CS 440: Introduction to Artificial Intelligence

Lecture 16

Matthew Stone

October 28, 2015

Supervised categorization—Recap

- Infer category of real-world object from features
- Start from examples
- Learn decision boundary
- Apply learned rule to new cases

Understanding classification via probability—Recap

We want to decide the most likely category

- Compute $P(C = c_1 | O = o)$
- Compute $P(C = c_2 | O = o)$, etc.
- Pick whichever one is the largest

Allows us to describe the optimum decision boundary

Naive Bayes assumption—Recap

- Ignore certain kinds of interactions in world
- Lets you use same data to learn multiple relationships
- Mathematically:

$$P(F_i|C) = P(F_i|C, F_1 \ldots F_{i-1})$$

- As a result:

$$P(F_1 \ldots F_n|C) = P(F_1|C)P(F_2|C) \ldots P(F_n|C)$$

Linear Classifiers–Recap

Equation (2D):

$$ax + by > 1$$

Or:

$$w_1 x_1 + w_2 x_2 > 1$$

Or (any number of dimensions):

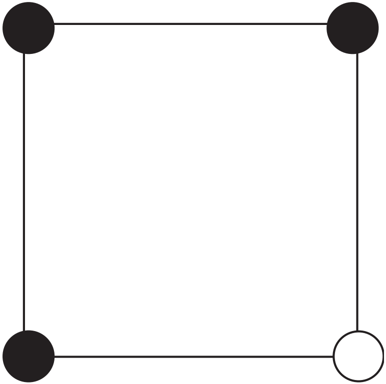
$$w \cdot x > 1$$

Interesting result

Discrete Naive Bayes models always have a single linear decision boundary

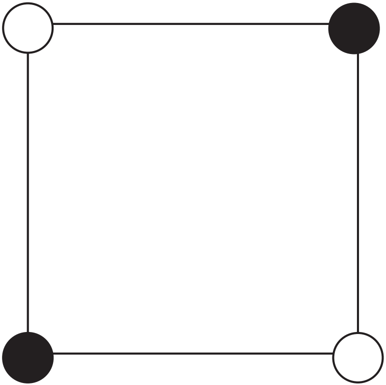
Visualization

Linearly separable:



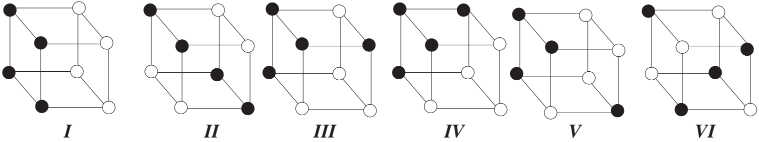
Visualization

Not linearly separable:



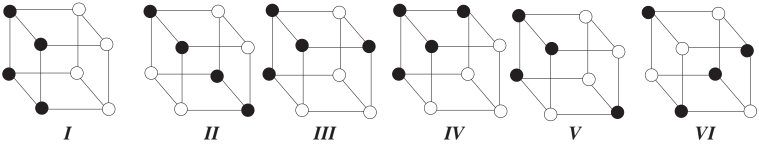
"XOR" function

More Complicated Cases



Which are linearly separable?

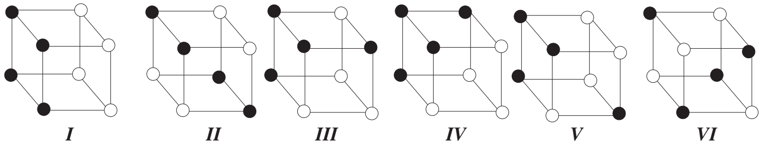
More Complicated Cases



Which are linearly separable?

- *I* and *IV* are clearly linearly separable
- Others are not

More Complicated Cases



No plane gives XOR intersected with linear decision boundary

- But *II* has XOR in front, in back
- But *III* has XOR along diagonal bottom-left to top-right (level in depth)
- But *V* has XOR in front, on bottom
- But *VI* has XOR on all sides

Interesting result

Naive Bayes models always have a single linear decision boundary

- vastly restricts the set of categorization rules they can encode
- flip side of powerful generalization

Linearity of Naive Bayes

Features as numbers

$$x_1 = \begin{cases} 1 \text{ if true} \\ 0 \text{ if false} \end{cases}$$

Same for x_2 (etc.)

Linearity of Naive Bayes

Look at $P(C = c_1 \,\&\, x_1 = 0 \,\&\, x_2 = 0)$

$$P(x_1 = 0 | C = c_1) P(x_2 = 0 | C = c_1) P(C = c_1)$$

Will be useful to work with $\log P(C = c_1 \,\&\, x_1 = 0 \,\&\, x_2 = 0)$.

$$\log P(x_1 = 0 | C = c_1) + \log P(x_2 = 0 | C = c_1) + \log P(C = c_1)$$

Call this k_1

Bit of Magic

Define w_{11} this way:

$$w_{11} := \log P(x_1 = 1 | C = c_1) - \log P(x_1 = 0 | C = c_1)$$

Consider $\log P(C = c_1 \ \& \ x_1 = 1 \ \& \ x_2 = 0)$

$$\begin{aligned}
&= \log P(x_1 = 1 | C = c_1) + \log P(x_2 = 0 | C = c_1) + \log P(C = c_1) \\
&= \log P(x_1 = 1 | C = c_1) - \log P(x_1 = 0 | C = c_1) + \\
&\quad \log P(x_1 = 0 | C = c_1) + \log P(x_2 = 0 | C = c_1) + \log P(C = c_1) \\
&= w_{11} x_1 + k_1
\end{aligned}$$

Bit of Magic

Define w_{12} this way:

$$w_{12} := \log P(x_2 = 1 | C = c_1) - \log P(x_2 = 0 | C = c_1)$$

Consider $\log P(C = c_1 \;\&\; x_1 = 0 \;\&\; x_2 = 1)$

$$\begin{aligned}
&= \log P(x_1 = 0 | C = c_1) + \log P(x_2 = 1 | C = c_1) + \log P(C = c_1) \\
&= \log P(x_1 = 0 | C = c_1) + \log P(x_2 = 1 | C = c_1) - \log P(x_2 = 0 | C = c_1) \\
&\quad \log P(x_2 = 0 | C = c_1) + \log P(C = c_1) \\
&= w_{12} x_2 + k_1
\end{aligned}$$

In fact...

Have the following general equation:

$$\log P(C = c_1 \,\&\, x_1 \,\&\, x_2) = w_{11}x_1 + w_{12}x_2 + k_1$$

As feature vector:

$$\log P(C = c_1 \,\&\, x_1 \,\&\, x_2) = w_1 x + k_1$$

Likewise

$$\log P(C = c_2 \,\&\, x_1 \,\&\, x_2) = w_2 x + k_2$$

Where we define w_2 and k_2 analogously for c_2.

Linearity of Naive Bayes

Rule:

$$\text{Decide } C = c_1 \text{ if } P(C = c_1 \,\&\, x_1 \,\&\, x_2) > P(C = c_2 \,\&\, x_1 \,\&\, x_2)$$

Take the log:

$$\text{Decide } C = c_1 \text{ if}$$
$$\log P(C = c_1 \,\&\, x_1 \,\&\, x_2) > \log P(C = c_2 \,\&\, x_1 \,\&\, x_2)$$

Use our formulas:

$$\text{Decide } C = c_1 \text{ if } w_1 x + k_1 > w_2 x + k_2$$
$$\text{Decide } C = c_1 \text{ if } (w_1 - w_2)x > k_2 - k_1$$

More General Linear Classifiers

Two ways to fit linear decision boundary

- Method 1: Naive Bayes
  - Measure probabilities under certain assumptions
  - Probabilities predict decision boundary
- Method 2: Optimize boundary directly
  - Find best boundary to separate training data by category
  - Use decision boundary for classification
  - Ignore probabilistic interpretation

“Generative” versus “discriminative” training

Alternative Algorithms

Perceptron

- Keep track of weight vector w
- Output yes if $w \cdot x > 1$
- General linear classifier

Repeatedly consider each point x_i in training data

- If x_i is classified correctly, continue
- If x_i is classified incorrectly:
  let ϵ be sign of error
  let λ be small correction factor
  do: $w \leftarrow w + \lambda \epsilon x_i$

Until weights converge

Perceptron

- Converges to good boundary
  if training data is linearly separable
- Useful in practice in other cases
  just decrease correction factor λ in later iterations
  (like simulated annealing)
- Influential as model of neurons from 1960s
- Difficult to generalize ideas beyond linear functions

Perceptron

Demo: `http://eecs.wsu.edu/~cook/ai/lectures/applets/perceptron/`

Alternative Algorithms

Optimize a decision boundary

- Maximum entropy models
  try to get posterior probabilities at observations
  to match the data as close as possible
- Support vector machines
  try to put observations as far as possible from the decision boundary

Kinds of Learning

So far: discrete

- Categorization problems
- World is in one of discrete set of states
- Need to infer state from sensor values

Alternative: continuous

- True state of the world is a quantitative value
- Still want to infer true state from sensor values

Case study: recommendation

Given

- User (or other reviewer)
- Movie (or other product)

Predict

- What numerical rating will the user give the movie?

Collaborative Filtering

Solving prediction problems from similarities

- Data set has tuples
  - User
  - Item
  - Ranking
- Get new pair
  - User who has rated some items
  - Item that other users have rated
  - User has never rated item
- Predict rating

Two Intuitions

Intuition 1. User-based.

- Some users have similar taste and have rated this item
- Make a prediction based on these ratings

Algorithmic breakdown

- Find users that are similar
- Aggregate their ratings of this item into prediction

Two Intuitions

Intuition 2. Item-based.

- Some items have similar properties and have been rated by this user
- Make a prediction based on these ratings

Algorithmic breakdown

- Find items that are similar
- Aggregate this user's ratings of these items into prediction

Nearest-Neighbor Techniques

Nearest-neighbor classification

- predict test point has majority label of closest training points

Nearest-neighbor clustering

- Iteratively find prototype points that are averages of the training points closest to them

Nearest-neighbor prediction

- predict test point at weighted average of values of closest training points

Point: similar ideas apply for supervised, unsupervised, classification, estimation

Making Predictions

- find nearest neighbors—most similar **users**
- get their predictions
- multiply by similarity
- sum up
- normalize by total similarity

Table 2-2. Creating recommendations for Toby

Critic	Similarity	Night	S.xNight	Lady	S.xLady	Luck	S.xLuck
Rose	0.99	3.0	2.97	2.5	2.48	3.0	2.97
Seymour	0.38	3.0	1.14	3.0	1.14	1.5	0.57
Puig	0.89	4.5	4.02			3.0	2.68
LaSalle	0.92	3.0	2.77	3.0	2.77	2.0	1.85
Matthews	0.66	3.0	1.99	3.0	1.99		
Total			12.89		8.38		8.07
Sim. Sum			3.84		2.95		3.18
Total/Sim. Sum			3.35		2.83		2.53

Making Predictions - method 2

- find nearest neighbors—most similar **items**
- get their predictions
- multiply by similarity
- sum up
- normalize by total similarity

Table 2-3. Item-based recommendations for Toby

Movie	Rating	Night	R.xNight	Lady	R.xLady	Luck	R.xLuck
Snakes	4.5	0.182	0.818	0.222	0.999	0.105	0.474
Superman	4.0	0.103	0.412	0.091	0.363	0.065	0.258
Dupree	1.0	0.148	0.148	0.4	0.4	0.182	0.182
Total		0.433	1.378	0.713	1.764	0.352	0.914
Normalized			3.183		2.473		2.598

Predictions and Similarity

One idea: Euclidean distance

	A	B	C
a	1	2	4
b	2	1	2
c	1	2	4
d	1	1	2
e	2	1	2

Questions: Who is closest to B?
Who would you want to know to predict B?

Predictions and Similarity

Another idea: How useful is one at predicting another?

	A	B	C
a	1	2	4
b	2	1	2
c	1	2	4
d	1	1	2
e	2	1	2

Now C is very similar to B!