COMS 4771 Spring 2017 Homework 5

Jin Tack Lim, jl4312

Problem 1.
(a)

- Initialization

  random $p^j(1), ..., p^j(D)$ where j = 1, ..., K

  $\pi_j = \frac{1}{K}$

- Expectation

  The probability that n-th document belongs to the class i. For t-th iteration, the superscript t for each of $\tau, \pi, p^j(1), ..., p^j(D)$ are omitted.

$$\tau_{n,i} = \frac{\pi_i \Pi_{d=1}^{D} p^i(d)^{x_n(d)}}{\sum_{j=1}^{K} \pi_j \Pi_{d=1}^{D} p^j(d)^{x_n(d)}}$$

- Maximization Update $p^i(1), ..., p^i(D)$ and π_i where i = 1, ..., K

$$\pi_i^{(t+1)} = \frac{\sum_{q=1}^{n} \tau_{q,i}^{(t)}}{n}$$

$$p^{i(t+1)}(d) = \frac{\sum_{q=1}^{n} \tau_{i,q} x(d)}{\sum_{q=1}^{n} \tau_{i,q} l_q}$$

  where d=1, ..., D, q = 1, ..., n

(b)
The marginal distribution of x_n is

$$\frac{\hat{x_n}}{l_n}$$

This can be interpreted as the binomial distribution for each dimensions. So if l_n is very large and $\frac{\hat{x_n}}{l_n}$ is very small, it can be approximated to the Poisson distribution, where $\lambda = \hat{x_n}$.
(c)

1. Similarity
   - Both of them can reduce the dimensionality.
2. Difference
   - PCA gives vectors in order such that the variance along the vector is high. (e.g. the given data has the highest variance along the first vector.) For (b), there's no guarantee.

- PCA gives vectors that are orthogonal to each other, since they are all eigenvectors. For (b), there's no guarantee.

(d)

Problem 2.

Problem 3.

We have q+2 layers except the convolution layer. Input to each layer is a^p and output is z^p where p = 1, ..., q+2.

$$R = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}(y_n - z_n^{q+2})^2$$

1. the update equation for V

$$\begin{aligned}\frac{\partial R}{\partial V} &= \frac{\partial R}{\partial z^{q+2}}\frac{\partial z^{q+2}}{\partial a^{q+2}}\frac{\partial a^{q+2}}{\partial V} \\ &= -\frac{1}{N}\sum_{n=1}^{N}(y - z^{q+2})z^{q+2}(1 - z^{q+2})z^{q+1}\end{aligned}$$

   Therefore, the update equation for V is

$$V^{t+1} = V^t - \eta\frac{\partial R}{\partial V}$$

2. the update equation for U

$$\begin{aligned}\frac{\partial R}{\partial U} &= \frac{\partial R}{\partial z^{q+1}}\frac{\partial z^{q+1}}{\partial a^{q+1}}\frac{\partial a^{q+1}}{\partial U} \\ &= \frac{\partial R}{\partial a^{q+2}}\frac{\partial a^{q+2}}{\partial z^{q+1}}\frac{\partial z^{q+1}}{\partial a^{q+1}}\frac{\partial a^{q+1}}{\partial U} \\ &= \left(-\frac{1}{N}\sum_{n=1}^{N}(y - z^{q+2})z^{q+2}(1 - z^{q+2})\right)Vz^{q+1}(1 - z^{q+1})z^q\end{aligned}$$

   The term put in the bracket is the one we can reuse from the previous derivation. (Backprop)

   Therefore, the update equation for U is

$$U^{t+1} = U^t - \eta\frac{\partial R}{\partial U}$$

3. the update equation for W

   This is only valid for the points X which are selected in the down-sampling layers. Since the down-sampling layers except the first one don't have weights, we get the derivatve in the first down sampling layer.

$$
\begin{aligned}
\frac{\partial R}{\partial W} &= (\frac{\partial R}{\partial z^1})\frac{\partial z^1}{\partial a^1}\frac{\partial a^1}{\partial W} \\
&= (\frac{\partial R}{\partial a^2}\frac{\partial a^2}{\partial z^1})\frac{\partial z^1}{\partial a^1}\frac{\partial a^1}{\partial W} \\
&= (\frac{\partial R}{\partial a^2})\frac{\partial z^1}{\partial a^1}\frac{\partial a^1}{\partial W}(\text{—— *note 1}) \\
&= (\frac{\partial R}{\partial a^{q+1}})\frac{\partial z^1}{\partial a^1}\frac{\partial a^1}{\partial W}(\text{—— *note 2}) \\
&= \Big(-\frac{1}{N}\sum_{n=1}^{N}(y - z^{q+2})z^{q+2}(1 - z^{q+2})Vz^{q+1}(1 - z^{q+1})\Big)\frac{\partial z^1}{\partial a^1}\frac{\partial a^1}{\partial W}(\text{—— *note 3}) \\
&= \Big(-\frac{1}{N}\sum_{n=1}^{N}(y - z^{q+2})z^{q+2}(1 - z^{q+2})Vz^{q+1}(1 - z^{q+1})\Big)z^1(1 - z^1)\frac{\partial a^1}{\partial W} \\
&= \Big(-\frac{1}{N}\sum_{n=1}^{N}(y - z^{q+2})z^{q+2}(1 - z^{q+2})Vz^{q+1}(1 - z^{q+1})\Big)z^1(1 - z^1)\sum_{k,l} x_{i+k,j+l}
\end{aligned}
$$

Therefore, the update equation for W is

$$W^{t+1} = W^t - \eta\frac{\partial R}{\partial W}$$

Here's detailed explanations for each note.

note 1: The term (da2/dz1) is 1 if this point survived in the last down-sampling layer and 0 if not. We are only considering points which survived, so it's 1.

note 2: Since the down-samplying layers don't have weight, the gradient is propogated without change. So we can use the gradient from q+1 layer.

note 3: We expand dR/da(q+1) from the previous derivative. No surprise.

References for the problem 3.

[1]: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/
[2]: http://cs231n.github.io/convolutional-networks/