Bayes Classifier, MAP, Naïve Bayes

Contents

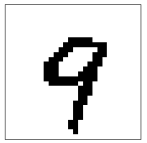
- MAP classification
  - Naïve Bayes

Reminder

- Next Tuesday – Review for Exam 1
- Next Thursday – Exam 1

Big Picture

- machine learning is about **learning functions from data**
- the inputs to the functions are called **features**

$$\boldsymbol{x} = \quad$$ 

- the true output is called a label

$$y \in \{0, 1, 2, \ldots, 9\}$$

- labeled examples of features make up *training* and *test* data

$$\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$$

- useful characterization:

$$(\boldsymbol{x}_i, y_i) \overset{i.i.d.}{\sim} p(\boldsymbol{x}, y)$$

- if we have a good approximation of $p(\boldsymbol{x}, y)$, we can design a good function for classification:

$$\widehat{y} = \arg\max_{y} p(y|\boldsymbol{x})$$

Maximum A-posteriori Probability (MAP)

- simple idea: given the data or features $\boldsymbol{x}$, what is the most probable class y?
- the *posterior* distribution $p(y|\boldsymbol{x})$ answers this

$$\widehat{y} = f(\boldsymbol{x}) = \arg\max_y p(y|\boldsymbol{x}) \qquad (f(x) \text{ is a function, not a pdf})$$

- in words: MAP picks the most probable class given the observation $\boldsymbol{x}$
- example: $\boldsymbol{x} =$ 

$$f(\boldsymbol{x}) = \arg\max_{i=0,\ldots,9} p(y = i|\boldsymbol{x} = \text{  })$$

$$p(y = 0|\boldsymbol{x} = \text{  }) =$$
$$p(y = 1|\boldsymbol{x} = \text{  }) =$$
$$\vdots$$
$$p(y = 9|\boldsymbol{x} = \text{  }) =$$

- challenge: estimating $p(y|\boldsymbol{x})$ from data.

Example – pear/apple classifier

x is top to bottom ratio

$y = 0$ if pear, $y = 1$ if apple

$x \in \{0.5, 1, 2\}$ and $y \in \{0,1\}$

- x is discrete.

$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$

counts:

		x = 0.5	1	2
y	0	10	30	0
	1	0	50	10

$$p(\boldsymbol{x}, y) = \frac{\sum_{i=1}^n \mathbb{I}_{\{\boldsymbol{x}_i = \boldsymbol{x}, y_i = y\}}}{n}$$

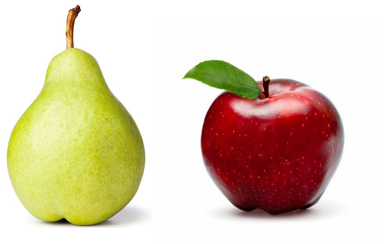
- classification: given x, what is the best guess for y?
- MAP rule:

$$\widehat{y} = \arg\max_y p(y|x)$$

$p(x, y)$

		x = 0.5	1	2
y	0	0.1	0.3	0
	1	0	0.5	0.1

Example – pear/apple classifier



x is top to bottom ratio

$y = 0$ if pear, $y = 1$ if apple

- new fruit with $x = 0.5$?

$p(x,y)$		x		
		0.5	1	2
y	0	0.1	0.3	0
	1	0	0.5	0.1

- new test fruit with $x = 1$?

- MAP rule:

$$\widehat{y} = \arg\max_y p(y|x)$$

- new test fruit with $x = 2$?

Maximum-a-posteriori (MAP)

- MAP estimate (Bayes optimal classification rule):

- x, y discrete

$$\widehat{y} = \arg\max_y p(y|\boldsymbol{x})$$

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x})}$$

$$\widehat{y} = \arg\max_y \frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x})}$$

$$\widehat{y} = \arg\max_y p(\boldsymbol{x}|y)p(y)$$

- y discrete, x continuous

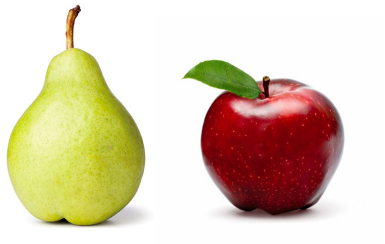
$$\widehat{y} = \arg\max_y p(y|\boldsymbol{x})$$

$$p(y|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|y)p(y)}{f(\boldsymbol{x})}$$

$$\widehat{y} = \arg\max_y \frac{f(\boldsymbol{x}|y)p(y)}{f(\boldsymbol{x})}$$

$$\widehat{y} = \arg\max_y f(\boldsymbol{x}|y)p(y)$$

Example – pear/apple classifier



x is top to bottom ratio

$y = 0$ if pear, $y = 1$ if apple

x is continuous

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$$

$$p(y = 0) = p(y = 1) = 0.5$$

- classification: given x, what is the best guess for y?
- MAP rule:

$$\widehat{y} = \arg\max_y p(y|x)$$

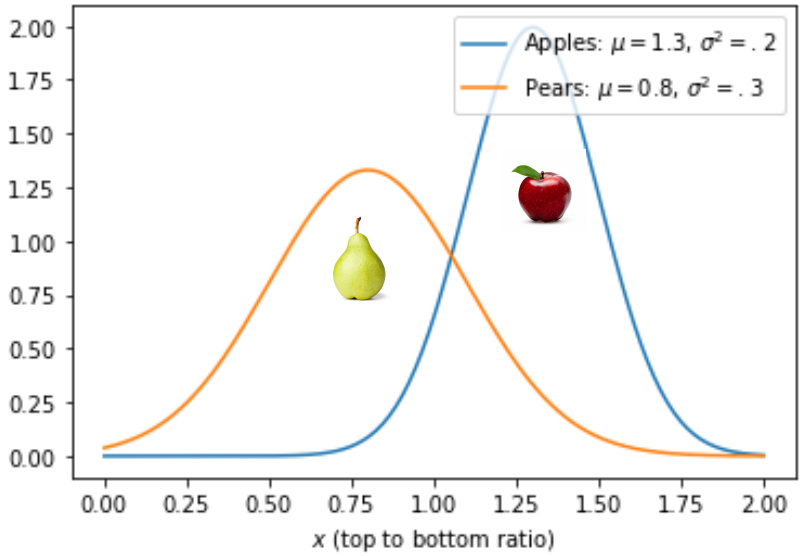
$$\widehat{y} = \arg\max_y f(x|y)p(y)$$

$$X|Y = 1 \sim \mathcal{N}(1.3, 0.2)$$

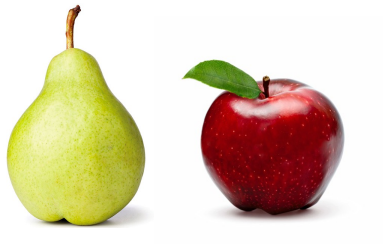
$$f(x|y = 1) = \frac{1}{\sqrt{2\pi(0.2)^2}} e^{-\frac{(x-1.3)^2}{2(0.2)^2}}$$

$$X|Y = 0 \sim \mathcal{N}(0.8, 0.3)$$

$$f(x|y = 0) = \frac{1}{\sqrt{2\pi(0.3)^2}} e^{-\frac{(x-0.8)^2}{2(0.3)^2}}$$



Example – pear/apple classifier



x is top to bottom ratio

$y = 0$ if pear, $y = 1$ if apple

$p(y = 0) = p(y = 1) = 0.5$

- MAP rule:

$$\widehat{y} = \arg\max_y p(y|x)$$

$$\widehat{y} = \arg\max_y f(x|y)p(y)$$

- if

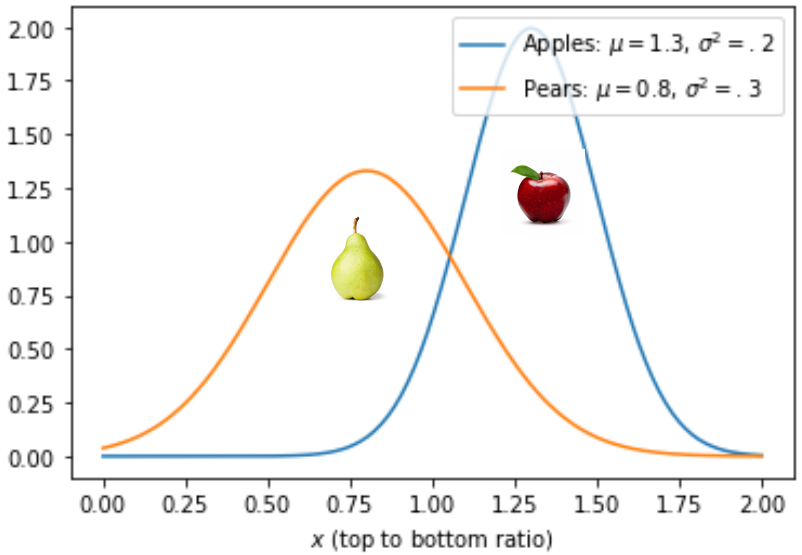
$$f(x|y = 1)p(y = 1) \geq f(x|y = 0)p(y = 0)$$

then $\widehat{y} = 1$ else $\widehat{y} = 0$.

- if

$$\frac{1}{\sqrt{2\pi(0.2)^2}} e^{-\frac{(x-1.3)^2}{2(0.2)^2}} \geq \frac{1}{\sqrt{2\pi(0.3)^2}} e^{-\frac{(x-0.8)^2}{2(0.3)^2}}$$

then $\widehat{y} = 1$ else $\widehat{y} = 0$.



- new test fruit with $x = 1.2$:

Learning in Low Dimensions

- how do we estimate $p(\boldsymbol{x}|y)$ from examples?

$$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$$

- if $\boldsymbol{x}$ low dimensional (i.e, apples vs. pears), no problem.
  - continuous: approximate it with a suitable distribution (i.e, Gaussian)
  - discrete: histogram estimate

$$p(\boldsymbol{x}|y) = \frac{\sum_{i=1}^n \mathbb{I}_{\{\boldsymbol{x}_i=\boldsymbol{x}, y_i=y\}}}{\sum_{i=1}^n \mathbb{I}_{\{y_i=y\}}}$$

$$p(\boldsymbol{x}, y) = \frac{\sum_{i=1}^n \mathbb{I}_{\{\boldsymbol{x}_i=\boldsymbol{x}, y_i=y\}}}{n}$$

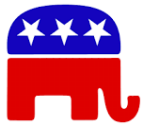
Example: Digital Advertising and Tracking Data

- example - user visited: cars.com, espn.com, hiking.com, stackexchange.com, ...

$$\boldsymbol{x} \in \mathbb{R}^{100{,}000} \qquad \mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$$

- classification: given x, is user a democrat or a republican?

$y = 0$ if democrat



$y = 1$ if republican

- MAP rule:

$$\widehat{y} = \arg\max_{y} p(y|\boldsymbol{x})$$

$$\widehat{y} = \arg\max_{y} p(\boldsymbol{x}|y)p(y)$$

Shortcomings ...

- how do we estimate $p(\boldsymbol{x}|y)$ from examples?

$$p(\boldsymbol{x}|y) = \frac{\sum_{i=1}^{n} \mathbb{I}_{\{\boldsymbol{x}_i=\boldsymbol{x}, y_i=y\}}}{\sum_{i=1}^{n} \mathbb{I}_{\{y_i=y\}}}$$

- what if $\boldsymbol{x}$ is a 28×28 black and white image?

$x =$ 

  - $2^{28\times28}$ possible values for $\boldsymbol{x}$
  - MNIST: 60,000 examples

- what if $\boldsymbol{x}$ browsing data, i.e, $\boldsymbol{x} \in \{0,1\}^{100{,}000}$?

  - $2^{100{,}000}$ possible values for $\boldsymbol{x}$
  - training data: 50,000 examples at most

- nearly all the values $\boldsymbol{x}$ won't appear in training data

$$p(\boldsymbol{x}, y) = \frac{\sum_{i=1}^{n} \mathbb{I}_{\{\boldsymbol{x}_i=\boldsymbol{x}, y_i=y\}}}{n}$$

Naive Bayes (Assume *Conditional* Independence)

- how can we estimate $p(\boldsymbol{x}|y)$ when $\boldsymbol{x}$ is high dimensional and we have limited training data?

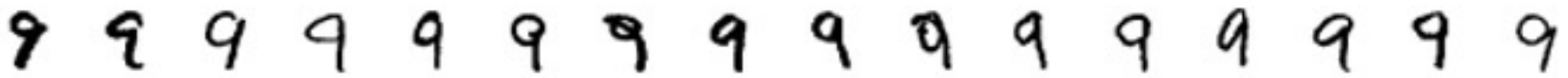
- naïve Bayes: features $x_1, x_2 \ldots$ are independent.

$$p(\boldsymbol{x}|y) \approx p(x_1|y) \times p(x_2|y) \ldots p(x_n|y) = \prod_{j=1}^{n} p(x_j|y)$$

- much easier to estimate marginals $p(x_j|y)$ than joint $p(\boldsymbol{x}|y)$

$$p(x_j|y) = \frac{\sum_{i=1}^{n} \mathbb{I}\{[\boldsymbol{x}_i]_j = x_j, y_i = y\}}{\sum_{i=1}^{n} \mathbb{I}\{y_i = y\}}$$

- MNIST: percentage of times pixel j is black (or white) when class is $y = 9$.

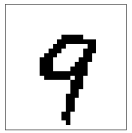


$$p(\boldsymbol{x}) \approx p(x_1) \times p(x_2) \times \cdots \times p(x_n)$$

- last step - use MAP:

$$\widehat{y} = \arg\max_y p(y|\boldsymbol{x}) = \arg\max_y \prod_j p(x_j|y)$$

Shortcomings of NB

- independence assumption is often really bad $x =$ 
- example: MNIST even conditioned on y, pixel values are highly correlated!
- still works reasonably well for classification, in spite of this shortcoming
- sometimes there are still no examples of a particular value
  - simple solution: add a 1 to both numerator and denominator of

$$p(x_j|y) = \frac{\sum_{i=1}^{n} \mathbb{I}\{[\boldsymbol{x}_i]_j = x_j\} + 1}{n+1}$$

Laplace smoothing

  - more precisely, add 1 to numerator and k to denominator, where $|\mathcal{X}| = k$
- numerically, better to work in with $\log p(\boldsymbol{x})$

$$\log p(\boldsymbol{x}) \approx \log\left(\prod_{j=1}^{n} p(x_j)\right) = \sum_{j=1}^{n} \log(p(x_j))$$

MAP is Bayes Optimal

- simple idea: given the data or features x, what is the most likely class y?
- MAP estimate:

$$\widehat{y} = \arg\max_y p(y|\boldsymbol{x})$$

- **claim**: MAP minimizes risk:
  - *risk* is the expected *loss*
  - misclassification loss: $\ell(\widehat{y}, y) = \mathbb{1}_{\{\widehat{y} \neq y\}}$

$$\ell(\widehat{y}, y) = \begin{cases} \widehat{y} \neq y & 1 \\ \text{else} & 0 \end{cases}$$

$$\begin{aligned} E[\ell(\widehat{y}, y)] &= E[\mathbb{1}_{\{\widehat{y} \neq y\}}] \\ &= \mathbb{P}(\widehat{y} \neq y) \\ &= 1 - \mathbb{P}(\widehat{y} = y) \\ &= 1 - \sum_{\boldsymbol{x}} p(x)\mathbb{P}(\widehat{y} = y|\boldsymbol{x}) \\ &\geq 1 - \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \max_y p(y|\boldsymbol{x}) \end{aligned}$$

where the last holds with equality if and only if MAP

$$\mathbb{P}(\widehat{y} = y|\boldsymbol{x}) = \max_y p(y|\boldsymbol{x})$$

$$\mathbb{P}(\widehat{y} = y|\boldsymbol{x}) \leq \max_y p(y|\boldsymbol{x}) \text{ for any classifier } \widehat{y}$$