\bar{\gamma}$ the functional margin
  - $\gamma = \frac{\bar{\gamma}}{||\mathbf{w}||}$ the geometrical margin

- Problem: $\bar{\gamma}$ and $\mathbf{w}$ interdepend!
  - Remedy: fix one, solve for the other
  - Common approach: $\bar{\gamma} = 1$.

Structural Hard-margin SVM

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}||\mathbf{w}||^2 \\ \text{s.t.} \quad & \forall_{i=1}^{n}, \forall_{\bar{\mathbf{y}} \neq \mathbf{y}_i} : \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \geq 1 \end{aligned}$$

- Converges only when data is linear separable
- Remedy: allow for pointwise relaxations of the margin constraint – introduce slack variables ξ_i for input examples

Structural Soft-margin SVM

$$\min_{\mathbf{w}} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

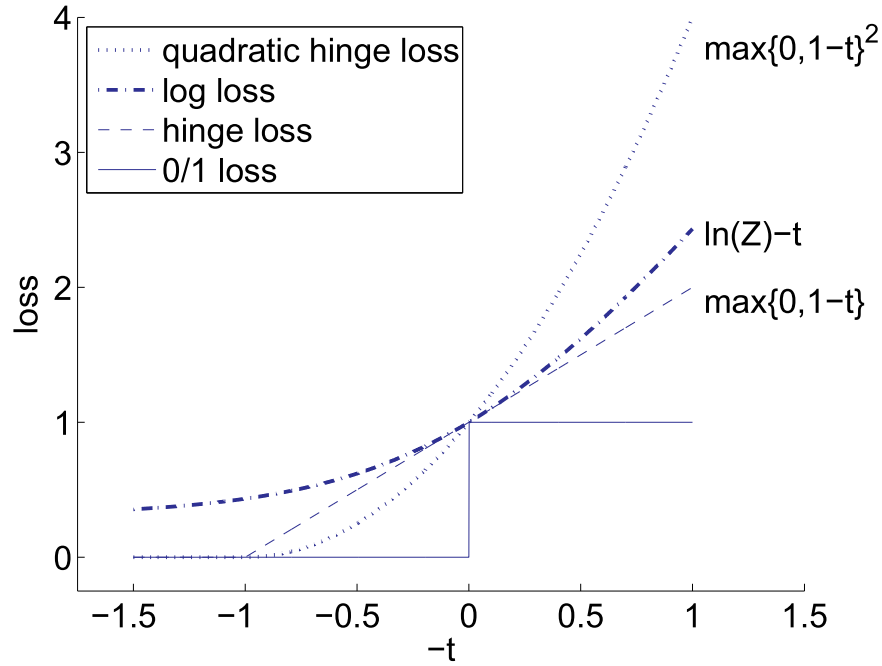
$$\text{s.t.} \quad \forall_{i=1}^{n}, \forall_{\bar{\mathbf{y}}\neq\mathbf{y}_i} : \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i)\rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{\mathbf{y}})\rangle \geq 1 - \xi_i$$

$$\forall_{i=1}^{n} : \xi_i \geq 0$$

- Maximize margin between true $\mathbf{y}_i$ and best runner-up $\bar{\mathbf{y}}$
  - Sum of slacks upper bounds 0/1 loss
  - Trade-off parameter $C > 0$

- Alternative formulation:
  - slack $\xi_{i\bar{\mathbf{y}}}$ are bound to constraint $\langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i)\rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{\mathbf{y}})\rangle$
  - computationally demanding

Hinge-loss



- SVM implicitely implements a hinge loss (solve for slacks)
- Hinge loss can be rescaled to incorporate arbitrary loss functions
  - Let $\Delta(\mathbf{y}_i, \hat{\mathbf{y}})$ denote a structural loss.
  - $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \Re_0^+$.
  - $\Delta(\mathbf{y}_i, \mathbf{y}_i) = 0$

Examplary Loss Functions

- 0/1 loss: $\Delta(\mathbf{y}, \bar{\mathbf{y}}) = [[\mathbf{y} == \bar{\mathbf{y}}]]$

- Hamming loss for sequences

$$\Delta(\mathbf{y}, \bar{\mathbf{y}}) = T - \sum_{t=1}^{T} [[y_t == \bar{y}_t]]$$

  – Property: decomposes across the cliques!

Risk Minimization

- We want to minimize the theoretical risk (the generalization error)

$$R(f) = \int_{\mathcal{X}\times\mathcal{Y}} \Delta\big(\mathbf{y}, \operatorname*{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}, \bar{\mathbf{y}})\big) dP(\mathbf{x}, \mathbf{y})$$

- In general, we don't know $P(\mathbf{x}, \mathbf{y})$
  – Remedy: Use training sample instead!

- Minimize the empirical risk

$$\hat{R}(f) = \sum_{i=1}^{n} \Delta\big(\mathbf{y}_i, \operatorname*{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}_i, \bar{\mathbf{y}})\big)$$

Idea

- SVMs minimize the (regularized) empirical risk:

$$\hat{R}(f) = \sum_{i=1}^{n} \Delta\big(\mathbf{y}_i, \operatorname*{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}_i, \bar{\mathbf{y}})\big)$$

- Sum of slacks $\sum_i \xi_i$ upper bounds empirical risk
- Slack variable ξ_i denotes the error for input $\mathbf{x}_i$
  – Now: Find maximal error wrt Δ

Margin-rescaling

- Taskar et al. (2004)
- Rescale the (functional) margin by actual loss

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \forall_{i=1}^{n}, \forall_{\bar{\mathbf{y}}\neq\mathbf{y}_i} : \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i)\rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{\mathbf{y}})\rangle \geq \Delta(\mathbf{y}_i, \bar{\mathbf{y}}) - \xi_i \\ & \forall_{i=1}^{n} : \xi_i \geq 0 \end{aligned}$$

  – Implicit hinge loss upper bounds Δ

  – Most strongly violated constraint:

$$\operatorname*{argmax}_{\bar{\mathbf{y}}\neq\mathbf{y}_i} \Big(\underbrace{\Delta(\mathbf{y}_i, \bar{\mathbf{y}}) - \big(\langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}})\rangle\big)}_{\xi_i} \Big)$$

Slack-rescaling

- Tsochantaridis et al. (2005)
- Rescale slack variables by actual loss

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \forall_{i=1}^{n}, \forall_{\bar{\mathbf{y}}\neq\mathbf{y}_i} : \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i)\rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{\mathbf{y}})\rangle \geq 1 - \frac{\xi_i}{\Delta(\mathbf{y}_i, \bar{\mathbf{y}})} \\ & \forall_{i=1}^{n} : \xi_i \geq 0 \end{aligned}$$

– Implicit hinge loss upper bounds Δ

– Most strongly violated constraint:

$$\operatorname*{argmax}_{\bar{\mathbf{y}}\neq\mathbf{y}_i} \Big(\underbrace{1 - \Delta(\mathbf{y}_i, \bar{\mathbf{y}}) \times \big(\langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}})\rangle\big)}_{\xi_i} \Big)$$

Implications

- Loss Δ decomposes across the cliques of the graph
  – Margin-rescaling is easily integrated into inference
  – Slack-rescaling difficult

- Loss not decomposable
  – Both difficult!

- In practice, slack-rescaling often better than margin-rescaling
  – rarely applicable (needs good approximation or enumerable sets)

Example

- Margin rescaling for sequences / Viterbi algorithm
- Remainder: 0/1 loss for simplicity

Towards Dual SVMs

- Intergrate constraints into objective
  – Apply Lagrange's Theorem
  – Lagrange multipliers $\alpha_i(\bar{\mathbf{y}})$ and μ_i

- Build Lagrangian:

$$\begin{aligned} L =& \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \\ & - \sum_{i=1}^{n}\sum_{\bar{\mathbf{y}}\neq\mathbf{y}_i} \alpha_i(\bar{\mathbf{y}})\langle \mathbf{w}, \Phi(\mathbf{x}_i,\mathbf{y}_i)\rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i,\bar{\mathbf{y}})\rangle - 1 + \xi_i \\ & - \sum_{i=1}^{n}\beta_i\xi_i \end{aligned}$$

- Minimum of Lagrangian is a saddle-point
  – max wrt α, μ, min wrt $\mathbf{w}, \xi$

Partial Derivatives: ξ_i

- Compute partial derivatives wrt ξ:

$$\frac{\partial L}{\partial \xi_i} = C - \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) - \beta_i \quad \stackrel{!}{=} \quad 0 \tag{1}$$

- Using the non-negativity of α and β yields

$$\forall_{i=1}^{n}: \quad 0 \leq \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) \leq C$$

Partial Derivatives: $\mathbf{w}$

- Compute partial derivatives wrt $\mathbf{w}$:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) \Big(\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \Big) \stackrel{!}{=} 0.$$

- We obtain:

$$\mathbf{w} = \sum_{i=1}^{n} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) \Big(\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \Big)$$

- Recall: dual perceptron!
  - Definition of $\mathbf{w}$ is equivalent
  - α's act like counters

Putting all together...

- Dual SVM for structured output variables:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) - \frac{1}{2} \sum_{i,j=1}^{n} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \sum_{\bar{\bar{\mathbf{y}}} \neq \mathbf{y}_j} \Big(\alpha_i(\bar{\mathbf{y}})\, \alpha_j(\bar{\bar{\mathbf{y}}}) \\ \langle \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \Phi(\mathbf{x}_j, \mathbf{y}_j) - \Phi(\mathbf{x}_j, \bar{\bar{\mathbf{y}}}) \rangle \Big)$$

  – subject to the constraints

$$\forall_{i=1}^{n} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) \leq C$$

$$\forall_{i=1}^{n}\, \forall i: \quad \alpha_i(\bar{\mathbf{y}}) \geq 0.$$

Optimization

- Dual SVM is a QP (quadratic program)
- Sketch of algorithm:

1 Loop
2 Loop $i = 1, \ldots, n$
3 Compute most strongly violated constraint $\bar{\mathbf{y}}$
4 If constraint causes more slack than before (ξ_i)
5 Add constraint $\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}})$ to working set,
6 Initialize $\alpha_i(\bar{\mathbf{y}})$
8 Solve QP
9 Until convergence

(For details see Tsochantaridis et al. JMLR,2005)

Comments

- Working set contains active constraints
  – Only a few of the constraints will be active
  – remove constraints if $\alpha_i(\bar{\mathbf{y}}) = 0$ after optimization

- Convergence in polynomial time
  – if computation of argmax takes at most polynomial time
  – proof: see Tsochantaridis et al. (2005)

- Implementation on the web:
  – SVM^{struct} by Thorsten Joachims (extension of SVM^{light})
  – `http://svmlight.joachims.org/svm_struct.html`

Named Entity Recognition

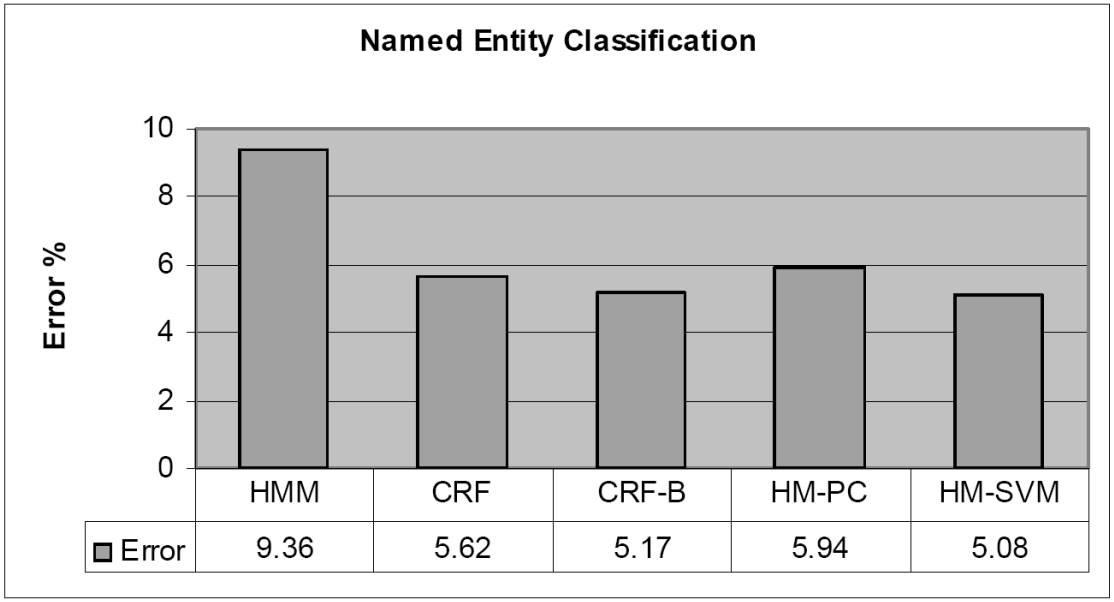


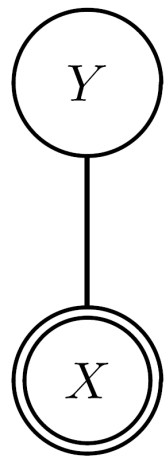
Figure 1. Test error of NER task over a window of size 3 using 5-fold cross validation.

Example: Como (O) contrapartida (O) Deutsche (C-B) Telekom (C-I) vender (O) al (O) consorcio (O) francs (O) su (O) participacion (O) del (O) por (O) ciento (O) en (O) el (O) empresa (O) mixta (O) britnica (O) MetroHoldings (C-B).

(see Altun et al. (2003))

CRF: Special Case

- Sequential approach =
  – a chain of multi-class classification problems
  – linked via label-label transitions



Multi-class Classification

- Let $\psi(\mathbf{x})$ the feature vector of observation $\mathbf{x}$
- Let $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ be the set of classes
- Then, the joint feature map is given by

$$\Phi(\mathbf{x}, y) = ([[y = \sigma_1]]\psi(\mathbf{x})^\mathsf{T}, \ldots, [[y = \sigma_k]]\psi(\mathbf{x})^\mathsf{T})^\mathsf{T}.$$

  – dimension of Φ is $dim(\Phi) = |\Sigma| dim(\psi)$
  – see also Crammer & Singer (2001), Weston & Watkins (1998)

Outlook

- What happens if only observations but no labels are observed?
  – i.e., training set consists of n observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$

- Probabilistic planning...