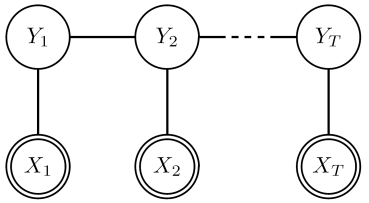
Introduction to Graphical Models
lecture 8 - Perceptron Algorithm for CRFs

Ulf Brefeld
TU Berlin

– supplements and extensions

Recall: Conditional MRF



$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{i=1}^{T} \psi^{obs}(X_i, Y_i) \prod_{i=2}^{T} \psi^{trans}(Y_{i-1}, Y_i)$$
$$\propto \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$$

Recall: Generalized Linear Models

- $\mathbf{x} =$ Bob jagt den Hund
- We want

$$[N, V, A, N] = \operatorname*{argmax}_{\bar{\mathbf{y}}} \langle \mathbf{w}, \Phi(\mathbf{x}, \bar{\mathbf{y}}) \rangle$$

- Equivalent representation:

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

Recall: Primal/Dual Perceptron

- Primal perceptron:
  – Decision function: $f(\mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$
  – Update rule: $\mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \hat{\mathbf{y}})$

- Dual perceptron:
  – Use relation: $\mathbf{w} = \sum_{i=1}^{n} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) \left(\Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \right)$
  – Decision function:

$$f(\mathbf{x}', \mathbf{y}') = \sum_{i=1}^{n} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) \left(\langle \Phi(\mathbf{x}_i, \mathbf{y}_i), \Phi(\mathbf{x}', \mathbf{y}') \rangle - \langle \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \Phi(\mathbf{x}', \mathbf{y}') \rangle \right)$$

  – Update rule: $\alpha_i(\hat{\mathbf{y}}) \leftarrow \alpha_i(\hat{\mathbf{y}}) + 1$

Primal/Dual Algorithm

1 loop $r = 1, \ldots, r_{max}$
2 loop $i = 1, \ldots, n$
3 Compute prediction $\hat{\mathbf{y}}$
4 If $\mathbf{y}_i \neq \hat{\mathbf{y}}$
5 Update $\mathbf{w}$ (primal) or $\alpha_i(\hat{\mathbf{y}})$ (dual)
6 End (if)
7 End loop (i)
8 End loop (r)

How to Compute the Prediction in Step 3?

- What is the relation between...
  - Viterbi algorithm
  - max-product algorithm
  - max-sum algorithm
  - scoring function $f(\mathbf{x}, \mathbf{y})$
  - ???
- How can we compute $\operatorname{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}, \bar{\mathbf{y}})$?
- Answer: use log-Viterbi = max-sum algorithm!

Recall: Homework

- Dual perceptron:
- Decompose $f(\mathbf{x}, \mathbf{y}) = f_1(\mathbf{x}, \mathbf{y}) + f_2(\mathbf{x}, \mathbf{y})$ with

$$f_1(\mathbf{x}, \mathbf{y}) = \sum_{\sigma,\tau} a(\sigma, \tau) \sum_s [[y^{s-1} = \sigma \wedge y^s = \tau]]$$

$$a(\sigma, \tau) = \sum_{i, \bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) \sum_t [[\bar{y}^{t-1} = \sigma \wedge \bar{y}^t = \tau]]$$

- and

$$f_2(\mathbf{x}, \mathbf{y}) = \sum_{s,\sigma} [[y^s = \sigma]] \sum_{i,t} b(i, t, \sigma) K_x(x^s, x_i^t),$$

$$b(i, t, \sigma) = \sum_{\mathbf{y} \neq \mathbf{y}_i} [[y^t = \sigma]] \alpha_i(\mathbf{y})$$

Recall: Viterbi Algorithm

- Computes: $\operatorname{argmax}_{y_1,\ldots,y_T} P(y_1, \ldots, y_T | x_1, \ldots, x_T)$
- Viterbi = max-product algorithm
  - define $\delta_{t+1}(\sigma) = \max_{y_1,\ldots,y_t} P(y_1, \ldots, y_{t+1} = \sigma, x_1, \ldots, x_{t+1})$
  - best score along a single path that ends in state σ at time $t + 1$
- log-Viterbi = max-sum algorithm
  - define $\delta_{t+1}(\sigma) = \max_{y_1,\ldots,y_t} \log P(y_1, \ldots, y_{t+1} = \sigma, x_1, \ldots, x_{t+1})$
  - apply $\delta_{t+1}(\sigma_i)$ recursively

Log-Viterbi Algorithm

- initialize $\delta_1(\sigma) = \log P(y_1 = \sigma) + \log P(x_1|y_1 = \sigma)$
- initialize $\psi_1(\sigma) = 0$
- loop $\sigma \in \Sigma$ and $t = 2, \ldots, T$:
  - $\delta_t(\sigma) = \left[\max_\tau \delta_{t-1}(\tau) + \log P(y_t = \sigma|y_{t-1} = \tau)\right] + \log P(x_t|y_t = \sigma)$
  - $\psi_t(\sigma) = \left[\operatorname{argmax}_\tau \delta_{t-1}(\tau) + \log P(y_t = \sigma|y_{t-1} = \tau)\right] + \log P(x_t|y_t = \sigma)$
- termination: $y_T^* = \operatorname{argmax}_\sigma \delta_T(\sigma)$
- loop $t = T, \ldots, 2$
  - $y_{t-1}^* = \psi_t(y_t^*)$

Scoring Function $f(\mathbf{x}, \mathbf{y})$

- Capture $\log P(y_1 = \sigma)$ implictely by adding constant label y_0.
- Observation probabilities:

$$\log P(x_t|y_t = \sigma) \propto \underbrace{\sum_j \sum_{s=1}^{T_j} \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} [[y^t = \sigma]] \alpha_i(\bar{\mathbf{y}}) k(x_t, x_{j,s})}_{b(\sigma, x_t)}$$

- Transition probabilities:

$$\log P(y_t = \tau|y_{t-1} = \sigma) \propto \underbrace{\sum_{i, \bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) \sum_t [[\bar{y}^{t-1} = \sigma \wedge \bar{y}^t = \tau]]}_{a(\sigma, \tau)}$$

It holds...

Theorem

Given n input-output pairs of sequences of length T_i for $1 \leq i \leq n$, let Σ denote the output alphabet with $|\Sigma| < \infty$. Let f be defined as

$$f(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n \sum_i \alpha_i(\bar{\mathbf{y}}) \left(\langle \Phi(\mathbf{x}_i, \mathbf{y}_i), \Phi(\mathbf{x}, \mathbf{y}) \rangle - \langle \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \Phi(\mathbf{x}, \mathbf{y}) \rangle\right),$$

where $\Phi(\mathbf{x}, \mathbf{y})$ denotes the joint feature map. Then for all $\alpha_i(\bar{\mathbf{y}}) \geq 0$ and any observation sequence $\mathbf{x}$ of length T,

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\bar{\mathbf{y}} \in \Sigma^T} f(\mathbf{x}, \bar{\mathbf{y}})$$

can be computed with a Viterbi algorithm in time $\mathcal{O}(T|\Sigma|^2)$.

Proof:

- The model f has the form

$$\begin{aligned} f(\mathbf{x}, \mathbf{y}) &= \sum_{i=1}^n \sum_i \alpha_i(\bar{\mathbf{y}}) \left(\langle \Phi(\mathbf{x}_i, \mathbf{y}_i), \Phi(\mathbf{x}, \mathbf{y}) \rangle - \langle \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \Phi(\mathbf{x}, \mathbf{y}) \rangle\right) \\ &= \sum_{i=1}^n \sum_i \alpha_i(\bar{\mathbf{y}}) \Bigg(\sum_{s,t} \left([[y_{i,s} = y_t]] - [[\bar{y}_s = y_t]]\right) k(x_{i,s}, x_t) \\ &\quad + \sum_{s,t} [[y_{i,s-1} = y_{t-1} \wedge y_{i,s} = y_t]] - [[\bar{y}_{s-1} = y_{t-1} \wedge \bar{y}_s = y_t]] \Bigg). \end{aligned}$$

- Make the dependency on labels $\sigma, \tau \in \Sigma$ explicit by summing over all transitions and observation states

$$\begin{aligned} f(\mathbf{x}, \mathbf{y}) = \sum_{\sigma, \tau \in \Sigma} \sum_{i, \bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) \Big(\sum_{s,t} \left([[y_{i,s} = \sigma]] - [[\bar{y}_s = \sigma]]\right) [[y_t = \tau]] k(x_{i,s}, x_t) \\ + \sum_{s,t} \Big([[y_{i,s-1} = \sigma \wedge y_{i,s} = \tau]] - [[\bar{y}_{s-1} = \sigma \wedge \bar{y}_s = \tau]] \Big) \\ \times [[y_{t-1} = \sigma \wedge y_t = \tau]] \Big). \end{aligned}$$

Proof Contd.

The transition scores from label σ to label τ are now given by

$$a(\sigma,\tau) = \sum_{i=1}^{n}\sum_{i}\alpha_i(\bar{\mathbf{y}})\Big(\sum_{t=1}^{T_i}[[y_{i,t-1}=\sigma \wedge y_{i,t}=\tau]] - [[\bar{y}_{t-1}=\sigma \wedge \bar{y}_t=\tau]]\Big)$$

and observation scores for label $y_s=\sigma$ and observation x_s by

$$b(\sigma,x) = \sum_{i=1}^{n}\sum_{t=1}^{T_i}\sum_{i}\alpha_i(\bar{\mathbf{y}})\left([[y_{i,t}=\sigma]] - [[\bar{y}_t=\sigma]]\right)k(x_{i,t},x).$$

...

The hypothesis $f(\mathbf{x},\mathbf{y})$ can be rewritten in terms of transition scores $a(\sigma,\tau)$ and observation scores $b(\sigma,x)$

$$f(\mathbf{x},\mathbf{y}) = \underbrace{\sum_{\sigma,\tau\in\Sigma} a(\sigma,\tau)\sum_{s=1}^{T}[[y_{s-1}=\sigma \wedge y_s=\tau]]}_{=:f_a(\mathbf{x},\mathbf{y})} + \underbrace{\sum_{s=1}^{T}\sum_{\sigma\in\Sigma}[[y_s=\sigma]]b(\sigma,x_s)}_{=:f_b(\mathbf{x},\mathbf{y})}.$$

where f_a weights the occurences of neighboring labels in $\mathbf{y}$ by corresponding scores of the model and f_b determines how well observations x_s fit to their labels y_s given the model. To decode the top scoring sequence we define

$$\delta_t(\sigma) = \max_{y_1,\ldots,y_{t-1}} f(\mathbf{x},y_1,\ldots,y_{t-1},y_t=\sigma), \tag{1}$$

that is, $\delta_t(\sigma)$ denotes the top scoring partial sequence up to position $t-1$ where $y_t=\sigma$.

Mathematical Induction: The Base Case

We first show by induction that

$$\delta_{t+1}(\sigma) = \max_{\tau\in\Sigma}\left[\delta_t(\tau)+a(\tau,\sigma)\right] + b(\sigma,x_{t+1}) \tag{2}$$

holds. The initialization is simply given by

$$\begin{aligned}\delta_0(\sigma) &= 0, \quad \forall\sigma\in\Sigma\\ \delta_1(\sigma) &= \max_{\tau\in\Sigma}\left[\delta_t(\tau)+a(\tau,\sigma)\right] + b(\sigma,x_{t+1})\\ &= a(\epsilon,\sigma) + b(\sigma,x_1).\end{aligned}$$

The Inductive Step

The recursion step is given for $2\le t\le T$ by

$$\begin{aligned}\delta_t(\sigma) &= \max_{y_1,\ldots,y_{t-1}} f(\mathbf{x},y_1,\ldots,y_{t-1},y_t=\sigma)\\ &= \max_{y_1,\ldots,y_{t-1}}\sum_{\tau,\bar\tau\in\mathcal{Y}} a(\tau,\bar\tau)\sum_{s=2}^{t-1}[[y_{s-1}=\tau\wedge y_s=\bar\tau]]\\ &\qquad + \sum_{\tau\in\Sigma} a(\tau,\sigma)[[y_{t-1}=\tau\wedge y_t=\sigma]]\\ &\qquad + \sum_{s=1}^{t-1}\sum_{\tau\in\Sigma}[[y_s=\tau]]b(\tau,x_s) + [[y_t=\sigma]]b(\sigma,x_t)\\ &= \max_{\sigma^\star}\max_{y_1,\ldots,y_{t-2}}\sum_{\tau,\bar\tau\in\mathcal{Y}} a(\tau,\bar\tau)\sum_{s=2}^{t-2}[[y_{s-1}=\tau\wedge y_s=\bar\tau]]\\ &\qquad + \sum_{\tau\in\Sigma} a(\tau,\sigma^\star)[[y_{t-2}=\tau\wedge y_{t-1}=\sigma^\star]]\\ &\qquad + a(\sigma^\star,\sigma)[[y^{t-1}=\sigma^\star\wedge y^t=\sigma]]\\ &\qquad + \sum_{s=1}^{t-2}\sum_{\tau\in\Sigma}[[y_s=\tau]]b(\tau,x_s) + b(\sigma^\star,x_{t-1}) + b(\sigma,x_t)\end{aligned}$$

The Inductive Step Contd.

$$= \max_{\sigma^\star} \left[\max_{y_1,\ldots,y_{t-2}} f(\mathbf{x}, y_1, \ldots, y_{t-2}, y_{t-1} = \sigma^\star) + a(\sigma^\star, \sigma) \right] + b(\sigma, x_t)$$
$$= \max_{\sigma^\star} \left[\delta_{t-1}(\sigma^\star) + a(\sigma^\star, \sigma) \right] + b(\sigma, x_t).$$

Thus, the top scoring sequence has the score

$$\max f(\mathbf{x}, \mathbf{y}) = \max_{\sigma \in \Sigma} \delta_T(\sigma).$$

We only sketch the extension to the argument of the maximum since it is analoguous to the regular Viterbi algorithm. We introduce path variables $\varphi_t(\sigma)$ that are initialized by $\varphi_1(\sigma) = \epsilon$ for all $\sigma \in \Sigma$.

Computing the Argmax

The sequence $\varphi_t(\sigma)$ is then defined recursively for $2 \leq t \leq T$ by

$$\varphi_t(\sigma) = \operatorname*{argmax}_{\sigma^\star \in \Sigma} \left[\delta_{t-1}(\sigma^\star) + a(\sigma^\star, \sigma) \right].$$

Once the $\delta_t(\sigma)$ of Theorem 1 are fixed, the optimal label sequence can be found by backtracking

$$y_T^\star = \operatorname*{argmax}_{\sigma \in \Sigma} \delta_T(\sigma)$$
$$y_t^\star = \varphi_{t+1}(y_{t+1}^\star) \quad \text{for} \quad t = T-1, \ldots, 1.$$

Conclusion

Given the transition matrix $[\mathbf{A}]_{\sigma,\tau} = a(\sigma, \tau)$ and the observation matrix $[\mathbf{B}_\mathbf{x}]_{\sigma,t} = b(\sigma, x_t)$ for input $\mathbf{x}$, the computation of δ and φ for a fixed t and $\sigma \in \Sigma$ involves visiting $|\Sigma|$ predecessors; thus, for a sequence of length T the time needed is in $\mathcal{O}(T|\Sigma|^2)$. This concludes the proof. □

Visualization

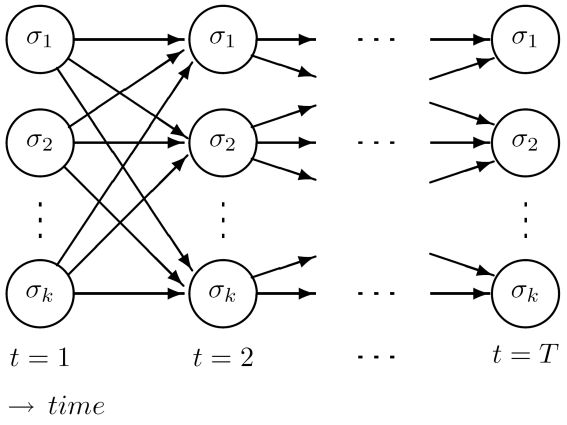


Figure: Visualization of a trellis over the alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$.

Summary

- Equivalence: Dual perceptron $f(\mathbf{x}, \mathbf{y})$ and Viterbi algorithm
  – similar proof for primal perceptron
- pos:
  – easy to implement
  – efficient training process
- neg:
  – depends on ordering
  – no confidences
  – only 0/1 loss

From Perceptrons to SVMs

- 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 $\bar{\gamma}$:

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

Optimization Problem

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

- 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

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

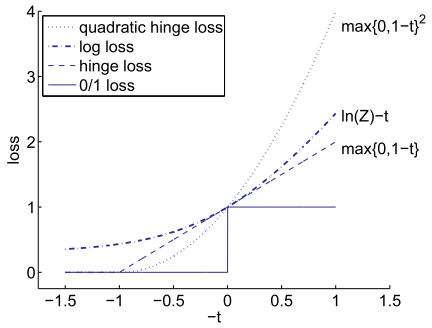
- Converges only when data is linear separable
- Remedy: allow for pointwise relaxations of the margin constraint
  – introduce slack variables ξ_i for input examples

Structural Soft-margin SVM

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

- Sum of slacks upper bounds 0/1 loss
- Now: maximize margin between true $\mathbf{y}_i$ and best runner-up $\bar{\mathbf{y}}$
- 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!

Margin-rescaling

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

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

  – Implicit hinge loss upper bounds Δ
  – Most strongly violated constraint:

$$\operatorname*{argmax}_{\bar{\mathbf{y}}} \left(\Delta(\mathbf{y}_i, \bar{\mathbf{y}}) - \left(\langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \right) \right)$$

Slack-rescaling

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

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

  – Implicit hinge loss upper bounds Δ
  – Most strongly violated constraint:

$$\operatorname*{argmax}_{\bar{\mathbf{y}}} \left(1 - \Delta(\mathbf{y}_i, \bar{\mathbf{y}}) \times \left(\langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle \right) \right)$$

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)