Discussion 9

Probabilistic Machine Learning, Spring 2018

1 Jensen's Inequality

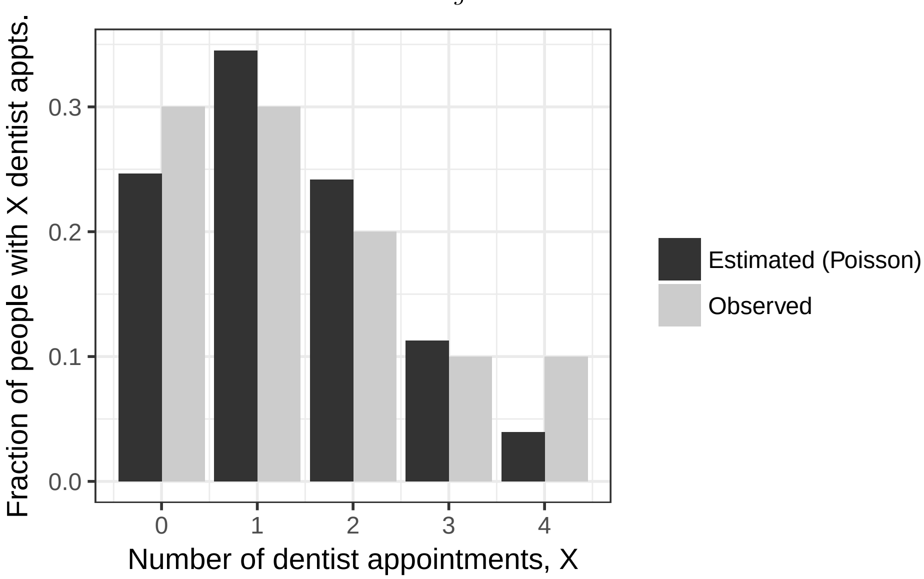
Let Y be a positive random variable and let $p > q \geq 1$. Relate $\mathbb{E}\left[Y^p\right]^{1/p}$ to $\mathbb{E}\left[Y^q\right]^{1/q}$ by an inequality.

2 Gaussian Mixture Models

Assume data is generated by two univariate Gaussian distributions, the first with mean 0 and variance 1, the second with mean 0 and variance 1/2. Let w denote the mixing weight. If there were a single observation x_1, what is the likelihood function and the maximum likelihood estimate $\hat{w}$ of w?

3 Expectation Maximization (EM) for a Mixture of Two Different Distributions

Figure 1



Suppose $N = 100$ people were surveyed about how many dentist appointments they made in the past year. One way to model this data to assume the responses $\{x_i\}_{i=1}^{100}$ are drawn i.i.d. from a Poisson distribution, so $p(X_i = x_i \mid \lambda) = \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}$. Figure 1 shows a histogram of the observed number appointments and the predicted number of appointments from maximizing the assumed likelihood.

(a) Derive the maximum likelihood estimator $\hat{\lambda}$ for λ.

(b) The numerical result is $\hat{\lambda} = 1.4$. Figure 1 shows the predicted number of visits with this value for λ. Is there anything suboptimal about how the model fits the data?

We will now try a new model. Suppose that each person i is one of two types, denoted by a latent variable Z_i:

- If $Z_i = 1$, then person i never goes to the dentist (with probability 1), so $p(X_i = x_i \mid Z_i = 1, \lambda) = \mathbb{1}_{[x_i=0]}$.
- If $Z_i = 2$, then $p(X_i = x_i \mid Z_i = 2, \lambda) = \text{Poisson}(\lambda)$, as before.

We model each person as a mixture of these two types, letting $w = p(Z_i = 1 \mid \lambda)$ and $1 - w = p(Z_i = 2 \mid \lambda)$ denote the mixture weights. In general, the presence of a latent variable like Z_i can make it difficult to maximize the likelihood. We use the EM algorithm as a remedy to this problem.

E-step

In this step we compute the probability of each type assignment for each person. That is, we compute $\gamma_{i,k} := p(Z_i = k \mid X_i = x_i, \lambda)$ for all people $i \in \{1, \ldots, N\}$ and all types $k \in \{1, 2\}$.

(c) Write a formula for $p(X_i = x_i \mid Z_i = k, \lambda)$, the likelihood of observing outcome x_i given that person i is of type $Z_i = k$.

(d) Write a formula for $p(X_i = x_i \mid \lambda)$, the likelihood of observing outcome x_i.

(e) Write a formula for the type assignments $\gamma_{i,k} := p(Z_i = k \mid X_i = x_i, \lambda)$.

M-step

In this step we maximize a lower bound of the log likelihood:

$$A(w, \lambda) = \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{i,k} \log \frac{p(X_i = x_i, Z_i = k \mid \lambda)}{\gamma_{i,k}}$$

over w and λ. Note that in this step the type assignments $\gamma_{i,k}$ are fixed.

(f) Maximize $A(w, \lambda)$ in λ. How does this compare to the maximum likelihood estimate when there was no latent variable (derived in part (a))?

(g) Maximize $A(w, \lambda)$ in w. How does this compare to the mixture of Gaussian distributions case, as derived in class?

4 Basics of Neural Networks

4.1

- A perceptron is guaranteed to perfectly learn a given linearly seperable function within a finite number of traning steps.
- For effective training of a neral network, the network should have at least 5-10 times as many weights as there are training samples.
- A single perceptron can coumpute the XOR function.
- The more hidden-layer units a BPN (BackPropagation Neural Network) has, the better it can predict desired outputs for new inputs that it was not trained with.

- In backpropagation learning, we should start with a small learning parameter η and slowly increase it during the learning process.
- A three-layer BPN with 5 neurons in each layer has a total of 50 connections and 50 weights.
- The backpropagation learning algorithm is based on the gradient-descent method.
- Some conflicts among training exemplars in a BPN can be resolved by adding features to the input vectors and adding input layer neurons to the network.

4.2

Derive the derivative of the tanh activation function

$$f(x) = \frac{2}{1+e^{-x}} - 1$$

Can it be expressed as a function of $f(x)$? Explain.