Computer Science

CSC380: Principles of Data Science

Predictive Modeling and Classification 3

Xinchen Yu

9. Given a distribution D with unknown mean μ and variance σ^2, and a set of n iid samples $X_1, \ldots, X_n$ drawn from it. Define $\tilde{\mu}_n = \frac{1}{n-1}\sum_{i=1}^{n} X_i$ as an estimator of μ.
(a) (4 points) Is $\tilde{\mu}_n$ an unbiased estimator of μ? Justify your answer.
(b) (6 points) Let $n = 4$. What is the bias, variance, and Mean Square Error (MSE) of $\tilde{\mu}_4$, respectively? Note: For variance, you can compute $Var[\tilde{\mu}_4]$, in other words, $Var[\frac{X_1+X_2+X_3+X_4}{3}]$. (You can have μ, σ^2 or numbers in the results).

Lecture statistics 3, page 7

$$\tilde{\mu}_n = \frac{1}{n-1}\sum_{i=1}^{n} X_i \qquad E[\tilde{\mu}_n] = E\left[\frac{1}{n-1}\sum_{i=1}^{n} X_i\right]$$

$$= \frac{1}{n-1} E\left[\sum_{i=1}^{n} X_i\right]$$

$$= \frac{1}{n-1}\sum_{i=1}^{n} E[X_i]$$

$$= \frac{1}{n-1}\sum_{i=1}^{n} \mu \quad = \frac{n\mu}{n-1}$$

$\tilde{\mu}_n$ is not an unbiased estimator of μ.

Midterm solutions

9. Given a distribution D with unknown mean μ and variance σ^2, and a set of n iid samples $X_1, \ldots, X_n$ drawn from it. Define $\tilde{\mu}_n = \frac{1}{n-1}\sum_{i=1}^{n} X_i$ as an estimator of μ.
(a) (4 points) Is $\tilde{\mu}_n$ an unbiased estimator of μ? Justify your answer.
(b) (6 points) Let $n = 4$. What is the bias, variance, and Mean Square Error (MSE) of $\tilde{\mu}_4$, respectively? Note: For variance, you can compute $Var[\tilde{\mu}_4]$, in other words, $Var[\frac{X_1+X_2+X_3+X_4}{3}]$. (You can have μ, σ^2 or numbers in the results).

$$\tilde{\mu}_4 = \frac{1}{3}(X_1 + X_2 + X_3 + X_4)$$

$$\begin{aligned}\textbf{Bias}(\tilde{\mu}_4) &= E[\tilde{\mu}_4] - \mu \\ &= \frac{4\mu}{3} - \mu \\ &= \frac{\mu}{3}\end{aligned}$$

$$\begin{aligned}\textbf{Var}[\tilde{\mu}_4] &= \textbf{Var}\left[\frac{1}{3}(X_1 + X_2 + X_3 + X_4)\right] \\ &= \frac{1}{9}\textbf{Var}[X_1 + X_2 + X_3 + X_4]\end{aligned}$$

Since the X_i are iid:

$$\begin{aligned}&= \frac{1}{9}(\textbf{Var}[X_1] + \textbf{Var}[X_2] + \textbf{Var}[X_3] + \textbf{Var}[X_4]) \\ &= \frac{1}{9}(4\sigma^2)\end{aligned}$$

$$\textbf{MSE}(\tilde{\mu}_4) = \textbf{Var}[\tilde{\mu}_4] + \textbf{Bias}(\tilde{\mu}_4)^2 = \frac{4\sigma^2 + \mu^2}{9}$$

Lecture statistics 3, page 8 and 18

Model Evaluation

Confusion Matrix

Suppose our classifier distinguishes between cats and non-cats.

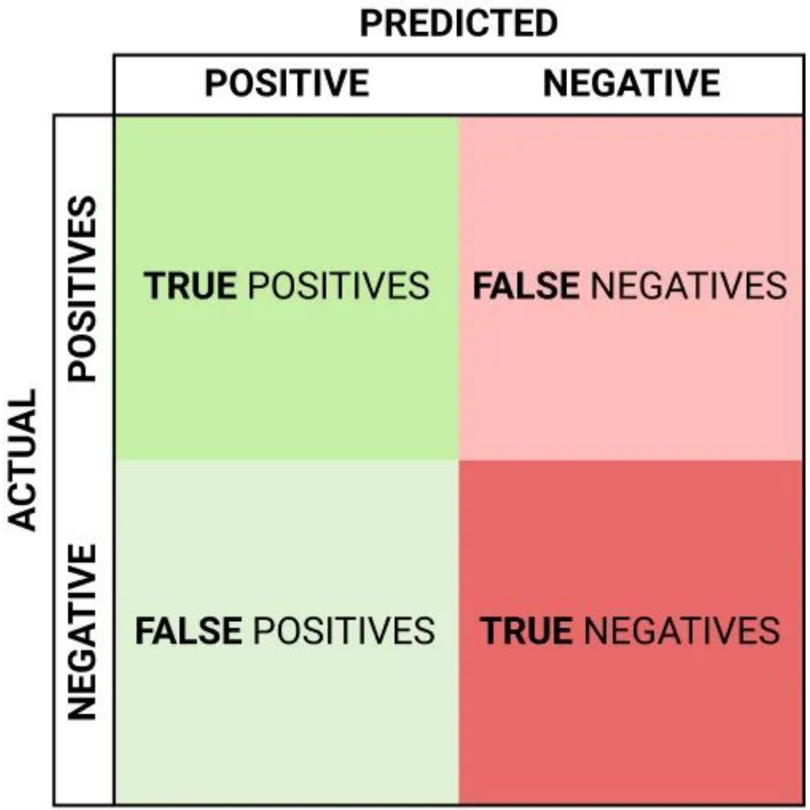
We can make the following table called **confusion matrix**:

Predicted class / Actual class	Cat	Non-cat
Cat	**6 true positives**	2 false negatives
Non-cat	1 false positive	**3 true negatives**

It tells us if classifier is biased towards certain mistakes (False Positives, False Neg.)

Good for investigating opportunities to improve the classifier.

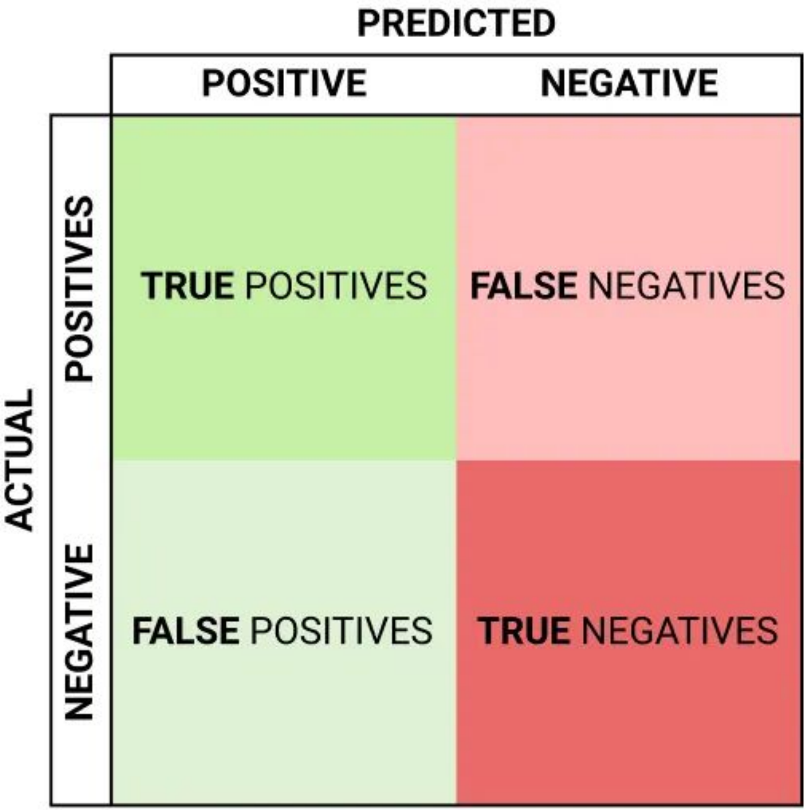
Evaluating Classifiers - Precision



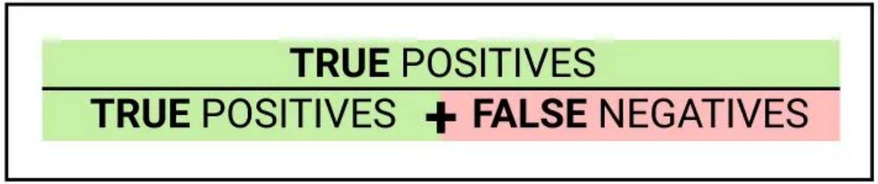
Precision: dividing the true positives by anything that was predicted as a positive.

$$\frac{\text{TRUE POSITIVES}}{\text{TRUE POSITIVES} + \text{FALSE POSITIVES}}$$

Evaluating Classifiers - Recall



Recall (or True Positive Rate): dividing the true positives by anything that should have been predicted as positive.



Evaluating Classifiers

F1 score symmetrically represents both precision and recall in one metric.

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}$$

- This is the *harmonic mean* of precision and recall
  - harmonic_mean(x,y)

$$\frac{1}{\frac{1}{2}(\frac{1}{x} + \frac{1}{y})}$$

- Gives equal importance to precision and recall – F1 may not be best when you care about one more than the other (e.g., in medical tests we care about recall)

Evaluation in Scikit-Learn

Evaluation functions live in `metrics`

`metrics.confusion_matrix`(y_true, y_pred, *)	Compute confusion matrix to evaluate the accuracy of a classification.
`metrics.dcg_score`(y_true, y_score, *[, k, ...])	Compute Discounted Cumulative Gain.
`metrics.det_curve`(y_true, y_score[, ...])	Compute error rates for different probability thresholds.
`metrics.f1_score`(y_true, y_pred, *[, ...])	Compute the F1 score, also known as balanced F-score or F-measure.
`metrics.fbeta_score`(y_true, y_pred, *, beta)	Compute the F-beta score.
`metrics.hamming_loss`(y_true, y_pred, *[, ...])	Compute the average Hamming loss.
`metrics.hinge_loss`(y_true, pred_decision, *)	Average hinge loss (non-regularized).
`metrics.jaccard_score`(y_true, y_pred, *[, ...])	Jaccard similarity coefficient score.
`metrics.log_loss`(y_true, y_pred, *[, eps, ...])	Log loss, aka logistic loss or cross-entropy loss.
`metrics.matthews_corrcoef`(y_true, y_pred, *)	Compute the Matthews correlation coefficient (MCC).
`metrics.multilabel_confusion_matrix`(y_true, ...)	Compute a confusion matrix for each class or sample.
`metrics.ndcg_score`(y_true, y_score, *[, k, ...])	Compute Normalized Discounted Cumulative Gain.
`metrics.precision_recall_curve`(y_true, ...)	Compute precision-recall pairs for different probability thresholds.
`metrics.precision_recall_fscore_support`(...)	Compute precision, recall, F-measure and support for each class.
`metrics.precision_score`(y_true, y_pred, *[, ...])	Compute the precision.
`metrics.recall_score`(y_true, y_pred, *[, ...])	Compute the recall.

Naïve Bayes

Naïve Bayes Overview

<u>Heads Up</u> This section will return to some math as we go in depth. But, much of it is that you already know with a new application (Naïve Bayes Classification) – **ask questions if you are lost**

- Introduction to Naïve Bayes Classifier
- Maximum Likelihood Estimation

Math Prep

- N RVs conditionally independent, given Z, if and only if:

$$p(X_1, \ldots, X_N \mid Z) = \prod_{i=1}^{N} p(X_i \mid Z)$$

Probability 3

- Bayes' rule

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

Probability 2

- Maximum likelihood estimation

Statistics 2

- Bernoulli and Gaussian distribution

Probability 3, 4
Statistics 2

Intuition

Training Data:

Person	height (feet)
male	6
male	5.92 (5'11")
male	5.58 (5'7")
male	5.92 (5'11")
female	5
female	5.5 (5'6")
female	5.42 (5'5")
female	5.75 (5'9")

Y X

Testing Data:

Person	height (feet)
?	6

Task: observe feature x_n , predict label $y_n \in (0,1)$

Choose the class with higher probability:

$$P(Y_n = 1|X_n = x_n) \quad \text{vs} \quad P(Y_n = 0|X_n = x_n)$$

$$P(Y_n = 1|X_n = x_n) = \frac{P(Y_n = 1, X_n = x_n)}{P(X_n = x_n)}$$

$$P(Y_n = 0|X_n = x_n) = \frac{P(Y_n = 0, X_n = x_n)}{P(X_n = x_n)}$$

How to learn a joint probability model for $P(Y, X)$?

Probabilistic Approach to ML

Training Data:

Person	height (feet)	weight (lbs)	foot size(inches)
male	6	180	12
male	5.92 (5'11")	190	11
male	5.58 (5'7")	170	12
male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

Features

Task: Observe features $x_1, \dots, x_D$ and predict class label $y \in \{1, \dots, C\}$

Model: Assume that the feature x and its label y follows certain type of distribution $\mathcal{D}$ with parameter θ.

$$(x, y) \overset{\text{i.i.d.}}{\sim} \mathcal{D}_\theta$$

Training Algorithm: Estimate θ e.g., MLE $\hat{\theta}$

To classify: Compute

$$\hat{y} = \arg \max_{c \in \{1,\dots,C\}} p(y = c \mid x; \hat{\theta})$$

what comes after semicolon is the parameter of the distribution

Naïve Bayes is a Specific Probabilistic Approach

Training Data:

Person	height (feet)	weight (lbs)	foot size(inches)
male	6	180	12
male	5.92 (5'11")	190	11
male	5.58 (5'7")	170	12
male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

Features

Task: Observe features $x_1, \ldots, x_D$ and predict class label $y \in \{1, \ldots, C\}$

Naïve Bayes Model: Treat features as *conditionally independent* given class label,

$$p(x,y) = p(y)p(x|y) = p(y)\prod_{d=1}^{D} p(x_d \mid y)$$

build individual models for these

To classify a given instance x: Bayes rule!

$$p(y = c \mid x) = \frac{p(y = c)p(x \mid y = c)}{p(x)}$$

Key concept in Naïve Bayes

$$p(x,y) = p(y)p(x|y) = p(y)\prod_{d=1}^{D} p(x_d \mid y)$$

Class prior distribution **Class conditional distribution**

Given one data point, it has 4 features (input), and the label is 0 (output)

$$p(x_1, x_2, x_3, x_4, y = 0) = p(y = 0) \cdot p(x_1, x_2, x_3, x_4 | y = 0)$$

$$= p(y = 0) \cdot p(x_1 | y = 0) \cdot p(x_2 | y = 0) \cdot p(x_3 | y = 0) \cdot p(x_4 | y = 0)$$

Class prior distribution

$$p(x,y) = p(y)p(x|y) = p(y)\prod_{d=1}^{D} p(x_d \mid y)$$

Class prior distribution **Class conditional distribution**

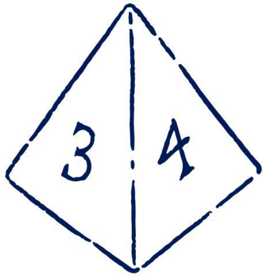
For the class prior distribution, take categorical distribution.

$$y \sim \text{Categorical}(\pi), \qquad \pi \in \mathbb{R}^C, \pi_c \geq 0, \sum_c \pi_c = 1$$

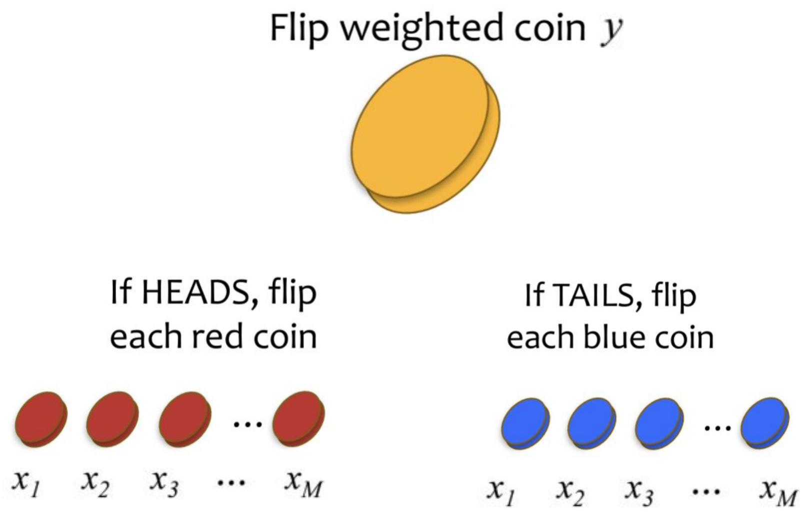
=> $p(y = c) = \pi_c$

Example: biased 4-sided die Y, given:

- P(Y=1) = 0.2, P(Y=2) = 0.3, P(Y=3) = 0.1, P(Y=4) = 0.4



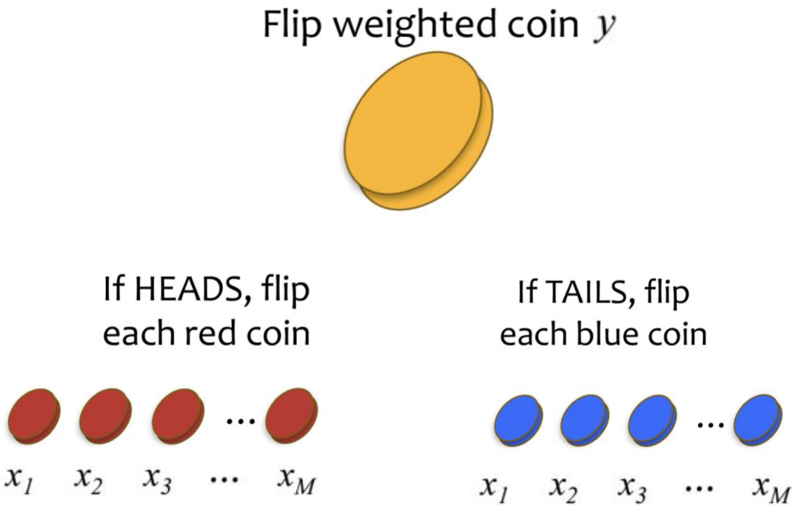
Class prior distribution



y	x_1	x_2	x_3	…	x_M
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

$p(y)$

Class conditional distribution

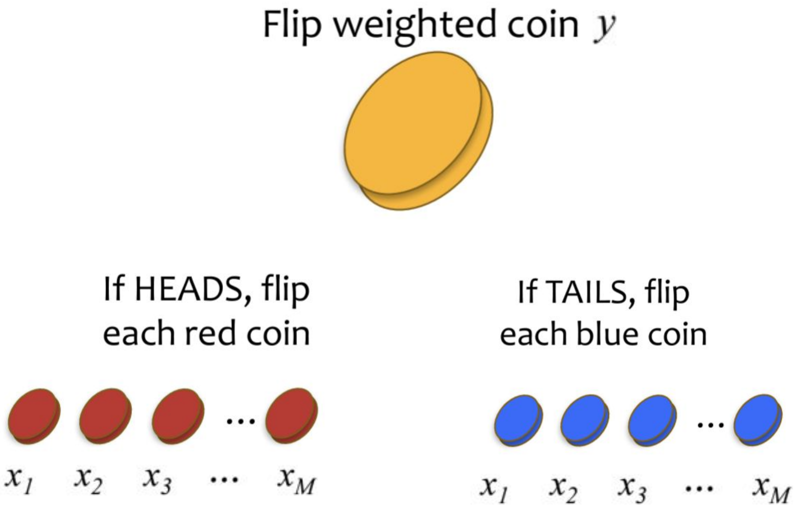


Each red / blue coin biases can be different.

y	x_1	x_2	x_3	…	x_M
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

$$p(x_1 | y = 1)$$

Class conditional distribution

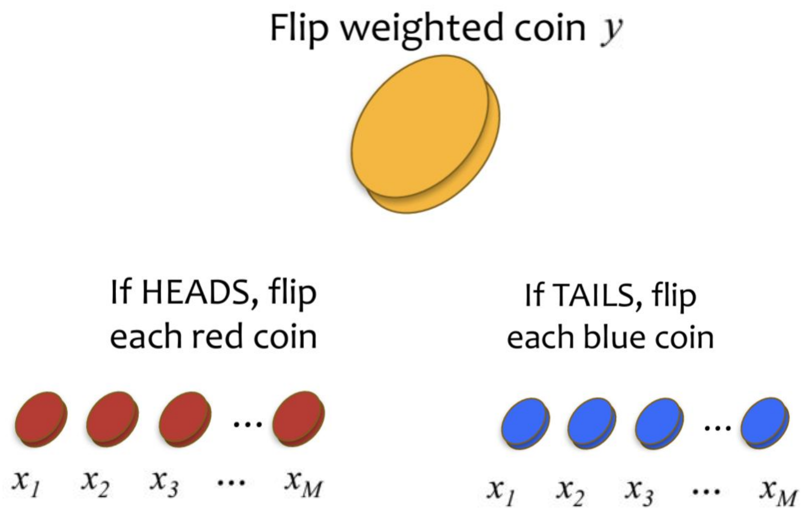


Each red / blue coin biases can be different.

y	x_1	x_2	x_3	…	x_M
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

$p(x_2|y = 1)$

Class conditional distribution



Each red / blue coin biases can be different.

y	x_1	x_2	x_3	…	x_M
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

$p(x_1|y=0)$

Class conditional distribution

Simplifying Assumption: "*Class conditional*" distribution assumes features are conditionally independent given class

$$p(x \mid y) = \prod_{d=1}^{D} p(x_d \mid y)$$

- "Naïve" as in general features are likely to be dependent.
- Every feature can have a different class-conditional distribution

$P(x_1|y)$ can be different from $P(x_2|y)$

Class conditional distribution

Features are typically not independent!

Example 1 If a recent news article contains word “Donald” it is much more likely to contain the word “Trump”.


• ELECTION 2016 •

MORE ELECTION COVERAGE •

Trump Spends Entire Classified National Security Briefing Asking About Egyptian Mummies

NEWS IN BRIEF August 18, 2016

VOL 52 ISSUE 32 · Politics · Politicians · Election 2016 · Donald Trump

Example 2 If flower <u>petal *width*</u> is very large then <u>petal *length*</u> is also likely to be high.

Source: Matt Gormley

Class conditional distribution

Simplifying Assumption: “*Class conditional*” distribution assumes features are conditionally independent given class

$$p(x \mid y) = \prod_{d=1}^{D} p(x_d \mid y)$$

- “Naïve” as in general features are likely to be dependent.
- Every feature can have a different class-conditional distribution

Doesn’t capture correlation among features. But why would it be a good idea?

- Easy computation: For C classes and D features only $O(CD)$ parameters
- Prevents overfitting
- Simplicity

Class conditional distribution

For real-valued features we can use Normal distribution:

$$p(x \mid y = c) = \prod_{d=1}^{D} \underbrace{\mathcal{N}(x_d \mid \mu_{cd}, \sigma_{cd}^2)}_{\text{Parameters of featured for class } c}$$

quiz candidate
Q: how many parameters?
2cd

For binary features $x_d \in \{0,1\}$ can use Bernoulli distributions:

$$p(x \mid y = c) = \prod_{d=1}^{D} \mathrm{Bernoulli}(x_d \mid \theta_{cd})$$

quiz candidate
Q: how many parameters?
cd

“Coin bias” for d[th] feature and class c

- K-valued discrete features: use Categorical.
- Can mix-and-match, e.g. some discrete, some continuous features

$$p(x \mid y = c) = \prod_{d=1}^{D'} \mathrm{Bernoulli}(x_d \mid \theta_{cd}) \prod_{d=D'+1}^{D} \mathcal{N}(x_d \mid \mu_{cd}, \sigma_{cd}^2)$$

Naïve Bayes Model : Maximum Likelihood

Fitting the model requires learning all parameters…

$$p(x, y = c) = \underbrace{p(y = c; \pi)}_{\text{Class Prior Parameters}} \prod_{d=1}^{D} \underbrace{p(x_d \mid \theta_{cd})}_{\text{Likelihood Parameters}}$$

Given training data $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ maximize the likelihood function,

$$\theta^{\text{MLE}} = \arg\max_{\pi,\theta} \log p(\mathcal{D}; \pi, \theta)$$

Naïve Bayes Model : Maximum Likelihood

$$\theta^{\mathrm{MLE}} = \arg\max_{\pi,\theta} \log p(\mathcal{D};\pi,\theta)$$

$(\mathcal{D} \coloneqq \{(x^{(i)}, y^{(i)})\}_{i=1}^{m})$

Since data are iid

$$= \arg\max_{\pi,\theta} \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)};\pi,\theta)$$

log(ab) = log a + log b

$$= \arg\max_{\pi,\theta} \sum_{i=1}^{m} \log p(x^{(i)}, y^{(i)};\pi,\theta)$$

Conditional probability + Naïve Bayes assumption

$$= \arg\max_{\pi,\theta} \sum_{i=1}^{m} \log p(y^{(i)};\pi) + \sum_{i=1}^{m}\sum_{d=1}^{D} \log p\left(x_d^{(i)} \middle| y^{(i)};\theta_{y^{(i)}d}\right)$$

θ_{cd}: parameter for feature d for class c

Find zero-gradient if concave, or gradient-based optimization otherwise

Example: Naïve Bayes with Bernoulli Features

Analogy:

y	x_1	x_2	...			x_D
5	0	1	1	0	1	0
2	1	0	0	0	0	0
3	1	1	1	1	0	0
...			...			
4	0	0	1	1	0	1

Example: Naïve Bayes with Bernoulli Features

Let each feature follow a Bernoulli distribution then the model is…

$$y \sim \text{Categorical}(\pi) \qquad x_d \mid y = c \sim \text{Bernoulli}(\theta_{cd})$$

The Naïve Bayes joint distribution is then:

$$p(\mathcal{D}; \pi, \theta) = \prod_{i=1}^{m} \left(p(y^{(i)}; \pi) \prod_{d} p\left(x_d^{(i)}; \theta_{y^{(i)}d} \right) \right)$$

$$= \prod_{i=1}^{m} \left(\prod_{c} \left(\pi_c^{\mathbf{I}\{y_i = c\}} \prod_{j} p(x_{ij} | \theta_{jc})^{\mathbf{I}\{y_i = c\}} \right) \right)$$

Write down log-likelihood and optimize…

Example: Naïve Bayes with Bernoulli Features

$j: feature,\ c: label,\ i: data$

$y \sim Categorical(\pi_c):\ p(y=c) = \pi_c$

$$p(y=1) = \pi_1$$
$$p(y=2) = \pi_2$$
$$p(y=3) = \pi_3 = 1 - \pi_1 - \pi_2$$

Q: how many parameters?

c-1+cj

y	x_1	x_2
1	0	1
3	1	0
3	1	1
2	0	0
1	1	0

$x|y \sim Bernoulli(\theta_{jc}):\ p(x|y) = \theta_{jc}{}^{x}\,(1-\theta_{jc})^{1-x}$

$x_{j=1}|y=1 \sim Bernoulli(\theta_{j=1,c=1})$ $\quad x_{j=2}|y=1 \sim Bernoulli(\theta_{j=2,c=1})$

$x_{j=1}|y=2 \sim Bernoulli(\theta_{j=1,c=2})$ $\quad x_{j=2}|y=2 \sim Bernoulli(\theta_{j=2,c=2})$

$x_{j=1}|y=3 \sim Bernoulli(\theta_{j=1,c=3})$ $\quad x_{j=2}|y=3 \sim Bernoulli(\theta_{j=2,c=3})$

$$\prod_{i=1}^{5}\Big(\Big(\pi_{c=1}^{I(y_i=1)}\cdot p(x_{i,j=1}|\theta_{j=1,c=1})\cdot p(x_{i,j=2}|\theta_{j=2,c=1})\Big)\cdot\Big(\pi_{c=2}^{I(y_i=2)}\cdot p(x_{i,j=1}|\theta_{j=1,c=2})\cdot p(x_{i,j=2}|\theta_{j=2,c=2})\Big)\cdot$$
$$\Big(\pi_{c=3}^{I(y_i=3)}\cdot p(x_{i,j=1}|\theta_{j=1,c=3})\cdot p(x_{i,j=2}|\theta_{j=2,c=3})\Big)\Big)$$

Bernoulli Naïve Bayes MLE

Let $m_c := \sum_{i=1}^{m} \mathbf{I}\{y^{(i)} = c\}$ be number of training examples in class c then,

$$\sum_{i=1}^{m} \log p(\mathcal{D}; \pi, \theta) = \sum_{c=1}^{C} m_c \log \pi_c + \sum_{c=1}^{C} \sum_{i:y^{(i)}=c} \sum_{d=1}^{D} \log p\left(x_d^{(i)}; \theta_{cd}\right)$$

Log-likelihood function is concave in all parameters so…

1. Take derivatives with respect to π and θ separately.
2. Set derivatives to zero and solve

$$\hat{\pi}_c = \frac{m_c}{m}$$

Fraction of training examples from class c

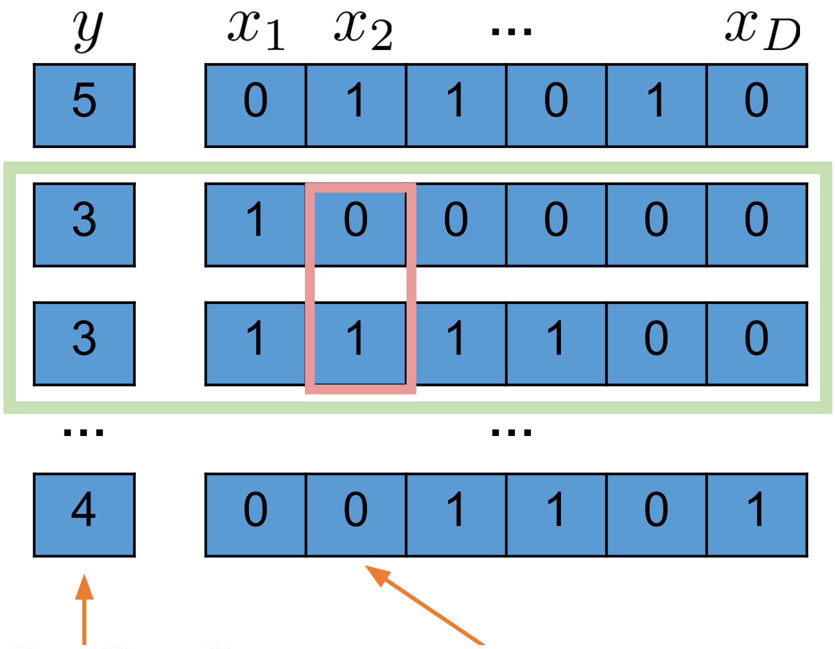
$$\hat{\theta}_{cd} = \frac{m_{cd}}{m_c}$$

Number of “heads” in training set from class c

$$m_{cd} = \sum_{i=1}^{m} I\{y^{(i)} = c, x_d^{(i)} = 1\}$$

Example: Naïve Bayes with Bernoulli Features

Analogy:



$$\hat{\pi}_c = \frac{m_c}{m}$$

$$\hat{\theta}_{cd} = \frac{m_{cd}}{m_c}$$

$$m_{cd} = \sum_{i=1}^{m} I\{y^{(i)} = c, x_d^{(i)} = 1\}$$

$$\widehat{\pi}_3 = \frac{\#\, number\ of\ y = 3}{\#\, total\ data}$$

$$\widehat{\theta}_{3,2} = \frac{\#\, number\ of\ y = 3\ and\ x_2 = 1}{\#\, number\ of\ y = 3}$$

Bernoulli Naïve Bayes: making prediction

$$\hat{\pi}_c = \frac{m_c}{m} \qquad \hat{\theta}_{cd} = \frac{m_{cd}}{m_c}$$

Given one data point, it has 4 features (input), compare the probabilities:

$$p(x_1, x_2, x_3, x_4, y = 0) = p(y = 0) \cdot p(x_1, x_2, x_3, x_4 | y = 0)$$

$$= p(y = 0) \cdot p(x_1 | y = 0) \cdot p(x_2 | y = 0) \cdot p(x_3 | y = 0) \cdot p(x_4 | y = 0)$$

$$p(x_1, x_2, x_3, x_4, y = 1) = p(y = 1) \cdot p(x_1, x_2, x_3, x_4 | y = 1)$$

$$= p(y = 1) \cdot p(x_1 | y = 1) \cdot p(x_2 | y = 1) \cdot p(x_3 | y = 1) \cdot p(x_4 | y = 1)$$

Naïve Bayes in Scikit-learn

Scikit-learn has separate classes each feature type

`sklearn.naive_bayes.GaussianNB`

Real-valued features

`sklearn.naive_bayes.MultinomialNB`

Discrete K-valued feature counts (e.g. multiple die rolls)

`sklearn.naive_bayes.BernoulliNB`

Binary features (e.g. coinflip)

`sklearn.naive_bayes.CategoricalNB`

Discrete K-valued features (e.g. single die roll)

For large training data that don't fit in memory use Scikit-learn's out-of-core learning

https://scikit-learn.org/stable/modules/naive_bayes.html