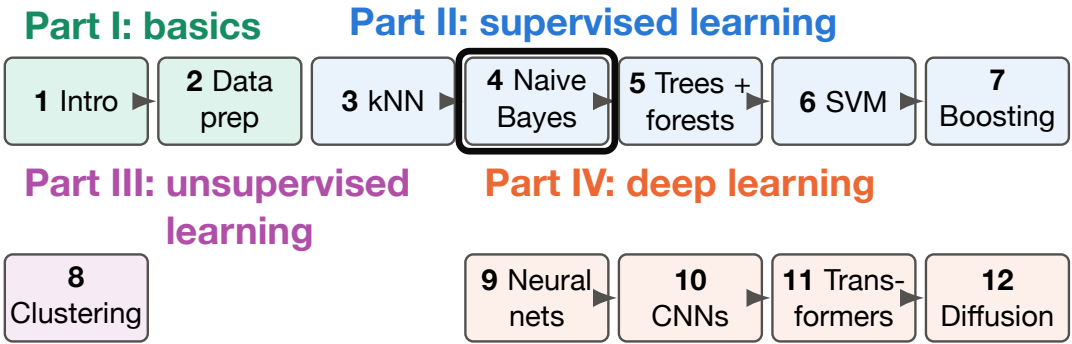
Naive Bayes classifier

229352 Lecture 4: naive Bayes

Course map


Part I: basics
Part II: supervised learning
1 Intro
2 Data prep
3 kNN
4 Naive Bayes
5 Trees + forests
6 SVM
7 Boosting
Part III: unsupervised learning
Part IV: deep learning
8 Clustering
9 Neural nets
10 CNNs
11 Trans-formers
12 Diffusion

A video game store

#	Genre	Price	Reviews	Buy
1	Action	Low	Good	Yes
2	Sports	High	Mixed	No
3	Puzzle	Low	Good	Yes
4	Action	High	Good	No
5	Sports	High	Good	Yes
6	Action	Low	Mixed	No
7	Puzzle	Low	Mixed	Yes
8	Action	High	Good	Yes
9	Sports	Low	Mixed	No
10	Sports	Low	Good	Yes

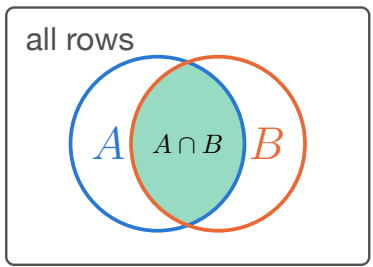
A user looked at 10 games.

Features: genre, price, reviews.

Class: did the user buy the game?

Made-up data

Conditional probability



$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Look only at the rows in B:
what share of them is also in A?

In counts:

$$P(A \mid B) = \frac{\#\,(A \text{ and } B)}{\#\,B}$$

Buying action games

#	Genre	Price	Reviews	Buy
1	Action	Low	Good	Yes
2	Sports	High	Mixed	No
3	Puzzle	Low	Good	Yes
4	Action	High	Good	No
5	Sports	High	Good	Yes
6	Action	Low	Mixed	No
7	Puzzle	Low	Mixed	Yes
8	Action	High	Good	Yes
9	Sports	Low	Mixed	No
10	Sports	Low	Good	Yes

$$P(\text{Yes} \mid \text{Action}) =$$

Classify by the larger probability

New game with features $x = (x_1, \ldots, x_d)$.

1. Estimate both conditional probabilities

$$P(\text{Yes} \mid x) \qquad \text{and} \qquad P(\text{No} \mid x)$$

2. Predict the class with the larger one:

$$\widehat{y} = \text{Yes} \;\text{ if }\; P(\text{Yes} \mid x) > P(\text{No} \mid x), \;\text{ else No}$$

A new game

Genre = Action, Price = High,

Reviews = Mixed

Will the user buy it?

No row matches all three

#	Genre	Price	Reviews	Buy
1	Action	Low	Good	Yes
2	Sports	High	Mixed	No
3	Puzzle	Low	Good	Yes
4	Action	High	Good	No
5	Sports	High	Good	Yes
6	Action	Low	Mixed	No
7	Puzzle	Low	Mixed	Yes
8	Action	High	Good	Yes
9	Sports	Low	Mixed	No
10	Sports	Low	Good	Yes

Shaded: action games.

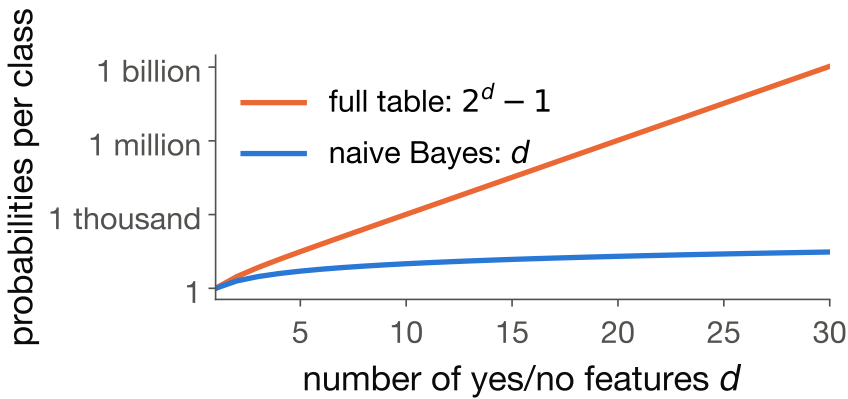
No action game is also High and Mixed.

Matching rows: 0

$$P(\text{Yes} \mid x) = 0/0$$

undefined

Too many combinations to count



10 rows cannot cover $3 \times 2 \times 2 = 12$ combinations

Bayes' theorem

$$\underbrace{P(Y \mid X)}_{\text{posterior}} = \frac{\overbrace{P(X \mid Y)}^{\text{likelihood}} \overbrace{P(Y)}^{\text{prior}}}{\underbrace{P(X)}_{\text{same for every class}}}$$

$X = (X_1, \ldots, X_d)$: the features; Y: the class

We can drop the denominator

Predict $\widehat{y} = \text{Yes}$ if

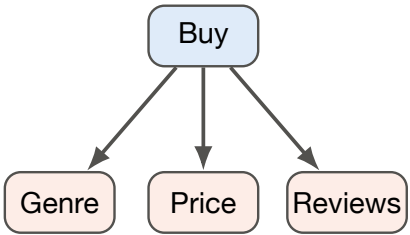
$$\frac{P(X \mid \text{Yes})\, P(\text{Yes})}{P(X)} > \frac{P(X \mid \text{No})\, P(\text{No})}{P(X)}$$

Both sides are divided by the same $P(X)$.
So predict Yes if

$$P(X \mid \text{Yes})\, P(\text{Yes}) > P(X \mid \text{No})\, P(\text{No})$$

But $P(X \mid Y)$ has the same counting problem.

The naive assumption



Given the class, the features are conditionally independent.

$$P(X_1, \ldots, X_d \mid Y) = P(X_1 \mid Y) \cdots P(X_d \mid Y)$$

Each factor needs one feature column and the class column.

In practice, features usually depend on each other.

The naive Bayes rule

$$\widehat{y} = \arg\max_{y} \; P(y) \prod_{j=1}^{d} P(x_j \mid y)$$

- $P(y)$: the share of training rows in class y
- $P(x_j \mid y)$: the share of class-y rows with value x_j in feature j
- score$(y) = P(y) \prod_j P(x_j \mid y)$: predict the class with the larger score

Summary: naive Bayes

- Predict the class y with the largest posterior $P(y \mid x)$.
- Bayes' theorem: $P(y \mid x) \propto P(x \mid y)\, P(y)$; the denominator $P(x)$ is the same for every class.
- Naive assumption: given y, the features are conditionally independent, so $P(x \mid y) = \prod_j P(x_j \mid y)$.
- Each $P(x_j \mid y)$ comes from counts in one feature column: far fewer rows are needed than for the full table.

Checkpoint 1

10 yes/no features, classes 0 and 1

How many probabilities does naive Bayes estimate?

Count only the free probabilities: $P(\text{no} \mid y) = 1 - P(\text{yes} \mid y)$

Checkpoint 1

10 yes/no features, classes 0 and 1

How many probabilities does naive Bayes estimate?

Count only the free probabilities: $P(\text{no} \mid y) = 1 - P(\text{yes} \mid y)$

21: the 20 values $P(x_j = \text{yes} \mid y)$, plus $P(y = 1)$

(the full table needs 2047)

Score for Yes

#	Genre	Price	Reviews	Buy
1	Action	Low	Good	Yes
2	Sports	High	Mixed	No
3	Puzzle	Low	Good	Yes
4	Action	High	Good	No
5	Sports	High	Good	Yes
6	Action	Low	Mixed	No
7	Puzzle	Low	Mixed	Yes
8	Action	High	Good	Yes
9	Sports	Low	Mixed	No
10	Sports	Low	Good	Yes

Action, High, Mixed

$$P(\text{Yes}) \times P(\text{Action} \mid \text{Yes}) \times P(\text{High} \mid \text{Yes}) \times P(\text{Mixed} \mid \text{Yes}) =$$

Score for No

#	Genre	Price	Reviews	Buy
1	Action	Low	Good	Yes
2	Sports	High	Mixed	No
3	Puzzle	Low	Good	Yes
4	Action	High	Good	No
5	Sports	High	Good	Yes
6	Action	Low	Mixed	No
7	Puzzle	Low	Mixed	Yes
8	Action	High	Good	Yes
9	Sports	Low	Mixed	No
10	Sports	Low	Good	Yes

Action, High, Mixed

$$P(\text{No}) \times P(\text{Action} \mid \text{No}) \times P(\text{High} \mid \text{No}) \times P(\text{Mixed} \mid \text{No}) =$$

Which class do we predict?

score(Yes) =

score(No) =

Prediction:

A numeric feature

#	Demo hours	Price	Buy
1	3.0	Low	Yes
2	1.5	High	No
3	4.0	Low	Yes
4	2.0	High	No
5	3.5	High	Yes
6	2.5	Low	No
7	3.0	Low	Yes
8	5.0	High	Yes
9	1.0	Low	No
10	4.5	Low	Yes

Demo hours: the number of hours the user played the free demo.

No game has exactly 3.2 demo hours, so counting fails.

Fit a bell curve to each class.

A bell curve for each class

$$f(x_j \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\Big(- \frac{(x_j - \mu_y)^2}{2\sigma_y^2} \Big)$$

numeric feature j; μ_y, σ_y^2: its mean and variance in class y

f is a density, not a probability. For a small width Δ:

$$P(x_j \le X_j \le x_j + \Delta \mid y) \approx f(x_j \mid y)\,\Delta$$

Δ is the same for every class: compare the heights $f(x_j \mid y)$.

Estimating the mean and the variance

N_y = number of rows in class y; x_{ij} = feature j of row i

$$\mu_y = \frac{1}{N_y} \sum_{i:\, y_i = y} x_{ij} \qquad \sigma_y^2 = \frac{1}{N_y - 1} \sum_{i:\, y_i = y} (x_{ij} - \mu_y)^2$$

Parameters for demo hours

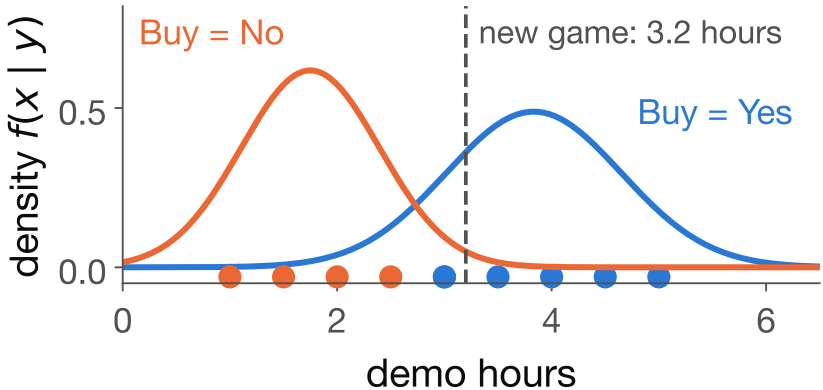
Buy = Yes: 3.0, 4.0, 3.5, 3.0, 5.0, 4.5 ($N_{\text{Yes}} = 6$)

$\mu_{\text{Yes}} = \frac{23}{6} = 3.83$ $\sigma^2_{\text{Yes}} = \frac{1}{5}\sum_i (x_i - 3.83)^2 = 0.67$

Buy = No: 1.5, 2.0, 2.5, 1.0 ($N_{\text{No}} = 4$)

$\mu_{\text{No}} = \frac{7}{4} = 1.75$ $\sigma^2_{\text{No}} = \frac{1}{3}\sum_i (x_i - 1.75)^2 = 0.42$

Fitted curves for demo hours



at 3.2 hours, compare the heights of the two curves

A new game with a demo

$$\text{Demo hours} = 3.2, \;\; \text{price} = \text{High}$$

$$\text{score(Yes)} = P(\text{Yes}) \times f(3.2 \mid \text{Yes}) \times P(\text{High} \mid \text{Yes})$$

$$\text{score(No)} = P(\text{No}) \times f(3.2 \mid \text{No}) \times P(\text{High} \mid \text{No})$$

Predict the class with the larger score.

Height of each curve at 3.2 hours

$\mu_{\text{Yes}} = 3.83,\ \sigma^2_{\text{Yes}} = 0.67, \qquad \mu_{\text{No}} = 1.75,\ \sigma^2_{\text{No}} = 0.42$

$f(3.2 \mid \text{Yes}) =$

$f(3.2 \mid \text{No}) =$

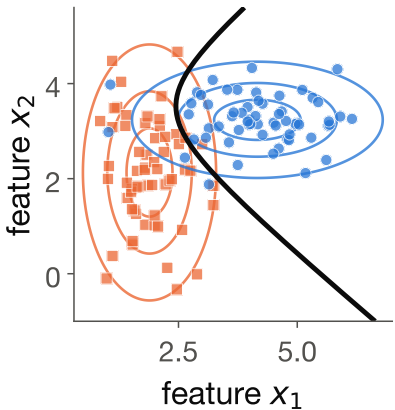
Probability of a high price

#	Demo hours	Price	Buy
1	3.0	Low	Yes
2	1.5	High	No
3	4.0	Low	Yes
4	2.0	High	No
5	3.5	High	Yes
6	2.5	Low	No
7	3.0	Low	Yes
8	5.0	High	Yes
9	1.0	Low	No
10	4.5	Low	Yes

$$P(\text{High} \mid \text{Yes}) =$$

$$P(\text{High} \mid \text{No}) =$$

Two numeric features



blue: Yes, orange: No

Contours: ellipses with axes parallel to x_1 and x_2.

Black: equal scores. This boundary is a curve; it is a straight line when each feature has the same variance in both classes.

Summary: Gaussian naive Bayes

- A numeric feature gets one bell curve per class, with mean μ_y and variance σ_y^2.
- In the naive Bayes rule, $P(x_j \mid y)$ is replaced by the height $f(x_j \mid y)$.
- Numeric and categorical features can be mixed: multiply their factors.

Positive or negative?

1	This movie is disappointing.	negative
2	I love everything about this movie.	positive
3	I would love to have that two hours of my life back.	negative
4	This is one of my favorite if not favorite films.	positive
5	I have seen so many bad low budget movies lately, but I love this one.	positive

A review as a bag of words



word	count
i	1
love	2
this	1
movie	1

We lowercase the text and drop punctuation; “movies” and “movie” are still different words.

Word order is dropped.

Naive Bayes for text

A review with L words $w_1, \ldots, w_L$ (i = position in the review):

$$\text{score}(y) = P(y) \prod_{i=1}^{L} P(w_i \mid y)$$

Assumption: given the class, each position holds a word drawn independently from the same distribution $P(w \mid y)$.

A word that appears twice contributes its factor twice.

Estimating the word probabilities

$$P(w \mid y) = \frac{\text{count}(w, y)}{\sum_{v \in V} \text{count}(v, y)}$$

count(w, y): how often the word w appears in the reviews of class y

V: the vocabulary, the set of all distinct words in the training reviews

$P(y)$: the share of the reviews in class y

Prior of the positive class

1	This movie is disappointing.	negative
2	I love everything about this movie.	positive
3	I would love to have that two hours of my life back.	negative
4	This is one of my favorite if not favorite films.	positive
5	I have seen so many bad low budget movies lately, but I love this one.	positive

$$P(\text{positive}) =$$

Probability of one word in a class

1	This movie is disappointing.	negative
2	I love everything about this movie.	positive
3	I would love to have that two hours of my life back.	negative
4	This is one of my favorite if not favorite films.	positive
5	I have seen so many bad low budget movies lately, but I love this one.	positive

$$P(\text{favorite} \mid \text{positive}) =$$

A new review: positive?

"I love, love this movie."

$P(y = \text{positive} \mid \text{review}) \propto$

A new review: negative?

"I love, love this movie."

$P(y = \text{negative} \mid \text{review}) \propto$

A zero count

New review: "Not **disappointing** at all. I love it."

"disappointing" appears only in a negative review.

$$\text{count}(\text{disappointing}, \text{positive}) = 0,$$

hence $P(\text{disappointing} \mid \text{positive}) = 0$.

So score(positive) $= 0$, whatever the other words are.

A word in no training review is not in V: it is dropped.

Laplace smoothing

Add $\alpha > 0$ to every count:

$$P(w \mid y) = \frac{\text{count}(w, y) + \alpha}{\sum_{v \in V} \left(\text{count}(v, y) + \alpha\right)} = \frac{\text{count}(w, y) + \alpha}{\sum_{v \in V} \text{count}(v, y) + \alpha \left|V\right|}$$

$$\alpha = 1: \quad P(\text{disappointing} \mid \text{positive}) = \frac{0 + 1}{N_{\text{pos}} + \left|V\right|} > 0$$

$\alpha = 1$ is Laplace smoothing. N_{pos}: number of words in the positive reviews.

Zero counts in the game data

#	Genre	Price	Reviews	Buy
1	Action	Low	Good	Yes
2	Sports	High	Mixed	No
3	Puzzle	Low	Good	Yes
4	Action	High	Good	No
5	Sports	High	Good	Yes
6	Action	Low	Mixed	No
7	Puzzle	Low	Mixed	Yes
8	Action	High	Good	Yes
9	Sports	Low	Mixed	No
10	Sports	Low	Good	Yes

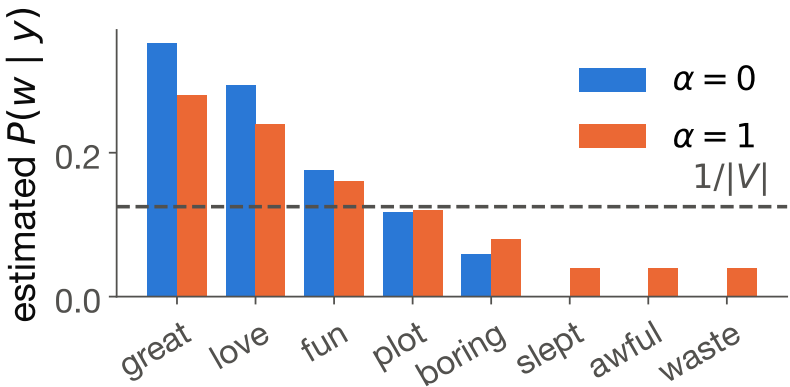
Shaded: puzzle games, all bought.

$P(\text{Puzzle} \mid \text{No}) = 0/4$, so score(No) $= 0$.

With $\alpha = 1$ ($m = 3$ genres play the role of $|V|$):

$$\frac{0+\alpha}{4+\alpha m} = \frac{0+1}{4+3} = \frac{1}{7}$$

Effect of smoothing



every estimate moves toward $1/|V|$

Multiplying many small numbers

200 words, each with $P(w \mid y) \approx 0.01$: $\quad 0.01^{200} = 10^{-400}$

The smallest positive double-precision number is about 5×10^{-324}: every score underflows to 0.

Compare sums of logs instead:

$$\log \text{score}(y) = \log P(y) + \sum_{i=1}^{L} \log P(w_i \mid y)$$

$\log$ is strictly increasing, so the class with the larger score also has the larger log score.

Summary: naive Bayes for text

- A review is a bag of words; each word w has a probability $P(w \mid y)$ in each class.
- Estimate $P(w \mid y)$ from word counts, with Laplace smoothing $\alpha > 0$ so no estimate is 0.
- Smoothing also removes zero probabilities in categorical features.
- Predict with sums of logs: $\log P(y) + \sum_i \log P(w_i \mid y)$.

Checkpoint 2

vocabulary: 1000 distinct words

positive reviews: 9000 words in total

“slept” appears in negative reviews only

$$\alpha = 1\text{:} \quad P(\text{slept} \mid \text{positive}) = ?$$

Checkpoint 2

vocabulary: 1000 distinct words

positive reviews: 9000 words in total

“slept” appears in negative reviews only

$$\alpha = 1: \quad P(\text{slept} \mid \text{positive}) = ?$$

$$\frac{0+1}{9000+1000} = 0.0001$$

Training naive Bayes for text

1. Collect the vocabulary V from the training reviews.
2. For each class y: $P(y)$ = share of the reviews in class y.
3. For each class y: concatenate its reviews; count(w, y) = number of times word w appears in them.
4. For each word w and class y:

$$P(w \mid y) = \frac{\text{count}(w, y) + \alpha}{\sum_{v \in V} \text{count}(v, y) + \alpha \, |V|}$$

Naive Bayes in scikit-learn

```
from sklearn.naive_bayes import CategoricalNB, GaussianNB
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer

CategoricalNB(alpha=1.0)  # categories, coded 0, 1, 2, ...
GaussianNB()              # numeric features

# token_pattern keeps one-letter words such as "i"
vec = CountVectorizer(token_pattern=r"(?u)\b\w+\b")
X = vec.fit_transform(train_reviews)  # word counts
model = MultinomialNB(alpha=1.0).fit(X, train_labels)
model.predict(vec.transform(["I love, love this movie."]))
```

`alpha` is the Laplace α; predictions use sums of logs.