CS 440: Introduction to Artificial Intelligence

Lecture 14

Matthew Stone

October 21, 2015

Supervised categorization—Recap

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

Analyzing Learning—Recap

Understanding how well algorithms work: Key concepts

- Probability of different outcomes
- Evidence available from training data
- Evidence available from features of test point

Analyzing Nearest Neighbor—Recap

Weaknesses

- Requires lots of data
  little generalization across items
- Responds badly to ambiguity
  makes random decisions, not likely ones

Understanding classification via probability

Item is drawn from underlying category

- Represented as random variable C
- Takes on one of a few possible values: c_1, c_2, etc.
- Have *prior* probabilities $P(C = c_1)$, $P(C = c_2)$ etc.
- Prior is overall weight of category throughout space

Understanding classification via probability

We get feature vector describing observation

- Represented as a random variable O
- Takes on discrete or continuous vector values o (too many possibilities to list)
- Depends only on category of item
- Have *likelihood* probabilities $P(O = o | C = c_1)$ etc.
- Often easy to represent and learn likelihood

Understanding classification via probability

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

Simple Case Study

Focus on using just two features

- Two categories (Category: cup or glass): $C = c_1$ or $C = g_2$
- Feature one (Temperature: warm or frosty): $T = w_1$ or $T = f_2$
- Feature two (Shape: handle or tube): $S = h_1$ or $S = t_2$

Analyze Each Example Type

Get data set of 100 vessels

T	S	C	count
$T = w_1$	$S = h_1$	$C = c_1$	32
$T = f_2$	$S = h_1$	$C = c_1$	8
$T = w_1$	$S = t_2$	$C = c_1$	16
$T = f_2$	$S = t_2$	$C = c_1$	4
$T = w_1$	$S = h_1$	$C = g_2$	4
$T = f_2$	$S = h_1$	$C = g_2$	6
$T = w_1$	$S = t_2$	$C = g_2$	12
$T = f_2$	$S = t_2$	$C = g_2$	18

Use counts to estimate probabilities

Aside

How do you get the data?

- ESP Game—CAPTCHAs
- Mechanical Turk
- Summer interns

Probability Tables

Start with joint distribution

T	S	C	$P(T\&S\&C)$
$T = w_1$	$S = h_1$	$C = c_1$	0.32
$T = f_2$	$S = h_1$	$C = c_1$	0.08
$T = w_1$	$S = t_2$	$C = c_1$	0.16
$T = f_2$	$S = t_2$	$C = c_1$	0.04
$T = w_1$	$S = h_1$	$C = g_2$	0.04
$T = f_2$	$S = h_1$	$C = g_2$	0.06
$T = w_1$	$S = t_2$	$C = g_2$	0.12
$T = f_2$	$S = t_2$	$C = g_2$	0.18

Then figure out conditional probabilities to make decisions

Probability Tables

Start with joint distribution

T	S	C	$P(T\&S\&C)$
$T = w_1$	$S = h_1$	$C = c_1$	0.32
$T = f_2$	$S = h_1$	$C = c_1$	0.08
$T = w_1$	$S = t_2$	$C = c_1$	0.16
$T = f_2$	$S = t_2$	$C = c_1$	0.04

T	S	C	$P(T\&S\&C)$
$T = w_1$	$S = h_1$	$C = g_2$	0.04
$T = f_2$	$S = h_1$	$C = g_2$	0.06
$T = w_1$	$S = t_2$	$C = g_2$	0.12
$T = f_2$	$S = t_2$	$C = g_2$	0.18

Then figure out conditional probabilities to make decisions

T	S	C	$P(C \mid T\&S)$
$T = w_1$	$S = h_1$	$C = c_1$	0.89
$T = f_2$	$S = h_1$	$C = c_1$	0.57
$T = w_1$	$S = t_2$	$C = c_1$	0.57
$T = f_2$	$S = t_2$	$C = c_1$	0.18

T	S	C	$P(C \mid T\&S)$
$T = w_1$	$S = h_1$	$C = g_2$	0.11
$T = f_2$	$S = h_1$	$C = g_2$	0.43
$T = w_1$	$S = t_2$	$C = g_2$	0.43
$T = f_2$	$S = t_2$	$C = g_2$	0.82

CS Questions

Suppose you have two classes, d features each with k values

- How big is the table that gives the joint distribution?
- How much data do you need to get good estimates?
- What conclusion do you draw about this method?

CS Questions

Suppose you have two classes, d features each with k values

- How big is the table that gives the joint distribution? $O(k^d)$
- How much data do you need to get good estimates? $O(k^d)$
- What conclusion do you draw about this method?
  It does not scale.

Problem is that the method does not *generalize*.

Generalization in Probabilistic Models

Key idea: indepedence assumptions

- Ignore certain kinds of interactions in world
- Assume that they are not important
- Lets you use same data to learn multiple relationships

Naive Bayes assumption

Each feature is independent of the others given the class

- Mathematically:

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

- As a result:

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

- Intuitively: features reflect class only, not other features
- Can be useful approximation for modeling and learning