67750: Advanced Practical Machine Learning	2018/9

Quiz

TA: Daniel Gissin

You have **two hours** to solve this quiz, which should be more than enough time. **Answer three out of the four possible sections**. You get 1 point for writing your ID on every page. You may either answer in English or in Hebrew.

Good luck!

1 The EM Algorithm (33 Points)

1.1 MLE of the Exponential Distribution

In class we calculated the MLE of a sample drawn from a Bernoulli distribution or a Gaussian distribution.

You are now the manager of the Rothberg cafeteria, and you collect N samples of the time it took from the moment person i asked for a salad until he got his salad. You assume, as one does, that the service time is distributed exponentially ($p(x) = \lambda e^{-\lambda x}$ for $x \geq 0$). Calculate the optimal parameter λ using MLE and show that:

$$\hat{\lambda}_{MLE} = \frac{1}{\bar{x}}$$

1.2 Mixture of Exponentials

There are k employees selling salads in the cafeteria, but your data doesn't contain who made each salad. Still, you want to try and estimate the service time distribution of each of your employees.

1. Write the probability distribution function of the "mixture of exponentials" model for a single service time, which assumes that first an employee is picked according to multinomial weights π and then the service time is sampled from that employee's exponential distribution.

2. What is the log likelihood function of the entire sample S under the mixture of exponentials model?

3. Formulate the EM update for this model.

2 Maximum Entropy Markov Models (33 Points)

2.1 Definition & Inference

For sequential data $(x_{1:T}, y_{1:T})$, we assume that the y variables are Markovian and hidden while the x variables are observed (as in the PoS tagging application from Ex2).

1. Define $\mathbb{P}(y_t|y_{t-1}, x_{1:T})$ according to the MEMM defined by the feature function ϕ and the weight vector w. Explicitly write the partition function which normalizes the distribution (Z).

2. Inference for the MEMM is done using Viterbi. Write the definition of $\pi_t(i)$, the value which we calculate in the Viterbi dynamic programming algorithm.

3. We can efficiently calculate $\pi_t(i)$ with a recursive formula, by using the values of π_{t-1}. Write the recursive formula for calculating $\pi_t(i)$, given π_{t-1} and the model parameters.

2.2 Learning

1. write the log probability of a single transition according to the MEMM ($log(\mathbb{P}(y_t|y_{t-1}, x_{1:T}))$).
2. Given a transition from your dataset $(y_{t-1}, y_t, x_{1:T})$, calculate the gradient of the log probability of the transition with respect to w. Show that it is the difference between the feature vector according to the real transition and the expected feature vector according to the model distribution (the expectation is over y_t).

3 Convnets (33 Points)

3.1 Neural Network Expressiveness

Consider two network layer architectures - the first is a fully connected layer followed by ReLU activation with an output of 100 neurons. The second is a convolutional layer of one weight filter followed by ReLU activation such that the output is 100 neurons (the image has 10×10 pixels).

Is one architecture more expressive than the other?

If the answer is no, show how any weight configuration of one layer can be expressed by the other.

If the answer is yes, show how one type of layer can express any configuration of the other and show a counter example for how the other layer can't express a specific configuration of the first layer.

3.2 Log Loss

A popular loss function for binary classification tasks is the log loss:

$$\ell(y, p) = -ylog(p) - (1 - y)log(1 - p)$$

Assume our networks final activation function is a sigmoid whose input comes from a fully connected layer:

$$p = \sigma(w^T x) = \frac{1}{1 + e^{-w^T x}}$$

Calculate the derivative of the log loss with respect to w. Simplify your solution as much as you can. It may be helpful to use the chain rule. You do not have to derive expressions we saw in class.

4 Clustering (33 Points)

4.1 k-means

1. Describe Lloyd's iterative algorithm for heuristically solving the k-means problem for the euclidean distance metric.
2. Prove that said algorithm converges in a finite number of steps.

4.2 Spectral Clustering

Spectral clustering performs k-means on an embedding of our original data. In the embedding, we move from the data matrix X to a distance matrix S and then to a non negative similarity matrix W. We then minimize the Laplacian of the graph whose adjacency matrix is described by W.

1. Define the graph Laplacian for a given adjacency matrix W.
2. Show that the vector of 1s is an eigenvector of L with an eigenvalue of 0.
3. Show that L is a PSD matrix (show that for any vector f, $f^T L f \geq 0$).

Good luck

(did you write your ID on all of the pages?)