Classifiers

Marisa Eisenberg

Machine Learning

- Focused on making quantitative inferences & predictions based on data

- Blends: statistics, linear algebra, artificial intelligence, signal processing

https://xkcd.com/1838/

Supervised and unsupervised learning

- **Unsupervised learning**
  - Often less focused on prediction, more focused on inferring structure from the data
  - Clustering, density estimation, dimension reduction
- **Supervised learning**
  - Given a training data set (where the outputs are known), predict the outputs for new data
  - Classification problems, regression problems
  - Goal is generalization of an input-output relationship

Supervised learning

$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$$

Input data vector
vector of features/attributes/predictors/ covariates

Training data
$(\vec{x}_1 y_1), \ldots, (\vec{x}_n y_n)$ input-output pairs - i.e. data vectors ($\vec{x}_i$) with labels (y_i)

Goal: given new input data $\vec{x}_i$, predict the output y_i

Classification

- A form of supervised learning
- Starts with a known set of classes
- Given training data that is labeled with which class it belongs to, predict which class new (unlabeled) data belongs to

Puppy or Bagel?

Labradoodle or Fried Chicken?

Chihuahua or Muffin?

Shar-Pei or Towel?

Shiba Inu or Marshmallow?

Parrot or Guacamole?

Classification

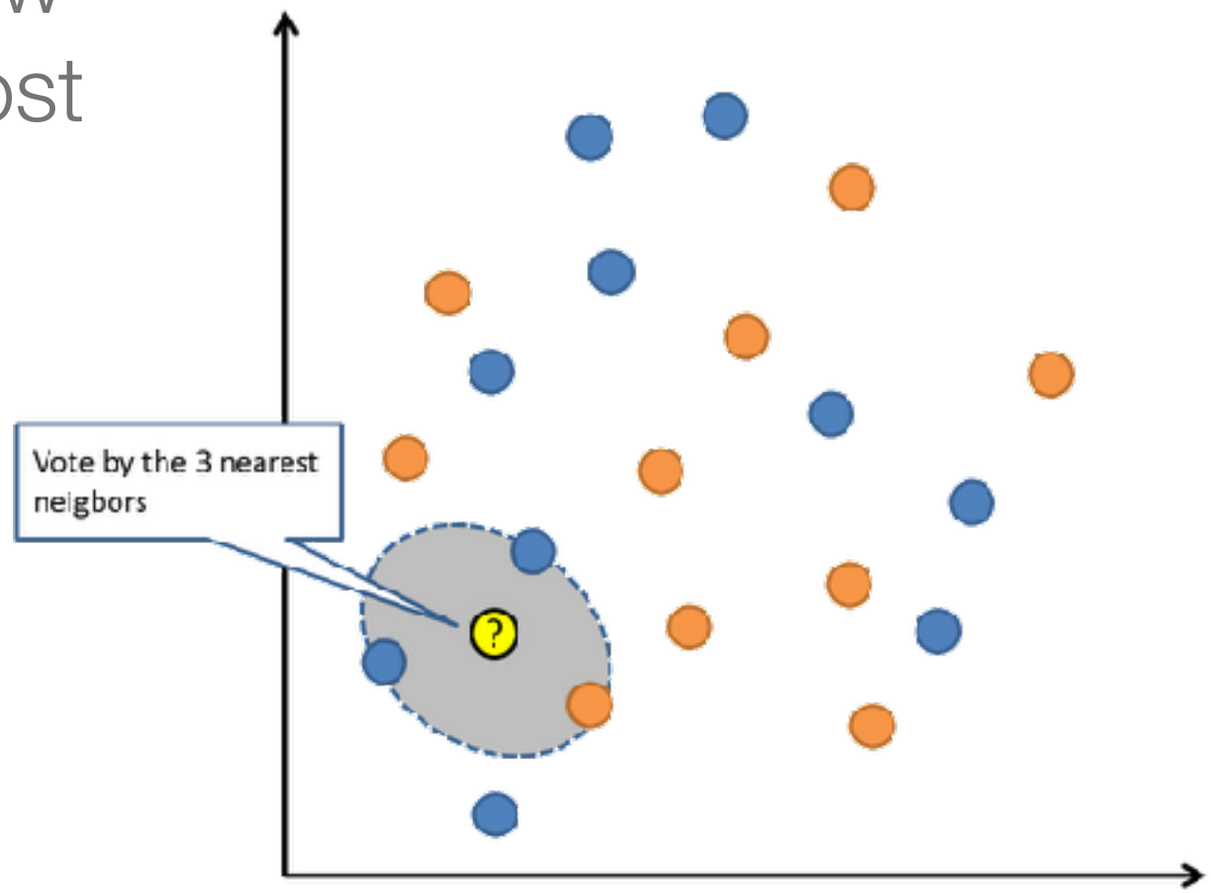
- Many (many!) different kinds
- Based on a distribution model or not
- Linear or nonlinear
- ‘Lazy’ vs ‘eager’ learners
  - Lazy learners (related: instance-based learning) load all the training data and then use the training data when new inputs come in to decide (often fast to train, but slow to operate on new data)
  - Eager learners build the classifier on the training data first, then just apply the classifier (e.g. some model) on new input data (often slow to train, but fast on new data)

Classification

- K-nearest-neighbors
- Bayes classifier-based methods
  - Naive Bayes
  - Linear discriminant analysis
  - Logistic regression
- Support vector machines
- Decision trees - relatedly: random forests
- Neural networks

k-Nearest-Neighbors (kNN)

- Decide output (class) for new data point based on the most common class among the nearest k neighbors in the training set



- Distance can be Euclidean, Hamming distance, etc.

- Weighted kNN - weight the nearest neighbors by their distance (e.g. 1/d)

https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623

k-Nearest-Neighbors (kNN)

- No explicit training phase, but the k nearest neighbors can be viewed as the training data
- Lazy learner/instance-based learner

Bayes Classifiers

- How to think about classifiers probabilistically?

- Suppose we have inputs and outputs:

$$X \in \mathbb{R}^d \qquad\qquad Y \in \{1, \ldots, K\}$$

- The joint distribution P_{XY} can be broken down:

$$P_{XY} = P_{X|Y} P_Y$$

Conditional distribution

Marginal distribution of Y (prior class distribution)

$$P_{XY} = P_{Y|X} P_X$$

Marginal of X

Posterior of Y given X

Bayes classifiers

- What is a classifier? A function $f : \mathbb{R}^d \to \{1, \ldots, K\}$ that takes input data and returns an output (class)

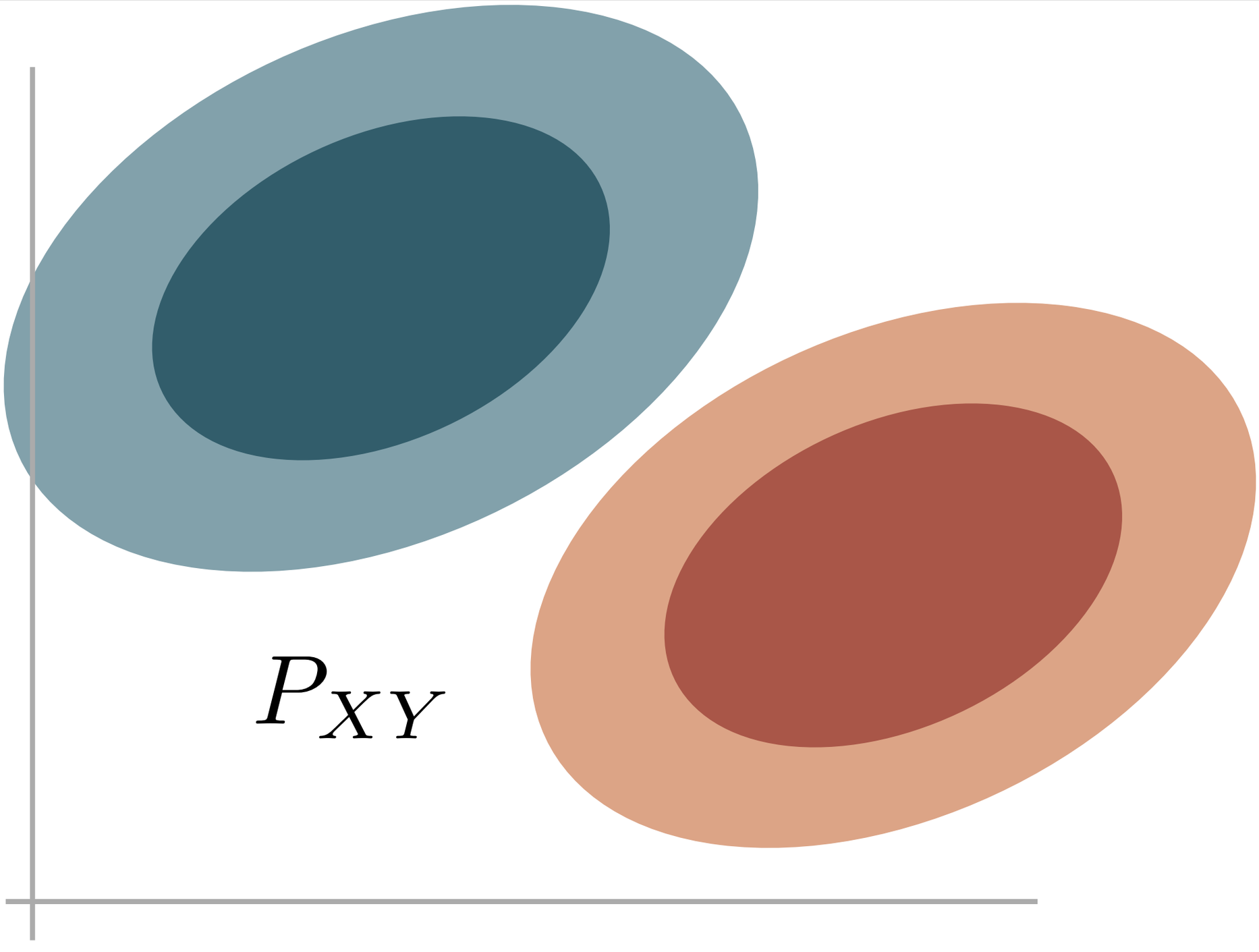
- How to choose the best classifier?

- One way is to minimize the classifier error:

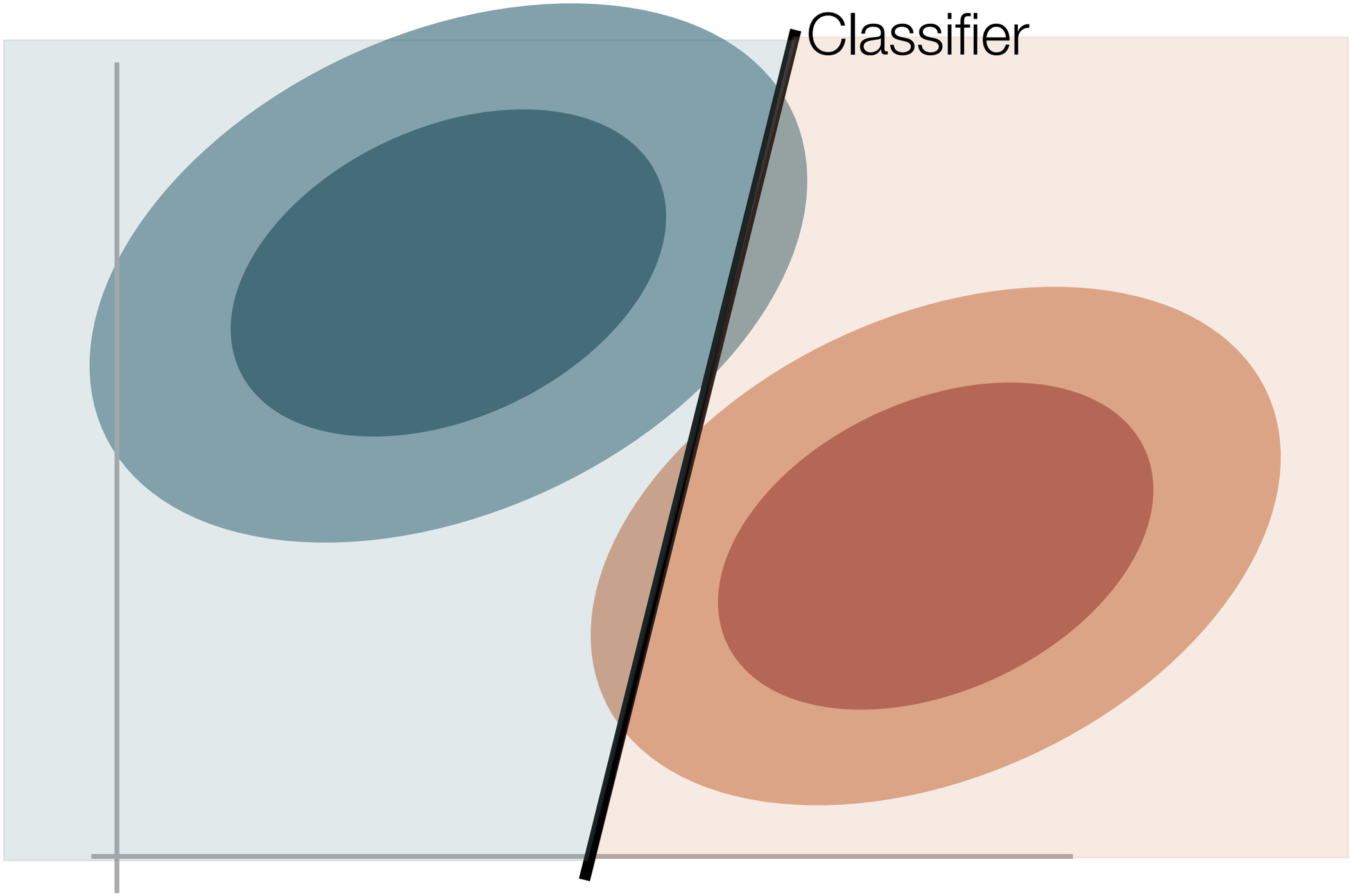
$$R(f) = P_{XY}(f(x) \neq Y)$$

- This is the part of the distribution where the input data does not match what the classifier predicts—i.e. this is the probability of the classifier being wrong

Bayes classifiers



Bayes classifiers



Bayes classifiers

- If R(f) is minimized for some f*, then f* is called a Bayes classifier

- $f^*(x) = \underset{y=1,\ldots,K}{\arg\max}\, P(x, y)$

$$f^*(x) = \underset{y=1,\ldots,K}{\arg\max}\, P(x|y)P(y)$$

$$f^*(x) = \underset{y=1,\ldots,K}{\arg\max}\, P(y|x)$$

- Many classifiers fill in these probabilities using different assumptions

Naive Bayes classifier

- Bayes classifier that assumes independence on the different input features x = (x1, …, xd) (i.e. it assumes x1, …, xd are independent)

- Strong assumption! But makes a lot of things easy

- Useful for predictors that have a finite range/categorical

- In particular, often used for text data! (like twitter data, also spam filters, etc.)

- Eager learner but relatively fast

Why is independence so useful?

- $f^*(x) = \underset{y=1,\ldots,K}{\arg\max}\, P(x|y)P(y)$

- How to calculate $P(x|y)$ when $x = (x_1, \ldots, x_d)$?

  $P(x|y) = P(x_1|x_2, \ldots, x_d, y)P(x_2|x_3, \ldots, x_d, y) \ldots P(x_d|y)$

- But because of independence, we can write:

  $P(x|y) = P(x_1|y) \ldots P(x_d|y)$

- Much simpler! We can estimate these from the training data

Naive Bayes classifier with text data

- Suppose we have a set of documents, with a subset that we have hand categorized (e.g. pro- or anti-vax, or by topic area, etc.)

- ‘Bag of words’ approach, the feature list is all words in all documents

- Each document is given by a vector $x = (x_1, \ldots, x_d)$, where x1, …, xd are 0 or 1 depending on if a word is present

Naive Bayes classifier with text data

- From the training data we can estimate

$$P(x|y)P(y) = P(x_1|y) \dots P(x_d|y)P(y)$$

- And choose the classifier that maxes this

Example

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes

https://www.geeksforgeeks.org/naive-bayes-classifiers/

Example

Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		**P(Yes)/P(No)**
Yes	9	9/14
No	5	5/14
Total	14	100%

Example

```
today = (Sunny, Hot, Normal, False)
```

$$P(SunnyOutlook|Yes)P(HotTemperature|Yes)P(NormalHumidity|Yes)P(NoWind|Yes)P(Yes)$$

$$\approx 0.0141$$

$$P(SunnyOutlook|No)P(HotTemperature|No)P(NormalHumidity|No)P(NoWind|No)P(No)$$

$$\approx 0.0068$$

So our classifier would say that today we play golf!

Other variations

- Gaussian naive Bayes assumes Gaussian distributions for the features (often used when working with continuous variables)

- Multinomial naive Bayes uses word frequency rather than just yes/no

- And many others