CPTS 580 Structured Prediction: Algorithms and Applications, Spring 2017
Homework #2
Due Date: Thu, Mar 9

NOTE 1: Please use a word processing software (e.g., Microsoft word or Latex) to write your answers and submit a printed copy to me. The rationale is that it is sometimes hard to read and understand the hand-written answers. Thanks for your understanding.

NOTE 2: Please ensure that all the graphs are appropriately labeled (x-axis, y-axis, and each curve). The caption or heading of each graph should be informative and self-contained.

1. (**100 points**) Please implement the online structured perceptron training algorithm for sequence labeling problems and experiment with two sequence labeling datasets: handwriting recognition, and text-to-speech mapping.

   In a sequence labeling problem, the structured input $x = (x_1, x_2, \cdots, x_T)$ is a sequence of input tokens, where each input token x_i is represented as a m-dimensional feature vector; and the structured output $y = (y_1, y_2, \cdots, y_T)$ is a sequence of output labels, where each output label y_i comes from a label set $\{1, 2, \cdots, k\}$. You were provided with a set of training examples $\mathcal{D} = \{(x, y^*\}$, where y^* is the correct structured output for the structured input x.

   You need to learn a scoring function $S(x, y) = w \cdot \phi(x, y)$, where $\phi(x, y) \in \Re^d$ is a joint feature representation over a structured input x and candidate structured output $y \in Y(x)$ and $w \in \Re^d$ corresponds to the weights (or parameters) of the cost function. Essentially, you need to learn the weights $w \in \Re^d$ from the given training data.

Algorithm 1 Best-First Beam Search Inference

Input: x = structured input, $\Phi(x, y)$ = joint feature function, (I, S) = search space definition, b = beam width, w = weights of features
Output: $\hat{y}$, the best scoring output

Initialization: $BEAM \leftarrow y_{start} = I(x)$
repeat
  $y_{select} \leftarrow \arg max_{y \in BEAM}\, w \cdot \Phi(x, y)$ // Selection
  $CANDIDATES \leftarrow BEAM \cup S(y_{select} \setminus \{y_{select}\}$ // Expansion
  $BEAM \leftarrow$ Top-b scoring outputs in $CANDIDATES$ // Pruning
until $BEAM$ contains a complete structured output or terminal
return best scoring complete structured output $\hat{y}$ in $BEAM$

Algorithm 2 Online Structured Perceptron Training

Input: $\mathcal{D}$ = Training examples, ϕ = joint feature function, b = beam width, η = learning rate, MAX = maximum training iterations
Output: w, weights of the scoring function

Initialize the weights of the scoring function $w = 0$
for MAX iterations or until convergence **do**
  for each training example $(x, y^*) \in \mathcal{D}$ **do**
    Make prediction: $\hat{y}$ = Beam-Search-Inference(x, ϕ, w, B)
    Compare $\hat{y}$ and y^* to check for error
    If error, perform weight update:
    $w = w + \eta(\phi(x, y^*) - \phi(x, \hat{y}))$
  end for
end for
return weights w

(a) Reuse the structured perceptron training algorithm from HW1 (corresponds to *standard training*). Implement the best-first beam search inference algorithm as shown in the above pseudo-code, and also its breadth-first beam search version.

(b) Modify the structured perceptron training algorithm to perform *early update* and *max-violation* update.

(c) Fix the feature representation ϕ to First-order: unary + pairwise features ($d = m \cdot k + k^2$), $\eta = 0.01$, and number of online iterations MAX=50.

(d) Plot the Hamming accuracy over the training and testing set as a function of the beam width b (1, 5, 10, 15, 25, 50, 100) for best-first beam search: standard update, early update, and max-violation update.

(e) Plot the Hamming accuracy over the training and testing set as a function of the beam width b (1, 5, 10, 15) for breadth-first beam search: standard update, early update, and max-violation update.

(e) List your observations about standard update vs. early update vs. max-violation update.

(f) List your observations about best-first beam search vs. breadth-first beam search based training and inference.