Probabilistic Models in Machine Learning

Mark Johnson

September 3, 2014

MACQUARIE UNIVERSITY

Readings for this week

- NLTK book, chapter 6 section 5 "Naive Bayes classifiers"
- Manning et al, *Information to Information Retrieval*, chapter 13
  - chapter 11 has an introduction to probability theory
  - chapter 12 explains how *language models* can be used for information retrieval

Probability and statistics

- *Probability theory*
  - the mathematical theory of *random phenomena*
- *Statistics*
  - a *statistic* is a summary (usually a number or a set of numbers) that *summarise* a set of data
- A *probabilistic model* predicts how likely certain kinds of events are
  - usually a probabilistic model has one or more adjustable parameters
  - try to *estimate* the values of these parameters *from data*
  - statistical theory provides ways of *estimating the value* of such adjustable parameters

Why are probabilistic models important in machine learning?

- In the 1980s, a large number of approaches were explored to handle *uncertainly* and incomplete or *partial knowledge*
- Probability and statistics turn out to be the most useful
  - probability lets us quantify the *degree of uncertainty*
  - there's a *well-developed mathematical theory* to build on
  - it's possible to develop probabilistic models with *rich internal structure*
  - no principled conflict between linguistic theory and probabilistic models
  - but many details still have to be worked out
  - and there are lots of interesting new models to explore!

Outline

A very brief introduction to probability and statistics

Naive Bayes classifiers

Generative models and Bayesian networks

Naive Bayes for supervised document classification

Summary

Event space

- An event e is a member of the *event space* or set $\mathcal{E}$ of *possible events*
  - Example: 1 roll of a die, so $\mathcal{E} = \{1, 2, 3, 4, 5, 6\}$
- In this example $\mathcal{E}$ is small, but in many real-life applications $\mathcal{E}$ is astronomical
  - in a question-answering system $\mathcal{E}$ might be the set of *all possible question/answer pairs* (i.e., all possible pairs of sequences of words)
  - in a machine translation application $\mathcal{E}$ might be the set of all possible *source language/target language sentence pairs*

Probability distributions

- A *probability distribution* P is a real-valued function on $\mathcal{E}$ where:
  a. for each $e \in \mathcal{E}$, $\mathrm{P}(e) \geq 0$ and $\mathrm{P}(e) \leq 1$
     i.e., probabilities are *real numbers between 0 and 1*
  b. $\sum_{e \in \mathcal{E}} \mathrm{P}(e) = 1$, i.e., probabilities *sum to 1*
  - Example (cont.):

$$\begin{array}{rclcrcl} \mathrm{P}(1) & = & 0.1 & & \mathrm{P}(4) & = & 0.1 \\ \mathrm{P}(2) & = & 0.1 & & \mathrm{P}(5) & = & 0.1 \\ \mathrm{P}(3) & = & 0.1 & & \mathrm{P}(6) & = & 0.5 \end{array}$$

- The chance that the event will be realised as e is $\mathrm{P}(e)$
  - Example (cont.): the chance of rolling a 5 is $\mathrm{P}(5) = 0.1$
  - *Frequentists* interpret probability to mean "if we repeatedly sample from $\mathcal{E}$, the fraction of samples that are e is $\mathrm{P}(e)$".
  - There are many other ways of understanding probabilities.

Normalisation

- Often we'll only know what the probability of an event is proportional to, i.e., we'll know $f(e)$ and that $\mathrm{P}(e) = c\, f(e)$
  - we get $\mathrm{P}(e)$ by *normalising* f, i.e.,

$$\begin{aligned} \mathrm{P}(e) &= \frac{1}{Z} f(e), \text{where} \\ Z &= \sum_{e \in \mathcal{E}} f(e) \end{aligned}$$

- Example (cont.):
  - Suppose we're only told that the probability of a 6 is 5 times larger than the probability of the other sides. That is:

$$\begin{aligned} f(1) &= 1 & f(3) &= 1 & f(5) &= 1 \\ f(2) &= 1 & f(4) &= 1 & f(6) &= 5 \end{aligned}$$

  - Then $Z = \sum_{e \in \mathcal{E}} f(e) = 10$, so

$$\begin{aligned} \mathrm{P}(1) &= 0.1 & \mathrm{P}(3) &= 0.1 & \mathrm{P}(5) &= 0.1 \\ \mathrm{P}(2) &= 0.1 & \mathrm{P}(4) &= 0.1 & \mathrm{P}(6) &= 0.5 \end{aligned}$$

Random variables

- A *random variable* is a function defined on the event space $\mathcal{E}$
  - Example (cont.): odd is the function

$$\begin{array}{lclclclclcl} \mathrm{odd}(1) &=& \mathrm{True} & \mathrm{odd}(2) &=& \mathrm{False} & \mathrm{odd}(3) &=& \mathrm{True} \\ \mathrm{odd}(4) &=& \mathrm{False} & \mathrm{odd}(5) &=& \mathrm{True} & \mathrm{odd}(6) &=& \mathrm{False} \end{array}$$

- If Y is a random variable and $y \in \mathcal{Y}$ is one of its values, then:

$$\mathrm{P}(Y{=}y) \;=\; \sum_{\substack{e \in \mathcal{E} \\ Y(e)=y}} \mathrm{P}(e)$$

  - Example (cont.):

$$\begin{array}{rcl} \mathrm{P}(\mathrm{odd}{=}\mathrm{False}) &=& \mathrm{P}(2) + \mathrm{P}(4) + \mathrm{P}(6) \;=\; 0.7 \\ \mathrm{P}(\mathrm{odd}{=}\mathrm{True}) &=& \mathrm{P}(1) + \mathrm{P}(3) + \mathrm{P}(5) \;=\; 0.3 \end{array}$$

- A *discrete* random variable is one that ranges over a countable set
  - in this course we'll only work with discrete random variables

Joint probability distributions

- If X and Y are random variables over the same event space $\mathcal{E}$, then their *joint probability distribution* $\mathrm{P}(X{=}x, Y{=}y)$ is:

$$\mathrm{P}(X{=}x, Y{=}y) \quad = \quad \sum_{\substack{e \in \mathcal{E} \\ X(e)=x \\ Y(e)=y}} \mathrm{P}(e)$$

  - Example (cont.): Define the random variable div_3 as follows:

$$\begin{array}{rclrclrcl} \mathrm{div}_3(1) & = & 0 & \mathrm{div}_3(2) & = & 0 & \mathrm{div}_3(3) & = & 1 \\ \mathrm{div}_3(4) & = & 1 & \mathrm{div}_3(5) & = & 1 & \mathrm{div}_3(6) & = & 2 \end{array}$$

Then:

$$\begin{array}{rcl} \mathrm{P}(\mathrm{odd}{=}\mathrm{False}, \mathrm{div}_3{=}0) & = & \mathrm{P}(2) \;=\; 0.1 \\ \mathrm{P}(\mathrm{odd}{=}\mathrm{False}, \mathrm{div}_3{=}1) & = & \mathrm{P}(4) \;=\; 0.1 \\ \mathrm{P}(\mathrm{odd}{=}\mathrm{False}, \mathrm{div}_3{=}2) & = & \mathrm{P}(6) \;=\; 0.5 \\ \mathrm{P}(\mathrm{odd}{=}\mathrm{True}, \mathrm{div}_3{=}0) & = & \mathrm{P}(1) \;=\; 0.1 \\ \mathrm{P}(\mathrm{odd}{=}\mathrm{True}, \mathrm{div}_3{=}1) & = & \mathrm{P}(3) + \mathrm{P}(5) \;=\; 0.2 \\ \mathrm{P}(\mathrm{odd}{=}\mathrm{True}, \mathrm{div}_3{=}2) & = & 0 \end{array}$$

Conditional probability distributions

- The *conditional probability distribution* $\mathrm{P}(X \mid Y)$ is defined as:

$$\mathrm{P}(X{=}x \mid Y{=}y) \quad = \quad \frac{\mathrm{P}(X{=}x, Y{=}y)}{\mathrm{P}(Y{=}y)}$$

- Informally, the conditional probability distribution $\mathrm{P}(X{=}x, Y{=}y)$ is what you get when you:
  - restrict the event space to the subset $\{e : e \in \mathcal{E}, Y(e) = y\}$
  - and renormalise
- Example (cont.):

$$\begin{aligned}\mathrm{P}(\mathrm{odd}{=}\mathrm{True} \mid \mathrm{div}_3{=}1) \quad &= \quad \frac{\mathrm{P}(\mathrm{odd}{=}\mathrm{True}, \mathrm{div}_3{=}1)}{\mathrm{P}(\mathrm{div}_3{=}1)} \\ &= \quad \frac{0.2}{0.3}\end{aligned}$$

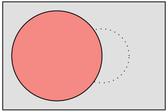
- An equivalent definition:

$$\mathrm{P}(X{=}x \mid Y{=}y)\,\mathrm{P}(Y{=}y) \quad = \quad \mathrm{P}(X{=}x, Y{=}y)$$

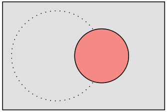
MACQUARIE UNIVERSITY

Joint and conditional probability pictures

- *Marginal distributions*

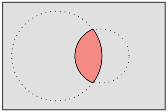


$P(X{=}\text{True})$



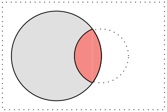
$P(Y{=}\text{True})$

- *Joint distribution*

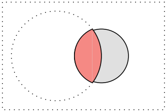


$P(X{=}\text{True}, Y{=}\text{True})$

- *Conditional distributions*



$P(Y{=}\text{True} \mid X{=}\text{True})$



$P(X{=}\text{True} \mid Y{=}\text{True})$

MACQUARIE UNIVERSITY

Bayes rule

- *Bayes rule* is a theorem that relates conditional probability distributions:

$$P(Y \mid X) = \frac{P(X \mid Y)\,P(Y)}{P(X)}$$

- Proof:

$$\begin{aligned} P(Y, X) &= P(X, Y) \\ P(Y \mid X)\,P(X) &= P(X \mid Y)\,P(Y), \text{so:} \\ P(Y \mid X) &= \frac{P(X \mid Y)\,P(Y)}{P(X)} \end{aligned}$$

- Example (cont.):

$$\begin{aligned} P(\mathrm{div}_3{=}1 \mid \mathrm{odd}{=}\mathrm{True}) &= \frac{P(\mathrm{odd}{=}\mathrm{True} \mid \mathrm{div}_3{=}1)\,P(\mathrm{div}_3{=}1)}{P(\mathrm{odd}{=}\mathrm{True})} \\ &= \frac{2/3 \cdot 3/10}{3/10} \\ &= 2/3 \end{aligned}$$

Independent random variables

- Informal explanation: two random variables X and Y are *independent* if knowing the value of X provides no information about the value of Y
- If X and Y are *independent* then:

$$\begin{aligned} \mathrm{P}(X, Y) &= \mathrm{P}(X)\,\mathrm{P}(Y), \text{ or equivalently:} \\ \mathrm{P}(Y \mid X) &= \mathrm{P}(Y) \end{aligned}$$

- Example: Suppose an event consists of a roll of a die *and* a flip of a coin, so an event might be $e = (4, \mathrm{Heads})$
  - let R be a random variable whose value is the roll of the die (so $\mathcal{R} = \{1, \ldots, 6\}$)
  - let F be a random variable whose value is the flip of the coin (so $\mathcal{F} = \{\mathrm{Heads}, \mathrm{Tails}\}$)
  - it might be reasonable to assume that R and F are independent, i.e.,

$$\mathrm{P}(R, F) = \mathrm{P}(R)\,\mathrm{P}(F)$$

MACQUARIE UNIVERSITY

Independent identically distributed random variables

- Two random variables X and Y are *independent and identically distributed* (i.i.d.) iff they are independent and *have the same distribution*

$$\mathrm{P}(X{=}v) \;=\; \mathrm{P}(Y{=}v) \text{ for all } v \in \mathcal{X} = \mathcal{Y}$$

  - i.i.d. random variables usually arise from *repeated samples from the same distribution*

I.i.d. variables example

- Suppose event consists of *10 rolls of same die*
  - an event might be $e = (2, 4, 2, 5, 1, 1, 6, 2, 1, 5)$
  - let X_j be the value of the jth roll, $j \in \{1, \ldots, 10\}$
  - if the X_j are i.i.d., then $\mathrm{P}(X_j{=}x) = \mathrm{P}(X_{j'}{=}x)$ for all $j, j' \in \{1, \ldots, 10\}$ and $x \in \{1, \ldots, 6\}$
- Let $p_x = \mathrm{P}(X_j{=}x)$ for all $x \in \{1, \ldots, 6\}$
  - p_x is the probability of rolling a x on any single roll
- Then

$$\mathrm{P}(X_1{=}x_1, X_2{=}x_2, \ldots, X_{10}{=}x_{10}) \quad = \quad p_{x_1} \cdot p_{x_2} \cdot \ldots \cdot p_{x_{10}} \;=\; \prod_{i=1}^{10} p_{x_i}$$

- suppose $\boldsymbol{p} = (0.1, 0.1, 0.1, 0.1, 0.1, 0.5)$ and $\boldsymbol{x} = (2, 4, 2, 5, 1, 1, 6, 2, 1, 5)$
- then
  $\mathrm{P}(\boldsymbol{X} = \boldsymbol{x}) = 0.1 \cdot 0.1 \cdot 0.1 \cdot 0.1 \cdot 0.1 \cdot 0.1 \cdot 0.5 \cdot 0.1 \cdot 0.1 \cdot 0.1 = 0.1^9 \cdot 0.5$

MACQUARIE UNIVERSITY

Probability of multiple i.i.d. events

- Suppose $\boldsymbol{X} = (X_1, \ldots, X_n)$ consists of n i.i.d. discrete random variables, with $\mathrm{P}(X_i{=}x) = p_x$
- Then the probability $\mathrm{P}(\boldsymbol{X}{=}\boldsymbol{x})$ of the sequence $\boldsymbol{x} = (x_1, \ldots, x_n)$ is:

$$\mathrm{P}(\boldsymbol{X}{=}\boldsymbol{x}) \;=\; p_{x_1} \cdot \ldots \cdot p_{x_n} \;=\; \prod_{i=1}^{n} p_{x_i}$$

- If n_v is the *number of times v appears in $\boldsymbol{x}$* then:

$$\mathrm{P}(\boldsymbol{X}{=}\boldsymbol{x}) \;=\; \prod_{v \in \mathcal{X}} p_v^{n_v}$$

- Example: 5 flips of a coin, with $p_{\mathrm{Tails}} = 0.8, p_{\mathrm{Heads}} = 0.2$
  - $\boldsymbol{x} = (\mathrm{Tails}, \mathrm{Heads}, \mathrm{Tails}, \mathrm{Tails}, \mathrm{Heads})$
  - so $n_{\mathrm{Tails}} = 3$ and $n_{\mathrm{Heads}} = 2$
  - so $\mathrm{P}(\boldsymbol{X}{=}\boldsymbol{x}) = 0.8^3 \cdot 0.2^2$

MACQUARIE UNIVERSITY

Statistics: estimating discrete probabilities

- Suppose we don't know $\boldsymbol{p}$, but observe a sample $\boldsymbol{x} = (x_1, \ldots, x_n)$ of n i.i.d. discrete random variables. Can we estimate $\boldsymbol{p}$?
- *Relative frequency estimator* (a.k.a. *maximum likelihood estimator*)

$$\begin{aligned} \hat{p}_v &= \frac{n_v}{n}, \quad \text{where:} \\ n_v &= \text{number of times } v \text{ appears in } (x_1, \ldots, x_n) \end{aligned}$$

  - $\boldsymbol{n}$ is a *vector of length* $m = |\mathcal{X}|$, where X is the set of values that each X_i range over
  - $\hat{\boldsymbol{p}}$ is a *vector of length* m too
  - *(estimates obtained from data are often written with hats)*
- Example:
  - data: $n = 10$ rolls of a die, so $\boldsymbol{x} = (2, 4, 2, 5, 1, 1, 6, 2, 1, 5)$
  - so $n_1 = 3, n_2 = 3, n_3 = 0, n_4 = 1, n_5 = 2, n_6 = 1$
  - so $\hat{p}_1 = 0.3, \hat{p}_2 = 0.3, \hat{p}_3 = 0, \hat{p}_4 = 0.1, \hat{p}_5 = 0.2, \hat{p}_6 = 0.1$

Maximum likelihood estimation

- *Maximum likelihood estimation* is a very general method for estimating parameter values from data
- It is provably *optimal* in many circumstances
- Given data $\boldsymbol{x}$, the *likelihood* $L(\boldsymbol{p})$ of parameters $\boldsymbol{p}$ is the *probability of the data w.r.t. the probability distribution specified by* $\boldsymbol{p}$

$$L_{\boldsymbol{x}}(\boldsymbol{p}) \quad = \quad \mathrm{P}_{\boldsymbol{p}}(\boldsymbol{x})$$

  - Example: $n = 3$ flips of a coin yield $\boldsymbol{x} = (\mathrm{Heads}, \mathrm{Tails}, \mathrm{Heads})$, so $L_{\boldsymbol{x}}(\boldsymbol{p}) = p_{\mathrm{Tails}} \cdot p_{\mathrm{Heads}}^2$

- The *maximum likelihood estimate* (MLE) $\hat{\boldsymbol{p}}$ is:

$$\hat{\boldsymbol{p}} \quad = \quad \underset{\boldsymbol{p}}{\mathrm{argmax}}\, L_{\boldsymbol{x}}(\boldsymbol{p})$$

  - Example (cont.): it's possible to show that $\hat{p}_{\mathrm{Heads}} = 2/3, \hat{p}_{\mathrm{Tails}} = 1/3$

The MLE and zero counts, and "add 1" smoothing

- MLE for a categorical distribution given i.i.d. data $\boldsymbol{x} = (x_1, \ldots, x_n)$ is:

$$\hat{p}_v = \frac{n_v}{n}, \quad \text{where:}$$
$$n_v = \text{the number of times } v \text{ appears in } \boldsymbol{x}$$

$\Rightarrow$ If $n_v = 0$ then $\hat{p}_v = 0$

- As we'll see, zero probability predictions can often cause problems
- *"Add 1" smoothing*, a.k.a. *Laplace smoothing*, estimates $\boldsymbol{p}$ as:

$$\hat{\hat{p}}_v = \frac{n_v + 1}{n + m}, \quad \text{where:}$$
$$m = |\mathcal{X}|, \quad \text{i.e., the number of values each } X_i \text{ ranges over}$$

- Example: $n = 3$ flips of a coin yield $\boldsymbol{x} = (\mathrm{Heads}, \mathrm{Tails}, \mathrm{Heads})$, so:
  - $m = |\mathcal{X}| = 2$
  - so $\hat{\hat{p}}_{\mathrm{Heads}} = 3/5$ and $\hat{\hat{p}}_{\mathrm{Tails}} = 2/5$

MACQUARIE UNIVERSITY

Outline

MACQUARIE UNIVERSITY

Using probabilistic models to build classifiers

- *Supervised classification:* given *training data* $D = ((x_1, y_1), \ldots, (x_n, y_n))$ where:
  - each x_i is a *data item* (e.g., a document)
  - and y_i is the corresponding *label*
  - predict the label y of a novel data item x
- *Probabilistic models can be used to build classifiers*
  - from D estimate a model $\widehat{\mathrm{P}}(Y \mid X)$
  - Use $\widehat{\mathrm{P}}$ to *predict the label $\hat{y}(x)$ on novel data item x*

$$\hat{y}(x) \quad = \quad \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \, \widehat{\mathrm{P}}(Y{=}y \mid X{=}x)$$

- It's possible to show that if $\widehat{\mathrm{P}}(Y|X) = \mathrm{P}(Y|X)$ (i.e., our estimated model is the true model of the data) then *this classifier has the highest accuracy possible* (i.e., Bayes optimal)

MACQUARIE UNIVERSITY

Bayesian inversion in classifiers

- Idea: use Bayes rule to "invert" conditional probability

$$\begin{aligned}\widehat{y}(x) &= \operatorname*{argmax}_{y \in \mathcal{Y}} \mathrm{P}(Y{=}y \mid X{=}x) && \text{(optimal classifier)}\\ &= \operatorname*{argmax}_{y \in \mathcal{Y}} \frac{\mathrm{P}(X{=}x \mid Y{=}y)\,\mathrm{P}(Y{=}y)}{\mathrm{P}(X{=}x)} && \textit{(Bayes rule)}\\ &= \operatorname*{argmax}_{y \in \mathcal{Y}} \mathrm{P}(X{=}x \mid Y{=}y)\,\mathrm{P}(Y{=}y) && (x \text{ is constant})\end{aligned}$$

- These equations are exact (i.e., no approximations here)
- $\mathrm{P}(Y{=}y)$ is the probability of a label y
  - if the set of possible labels $\mathcal{Y}$ is small, estimate $\mathrm{P}(Y)$ from D
  - E.g., $\widehat{\mathrm{P}}(Y{=}y) = n_y/n$, i.e., fraction of data items with label y
- $\mathrm{P}(X{=}x \mid Y{=}y)$ is the probability of data item x given label y
  - usually can't be directly estimated from training data
  - because $\mathcal{X}$ too large to observe all possible (x, y) combinations in D

Representing data items as collections of features

- Problem: set of data items $\mathcal{X}$ is *too large* treat each item atomically
- Idea: treat data item X as a *collection of features* $(F_1, \ldots, F_m)$
- Example: in the *name gender classification* problem
  - X is a name (a character sequence) and Y is its gender
  - there 2 features:
    - F_1 is the last character in X
    - F_2 is the first vowel in X
  - so if $X =$ 'Steven', then $F_1 =$ 'n' and $F_2 =$ 'e'.
- Most probabilistic classifiers use features to handle *sparse data*
- "Naive" aspect of naive Bayes: assume *features are independent* given Y

The "naive" assumption in naive Bayes classifiers

- Optimal classifier: (from before, with $X = \boldsymbol{F}$)

$$\widehat{y}(\boldsymbol{f}) \quad = \quad \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \, \mathrm{P}(\boldsymbol{F}{=}\boldsymbol{f} \mid Y{=}y) \, \mathrm{P}(Y{=}y)$$

  - $\boldsymbol{F} = (F_1, \ldots, F_m)$ is a vector of features
  - $\mathrm{P}(Y{=}y)$ is easy to estimate from D
- "Naive" assumption in naive Bayes:

$$\mathrm{P}(\boldsymbol{F}{=}\boldsymbol{f} \mid Y{=}y) \quad \approx \quad \prod_{j=1}^{m} \mathrm{P}(F_j{=}f_j \mid Y{=}y)$$

  - i.e., assume the *features F_j are independent given Y*
  - usually not true, but naive Bayes classifiers often work well
- A Naive Bayes classifier is one that uses the "naive Bayes" approximation for $\mathrm{P}(\boldsymbol{F}{=}\boldsymbol{f} \mid Y{=}y)$, so:

$$\widehat{y}(\boldsymbol{f}) \quad = \quad \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \, \mathrm{P}(Y{=}y) \prod_{j=1}^{m} \mathrm{P}(F_j{=}f_j \mid Y{=}y)$$

Naive Bayes classifier for name gender

- Data items X are *names*, labels Y are their *gender*
- 2 features: F_1 (last character) and F_2 (first vowel)
- "Naive" Bayes assumption:

$$\mathrm{P}(\boldsymbol{F}{=}\boldsymbol{f} \mid Y{=}y) \;\approxeq\; \mathrm{P}(F_1{=}f_1 \mid Y{=}y)\,\mathrm{P}(F_2{=}f_2 \mid Y{=}y)$$

- For example, if X = 'Steven', F_1 = 'n' and F_2 = 'e'. So:

$$\begin{aligned} &\mathrm{P}(\boldsymbol{F}{=}(\text{'n'}, \text{'e'}) \mid Y = \text{'male'}) \\ &\quad \approxeq \; \mathrm{P}(F_1{=}\text{'n'} \mid Y = \text{'male'})\,\mathrm{P}(F_2{=}\text{'e'} \mid Y = \text{'male'}) \end{aligned}$$

- Use Naive Bayes assumption in classifier formula to predict label $\widehat{y}$:

$$\begin{aligned} \widehat{y}(\boldsymbol{f}) &= \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, \mathrm{P}(\boldsymbol{F}{=}\boldsymbol{f} \mid Y{=}y)\,\mathrm{P}(Y{=}y) \\ &= \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, \mathrm{P}(F_1{=}f_1 \mid Y{=}y)\,\mathrm{P}(F_2{=}f_2 \mid Y{=}y)\,\mathrm{P}(Y{=}y) \end{aligned}$$

Calculating the quantities needed for NB classifier

- For Naive Bayes classifier, need to calculate:

$$\mathrm{P}(Y{=}y) \prod_{j=1}^{m} \mathrm{P}(F_j{=}f_j \mid Y{=}y)$$

- Estimate $\mathrm{P}(Y{=}y) = p_y$ as follows:

$$\begin{aligned} \hat{p}_y &= n_{Y=y}/n, \quad \text{where:} \\ n &= \text{number of data items in training data, and} \\ n_{Y=y} &= \text{number of data items with class label } y \end{aligned}$$

- Estimate $\mathrm{P}(F_j{=}f \mid Y{=}y) = q_{j,f,y}$ as follows:

$$\begin{aligned} \hat{q}_{j,f,y} &= n_{F_j=f,Y=y}/n_{Y=y}, \quad \text{where:} \\ n_{F_j=f,Y=y} &= \text{number of data items where feature } F_j \text{ has value } f \end{aligned}$$

Add-1 smoothing for NB classifiers

- If any of the probabilities in an NB classifier are zero, this causes the class to be ruled out:

$$\mathrm{P}(Y{=}y) \prod_{j=1}^{m} \mathrm{P}(F_j{=}f_j \mid Y{=}y)$$

  Add-1 smoothing can avoid this.
- Estimate $\mathrm{P}(Y{=}y) = p_y$ as follows:

$$\begin{aligned} \hat{\hat{p}}_y &= \frac{n_{Y=y} + 1}{n + |\mathcal{Y}|}, \quad \text{where:} \\ n &= \text{number of data items in training data, and} \\ n_{Y=y} &= \text{number of data items with class label } y \end{aligned}$$

- Estimate $\mathrm{P}(F_j{=}f \mid Y{=}y) = q_{j,f,y}$ as follows:

$$\begin{aligned} \hat{\hat{q}}_{j,f,y} &= \frac{n_{F_j=f,Y=y} + 1}{n_{Y=y} + |\mathcal{F}_j|}, \quad \text{where:} \\ n_{F_j=f,Y=y} &= \text{number of data items where feature } F_j \text{ has value } f \end{aligned}$$

Naive Bayes classifiers in a broader context

- The "naive" assumption of feature independence (given the label) makes naive Bayes classifiers very easy to train
- More sophisticated classifiers don't assume feature independence (e.g., logistic regression, support vector machines)
- But naive Bayes can sometimes be very competitive, even when the "naive" feature independence assumption is not true
- On very large data sets, sometimes naive Bayes is used because more sophisticated methods would be infeasible
- Many other important models also make a "naive Bayes" independence assumption (e.g., Hidden Markov Models, Topic Models).

Outline

Generative models

- A *generative model* is a probabilistic model where *the joint distribution is factored into product of conditional probability distributions*
- The naive Bayes model is a generative model:

$$\mathrm{P}(Y, F_1, \ldots, F_m) \;=\; \mathrm{P}(Y)\,\mathrm{P}(F_1 \mid Y) \ldots \mathrm{P}(F_m \mid Y)$$

- Factoring makes it possible to estimate each conditional distributions independently
  - but independence assumptions may cause model to be *badly biased*
  - the major alternative approach (called "discriminative models") couples factors via a shared normalisation factor (the "partition function")

Bias and variance in Naive Bayes

- Any distribution can factored into a product of conditional distributions:

$$\mathrm{P}(X_1, X_2, \ldots, X_n) \;=\; \mathrm{P}(X_1)\,\mathrm{P}(X_2 \mid X_1)\;\ldots\;\mathrm{P}(X_n \mid X_1, \ldots, X_{n-1})$$

- *Independence assumptions* reduce the size of the models, but may introduce bias, e.g.:

$$\mathrm{P}(X_n \mid X_1, \ldots, X_{n-1}) \;=\; \mathrm{P}(X_n \mid X_1)$$

- Example: naive Bayes:
    - How many parameters are required to represent $\mathrm{P}(Y, F_1, \ldots, F_m)$ directly.
    - How many parameters are required to represent each factor in the exact conditional factorisation:

$$\begin{aligned}&\mathrm{P}(Y, F_1, F_2 \ldots, F_m)\\ &\quad= \;\mathrm{P}(Y)\,\mathrm{P}(F_1 \mid Y)\,\mathrm{P}(F_2 \mid Y, F_1)\;\ldots\;\mathrm{P}(F_m \mid Y, F_1, \ldots, F_{m-1})\end{aligned}$$

    - We get the naive Bayes model if we assume

$$\mathrm{P}(F_j \mid Y, F_1, \ldots, F_{j-1}) \;=\; \mathrm{P}(F_j \mid Y)$$

      How many parameters does this model require?

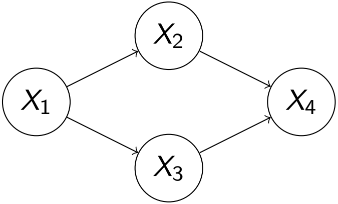
MACQUARIE UNIVERSITY

Why Bayes nets?

- *Bayesian networks* are a *graphical notation* for writing generative probabilistic models
- Every generative model can be written as:

$$\mathrm{P}(X_1, \ldots, X_m) \;=\; \prod_{i=1}^{m} \mathrm{P}(X_i \mid Parents_i)$$

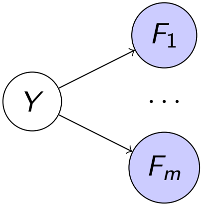
  where $Parents_i$ is a subset of $X_1, \ldots, X_{i-1}$
- Bayes nets represent this as a *directed acyclic graph* where:
  - each variable X_i is represented as a node
  - there is an edge from X_j to X_i iff $X_j \in Parents_i$
- Example:
  $\mathrm{P}(X_1, X_2, X_3, X_4) = \mathrm{P}(X_1)\,\mathrm{P}(X_2 \mid X_1)\,\mathrm{P}(X_3 \mid X_1)\,\mathrm{P}(X_4 \mid X_2, X_3)$



Naive Bayes as a Bayes net

- The naive Bayes model:

$$\mathrm{P}(Y, F_1, \ldots, F_m) = \mathrm{P}(Y)\,\mathrm{P}(F_1 \mid Y) \ldots \mathrm{P}(F_m \mid Y)$$

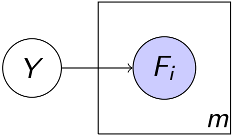


- Shaded nodes represent variables whose *values are known when inference is performed*

Plate notation

- *Plate notation* abbreviates repeated subsets of variables and dependencies
- The naive Bayes model:

$$\mathrm{P}(Y, F_1, \ldots, F_m) = \mathrm{P}(Y)\,\mathrm{P}(F_1 \mid Y) \ldots \mathrm{P}(F_m \mid Y)$$



Outline

Two ways of using NB for document classification

- Training data D consists of a *corpus of documents X where each document has a class label Y*
- Two different kinds of naive Bayes models (at least) that define features $\boldsymbol{F}$ in different ways
- *Bernoulli word features* introduce a feature F_w for each word type w
  - $F_w = \mathrm{True}$ if document X contains w
  - $F_w = \mathrm{False}$ if document X does not contain w
  - A Bernoulli random variable is one with exactly two values (this model *ignores word frequencies*)
- *Multinomial features* introduce a feature F_j for each *position j* in the document and *assumes that the F_j are i.i.d.*
  - i.e., each word in the document is generated from the *same distribution* $\mathrm{P}(W|Y)$ over words
  - (a multinomial random variable is distribution produced by repeatedly sampling from a finite distribution, e.g., rolls of a die)

Why use the naive Bayes approximation?

- Both the Bernoulli and the multinomial naive Bayes document classification models assume the naive Bayes approximation
  - the Bernoulli model assumes that *the occurence of a word is independent of the occurence of other words* given the document label Y
  - the multinomial model assumes that *the word in a particular position is independent of the other words in the document* given the document label Y
- *It's easy to show these assumptions are false*
  - they are a (structural) *bias* in our model
- But we make them because:
  - these models are *computationally and statistically tractable*, and
  - they do a fairly good job of document classification
  - recall bias/variance trade-off (NB has high bias but low variance)

Estimating a naive Bayes classifier model

- Recall the *naive Bayes decision rule*:

$$\widehat{y}(\boldsymbol{f}) \quad = \quad \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \, \mathrm{P}(Y{=}y) \prod_{j=1}^{m} \mathrm{P}(F_j{=}f_j \mid Y{=}y)$$

  where *Y is the label* and $\boldsymbol{F} = (F_1, \ldots, F_m)$ *are the features*
- The *class probabilities* $\mathrm{P}(Y)$ are estimated in the same way in both the Bernoulli and multinomial classifiers

$$\begin{aligned} \mathrm{P}(Y{=}y) &= p_y \\ \hat{p}_y &= n_y / n, \quad \text{where:} \\ n &= \text{number of documents in training data, and} \\ n_y &= \text{number of documents in training data with class label } y \end{aligned}$$

  - $\boldsymbol{n}$ and $\boldsymbol{p}$ are both *vectors of length* $|\mathcal{Y}|$
- "Add 1" smoothing can also be used to estimate $\mathrm{P}(Y{=}y)$

$$\hat{\hat{p}}_y \quad = \quad (n_y + 1)/(n + |\mathcal{Y}|)$$

MACQUARIE UNIVERSITY

Estimating a Bernoulli NB document classifier

- In a Bernoulli NB document classifier, there is *a feature F_w for each word* in the vocabulary $\mathcal{W}$
  - $F_w = \mathrm{True}$ if w is in the document, and False otherwise

$$
\begin{aligned}
\mathrm{P}(F_w{=}\mathrm{True} \mid Y{=}y) &= q_{w,y} \quad \text{for } w \in \mathcal{W}, y \in \mathcal{Y} \\
\hat{q}_{w,y} &= \frac{n_{w,y}}{n_y} \quad \text{where:} \\
n_{w,y} &= \text{the number of documents with label } y \text{ that} \\
& \text{contain word } w\text{, and} \\
n_y &= \text{the number of documents with label } y
\end{aligned}
$$

  - The n_y form a vector of length $|\mathcal{Y}|$
  - The $q_{w,y}$ and $n_{w,y}$ form *matrices* of dimensions $(|\mathcal{W}|, |\mathcal{Y}|)$
- Smoothing (e.g., "add 1" smoothing) is often *essential* for q:

$$
\hat{\hat{q}}_{w,y} = \frac{n_{w,y} + 1}{n_y + 2}
$$

MACQUARIE UNIVERSITY

Bernoulli NB document classification example (1)

D =

label	*text*
sports	run kick ball run
finance	buy sell sell
sports	kick ball

$$n = 3$$

$$n_{\text{sports}} = 2 \qquad n_{\text{finance}} = 1$$

$$\hat{\hat{p}}_{\text{sports}} = 3/5 \qquad \hat{\hat{p}}_{\text{finance}} = 2/5$$

$\boldsymbol{n}$ =

	run	*kick*	*ball*	*buy*	*sell*
sports	1	2	2	0	0
finance	0	0	0	1	1

$\hat{\hat{\boldsymbol{q}}}$ =

	run	*kick*	*ball*	*buy*	*sell*
sports	2/4	3/4	3/4	1/4	1/4
finance	1/3	1/3	1/3	2/3	2/3

MACQUARIE UNIVERSITY

Running a Bernoulli NB document classifier

- The *naive Bayes classifier decision rule*: given a document x

$$
\begin{aligned}
\hat{y}(x) &= \operatorname*{argmax}_{y \in \mathcal{Y}} \mathrm{P}(Y{=}y) \prod_{w \in \mathcal{W}} \mathrm{P}(F_w{=}f_w \mid Y{=}y) \\
&= \operatorname*{argmax}_{y \in \mathcal{Y}} p_y \left(\prod_{w \in x} q_{w,y} \right) \left(\prod_{w \in \mathcal{W} \setminus x} 1 - q_{w,y} \right) \\
&= \operatorname*{argmax}_{y \in \mathcal{Y}} \frac{n_y + 1}{n + |\mathcal{Y}|} \left(\prod_{w \in x} \frac{n_{w,y} + 1}{n_y + 2} \right) \left(\prod_{w \in \mathcal{W} \setminus x} \frac{n_y - n_{w,y} + 1}{n_y + 2} \right)
\end{aligned}
$$

- $\mathcal{W}$ is the vocabulary (set of word types)
- $q_{w,y}$ is probability of a document with label y containing word w, so $1 - q_{w,y}$ is probability of a document with label y *not* containing word w
- In the last line I used "add-1" estimates for $\boldsymbol{p}$ and $\boldsymbol{q}$

Bernoulli NB document classification example (2)

$$\hat{\hat{p}}_{\text{sports}} = 3/5 \qquad \hat{\hat{p}}_{\text{finance}} = 2/5$$

$\hat{\hat{q}}$ =

	run	kick	ball	buy	sell
sports	2/4	3/4	3/4	1/4	1/4
finance	1/3	1/3	1/3	2/3	2/3

$$x = \text{run run buy}$$

$$\begin{aligned}
\text{Score}(x, \text{sports}) &= \hat{\hat{p}}_{\text{sports}} \cdot \hat{\hat{q}}_{\text{run,sports}} \cdot \hat{\hat{q}}_{\text{buy,sports}} \\
&\quad \cdot (1 - \hat{\hat{q}}_{\text{kick,sports}}) \cdot (1 - \hat{\hat{q}}_{\text{ball,sports}}) \cdot (1 - \hat{\hat{q}}_{\text{sell,sports}}) \\
&= 3/5 \cdot 2/4 \cdot 3/4 \cdot 1/4 \cdot 1/4 \cdot 3/4 \\
&\approxeq 0.011 \\
\text{Score}(x, \text{finance}) &= \hat{\hat{p}}_{\text{finance}} \cdot \hat{\hat{q}}_{\text{run,finance}} \cdot \hat{\hat{q}}_{\text{buy,finance}} \\
&\quad \cdot (1 - \hat{\hat{q}}_{\text{kick,finance}}) \cdot (1 - \hat{\hat{q}}_{\text{ball,finance}}) \cdot (1 - \hat{\hat{q}}_{\text{sell,finance}}) \\
&= 2/5 \cdot 1/3 \cdot 2/3 \cdot 2/3 \cdot 2/3 \cdot 1/3 \\
&\approxeq 0.013 \qquad \text{so:} \\
\widehat{y}(x) &= \text{finance}
\end{aligned}$$

Avoiding floating point underflow when calculating probabilities

- If vocabulary $\mathcal{W}$ is large, *floating point arithmetic may underflow*

$\Rightarrow$ Use logarithms in these calculations to avoid underflow

$$
\begin{aligned}
\widehat{y}(x) &= \operatorname*{argmax}_{y \in \mathcal{Y}} \log \left(\mathrm{P}(Y{=}y) \prod_{w \in \mathcal{W}} \mathrm{P}(F_w{=}f_w \mid Y{=}y) \right) \\
&= \operatorname*{argmax}_{y \in \mathcal{Y}} \log\left(\mathrm{P}(Y{=}y)\right) + \sum_{w \in \mathcal{W}} \log\left(\mathrm{P}(F_w{=}f_w \mid Y{=}y)\right) \\
&= \operatorname*{argmax}_{y \in \mathcal{Y}} \log(p_y) + \left(\sum_{w \in x} \log(q_{w,y}) \right) + \left(\sum_{w \in \mathcal{W} \setminus x} \log(1 - q_{w,y}) \right)
\end{aligned}
$$

MACQUARIE UNIVERSITY

Bernoulli NB document classification example (3)

$$\hat{\hat{p}}_{\text{sports}} = 3/5 \qquad \hat{\hat{p}}_{\text{finance}} = 2/5$$

$\hat{\hat{q}}$ =

	run	*kick*	*ball*	*buy*	*sell*
sports	2/4	3/4	3/4	1/4	1/4
finance	1/3	1/3	1/3	2/3	2/3

$$x = \text{run run buy}$$

$$\begin{aligned}
\text{logScore}(x, \text{sports}) &= \log(\hat{\hat{p}}_{\text{sports}}) + \log(\hat{\hat{q}}_{\text{run,sports}}) + \log(\hat{\hat{q}}_{\text{buy,sports}}) \\
&\quad + \log(1 - \hat{\hat{q}}_{\text{kick,sports}}) + \log(1 - \hat{\hat{q}}_{\text{ball,sports}}) + \log(1 - \hat{\hat{q}}_{\text{sell,sports}} \\
&\approxeq -0.5 + -0.7 + -0.3 + -1.4 + -1.4 + -0.3 \\
&\approxeq -4.5 \\
\text{logScore}(x, \text{finance}) &= \log(\hat{\hat{p}}_{\text{finance}}) + \log(\hat{\hat{q}}_{\text{run,finance}}) + \log(\hat{\hat{q}}_{\text{buy,finance}}) \\
&\quad + \log(1 - \hat{\hat{q}}_{\text{kick,finance}}) + \log(1 - \hat{\hat{q}}_{\text{ball,finance}}) + \log(1 - \hat{\hat{q}}_{\text{sell,finan}} \\
&\approxeq -0.9 + -1.1 + -0.4 + -0.4 + -0.4 + -1.1 \\
&\approxeq -4.3 \qquad \text{so:} \\
\hat{y}(x) &= \text{finance}
\end{aligned}$$

MACQUARIE UNIVERSITY

Why smoothing is important in an NB classifier

- It's important that the estimates of $\mathrm{P}(F_j \mid Y)$ be smoothed
- Suppose w is a rare word that doesn't appear in any documents with class label y in the training data
  - $\Rightarrow$ $n_{w,y} = 0$, so $\hat{q}_{w,y} = 0$,
    i.e., our estimate of $\mathrm{P}(F_w{=}\mathrm{True} \mid Y{=}y)$ is zero
  - $\Rightarrow$ the NB classifier will never predict y if the document contains w
    - this is undesirable because it's possible w was missing from documents with label y "by chance", i.e., sparse data
- It gets worse. Suppose w is a rare word that *only* appears in documents with label y, and w' is a rare word that only appears in documents with a different label y'.
  - what happens if a document turns up containing both w and w'?
    - $\hat{q}_{w,y'} = 0$ because no document in y' contains w
    - $\hat{q}_{w',y} = 0$ because no document in y contains w'
  - $\Rightarrow$ *every class gets a score of zero*
  - $\Rightarrow$ NB cannot choose a label for document

Bernoulli NB document classification example (4)

$D =$

label	*text*
sports	run kick ball run
finance	buy sell sell
sports	kick ball

$$\hat{p}_{\text{sports}} = 2/3 \qquad \hat{p}_{\text{finance}} = 1/3$$

$\hat{\boldsymbol{q}} =$

	run	*kick*	*ball*	*buy*	*sell*
sports	1/2	2/2	2/2	0/2	0/2
finance	0/1	0/1	0/1	1/1	1/1

$$x = \text{run run buy}$$

$$\begin{aligned}
\text{Score}(x, \text{sports}) &= \hat{p}_{\text{sports}} \cdot \hat{q}_{\text{run,sports}} \cdot \hat{q}_{\text{buy,sports}} \\
&\quad \cdot (1 - \hat{q}_{\text{kick,sports}}) \cdot (1 - \hat{q}_{\text{ball,sports}}) \cdot (1 - \hat{q}_{\text{sell,sports}}) \\
&= 2/3 \cdot 1/2 \cdot 0 \cdot 0 \cdot 0 \cdot 1 \\
&= 0 \\
\text{Score}(x, \text{finance}) &= \hat{p}_{\text{finance}} \cdot \hat{q}_{\text{run,finance}} \cdot \hat{q}_{\text{buy,finance}} \\
&\quad \cdot (1 - \hat{q}_{\text{kick,finance}}) \cdot (1 - \hat{q}_{\text{ball,finance}}) \cdot (1 - \hat{q}_{\text{sell,finance}}) \\
&= 1/3 \cdot 0 \cdot 1 \cdot 1 \cdot 1 \cdot 0 \\
&= 0 \qquad \text{so:}
\end{aligned}$$

score of both labels is zero – we can't pick a winner!

Multinomial naive Bayes document classifier

- In the multinomial model, a document x with m_x words is represented as a sequence of m_x features, where:
  - *the value of feature F_j is word w_j at position j*
- Formula for multinomial NB document classifier:

$$
\begin{aligned}
\widehat{y}(x) &= \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \, \mathrm{P}(Y{=}y) \prod_{j=1}^{m_x} \mathrm{P}(F_j{=}w_j \mid Y{=}y), \quad \text{where:} \\
\mathrm{P}(F_j{=}w \mid Y{=}y) &= q'_{w,y} \\
\hat{q}'_{w,y} &= \frac{n'_{w,y}}{n'_y} \\
n'_y &= \text{total number of words in all documents labelled } y \\
n'_{w,y} &= \text{number of occurences of } w \text{ in documents labelled } y
\end{aligned}
$$

Smoothing in a multinomial NB document classifier

- Smoothing $\mathrm{P}(F_j|Y)$ is important in a multinomial NB document classifier (just as in a Bernoulli NB document classifier)
- "Add 1" smoothing for multinomial NB document classifier:

$$\begin{aligned}
\widehat{y}(x) &= \operatorname*{argmax}_{y \in \mathcal{Y}} \mathrm{P}(Y{=}y) \prod_{j=1}^{m_x} \mathrm{P}(F_j{=}w_j \mid Y{=}y), \quad \text{where:} \\
\mathrm{P}(F_j{=}w \mid Y{=}y) &= q'_{w,y} \\
\hat{\hat{q}}'_{w,y} &= \frac{n'_{w,y} + 1}{n'_y + |\mathcal{W}|} \\
n'_y &= \text{total number of words in all documents labelled } y \\
n'_{w,y} &= \text{number of occurences of } w \text{ in documents labelled } y \\
\mathcal{W} &= \text{vocabulary used in classifier}
\end{aligned}$$

Multinomial NB document classification example (1)

$D =$

label	*text*
sports	run kick ball run
finance	buy sell sell
sports	kick ball

$$n = 3 \qquad |\mathcal{W}| = 5$$

$$n'_{\text{sports}} = 6 \qquad n'_{\text{finance}} = 3$$

$$\hat{\hat{p}}_{\text{sports}} = 3/5 \qquad \hat{\hat{p}}_{\text{finance}} = 2/5$$

$\boldsymbol{n}' =$

	run	*kick*	*ball*	*buy*	*sell*
sports	2	2	2	0	0
finance	0	0	0	1	2

$\hat{\hat{\boldsymbol{q}}} =$

	run	*kick*	*ball*	*buy*	*sell*
sports	3/11	3/11	3/11	1/11	1/11
finance	1/8	1/8	1/8	2/8	3/8

Multinomial NB document classification example (2)

$$\hat{\hat{p}}_{\text{sports}} = 3/5 \qquad \hat{\hat{p}}_{\text{finance}} = 2/5$$

$\hat{\hat{q}}$ =

	run	kick	ball	buy	sell
sports	3/11	3/11	3/11	1/11	1/11
finance	1/8	1/8	1/8	2/8	3/8

$$\begin{aligned}
x &= \text{run run buy} \\
\text{Score}(x, \text{sports}) &= \hat{\hat{p}}_{\text{sports}} \cdot \hat{\hat{q}}'_{\text{run,sports}} \cdot \hat{\hat{q}}'_{\text{run,sports}} \cdot \hat{\hat{q}}'_{\text{buy,sports}} \\
&= 3/5 \cdot 3/11 \cdot 3/11 \cdot 1/11 \\
&\approxeq 0.004 \\
\text{Score}(x, \text{finance}) &= \hat{\hat{p}}_{\text{finance}} \cdot \hat{\hat{q}}'_{\text{run,finance}} \cdot \hat{\hat{q}}'_{\text{run,finance}} \cdot \hat{\hat{q}}'_{\text{buy,finance}} \\
&= 2/5 \cdot 1/8 \cdot 1/8 \cdot 2/8 \\
&\approxeq 0.002 \qquad \text{so:} \\
\widehat{y}(x) &= \text{sports}
\end{aligned}$$

Avoiding floating point underflow when calculating probabilities

- Unless the documents are very short, the probabilities will underflow floating-point calculations
  - calculate log probabilities instead of probabilities

$$
\begin{aligned}
\widehat{y}(x) &= \operatorname*{argmax}_{y \in \mathcal{Y}} \log(\mathrm{P}(Y{=}y)) + \sum_{j=1}^{m_x} \log\left(\mathrm{P}(F_j{=}w_j \mid Y{=}y)\right) \\
&= \operatorname*{argmax}_{y \in \mathcal{Y}} \log(p_y) + \sum_{j=1}^{m_x} \log(q'_{w_j,y})
\end{aligned}
$$

MACQUARIE UNIVERSITY

More efficient computation in multinomial NB classifier

- Suppose instead of representing a document by features $\boldsymbol{F}$, where the value of feature F_j is the word w_j at position j,
  we represent a document by features $\boldsymbol{G}$, where
  - there is a feature G_w for each $w \in \mathcal{W}$, and
  - the value g_w of G_w is *the number of times w appears in the document*
- Then it's easy to show that:

$$\sum_{j=1}^{m_x} \log(q'_{w_j,y}) \quad = \quad \sum_{w \in \mathcal{W}} g_w \log(q'_{w_j,y})$$

- This means the multinomial NB classifier can be computed as:

$$\widehat{y}(x) \quad = \quad \operatorname*{argmax}_{y \in \mathcal{Y}} \log(p_y) + \sum_{w \in \mathcal{W}} g_w \log(q'_{w_j,y})$$

- This saves time *if the document is represented as a vector of word-count pairs*

MACQUARIE UNIVERSITY

Multinomial NB document classification example (3)

$$\hat{\hat{p}}_{\text{sports}} = 3/5 \qquad \hat{\hat{p}}_{\text{finance}} = 2/5$$

$\hat{\hat{q}} =$

	run	*kick*	*ball*	*buy*	*sell*
sports	3/11	3/11	3/11	1/11	1/11
finance	1/8	1/8	1/8	2/8	3/8

$$x = \text{run run buy, so } g_{\text{run}} = 2 \text{ and } g_{\text{buy}} = 1$$

$$\begin{aligned}
\text{logScore}(x, \text{sports}) &= \log(\hat{\hat{p}}_{\text{sports}}) + g_{\text{run}} \log(\hat{\hat{q}}'_{\text{run,sports}}) + g_{\text{buy}} \log(\hat{\hat{q}}'_{\text{buy,spo}} \\
&\approxeq -0.5 + 2 \cdot -1.3 + -2.4 \\
&\approxeq -5.5 \\
\text{logScore}(x, \text{finance}) &= \log(\hat{\hat{p}}_{\text{finance}}) + g_{\text{run}} \log(\hat{\hat{q}}'_{\text{run,finance}}) + g_{\text{buy}} \log(\hat{\hat{q}}'_{\text{buy,fi}} \\
&= -0.9 + 2 \cdot -2.0 + -1.4 \\
&= -6.5 \\
\hat{y}(x) &= \text{sports}
\end{aligned}$$

Outline

MACQUARIE UNIVERSITY

Summary

- Probabilities give us a way to *quantify uncertainty*
  - machine learning involves combining weak or uncertain information from many sources
- *Conditional probabilities* describe the probability of one event given that another event occurs
- Two random variables are *independent* if knowing the value of one provides no information about the value of the other
- *Bayes rule* enables us to *invert* conditional probability distributions
- A *naive Bayes model*:
  - uses Bayes rule to define the probability of the class label in terms of the probability of the features given the class label
  - assumes that the probability of each feature is independent given the class label
- The independence assumption makes naive Bayes classifiers very easy to train
  - more sophisticated classifiers (e.g., logistic regression, support vector machines) don't make this independence assumption