COMS E6998-3, Spring 2012, problem set 1

Due date: 5pm, 24th February 2011

Question 1 (40 points)

In conditional random fields (CRFs), a key idea was to define a "global" feature vector

$$\underline{\Phi}(\underline{x}, \underline{s})$$

that maps an input sequence $\underline{x}$ paired with an output sequence $\underline{s}$ to a d-dimensional feature vector. In *bigram* models for sequence modeling, we define

$$\underline{\Phi}(\underline{x}, \underline{s}) = \sum_{j=1}^{n} \underline{\phi}(\underline{x}, j, s_{j-1}, s_j)$$

where $\underline{\phi}$ is a *local* feature-vector definition. We described a decoding algorithm for CRFs that take this form, and also an algorithm for parameter estimation.

In this question we'll consider a *trigram* model for conditional random fields, where

$$\underline{\Phi}(\underline{x}, \underline{s}) = \sum_{j=1}^{n} \underline{\phi}(\underline{x}, j, s_{j-2}, s_{j-1}, s_j)$$

where $\underline{\phi}$ is again a *local* feature-vector definition, which can now consider sequences of three tags (s_{j-2}, s_{j-1}, s_j).

Question (15 points): Give pseudo-code for a dynamic-programming algorithm for decoding for the trigram model.

Question (15 points): In class we described a parameter estimation method for bigram CRF models. Describe an analogous parameter estimation method for trigram models. Your method should use analogous terms to the $q^i_j(a, b)$ terms employed for bigram models (see the notes on CRFs). (Note that we do not require you to derive a variant of the forward-backward algorithm for calculation of these q terms; it is sufficient to define them correctly.)

Question (10 points): Give pseudo code for a perceptron-based algorithm for parameter estimation for the trigram model.

Question 2 (20 points)

Consider a sequence modeling task where we have the following training data:

- 100 examples where $x_1 = \texttt{a}, x_2 = \texttt{b}$, and $s_1 = \texttt{A}, s_2 = \texttt{B}$.
- 100 examples where $x_1 = \texttt{a}, x_2 = \texttt{c}$, and $s_1 = \texttt{A}, s_2 = \texttt{C}$.
- 800 examples where $x_1 = \texttt{c}, x_2 = \texttt{d}$, and $s_1 = \texttt{B}, s_2 = \texttt{D}$.

Question (10 points): We first train a bigram HMM for the sequence modeling problem. List all non-zero parameters for the HMM. What is the output from the HMM on the three input sequences `a b`, `a c`, and `c d`?

Question (10 points): Describe features for a bigram CRF for the sequence modeling problem, which models the data correctly. (By "correctly" we mean that the output from the model on the input sequences `a b`, `a c`, and `c d` is `A B`, `A C`, and `B D` respectively.)

Question 3 (30 points)

This question again concerns log-linear models. To recap the details from the lecture notes: we have a set $\mathcal{X}$ of possible inputs, and a finite set $\mathcal{Y}$ of possible labels. We have a feature vector $f(x, y) \in \mathbb{R}^d$ for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We have a parameter vector $v \in \mathbb{R}^d$. The log-linear model defines the conditional probability as

$$p(y|x; v) = \frac{\exp\left(v \cdot f(x, y)\right)}{\sum_{y \in \mathcal{Y}} \exp\left(v \cdot f(x, y)\right)}$$

To estimate the parameters of the model, we have a set of training examples $(x^{(i)}, y^{(i)})$ for $i \in \{1 \ldots n\}$. The regularized log-likelihood function is

$$L(v) = \sum_{i=1}^{n} \log p(y_i|x_i; v) - \frac{\lambda}{2} \sum_j |v_j|$$

where $\lambda > 0$ is a parameter. (Here we use $|v_j|$ to refer to the absolute value of v_j.)

The optimal parameters are

$$v^* = \arg\max_{v \in \mathbb{R}^d} L(v)$$

Note that this is different from the method described in lecture, where we used a regularizer of the form

$$\frac{\lambda}{2} \sum_j v_j^2$$

Note also that the term $\sum_j |v_j|$ is not differentiable. You should be able to complete this question *without attempting to take derivatives of* $L(v)$.

Question (10 points) Assume that for feature f_1, we have $f_1(x_i, y) = 0$ for all $i \in \{1 \ldots n\}, y \in \mathcal{Y}$. What is the value of v_1^*? Make sure to justify your answer (5 out of 10 points will be given for the justification).

Question (10 points) Assume that for feature f_2, we have $f_2(x_i, y) = 10$ for all $i \in \{1 \ldots n\}, y \in \mathcal{Y}$. What is the value of v_2^*? Make sure to justify your answer (5 out of 10 points will be given for the justification).

Question (10 points) Assume that for feature f_3, we have $f_3(x_i, y) = i$ for all $i \in \{1 \ldots n\}, y \in \mathcal{Y}$. What is the value of v_3^*? Make sure to justify your answer (5 out of 10 points will be given for the justification).

Question 4 (30 points)

Figure 1 shows the Pegasos algorithm, as introduced in lecture. We have augmented the algorithm to include the following variables:

- Inputs: training set $\{(\underline{x}_i, y_i)\}_{i=1}^n, T$
- Initialization: $\underline{\theta}_1 = \underline{0}$, $M_1 = 0$
- For $t = 1 \ldots T$:
  1. Pick an example $i \in \{1 \ldots n\}$ uniformly at random
  2. If $y_i(\underline{\theta}_t \cdot \underline{x}_i) < 1$

$$\begin{aligned} \underline{\theta}_{t+1} &= \left(1 - \frac{1}{t}\right)\underline{\theta}_t + \frac{1}{\lambda t} y_i \underline{x}_i \\ M_{t+1} &= M_t + 1 \\ \underline{x}'_{M_{t+1}} &= \underline{x}_i \\ y'_{M_{t+1}} &= y_i \end{aligned}$$

     else if $y_i(\underline{\theta}_t \cdot \underline{x}_i) \geq 1$

$$\underline{\theta}_{t+1} = \left(1 - \frac{1}{t}\right)\underline{\theta}_t$$

$$M_{t+1} = M_t$$

- Return $\underline{\theta}_{T+1}$

Figure 1: The Pegasos algorithm, as described in lecture. The algorithm has the following additional variables: M_t for $t \geq 1$ is the number of cases where $y_i(\underline{\theta}_t \cdot \underline{x}_i) < 1$ up to iteration t of the algorithm. x'_i, y'_i for $i = 1 \ldots M_t$ is the sequence of examples up to iteration t where $y_i(\underline{\theta}_t \cdot \underline{x}_i) < 1$.

- M_t for $t \geq 1$ is the number of cases up to iteration t where the condition $y_i(\underline{\theta}_t \cdot \underline{x}_i) < 1$ is reached.
- x'_i, y'_i for $i = 1 \ldots M_t$ is a record of the examples where the condition $y_i(\underline{\theta}_t \cdot \underline{x}_i) < 1$ was reached.

The question is as follows: Prove by induction that for any $t \geq 2$,

$$\underline{\theta}_t = \frac{1}{\lambda(t-1)} \sum_{i=1}^{M_t} y'_i \underline{x}'_i$$

Question 5 (30 points)

Figure 2 gives the structured perceptron, as described in lecture. We gave the following definition of separability, which is a generalization of the definition for the perceptron for binary classification:

Definition: The training set $\{(\underline{x}^i, \underline{s}^i)\}_{i=1}^n$ is separable with margin $\delta > 0$, if there exists some parameter vector $\underline{w}$ such that:

1. $||\underline{w}||^2 = 1$

- Input: labeled examples, $\{(\underline{x}^i, \underline{s}^i)\}_{i=1}^n$.
- Initialization: $\underline{w} = \underline{0}$
- For $t = 1 \dots T$, for $i = 1 \dots n$:
  - Use the Viterbi algorithm to calculate

$$\underline{s}^* = \arg\max_{\underline{s} \in \mathcal{Y}} \quad \underline{w} \cdot \underline{\Phi}(\underline{x}^i, \underline{s}) = \arg\max_{\underline{s} \in \mathcal{Y}} \quad \sum_{j=1}^{m} \underline{w} \cdot \underline{\phi}(\underline{x}, j, s_{j-1}, s_j)$$

  - Updates:

$$\begin{aligned} \underline{w} &= \underline{w} + \underline{\Phi}(\underline{x}^i, \underline{s}^i) - \underline{\Phi}(\underline{x}^i, \underline{s}^*) \\ &= \underline{w} + \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s^i_{j-1}, s^i_j) - \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s^*_{j-1}, s^*_j) \end{aligned}$$

- Return $\underline{w}$

Figure 2: The structured perceptron algorithm.

2. For all $i = 1 \dots n$, for all $s_1 \dots s_m$ such that $s_j \neq s^i_j$ for some j,

$$\underline{w} \cdot \underline{\Phi}(\underline{x}^i, \underline{s}^i) - \underline{w} \cdot \underline{\Phi}(\underline{x}^i, \underline{s}) \geq \delta$$

We then gave the following theorem:

Theorem: Assume that the training set is separable with margin δ, and that for all i, for all state sequences $\underline{s} = s_1 \dots s_m$,

$$||\underline{\Phi}(\underline{x}^i, \underline{s}^i) - \underline{\Phi}(\underline{x}^i, \underline{s})||^2 \leq R^2$$

Then the structured perceptron (see algorithm in figure 2) makes at most

$$\frac{R^2}{\delta^2}$$

mistakes. (A "mistake" occurs each time $\underline{s}^* \neq \underline{s}^i$ in the algorithm.)

Question: Give a proof of the theorem. (The proof should be similar to the proof for the perceptron for binary classification; see the note on the class webpage.)