Naive Bayes Classifier

(Brief Intro: to be continued in the next lecture)

Naive Bayes classifier

- Probability of class label:
  - $p(C_k)$: Constant (e.g., Bernoulli)
- Conditional probability of data given the class
  - Naive Bayes assumption: $p(\mathbf{x} \mid C_k)$ is factorized (Each coordinate of $\mathbf{x}$ is conditionally independent of other coordinates given the class label)

$$P(x_1, ..., x_M|C_k) = P(x_1|C_k) \cdots P(x_M|C_k) = \prod_{j=1}^{M} P(x_j|C_k)$$

- Classification: use Bayes rule

$$\text{(binary)} \quad P(C_1|\mathbf{x}) \quad = \quad \frac{P(C_1, \mathbf{x})}{P(\mathbf{x})} = \frac{P(C_1, \mathbf{x})}{P(C_1, \mathbf{x}) + P(C_2, \mathbf{x})}$$

Naive Bayes classifier

- When classifying, we can simply find the class C_k that maximizes $P(C_k|\mathbf{x})$ using the Bayes rule:

$$\begin{aligned}\arg\max_k P(C_k|\mathbf{x}) &= \arg\max_k P(C_k, \mathbf{x}) \\ &= \arg\max_k P(C_k)P(\mathbf{x}|C_k) \\ &= \arg\max_k P(C_k) \prod_{j=1}^{M} P(x_j|C_k)\end{aligned}$$

Naive Bayes assumption

Example: Naive Bayes for real-valued inputs

- Probability of class label:
  - $p(C_k)$: Constant (e.g., Bernoulli)
- Conditional probability of data given the class
  - Naive Bayes assumption: $P(\mathbf{x}|C_k)$ is factorized (e.g., 1D Gaussian)

$$\begin{aligned}P(x_1, \ldots, x_M|C_k) &= P(x_1|C_k) \cdots P(x_M|C_k) \\ &= \prod_{j=1}^{M} P(x_j|C_k) \\ &= \prod_{j=1}^{M} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)\end{aligned}$$

  - Note: this is equivalent to GDA with diagonal covariance!!

Comparison: Discriminative vs. Generative

- The *generative* approach is typically model-based, and it can generate synthetic data from $p(\mathbf{x}|\ C_k)$.
  - By comparing the synthetic data and real data, we get a sense of how good the generative model is.
- The *discriminative* approach will typically have fewer parameters to estimate and have less assumptions about data distribution.
  - Linear (e.g. logistic regression) v/s quadratic (e.g., Gaussian discriminant analysis) in the dimension of the input.
  - Less generative assumptions about the data (however, constructing the features may require domain knowledge)

Naive Bayes classifier

- Probability of class label: $P(C_1) = p,\ P(C_2) = 1-p$
  - $p(C_k)$: Constant (e.g., Bernoulli)
- Conditional probability of data given the class
  - Naive Bayes assumption: $p(\mathbf{x} \mid C_k)$ is factorized (Each coordinate of $\mathbf{x}$ is conditionally independent of other coordinates given the class label)

$$P(x_1, ..., x_M|C_k) = P(x_1|C_k) \cdots P(x_M|C_k) = \prod_{j=1}^{M} P(x_j|C_k)$$

- Classification: use Bayes rule

$$\text{(binary)} \quad P(C_1|\mathbf{x}) \quad = \quad \frac{P(C_1, \mathbf{x})}{P(\mathbf{x})} = \frac{P(C_1, \mathbf{x})}{P(C_1, \mathbf{x}) + P(C_2, \mathbf{x})}$$

Example: Spam mail classification

- Label: y=1 (spam), y=0 (non-spam)
- Features: $x_j \in \{0,1\}^M$

$$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow m\text{th}$$

  - x_j: *j*-th word in the mail, where M is the vocabulary size.
  - Multinomial variable (M-dimensional binary vector with only one coordinate with 1)
- Naive Bayes Assumption:
  - Given a class label y, each word in a mail is a independent multinomial variable.

Naive Bayes Spam classifier

- Model

$P(y)$

$$P(\text{spam}) = Bernoulli(\phi)$$

$P(X=\text{word}|\text{spam})$

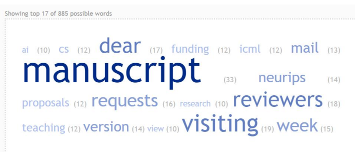
$$P(\text{word}|\text{spam}) = Multinomial(\mu_1^s, \ldots, \mu_M^s)$$
$$P(\text{word}|\text{nonspam}) = Multinomial(\mu_1^{ns}, \ldots, \mu_M^{ns})$$

$P(X=\text{word}|\text{nonspam})$

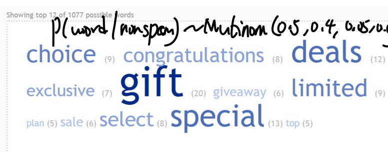
ex: M=4 ; vocabulary = {monu, piazza, gift, special}

P(word|spam) ~ Multinom(0.001, 0.001, 0.454, 0.546)

P(word|nonspam) ~ Multinom(0.5, 0.4, 0.05, 0.05)



top words from my **non-spam** emails



top words from my **spam** emails

Naive Bayes Spam classifier

- Model

$$P(\text{spam}) = Bernoulli(\phi)$$
$$P(\text{word}|\text{spam}) = Multinomial(\mu_1^s, \ldots, \mu_M^s)$$
$$P(\text{word}|\text{nonspam}) = Multinomial(\mu_1^{ns}, \ldots, \mu_M^{ns})$$

- Goal

Find ϕ, μ^s, μ^{ns} that best fits the data $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ by maximizing the joint likelihood:

$$\prod_{i=1}^{N} P(\mathbf{x}^{(i)}, y^{(i)})$$

- Joint Likelihood (joint probability of inputs/labels)
  - Note that the joint likelihood is conditioned on parameters ϕ, μ^s, μ^{ns}

13

Naive Bayes Spam classifier

- Model

$$P(\text{spam}) = Bernoulli(\phi)$$
$$P(\text{word}|\text{spam}) = Multinomial(\mu_1^s, \ldots, \mu_M^s)$$
$$P(\text{word}|\text{nonspam}) = Multinomial(\mu_1^{ns}, \ldots, \mu_M^{ns})$$

- Goal

Find ϕ, μ^s, μ^{ns} that best fits the data $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$

- Likelihood - conditioned on parameters ϕ, μ^s, μ^{ns}

$$\prod_{i=1}^{N} P(\mathbf{x}^{(i)}, y^{(i)})$$
$$= \prod_{i=1}^{N} P(\mathbf{x}^{(i)}|y^{(i)}) P(y^{(i)})$$
$$= \underbrace{\left(\boxed{\prod_{i:y^{(i)}=1} P(\mathbf{x}^{(i)}|y^{(i)}) P(y^{(i)})}\right)}_{\text{Spam}} \underbrace{\left(\prod_{i:y^{(i)}=0} P(\mathbf{x}^{(i)}|y^{(i)}) P(y^{(i)})\right)}_{\text{Non-spam}}$$

14

Naive Bayes Spam classifier

- Likelihood - spam

$$\left(\prod_{i:y^{(i)}=1} P(\mathbf{x}^{(i)}|y^{(i)}) P(y^{(i)})\right)$$

$x_k^{(i)}$ — i-th mail, k-th word

- Naive Bayes assumption:

$$P(\text{spam}) = Bernoulli(\phi)$$
$$P(\text{word}|\text{spam}) = Multinomial(\mu_1^s, \ldots, \mu_M^s)$$

$$P(\mathbf{x}^{(i)}|y^{(i)} = 1) = \prod_{k=1}^{\text{len}(\mathbf{x}^{(i)})} P(x_k^{(i)}|y^{(i)} = 1)$$
$$= \prod_{k=1}^{\text{len}(\mathbf{x}^{(i)})} \prod_{j=1}^{M} (\mu_j^s)^{\mathbb{I}(x_k^{(i)} = \text{“}j\text{”th word})}$$

找到 $x_k^{(i)}$ 是第 j^{th} word

value $= \mu_j^s$

$= \prod_{k=1}^{\text{len}(x^{(i)})} \mu_j^s$ where j is the order of $x_k^{(i)}$ in the voab

$$P(y^{(i)} = 1) = \phi$$

15

Naive Bayes Spam classifier

- Likelihood - spam (cont')

$$\left(\prod_{i:y^{(i)}=1} P\left(\mathbf{x}^{(i)} \mid y^{(i)}\right) P\left(y^{(i)}\right)\right)$$
$$= \left(\prod_{i:y^{(i)}=1}^{N} \prod_{k=1}^{\text{len}(x^{(i)})} \prod_{j=1}^{M} \left(\mu_j^s\right)^{\mathbb{I}\left(x_k^{(i)} = \text{“}j\text{”th word}\right)} \phi\right) \text{const}$$

$$\Rightarrow = \left(\prod_{i:y^{(i)}=1}^{N} \prod_{k=1}^{\text{len}(x^{(i)})} \prod_{j=1}^{M} (\mu_j^s)^{I(--)}\right) \left(\prod_{i:y^{(i)}=1}^{N} \phi\right)$$

16

Naive Bayes Spam classifier

- Likelihood - spam (cont')

$$\left(\prod_{i:y^{(i)}=1} P\left(\mathbf{x}^{(i)} \mid y^{(i)}\right) P\left(y^{(i)}\right)\right)$$
$$= \left(\prod_{i:y^{(i)}=1}^{N} \prod_{k=1}^{\text{len}(x^{(i)})} \prod_{j=1}^{M} \left(\mu_j^s\right)^{\mathbb{I}\left(x_k^{(i)} = \text{“}j\text{”th word}\right)} \phi\right)$$
$$= \left(\prod_{i:y^{(i)}=1}^{N} \prod_{k=1}^{\text{len}(x^{(i)})} \prod_{j=1}^{M} \left(\mu_j^s\right)^{\mathbb{I}\left(x_k^{(i)} = \text{“}j\text{”th word}\right)}\right) \left(\prod_{i:y^{(i)}=1}^{N} \phi\right)$$
$$= \left(\prod_{j=1}^{M} \left(\mu_j^s\right)^{\sum_{i:y^{(i)}=1}^{N} \sum_{k=1}^{\text{len}(x^{(i)})} \mathbb{I}\left(x_k^{(i)} = \text{“}j\text{”th word}\right)}\right) \left(\prod_{i:y^{(i)}=1}^{N} \phi\right)$$
$$= \left(\prod_{j=1}^{M} \left(\mu_j^s\right)^{N_j^{\text{spam}}}\right) \phi^{N^{\text{spam}}}$$

Definition:
N^{spam}: total # examples for spam
N_j^{spam}: total # of word j from the entire spam emails

19

Naive Bayes Spam classifier

- Likelihood - non-spam

$$\left(\prod_{i:y^{(i)}=1} P(\mathbf{x}^{(i)}|y^{(i)}) P(y^{(i)})\right) \left(\prod_{i:y^{(i)}=0} P(\mathbf{x}^{(i)}|y^{(i)}) P(y^{(i)})\right)$$

- Similarly for non-spam mails,

$$P(\text{spam}) = Bernoulli(\phi)$$
$$P(\text{word}|\text{nonspam}) = Multinomial(\mu_1^{ns}, \ldots, \mu_M^{ns})$$

$$P(\mathbf{x}^{(i)}|y^{(i)} = 0) = \prod_{k=1}^{\text{len}(\mathbf{x}^{(i)})} P(x_k^{(i)}|y^{(i)} = 0)$$
$$= \prod_{k=1}^{\text{len}(\mathbf{x}^{(i)})} \prod_{j=1}^{M} (\mu_j^{ns})^{I(x_k^{(i)} = \text{“}j\text{”th word})}$$
$$P(y^{(i)} = 0) = 1 - \phi$$

20

Maximum likelihood estimation

- Putting together:

$$\prod_{i=1}^{N} P(\mathbf{x}^{(i)}, y^{(i)})$$
$$= \left(\prod_{i:y^{(i)}=1} P(\mathbf{x}^{(i)}|y^{(i)}) P(y^{(i)})\right) \left(\prod_{i:y^{(i)}=0} P(\mathbf{x}^{(i)}|y^{(i)}) P(y^{(i)})\right)$$
$$= \left(\phi^{N^{spam}} \prod_{word\, j} (\mu_j^s)^{N_j^{spam}}\right) \left((1-\phi)^{N^{nonspam}} \prod_{word\, j} (\mu_j^{ns})^{N_j^{nonspam}}\right)$$

Recall:
N^{spam}: total # examples for spam
$N^{nonspam}$: total # examples for non-spam

N_j^{spam}: total # word j from the entire spam emails
$N_j^{nonspam}$: total # word j from the entire nonspam emails

22

Maximum likelihood estimation

- Putting together:

$$\prod_{i=1}^{N} P(\mathbf{x}^{(i)}, y^{(i)})$$
$$= \left(\prod_{i:y^{(i)}=1} P(\mathbf{x}^{(i)}|y^{(i)}) P(y^{(i)})\right) \left(\prod_{i:y^{(i)}=0} P(\mathbf{x}^{(i)}|y^{(i)}) P(y^{(i)})\right)$$
$$= \left(\phi^{N^{spam}} \prod_{word\, j} (\mu_j^s)^{N_j^{spam}}\right) \left((1-\phi)^{N^{nonspam}} \prod_{word\, j} (\mu_j^{ns})^{N_j^{nonspam}}\right)$$

- Joint Log-likelihood

$$\log P(\mathcal{D})$$
$$= \log \prod_{i=1}^{N} P(x^{(i)}, y^{(i)})$$
$$= N^{spam} \log \phi + \sum_{word\, j} N_j^{spam} \log \mu_j^s + N^{nonspam} \log(1-\phi) + \sum_{word\, j} N_j^{nonspam} \log \mu_j^{ns}$$

23

Maximum likelihood estimation

- Joint Log-likelihood

$$\begin{aligned} &\log P(\mathcal{D}) \\ =\ &\log \prod_{i=1}^{N} P(x^{(i)}, y^{(i)}) \\ =\ &N^{spam}\log\phi + \sum_{word\,j} N_j^{spam}\log\mu_j^s + N^{nonspam}\log(1-\phi) + \sum_{word\,j} N_j^{nonspam}\log\mu_j^{ns} \end{aligned}$$

- Maximum-likelihood
  - Take the derivative of log-likelihood w.r.t. the parameters, and set it to zero.

Maximum likelihood estimation

- From $\frac{\partial l}{\partial \phi} = \frac{1}{\phi}N^{spam} - \frac{1}{1-\phi}N^{nonspam} = 0$

  We get $\phi = \frac{N^{spam}}{N^{spam} + N^{nonspam}}$

  ML

- Removing dependent variables:

$$\sum_{word\,j=1}^{M} N_j^{spam}\log\mu_j^s = \sum_{word\,j=1}^{M-1} N_j^{spam}\log\mu_j^s + N_M^{spam}\log(1-\sum_{j=1}^{M-1}\mu_j^s)$$

$$\frac{\partial}{\partial\mu_j^s}\left(\sum_{word\,j=1}^{M} N_j^{spam}\log\mu_j^s\right) = \frac{N_j^{spam}}{\mu_j^s} - \frac{N_M^{spam}}{1-\sum_{j=1}^{M-1}\mu_j^s} = 0$$

$$\text{s.t.} \quad \sum_j \mu_j^s = 1$$

$$\sum_j \mu_j^{ns} = 1$$

Maximum likelihood estimation

- From $\frac{\partial l}{\partial \phi} = \frac{1}{\phi}N^{spam} - \frac{1}{1-\phi}N^{nonspam} = 0$

  We get $\phi = \frac{N^{spam}}{N^{spam} + N^{nonspam}}$

- Removing dependent variables:

$$\sum_{word\,j=1}^{M} N_j^{spam}\log\mu_j^s = \sum_{word\,j=1}^{M-1} N_j^{spam}\log\mu_j^s + N_M^{spam}\log(1-\sum_{j=1}^{M-1}\mu_j^s)$$

$$\frac{\partial}{\partial\mu_j^s}\left(\sum_{word\,j=1}^{M} N_j^{spam}\log\mu_j^s\right) = \frac{N_j^{spam}}{\mu_j^s} - \frac{N_M^{spam}}{1-\sum_{j=1}^{M-1}\mu_j^s} = 0$$

$$\Rightarrow \frac{N_j^{spam}}{\mu_j^s} = constant, \forall j$$

$$\Rightarrow \mu_j^s = \frac{N_j^{spam}}{\sum_j N_j^{spam}}$$

Maximum likelihood estimation

- Summary:

$$P(spam) = \phi = \frac{N^{spam}}{N^{spam} + N^{nonspam}}$$

$$P(word = j|spam) = \mu_j^s = \frac{N_j^{spam}}{\sum_j N_j^{spam}}$$

(MLE)

$$P(word = j|non-spam) = \mu_j^{ns} = \frac{N_j^{nonspam}}{\sum_j N_j^{nonspam}}$$

Recall:
N^{spam}: total # examples for spam
$N^{nonspam}$: total # examples for non-spam

N_j^{spam}: total # word j from the entire spam emails
$N_j^{nonspam}$: total # word j from the entire nonspam emails

Laplace Smoothing

- Maximum likelihood is problematic when a specific word count is 0
  - Leads to probability of 0!
- Solution: Put “imaginary” counts for each word
  - prevent zero probability estimates (overfitting)!
  - E.g.: Adding “1” as imaginary count for each word

$$P(spam) = \phi = \frac{N^{spam}}{N^{spam} + N^{nonspam}}$$

$$P(word = j|spam) = \mu_j^s = \frac{N_j^{spam}\boxed{+1}}{\sum_j N_j^{spam}\boxed{+M}}$$

$$P(word = j|non-spam) = \mu_j^{ns} = \frac{N_j^{nonspam}\boxed{+1}}{\sum_j N_j^{nonspam}\boxed{+M}}$$