COMP 652 - ECSE 608: Machine Learning - Assignment 3

Posted November 14, 2017
Due December 6, 2017

---

1. [25 points] **EM algorithm**

   In this question we will explore a mixture model for modeling text. Suppose you have a vocabulary of M words. We consider each word in a document as a random variable W whose value is a vector of M components, such that $W(i) = 1$ if the value of W is the ith word in the vocabulary, and 0 otherwise. Hence, $\sum_{i=1}^{M} W(i) = 1$ (this is also known as a one-hot encoding). Suppose the words are generated from a discrete mixture of K latent topics:

   $$P(W) = \sum_{k=1}^{K} \pi_k P(W|\mu_k)$$

   where π_k is the probability for the kth latent topic and $P(W|\mu_k)$ is modeled as:

   $$P(W|\mu_k) = \prod_{i=1}^{M} (\mu_k(i))^{W(i)}$$

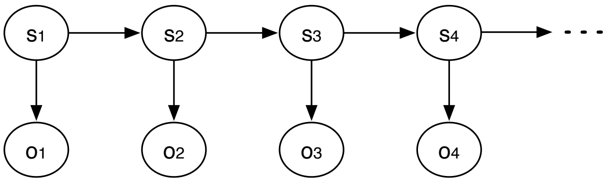
   Hence, we generate a word by drawing a topic k from π and then drawing the word from the topic's distribution, according to μ_k (i.e. the ith component of $\mu_k \in \mathbb{R}^M$ is the probability of drawing the ith word knowing that the topic is k).

   (a) [5 points] Suppose we have documents consisting of N words, which have been drawn i.i.d. according to this process. Suppose that for each document we have a given topic, which is known. Compute the maximum likelihood estimators for the π and μ parameters.

   (b) [15 points] Suppose now that the topics are not known, and in fact, one document may cover multiple topics. Derive an expectation maximization algorithm for learning the parameters π and μ. In this case, for the expectation step, you need to compute the probability of the topic associated with each word W_j, in order to complete the data, and in the maximization step, you need to re-compute the parameters that maximize the likelihood of the data.

   (c) [5 points] The assumption that words are drawn iid from a topic is quite strong. It would make more sense to assume a word's probability is conditioned on the topic as well as the previous word in the document. Explain how many parameters would the model have in this case, and what is the bias-variance trade-off compared to the previous model.

2. [25 points] **Belief propagation and Forward-Backward algorithm in HMMs.**



   (a) [5 points] Consider the HMM directed graphical model we saw in class. Any directed graphical model can be transformed into a factor graph by putting a factor node on each edge. We will denote the factor nodes of edges connecting O_i to S_i by g_i and the ones connecting S_{i-1} to S_i by f_i. How shoud you define the potential functions associated to these factor nodes (as a function of the HMM parameters, i.e. transition/emission/initial probabilities) in order to obtain the same factorization of the joint distribution as the one induced by the directed graphical model.

(b) [10 points] We will perform belief propagation in this graphical model to infer the probability of each state S_i given a sequence of observations $o_1, \cdots, o_t$. We will implement message passing in the following order:

1) Send $\mu_{g_i \to S_i}$ for $i = 1, \cdots, t$
2) For $i = 2, 3, \cdots, t$ (in this specific order), send $\mu_{f_i \to S_i}$
3) For $i = t, t-1, \cdots, 2$ (in this specific order), send $\mu_{S_i \to f_i}$

where $\mu_{f \to V}$ denote the message from a factor f to a node V and $\mu_{V \to f}$ the one from a node V to a factor f.

Give the values of each of these messages as a function of the HMM parameters (i.e. transition probabilities, emission probabilities and initial probabilities).

(c) [5 points] How do you compute the probability $P(S_i = s \mid O_1 = o_1, \cdots, O_t = o_t)$ for any index i and any state s after you performed belief propagation in the previous question?

(d) [5 points] Comment on the relation between the belief propagation algorithm from the previous question and the forward-backward algorithm.

3. [35 points] **Neural Networks and Convolutional Neural Networks**

In this question, we will explore neural networks and convolutional neural networks, and understand how the backpropagation algorithm works.

A neural network, also known as Multi-Layer Perceptrons (MLP), with one hidden layer is to be used for a multi-class classification problem. We can explicitly write down the set of parameters for the model: we denote the matrix of weights from input to the hidden layer as W and from the hidden layer to the output layer as V. There are K classes, the input of the network is a d-dimensional feature vector and there are n examples to train the MLP, $x_1, ...x_n \in \mathbb{R}^d$. For each of the training samples there is a target vector $y_1, ...y_n \in \mathbb{R}^K$, which uses a 1-out-of-K coding for the class of each of the training examples.

(a) What needs to be considered when designing the neural network? Your answer should include a discussion of the number of MLP parameters for the designed neural network.

(b) The MLP parameters are to be trained using cross-entropy. This cost-function has the form

$$E = -\sum_{i=1}^{I} \sum_{j=1}^{J} a_{i,j} \log(b_{i,j}) \tag{1}$$

i. For this expression describe what the variables I, J, $a_{i,j}$ and $b_{i,j}$ represent.
ii. Write down the likelihood term for this model. Comments?
iii. To avoid overfitting, add an L2 regulariser to the model. Write down the overall objective function that needs to be minimized.
iv. An alternative to L2 regularisation in neural networks is to use Dropout during training. Explain how Dropout works as a regulariser in model fitting.
v. Explain the difference between using a L2 and a Dropout regulariser to avoid overfitting.
vi. With the expression for the overall objective function, find the derivatives with respect to the parameters needed to implement the the backpropagation algorithm.

(c) The Hessian matrix is to be used in the optimisation process for the model parameters in $b(ii)$.

i. How is the Hessian matrix derived, and how can it be used to improve the training of the MLP.
ii. Discuss the computational cost of using the Hessian in optimising the output-layer weights.

A *convolutional neural network (CNN)* is to be trained for pedestrian detection. A labelled dataset has been collected that contains greyscale images $\{Z^n\}_{n=1}^N$ and binary labels $\{t^n\}_{n=1}^N$ which indicate whether pedestrians are present in each image.

The network contains three stages. The first stage carries out a 2D convolutional between the image pixels Z^n_{ij} and convolutional weights $W_{i,j}$.

$$a^n_{i,j} = \sum_{k,l} W_{k,l} Z^n_{i-k,j-l} \tag{2}$$

The second stage applies a point-wise non-linearity $y^n_{i,j} = f(a^n_{i,j})$.

The third stage applies a set of output weights $V_{i,j}$ and a logistic non-linearity in order to form the scalar output of the network. *Note that there is no pooling stage in this architecture.*

$$x^n = \frac{1}{1 + \exp(-\sum_{i,j} V_{i,j} y^n_{i,j})} \tag{3}$$

The network's weights will be trained using the following objective function.

$$G(V, W) = -\sum_{n=1}^{N} (t^n \log(x^n)) + (1 - t^n \log(1 - x^n) + \frac{\alpha}{2} \sum_{i,j} V^2_{i,j} + \frac{\beta}{2} \sum_{i,j} W^2_{i,j} \tag{4}$$

(a) Explain the three stages of the convolutional network model in this architecture, and why we need these three stages. What are the benefits of using a CNN compared to using a neural network with multiple hidden layers?

(b) Describe how to train the network's convolutional weights W using gradient descent. Derive the gradient required to implement gradient descent. Simplify your expression and interpret the terms.

(c) Describe enhancements to the architecture of the network that might improve its ability to perform pedestrian detection.

4. [20 points] **A midterm preparation question**

For each of the learning problems outlined below, specify what is the best learning algorithm to use and why. Note that you should give *one* algorithm for each problem, even if there are several correct answers.

(a) You have about 1000 training examples in a 6-dimensional continuous feature space and the output is binary. You only expect to be asked to classify 100 test examples.

(b) You have a problem similar to the previous one except that the inputs are now of dimension 10,000 and you know that the relation between inputs and outputs is linear.

(c) You want to predict the values of several stock indexes for the next 3 days from their values in the preceding week.

(d) You want to design an algorithm to predict whether a machine in a factory is likely to have a failure given various available measurements and you want to use this algorithm in the following way: if the likelihood of a failure is above 85%, you ask for a technician to do a preemptive maintenance check. You would also like to be able to leverage the fact that you know the likelihood of a random failure of any machine.