Assignment 4: Hidden Markov Models

CS485/685 – Winter 2016

Out: March 8, 2016
Due: March 21, (11:59 pm), 2016. Submit an electronic copy of your assignment via LEARN.
Late assignments may be submitted within 24 hrs for 50% credit.

1. **[30 pts]** Consider a Hidden Markov Model parametrized by $\Pr(y_t|y_{t-1})$ and $\Pr(x_t|y_t)$. It satisfies the Markov property since the current state y_t depends only on the previous state y_{t-1}. In practice, the current state may depend on earlier states and therefore the Markov property is not satisfied. It turns out that it is possible to ensure that the Markov property holds by augmenting the set of states.

   (a) **[15 pts]** Show how to rewrite a process parametrized by $\Pr(y_t|y_{t-1}, y_{t-2})$ and $\Pr(x_t|y_t)$ into an HMM parametrized by $\Pr(y'_t|y'_{t-1})$ and $\Pr(x_t|y'_t)$.

   (b) **[15 pts]** Is it possible to do this reparametrization without increasing the number of parameters? Justify your answer by counting the number of parameters before and after the transformation.

2. **[20 pts]** In some applications, HMMs are used to model a sequence of observations while the hidden states are treated as latent variables without any meaning. For instance, in gesture recognition, an HMM can be used to model a sequence of hand postures (e.g., location, position and orientation) defining a gesture. Specify a mathematical objective to train an HMM to maximize the likelihood of a set of sequences of hand postures. More precisely, formulate an optimization problem that could be used to learn the parameters of the HMM despite the fact that the hidden states are never observed.

3. **[50 pts]** HMM Implementation

   In this question, you will experiment with a Hidden Markov Model (HMM) and compare it to the mixture of Gaussians model that you implemented in Assignment 2. Download the dataset posted on the course website. It consists of several sequences of continuous inputs and discrete outputs. The goal is to infer the outputs based on the inputs. As a baseline, train a mixture of Gaussians model and classify each instance separately. Then train a hidden Markov Model and classify the instances by taking into account the correlations between them.

   (a) **[10 pts]** Train a Hidden Markov Model by supervised maximum likelihood learning with the training set. Since the inputs are continuous, estimate the mean and covariance matrix of the Gaussian emission distributions. Since the outputs are discrete, estimate the parameters of the multinomial transition distributions and initial state distribution. Similarly, train a mixture of Gaussians model.

      What to hand in:

      - Printout of your code.
      - Printout of the parameters of the HMM and Mixture of Gaussians.

   (b) **[20 pts]** Implement the Forward algorithm for monitoring with your HMM. More precisely, estimate the probability of the class at each step based on the current and previous inputs by computing $\Pr(y_t|x_{1..t})$. In comparison, estimate the probability of the class at each step based on the current input only with the mixture of Gaussian models by computing $Pr(y_t|x_t)$. For both models, return the class that has the highest probability at each step.

      What to hand in:

      - Printout of your code.

- Monitoring accuracy (percentage of correctly classified instances in the test set) of the HMM and Mixture of Gaussian models.
- Discuss the results.

(c) **[20 pts]** Implement the Viterbi algorithm for simultaneous classification of all the instances with the HMM. More precisely, find the sequence of outputs that has the highest probability given all the inputs by computing $argmax_{y_{0..t}} \Pr(y_{0..t}|x_{1..t})$.

What to hand in:

- Printout of your code.
- Joint classification accuracy (percentage of correctly classified instances in the test set).
- Discuss and compare the results found for monitoring and joint classification.