CIS606 – Lecture 5

Woon Wei Lee, Jacob Crandall
Spring 2014, 10:00m-11:15pm,
Mondays and Thursdays

For today:

- Mixture of Naïve Bayes model
- EM algorithm - derivation of MNB update rules

Refresher on the Naïve Bayes

- **We will be focusing on the Multivariate Bernoulli "version" of Naïve Bayes**
  - For a multivariate variable $\boldsymbol{x} \sim \{x_1, x_2, x_3 \ldots x_n\}$**:**

$$P(c_i|x) = \frac{p(x|c_i)\,p(c_i)}{p(x)}$$

$$= \frac{p(x_1, x_2, \cdots x_n|c_i)\,p(c_i)}{p(x)}$$

  - Problem: how do we find joint likelihood over all the features
    - No easy analytical form
    - Typically high dimensional, sparse, etc.
  - Naïve Bayes assumption → variables are independent
    - Permits following simplication:

$$P(c_i|x) = \frac{p(x_1, x_2, \cdots x_n|c_i)\,p(c_i)}{p(x)}$$

$$= \frac{p(x_1|c_i) \cdot p(x_2|c_i) \cdots p(x_n|c_i)\,p(c_i)}{p(x)}$$

Mixture of Naïve Bayes (MNB)

- **Previous incarnation, was purely as a classification technique:**
  - Find $p(x_i|c_j)$ for each variable/feature and class
  - Estimate $p(x_1 \ldots x_n|c_j)$ for each class and hence find the most probable class.
  - Firmly in the “supervised learning” category
- **We can also use the Naïve Bayes assumption as a tool for probabilistic modeling.**
  - Let's assume the following distribution function

$$P(x;\theta)=\sum_{i=1}^{k} p(x|c_i;\theta)\,p(c_i;\theta)$$

$$=\sum_{i=1}^{k} p(x_1, x_2, \cdots x_n|c_i;\theta)\,p(c_i;\theta)$$

$$=\sum_{i=1}^{k}\left[\prod_{j=1}^{m} p(x_j|c_i)\,p(c_i;\theta)\right]$$

  - i.e. the total data density is the sum of a number of component densities, each of which is modelled using the Naïve Bayes assumption

(Cont'd)

- **This is part of the large and very important group of *mixture models***
  - Gaussian Mixtures
  - Mixtures of experts
  - Mixtures of factor analyzers, etc etc.
- **In this case, the learning process is now an unsupervised learning process:**
  - Objective is to find:

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{k} \left[\prod_{j=1}^{m} p(x_j|c_i)\, p(c_i;\theta) \right]$$

  - Basic set-up:
    - Hidden variable: z ~ “class labels” c_j
    - Parameters
      - Class-conditional distributions $p(x_i|z=j)$

        (we are taking the multivariate bernoulli process model, so x={1,0})
      - Prior distribution on each of the classes, p(z)
    - As usual, θ denote the parameter vector

Learning algorithm for MNB

- **First, let's rewrite the previous equation in a more familiar form:**

$$p(x_j | c_i)\, p(c_i ; \theta) \rightarrow p(x_j | z ; \theta)\, p(z ; \theta) = p(x_j , z ; \theta)$$

- **Maximum likelihood optimization:**

$$\theta^* = \underset{\theta}{arg\ max} \log p(\boldsymbol{X} ; \theta) \qquad \text{(incomplete data likelihood)}$$

$$= \underset{\theta}{arg\ max} \log \sum_{z=1}^{k} p(\boldsymbol{X}, z ; \theta)$$

$$= \underset{\theta}{arg\ max} \sum_{i=1}^{N} \log \sum_{z=1}^{k} \left[\prod_{j=1}^{m} p(x_j(i), z ; \theta) \right]$$

- **As usual, we get the pesky "log of sums" expression**
  - To derive tractable maximum likelihood solution

    → EM algorithm!

E-step

- **First, introduce the variational distribution Q(z)**

$$\log \sum_{z=1}^{k} p(\boldsymbol{X}, z; \theta) = \log \sum_{z=1}^{k} Q(z) \frac{p(\boldsymbol{X}, z; \theta)}{Q(z)}$$

- **Next, to create tight lower bound, set:**

$$Q(z; x(i), \hat{\theta}) = p(z | x(i); \hat{\theta}) \quad = \frac{p(x(i) | z; \hat{\theta})\, p(z; \hat{\theta})}{p(x(i) | \hat{\theta})}$$

$$= \frac{p(x(i) | z; \hat{\theta})\, p(z; \hat{\theta})}{\sum_{z} p(x(i), z | \hat{\theta})}$$

$$= \frac{\prod_{j=1}^{m} p(x_j(i) | z; \hat{\theta})\, p(z; \hat{\theta})}{\sum_{z=1}^{k} \left[\prod_{j=1}^{m} p(x_j(i) | z; \hat{\theta})\, p(z; \hat{\theta}) \right]}$$

M-step

- **Next, maximize the expectation of the complete data log likelihood**

$$\theta^* = \arg\max_{\theta} E_{Q(z;x,\hat{\theta})}\left[\log p(\boldsymbol{X}, z; \theta)\right]$$

$$= \arg\max_{\theta} \sum_{i=1}^{N} \sum_{z=1}^{k} Q(z; x(i), \hat{\theta}) \log\left[\prod_{j=1}^{m} p(x_j(i), z; \theta)\right]$$

$$= \arg\max_{\theta} \sum_{i=1}^{N} \sum_{z=1}^{k} Q(z; x(i), \hat{\theta}) \sum_{j=1}^{m} \left[\log p(x_j(i) | z; \theta) + \log p(z; \theta)\right]$$

- **Also:**
  - We are assuming the multivariate Bernoulli model

    → x is one of {1,0}, let's take p_{jz} to be $p(x_j=1|z)$
  - As such, note that:

$$\log p(x_j | z; \theta) = x_j\, p_{jz} + (1 - x_j)(1 - p_{jz})$$

(Cont'd)

- **Hence, expression becomes:**

$$\theta^* = \underset{\theta}{arg\,max}\, E_{Q(z;x_i,\hat{\theta})}\left[\log p(\boldsymbol{X}, z;\theta)\right]$$

$$= \underset{\theta}{arg\,max} \sum_{i=1}^{N}\sum_{z=1}^{k} Q(z;x(i),\hat{\theta}) \sum_{j=1}^{m} \log\left(x_j(i)\,p_{jz} + (1-x_j(i))(1-p_{jz})\right) \quad \text{A}$$

$$\ldots + \sum_{i=1}^{N}\sum_{z=1}^{k} Q(z;x(i),\hat{\theta}) \sum_{j=1}^{m} \log p(z;\theta) \quad \text{B}$$

- **Let's call the expression in the argmax →** $R(\theta;\hat{\theta})$
  - Taking derivative w.r.t. P_{jz} (only expression (A) involved):

$$\frac{\partial R(\theta;\hat{\theta})}{\partial p_{jz}} = \sum_{i=1}^{N} \frac{Q(z;x(i),\hat{\theta})(x_j(i)-(1-x_j(i)))}{\left(x_j(i)\,p_{jz}+(1-x_j(i))(1-p_{jz})\right)}$$

(Cont'd again)

- **To optimize, set to zero:**

$$\frac{\partial R(\theta;\hat{\theta})}{\partial p_{jz}} = \sum_{i=1}^{N} \frac{Q(z;x(i),\hat{\theta})(x_j(i)-(1-x_j(i)))}{\left(x_j(i)\,p_{jz}+(1-x_j(i))(1-p_{jz})\right)}$$

- **Also:**

$$\frac{Q(z;x(i),\hat{\theta})(x_j(i)-(1-x_j(i)))}{\left(x_j(i)\,p_{jz}+(1-x_j(i))(1-p_{jz})\right)} = \begin{cases} (x_j(i)=1) \;\ldots\ldots\ldots\; \dfrac{Q(z;x(i),\hat{\theta})}{p_{jz}} \\ (x_j(i)=0) \;\ldots\ldots\ldots\; -\dfrac{Q(z;x(i),\hat{\theta})}{(1-p_{jz})} \end{cases}$$

- **Therefore:**

$$\sum_{i=1}^{N} \frac{Q(z;x_i,\hat{\theta})}{\left(x_j(i)\,p_{jz}+(1-x_j(i))(1-p_{jz})\right)} = \frac{\sum_{\{x:x=1\}} Q(z;x_i,\hat{\theta})}{p_{jz}} - \frac{\sum_{\{x:x=0\}} Q(z;x_i,\hat{\theta})}{(1-p_{jz})}$$

Masdar INSTITUTE

(Cont'd yet again)

- **Setting this to zero:**

$$\frac{\sum_{\{x:x=1\}} Q(z;x(i),\hat{\theta})}{p_{jz}} - \frac{\sum_{\{x:x=0\}} Q(z;x(i),\hat{\theta})}{(1-p_{jz})} = 0$$

$$\Rightarrow \quad \frac{\sum_{\{x:x=1\}} Q(z;x(i),\hat{\theta})}{p_{jz}} = \frac{\sum_{\{x:x=0\}} Q(z;x(i),\hat{\theta})}{(1-p_{jz})}$$

- **Therefore:**

$$\frac{p_{jz}}{1-p_{jz}} = \frac{\sum_{\{x:x=1\}} Q(z;x(i),\hat{\theta})}{\sum_{\{x:x=0\}} Q(z;x(i),\hat{\theta})}$$

$$\Rightarrow \quad p_{jz} = \frac{\sum_{\{x:x=1\}} Q(z;x(i),\hat{\theta})}{\sum_{\{x:x=0\}} Q(z;x(i),\hat{\theta}) + \sum_{\{x:x=1\}} Q(z;x(i),\hat{\theta})}$$

(Cont'd)

- **Refresher:**

$$\theta^* = arg\ \max_{\theta} \sum_{i=1}^{N} \sum_{z=1}^{k} Q(z;x(i),\hat{\theta}) \sum_{j=1}^{m} \log\left(x_j(i)\,p_{jz} + (1-x_j(i))(1-p_{jz})\right) + \ldots \quad \text{A}$$

$$\ldots + \sum_{i=1}^{N} \sum_{z=1}^{k} Q(z;x(i),\hat{\theta}) \sum_{j=1}^{m} \log p(z;\theta) \quad \text{B}$$

- **Taking derivative w.r.t. p(z;θ) (only expression (B) involved):**

$$\frac{\partial R(\theta;\hat{\theta})}{\partial p(z|\theta)} = \frac{\sum_{i=1}^{N} Q(z;x(i),\hat{\theta})}{p(z|\theta)}$$

- **Problem! Setting to zero gives p(z;θ)=∞**
  - Because setting this "maximizes" the likelihood..
  - Can be solved by adding the following constraint expression to (B):

$$\text{B} \Rightarrow \ldots + \sum_{i=1}^{N} \sum_{z=1}^{k} Q(z;x(i),\hat{\theta}) \sum_{j=1}^{m} \log p(z;\theta) - \lambda\left(\sum_{z} p(z;\theta) - 1\right)$$

Masdar INSTITUTE

(Cont'd)

- **Taking derivative w.r.t. p(z;θ), and setting to zero:**

$$\frac{\partial R(\theta;\hat{\theta})}{\partial p(z|\theta)} = \frac{\sum_{i=1}^{N} Q(z;x(i),\hat{\theta})}{p(z|\theta)} - \lambda = 0$$

- **Substituting this expression back into our constraint**:

$$p(z|\theta) = \frac{1}{\lambda}\sum_{i=1}^{N} Q(z;x(i),\hat{\theta})$$

$$\sum_{z} p(z|\theta) = 1 \quad \Rightarrow \quad \sum_{z} \frac{1}{\lambda}\sum_{i=1}^{N} Q(z;x(i),\hat{\theta}) = 1$$

$$\lambda = \sum_{z} \sum_{i=1}^{N} Q(z;x(i),\hat{\theta})$$

$$p(z|\theta) = \frac{\sum_{i=1}^{N} Q(z;x(i),\hat{\theta})}{\sum_{z}\sum_{i=1}^{N} Q(z;x(i),\hat{\theta})}$$

All done!

- **E-Step:**

$$Q(z;x(i),\hat{\theta}) = \frac{\prod_{j=1}^{m} p(x_j(i)|z;\hat{\theta})\, p(z;\hat{\theta})}{\sum_{z=1}^{k}\left[\prod_{j=1}^{m} p(x_j(i)|z;\hat{\theta})\, p(z;\hat{\theta})\right]}$$

- **M-Step:**

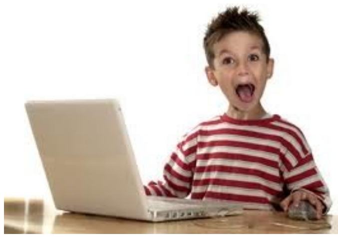
$$p_{jz} = \frac{\dfrac{\sum_{\{x:x=1\}} Q(z;x(i),\hat{\theta})}{\sum_{\{x:x=0\}} Q(z;x(i),\hat{\theta})}}{1+\dfrac{\sum_{\{x:x=1\}} Q(z;x(i),\hat{\theta})}{\sum_{\{x:x=0\}} Q(z;x(i),\hat{\theta})}}$$

$$p(z|\theta) = \frac{\sum_{i=1}^{N} Q(z;x(i),\hat{\theta})}{\sum_{z}\sum_{i=1}^{N} Q(z;x(i),\hat{\theta})}$$

And... it actually works!

```
# Trains a model using mixture of naive bayes learning
# data in dim x numdocs
# k is number of hidden naive bayes models to use
function results=em_mixture_nbayes(data,k)

# Setting stuff up
[dim,numdocs]=size(data);
qz=zeros(k,numdocs);
px=rand(dim,k);
num_iter=80;
pz=repmat(1/k,k,1); # This is p(z;theta) (prior for z) -> assumed to be uniform

# Starting the iteration
for count=1:num_iter

  ########################################
  # E-step
  # qz_{i,j}=p(z_i|x_j,current_px_values)
  ########################################

  for kk=1:k
    pxmat=repmat(px(:,kk),1,numdocs);
    #qz(kk,:)=prod(data.*pxmat+~data.*(1-pxmat)); <- possibly buggy
    qz(kk,:)=pz(kk)*prod(data.*pxmat+~data.*(1-pxmat));
  endfor

  # Calculating the log-likelihood p(x|px)
  log_likelihood=sum(log(sum(qz)),2);

  # Calculating the p(z_i|x_j,current_px_values) (i.e. Q*(z;theta))
  qz=qz./repmat(sum(repmat(pz,1,numdocs).*qz),k,1);

  # Generating output
  if mod(count-1,10)==0
    disp(["Iteration number ",num2str(count),". Log-Likelihood:",num2str(log_likelihood)]);
    fflush(stdout);
  endif

  ########################################
  # M-step
  ########################################

  for kk=1:k

    pdiv1minusp=sum(data.*repmat(qz(kk,:),dim,1),2)./sum((~data).*repmat(qz(kk,:),dim,1),2);
    px(:,kk)=pdiv1minusp./(1+pdiv1minusp);
    pz=sum(qz,2)/sum(sum(qz));

  endfor

endfor

disp("p(x_i|z_j) is:");
px
disp("");
disp("p(z_i_current is:");
pz
```

1 page of code...



“True” distributions

EM algorithm

Recovered p_{jz}'s

Recovered p(z)'s

```
octave:6> dists
dists =

   0.30000   0.70000   0.70000   0.20000   0.20000
   0.60000   0.10000   0.10000   0.90000   0.90000
   0.10000   0.80000   0.80000   0.30000   0.30000
   0.90000   0.20000   0.20000   0.20000   0.20000
   0.50000   0.20000   0.20000   0.60000   0.60000
   0.20000   0.50000   0.50000   0.20000   0.20000

octave:7> em_mixture_nbayes(data,3);
Iteration number 1. Log-Likelihood:-2.5404e+05
Iteration number 11. Log-Likelihood:-1.8915e+05
Iteration number 21. Log-Likelihood:-1.8839e+05
Iteration number 31. Log-Likelihood:-1.8833e+05
Iteration number 41. Log-Likelihood:-1.8831e+05
Iteration number 51. Log-Likelihood:-1.8831e+05
Iteration number 61. Log-Likelihood:-1.8831e+05
Iteration number 71. Log-Likelihood:-1.883e+05
p(x_i|z_j) is:
px =

   0.696265   0.194033   0.294410
   0.093630   0.922830   0.605943
   0.801575   0.311763   0.102903
   0.199731   0.189878   0.809579
   0.200851   0.607593   0.494048
   0.494811   0.200901   0.207522


p(z_i_current is:
pz =

   0.36652
   0.35095
   0.28253

octave:8> 
```