The Expectation-Maximization Algorithm:
Bernoulli Mixture Models Case Study and the General Case
Naïve Bayes: Review

• Training data:
  
  \( (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)}) \)

  • \( x \): an \( p \)-dimensional vector of binary variables
  • \( y \): a discrete label

• Assumption:
  
  \[ P(x_1, \ldots, x_p | y = c) = \prod_{i=1}^{p} p(x_i | y = c) \]

• Estimate:
  
  \[ P(x_j = 1 | y = c) \approx \frac{\text{count}(x_j=1, y=c)}{\text{count}(y=c)} \]
  
  \[ P(y = c) \approx \frac{\text{count}(y=c)}{m} \]

• Predict:
  
  \[ P(y = c | x) = \frac{P(y=c)P(x|y=c)}{\sum_{c'} P(y=c')P(x|y=c')} \]

Parameters:

\[ \theta_{j,c} = P(x_j = 1 | y = c) \]

\[ \pi_c = P(y = c) \]
Naïve Bayes: Num. of Parameters

- $\theta_{j,c} = P(x_j \mid y = c)$
  - $\text{classes} \times \text{dim}(x)$ parameters
  - $P(x_j = 0 \mid y = c) = 1 - \theta_{i,j,c}$
- $\pi_c = P(y = c)$
  - $(\text{classes} - 1)$ parameters
  - $\pi_1 = 1 - \sum_{c'=2\ldots\text{classes}} \pi_{c'}$
- A total of $\text{classes} \times \text{dim}(x) + \text{classes} - 1$ parameters to estimate
What if we don’t know the labels?

• If we know the parameters, we can guess the labels

\[
P(y = c | x) = \frac{P(y = c) P(x | y = c)}{\sum_{c'} P(y = c') P(x | y = c')}
\]

• Can guess \(y = 1\) if \(P(y = c | x) > 0.5\), or just be happy with the probability that \(y = c\): the expectation of an indicator variable that checks if the class is \(c\)

  • \(E[I[y = c]|x] = P(y = c|x)\)

• If we know the labels, we can estimate the parameters

\[
I[y = c] = \begin{cases} 
1, & y = c \\
0, & otherwise
\end{cases}
\]
Expectation-Maximization

• Start with a random guess of the parameters $\theta$ and $\pi$

• Repeat:

  - For each example $i$ in the training set, compute
    \[ E_{\theta,\pi}[I[y^{(i)} = c]|x^{(i)}] = P_{\theta,\pi}(y^{(i)} = c|x^{(i)}) \]
    for every class $c$

  - Compute the expected number of examples for every class $c$ and feature $j$
    \[ \text{count} (x_j = 1, y = c) = E_{\theta,\pi} \left[ \sum_{i|x_j^{(i)} = 1} I[y^{(i)} = c]|x^{(i)} \right] = \sum_{i|x_j^{(i)} = 1} E_{\theta,\pi} [I[y^{(i)} = c]|x^{(i)}] \]
    \[ \text{count} (y = c) = E_{\theta,\pi} \left[ \sum_i I[y^{(i)} = c]|x^{(i)} \right] \]

• Re-estimate the $\theta$ and $\pi$ using the new counts
E-step

\[ E_{\theta,\pi}[I[y^{(i)} = c]|x^{(i)}] = P_{\theta,\pi}(y^{(i)} = c|x^{(i)}) \]

• Assume you know the parameters, estimate the labels

• We use *soft assignment*: a point can be assigned to \( y = 1 \) with probability 0.9 and to \( y = 0 \) with probability 0.1
M-step

\[
\widehat{\text{count }} (x_j = 1, y = c) = E_{\theta, \pi} \left[ \sum_{i | x_j^{(i)} = 1} I[y^{(i)} = c] | x^{(i)} \right] = \sum_{i | x_j^{(i)} = 1} E_{\theta, \pi} \left[ I[y^{(i)} = c] | x^{(i)} \right]
\]

\[
\widehat{\text{count }} (y = c) = E_{\theta, \pi} \left[ \sum_i I[y^{(i)} = c] | x^{(i)} \right]
\]

Re-estimate the \( \theta \) and \( \pi \) using the new counts

• Compute the counts for each class and feature, assuming that the soft assignments from the E-step are correct

• Re-estimate \( \theta \) and \( \pi \)
The EM Algorithm: Summary

• Initialize \( \pi \) and \( \theta \)
• Repeat
  • E-step: compute soft assignments for each training sample
  • M-step: re-estimate \( \pi \) and \( \theta \) based on the new soft assignments
Why does it work?

- Intuitively, the E-step computes the best assignments under the current $\pi$ and $\theta$
- The M-step computes the best $\pi$ and $\theta$ given the current assignments
- It can be shown* that the EM algorithm optimizes an upper bound on the marginal probability of the data

*But we aren’t doing it in this class
Probability of the data

\[ P_{\pi, \theta}(x) = \prod_i P_{\pi, \theta}(x^{(i)}) = \prod_i \sum_y P_{\pi, \theta}(x^{(i)}, y) \]

• Finding \( \pi \) and \( \theta \) that maximize the probability of the data means finding a model for which the data we observe is likely
Interpreting $\pi$ and $\theta$

- We don’t know the names of labels, but for each “anonymous” label, we obtain the probability of each keyword appearing
Sample results

• $\theta_A = 0.6, \theta_B = 0.4$
• $P(\text{password}|A) = 0.5, P(\text{send}|A) = 0.6, P(\text{paper}|A) = 0.1, P(\text{password}|B) = 0.1, P(\text{send}|B) = 0.6, P(\text{paper}|B) = 0.3$
• Interpretation: label A means “spam”, label B means “not spam”
The EM Algorithm in General

- We observe the data $x$, and have latent (unobserved) data $y$. For (unknown) parameters $\theta$, we have the distribution $P(x, y|\theta)$.
- We want to learn $\theta$ using Maximum Likelihood: find the $\theta$ for which $P(x|\theta) = \sum_y P(x, y|\theta)$ is maximized.
- If we know $y$, it’s easy to find $\theta$ using Maximum likelihood.
- If we know $\theta$, it’s easy to find $P(y|x, \theta)$.
The EM Algorithm in General

• E Step: using the current $\theta$, estimate the “responsibility” of each cluster for each point

• M Step: maximize $P(\text{responsibilities}|x, \theta)$ with respect to $\theta$