FYS-2021 Machine Learning

Bayes Rule and Classification (Part 2)

Slides by Stine Hansen,
Lecture by Elisabeth Wetzer

UiT The Arctic University of Norway

Repetition Random variables

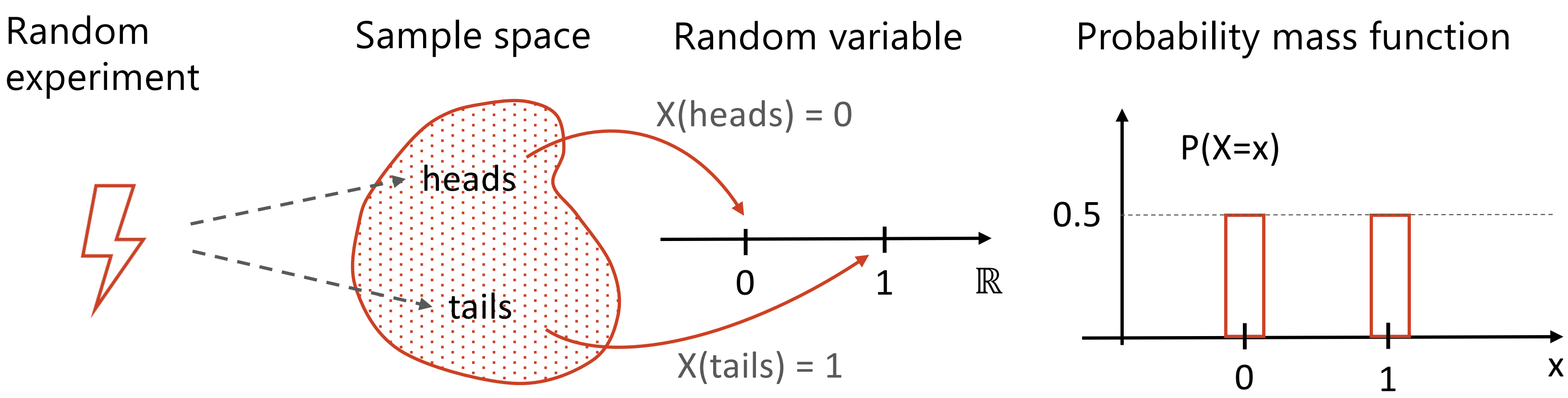
$X = \begin{cases} 0 \text{ if heads} \\ 1 \text{ if tails} \end{cases}$

A random variable is a function that assigns a value to each outcome of a random experiment.

coin toss

pmf

A probability distribution describes how the probabilities are distributed over the possible values of a random variable.



Repetition Bayes Decision rule

$$\underbrace{p(C_i|x)}_{\text{posterior}} = \frac{\overbrace{p(x|C_i)}^{\text{Likelihood}}\ \overbrace{p(C_i)}^{\text{Prior}}}{\underbrace{p(x)}_{\text{Evidence}}}$$

$$x \rightarrow C_1 \quad if \qquad p(C_1|x) > p(C_2|x)$$

$$\frac{p(x|C_1)\,p(C_1)}{p(x)} > \frac{p(x|C_2)\,p(C_2)}{p(x)}$$

$$p(x|C_1)\,p(C_1) > p(x|C_2)\,p(C_2)$$

$$\boxed{x \rightarrow C_i \quad if \quad p(x|C_i)\,p(C_i) > p(x|C_j)\,p(C_j)\ \forall\, j \neq i}$$

Repetition Estimating $p(C_i)$ and $p(x|C_i)$ from data

To train a Bayes classification model = To estimate $p(C_i)$ and $p(x|C_i)$ from training data $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^{N}$!

Scaling Shape/position

Prior class probabilities, $p(C_i)$

is estimated as the frequency of class C_i in the training data:

$$p(C_i) = \frac{\#\mathcal{D}\{y^i = C_i\}}{N}$$

Number of training samples belonging to class C_i

Likelihood terms, $p(x|C_i)$

is estimated via **maximum likelihood**!

Repetition Two-class univariate Gaussian Bayes classifier

Training

MLEs for likelihood terms:

$\hat{\mu}_{+} = \frac{1}{N}\sum_{j=1}^{N} x^{j}$ $\hat{\mu}_{\div} = \frac{1}{M}\sum_{j=1}^{M} x^{j}$

$\hat{\sigma}_{+}^{2} = \frac{1}{N}\sum_{j=1}^{N}\left(x^{j}-\mu\right)^{2}$ $\hat{\sigma}_{\div}^{2} = \frac{1}{M}\sum_{j=1}^{M}\left(x^{j}-\mu\right)^{2}$

Prior class probabilities:

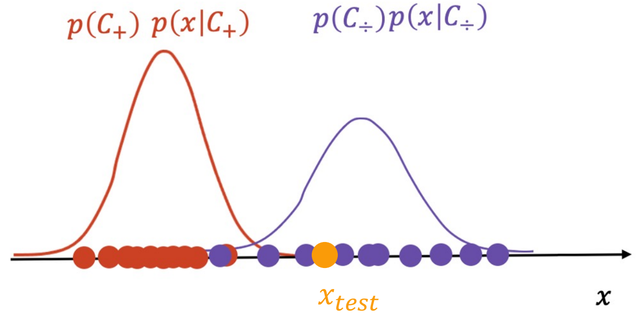
$\hat{p}(C_{+}) = \frac{N}{N+M}$ $\hat{p}(C_{\div}) = \frac{M}{N+M}$

Testing

Discriminant functions:

$$g_{+}(x_{test}) = p(x_{test}|C_{+})p(C_{+}) = \left(\frac{1}{\sqrt{2\pi}\,\hat{\sigma}_{+}} e^{-\frac{1}{2}\left(\frac{x_{test}-\hat{\mu}_{+}}{\hat{\sigma}_{+}}\right)^{2}}\right)\left(\frac{N}{N+M}\right)$$

$$g_{-}(x_{test}) = p(x_{test}|C_{-})p(C_{-}) = \left(\frac{1}{\sqrt{2\pi}\,\hat{\sigma}_{\div}} e^{-\frac{1}{2}\left(\frac{x_{test}-\hat{\mu}_{\div}}{\hat{\sigma}_{\div}}\right)^{2}}\right)\left(\frac{M}{N+M}\right)$$



What about multivariate data?

Tuesday: Focused on univariate data. However, a richer representation can lead to more accurate models.

Today: Extend the Bayes classifier to multivariate data!

Road map

1. Probabilistic thinking and Random vectors
2. Feature types
3. Multivariate Bayes classifier
4. The naïve Bayes assumption

   Break

5. (Multivariate) Bernoulli Naïve Bayes Classifier
6. Practical application: Text classification
7. Practical connection (when implementing)

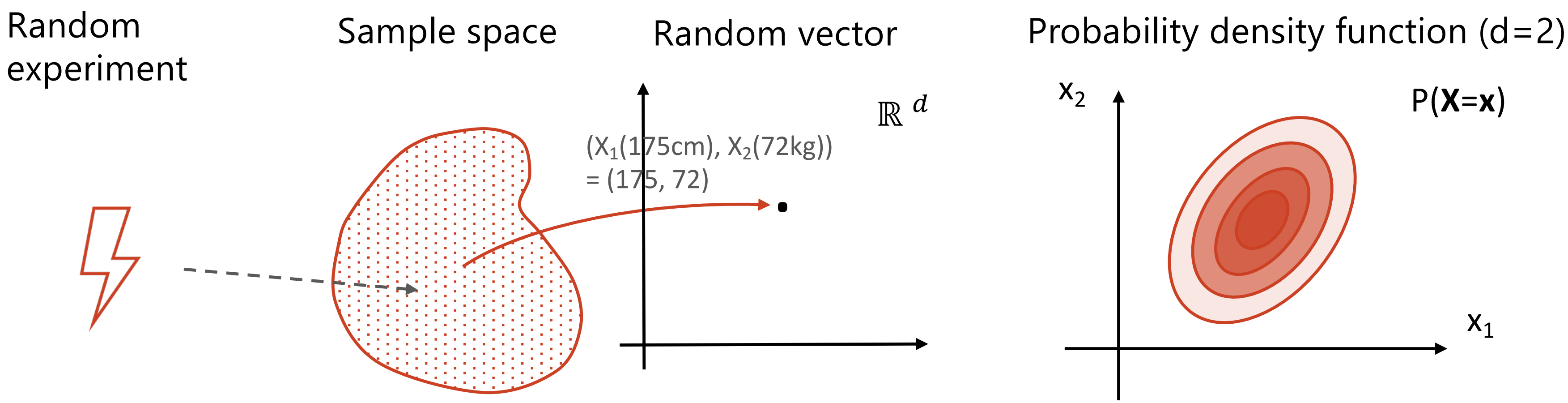
Probabilistic thinking: Random vectors

$\boldsymbol{X} = [X_1, X_2]$

A random vector is a collection of random variables and assigns a real valued vector to each outcome of a random experiment.

sample a random individual and measure height/weight

A joint probability distribution describes how the probabilities are distributed over the possible combinations of values of the random variables.



Multivariate data: Feature types

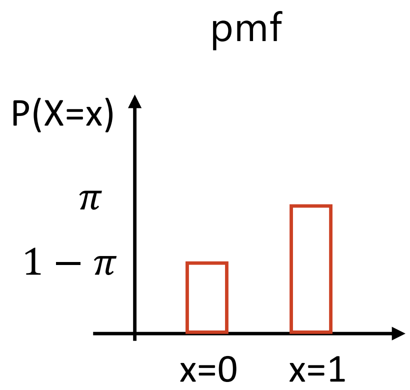
Continuous variables

such as height, weight, time, etc.

→ FYS-3012 Pattern Recognition

Discrete variables

such as number of siblings, hair color, drinks coffee (Y/N), etc.



Ex: Binary features, $X \in \{0,1\}$ → Bernoulli(π) distribution:

$$p_\pi(x) = \begin{cases} \pi & if \quad x = 1 \\ 1-\pi & if \quad x = 0 \end{cases} \leftrightarrow p_\pi(x) = \pi^x(1-\pi)^{1-x} \text{ for } x \in \{0,1\}$$

Notation

Description	Notation
Random variable	X
Realization of random variable	x
Set of N realizations (samples)	$\{x^1, x^2, \ldots, x^N\}$
Class i	C_i
Model parameters	θ
Number of features/variables	d
Random vector	$\mathbf{X} = [X_1, X_2, \ldots, X_d]$
Realization of random vector	$\mathbf{x} = [x_1, x_2, \ldots, x_d]$
Set of N realizations (samples)	$\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$
*k*th feature in the *j*th sample	x_k^j

Multivariate Bayes classifier

same decision rule as before

$$\boxed{\boldsymbol{x} \rightarrow C_i \quad if \quad p(\boldsymbol{x}|C_i)\, p(C_i) > p\left(\boldsymbol{x}\middle|C_j\right) p\left(C_j\right) \forall\, j \neq i}$$

need to estimate $p(C_i)$ and $p(\boldsymbol{x}|C_i)$

same as before

need to make some assumptions!

Estimating $p(\boldsymbol{x}|C_i)$

For $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{d}}$

$$p(\boldsymbol{x}|C_i) = p(x_1, x_2, \ldots, x_d|C_i)$$

$$= p(x_1|C_i) \cdot p(x_2|C_i, x_1) \cdot p(x_3|C_i, x_1, x_2) \cdot \ldots \cdot p(x_d|C_i, x_1, x_2, \ldots, x_{d-1})$$ chain rule

$$= \prod_{k=1}^{d} p(x_k|C_i)$$ Naïve Bayes assumption: Assuming conditional independence

Assuming N observed samples from class C_i, $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$

$$\hat{\theta}_{MLE} = arg\max_{\theta} L(\theta|\mathcal{D})$$

log(a*b) = log(a) + log(b)

$$L(\theta|\mathcal{D}) = \sum_{j=1}^{N} \log p_\theta(\boldsymbol{x}^j|C_i) = \sum_{j=1}^{N} \log\left[\prod_{k=1}^{d} p_{\theta_k}(x_k^j|C_i)\right] = \sum_{k=1}^{d} \sum_{j=1}^{N} \log p_{\theta_k}(x_k^j|C_i)$$

as before

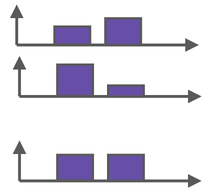
log likelihood for the multivariate case!

Estimating $p(\boldsymbol{x}|C_i)$ when features are Bernoulli distributed

log likelihood, $L(\theta|\mathcal{D})$

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \sum_{k=1}^{d} \sum_{j=1}^{N} \log p_{\theta_k}(x_k^j|C_i) \quad \rightarrow \text{solve:} \quad \frac{\partial}{\partial\theta} L(\theta|\mathcal{D}) = 0$$

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_d \end{bmatrix} \begin{matrix} \sim\text{Bernoulli}(\pi_1) \\ \sim\text{Bernoulli}(\pi_2) \\ \vdots \\ \sim\text{Bernoulli}(\pi_d) \end{matrix}$$



Each feature is Bernoulli distrubuted

Example: Assume we have N observed samples from class C_i, $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^N\}$, and that the features are Bernoulli distributed $p_{\theta_k}(x_k|C_i) \sim Bernoulli(\pi_k)$

$$L(\theta|\mathcal{D}) = \sum_{k=1}^{d} \sum_{j=1}^{N} \log p_{\theta_k}(x_k^j|C_i)$$

$$= \sum_{k=1}^{d} \sum_{j=1}^{N} \log\left[\pi_k^{x_k^j}(1-\pi_k)^{1-x_k^j}\right]$$

$$= \sum_{k=1}^{d} \sum_{j=1}^{N} \left(\log\left[\pi_k^{x_k^j}\right] + \log\left[(1-\pi_k)^{1-x_k^j}\right]\right)$$

$$= \sum_{k=1}^{d} \sum_{j=1}^{N} \left(x_k^j \log \pi_k + (1-x_k^j)\log(1-\pi_k)\right)$$

Here: θ=[π_1, π_2, …, π_d]

$$\frac{\partial}{\partial \pi_m} L(\theta|\mathcal{D}) = 0 \rightarrow \quad \hat{\pi}_m = \frac{1}{N}\sum_{j=1}^{N} x_m^j$$

Probability of feature m taking the value 1 in the observed samples!

Road map

1. Probabilistic thinking and Random vectors
2. Feature types
3. Multivariate Bayes classifier
4. The naïve Bayes assumption

Break

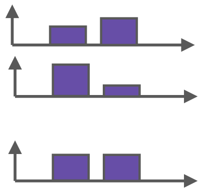
5. (Multivariate) Bernoulli Naïve Bayes Classifier
6. Practical application: Text classification
7. Practical connection (when implementing)

Estimating $p(\boldsymbol{x}|C_i)$ when features are Bernoulli distributed

log likelihood, $L(\theta|\mathcal{D})$

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \sum_{k=1}^{d}\sum_{j=1}^{N} \log p_{\theta_k}(x_k^j|C_i) \quad \rightarrow \text{solve:} \quad \frac{\partial}{\partial\theta} L(\theta|\mathcal{D}) = 0$$

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_d \end{bmatrix} \begin{matrix} \sim \text{Bernoulli}(\pi_1) \\ \sim \text{Bernoulli}(\pi_2) \\ \vdots \\ \sim \text{Bernoulli}(\pi_d) \end{matrix}$$



Each feature is Bernoulli distrubuted

Example: Assume we have N observed samples from class C_i, $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^N\}$, and that the features are Bernoulli distributed $p_{\theta_k}(x_k|C_i) \sim Bernoulli(\pi_k)$

$$L(\theta|\mathcal{D}) = \sum_{k=1}^{d}\sum_{j=1}^{N} \log p_{\theta_k}(x_k^j|C_i)$$

$$= \sum_{k=1}^{d}\sum_{j=1}^{N} \log\left[\pi_k^{x_k^j}(1-\pi_k)^{1-x_k^j}\right]$$

$$= \sum_{k=1}^{d}\sum_{j=1}^{N} \left(\log\left[\pi_k^{x_k^j}\right] + \log\left[(1-\pi_k)^{1-x_k^j}\right]\right)$$

$$= \sum_{k=1}^{d}\sum_{j=1}^{N} \left(x_k^j \log \pi_k + (1-x_k^j)\log(1-\pi_k)\right)$$

Here: $\theta=[\pi_1, \pi_2, \dots, \pi_d]$

$$\frac{\partial}{\partial \pi_m} L(\theta|\mathcal{D}) = 0 \rightarrow \quad \hat{\pi}_m = \frac{1}{N}\sum_{j=1}^{N} x_m^j$$

Probability of feature m taking the value 1 in the observed samples!

(Multivariate) Bernoulli Naïve Bayes classifier

Putting it all together

Assuming we have N training samples from class C_+, $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$, and M training samples from class $C_\div$, $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^M\}$, and that the features are Bernoulli distributed, $p(x_k|C_+) \sim Bernoulli(\pi_k)$, and $p(x_k|C_-) \sim Bernoulli(\pi_k)$, for $k = 1, \ldots, d$.

Training:

MLEs for likelihood terms for each feature $m = 1, \ldots, d$ are given by

$$\hat{\pi}_m = \frac{1}{N}\sum_{j=1}^{N} x_m^j \qquad \hat{\pi}_m = \frac{1}{M}\sum_{j=1}^{M} x_m^j$$

Prior class probability estimates are given by

$$p(C_+) = \frac{N}{N+M} \qquad p(C_\div) = \frac{M}{N+M}$$

(Multivariate) Bernoulli Naïve Bayes classifier

Test time (inference)

$$\boxed{\begin{gathered}\boldsymbol{x} \to C_i \quad if \\ p(\boldsymbol{x}|C_i)\, p(C_i) > p(\boldsymbol{x}|C_j)\, p(C_j)\ \forall\, j \neq i\end{gathered}}$$

Decision rule

When we have some test sample $\boldsymbol{x}_{test}$ that we want to classify, we need to evaluate

$$g_i(\boldsymbol{x}_{test}) = p(\boldsymbol{x}_{test}|C_i)\, p(C_i)$$ (discriminant function)

for all i. Then chose class C_i if

$$g_i(\boldsymbol{x}_{test}) = \max_k g_k(\boldsymbol{x}_{test})$$

$$g_+(\boldsymbol{x}^{test}) = \left(\prod_{k=1}^{d} \hat{\pi}_k{}^{x_k^{test}} (1-\hat{\pi}_k)^{1-x_k^{test}}\right)\left(\frac{N}{N+M}\right)$$

$$g_-(x_{test}) = \left(\prod_{k=1}^{d} \hat{\pi}_k{}^{x_k^{test}} (1-\hat{\pi}_k)^{1-x_k^{test}}\right)\left(\frac{M}{N+M}\right)$$

(Multivariate) Bernoulli Naïve Bayes classifier

Test time (inference)

$$\boxed{\begin{array}{c} \boldsymbol{x} \rightarrow C_i \quad if \\ p(\boldsymbol{x}|C_i)\, p(C_i) > p(\boldsymbol{x}|C_j)\, p(C_j)\; \forall\, j \neq i \end{array}}$$

Decision rule

When we have some test sample $\boldsymbol{x}_{test}$ that we want to classify, we need to evaluate

$$g_i(\boldsymbol{x}_{test}) = \log[p(\boldsymbol{x}_{test}|C_i)\, p(C_i)]$$

(discriminant function)

for all i. Then chose class C_i if

$$g_i(\boldsymbol{x}_{test}) = \max_k g_k(\boldsymbol{x}_{test})$$

$$g_+(\boldsymbol{x}^{test}) = \sum_{k=1}^{d}[x_k^{test} \log(\hat{\pi}_k) + (1 - x_k^{test}) \log(1 - \hat{\pi}_k)] + \log\frac{N}{N+M}$$

$$g_-(x_{test}) = \sum_{k=1}^{d}[x_k^{test} \log(\hat{\pi}_k) + (1 - x_k^{test}) \log(1 - \hat{\pi}_k)] + \log\frac{M}{N+M}$$

Important: What happens when. $\hat{\pi}_k = \frac{1}{N}\sum_j x_k^j$ is 0 or 1?

Problem with zero/one probabilities

$$g_i(\boldsymbol{x}) = \sum_{k=1}^{d}[x_k \log(\hat{\pi}_k) + (1 - x_k)\log(1 - \hat{\pi}_k)] + \log\frac{N}{N+M}$$

log(0) is not defined!

Can do a count-adjustment (Laplace smoothing) to alleviate the problem by letting

$$\hat{\pi}_m = \frac{\sum_{j=1}^{N} x_m^j + \alpha}{N + (2\alpha)}$$

Practical Application: Text classification

E.g. classify which category a news article belongs to?
which e-mails are spam?
which comments contain hate speech?

The **first step** is to represent the text documents numerically!

$\rightarrow \boldsymbol{x} = [x_1, x_2, \dots, x_d]$

E.g. “Bag-of-Words” representation

The Bag-of-Words (BoW) model

is a framework for representing text data as numerical vectors.

Example: Binary BoW representations.

I like cats.

Dogs are friendly.

Cats and dogs are pets.

Documents

Unique words (vocabulary)

I	like	cats	dogs	are	friendly	and	pets
1	1	1	0	0	0	0	0
0	0	0	1	1	1	0	0
0	0	1	1	1	0	1	1

each word in the vocabulary is present (1) or absent (0) in the respective sentence

Binary BoW representation

Bag-of-Words representations

Documents are represented as d-dimensional feature vectors $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{d}}$, where d is the number of words in the vocabulary.

+ simple and effective
- disregards order and structure

(Multivariate) Bernoulli Naïve Bayes classifier for text classification

Assuming we have the d-dimensional binary BoW-representations of N (training) samples from class C_+, $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^N\}$, and M training samples from class $C_\div$, $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^M\}$, and that the features are Bernoulli distributed, $p(x_k|C_i) \sim Bernoulli(\pi_k)$.

Remember: Naïve Bayes assumption
$p_\theta(\boldsymbol{x}|C_i) = \prod_{k=1}^{d} p_{\theta_k}(x_k|C_i)$

(what does this mean in the context of text classification?)

How many parameters do we have to estimate?

Answer: $(d+1) * 2$ parameters

$\hat{\pi}_m = \frac{1}{N}\sum_{j=1}^{N} x_m^j$ and $\hat{\pi}_m = \frac{1}{M}\sum_{j=1}^{M} x_m^j$ for $m = 1, \dots, d$

Probability of feature (word) m appearing in documents in class $C_{+/\div}$

and $p(C_+) = \frac{N}{N+M}$ and $p(C_\div) = \frac{M}{N+M}$

Probability of class $C_{+/\div}$

(Multivariate) Bernoulli Naïve Bayes classifier for text classification: Example

Labeled training data:

I enjoyed the movie.

The book was poor.

The movie was poor.

The book was good.

The movie was bad.

I disliked the book.

The movie was good.

The book was excellent.

Unlabeled test data:

I enjoyed the book.

(Multivariate) Bernoulli Naïve Bayes classifier for text classification: Example

	bad	book	disliked	enjoyed	excellent	good	I	movie	poor	the	was	
I enjoyed the movie.	0	0	0	1	0	0	1	1	0	1	0	$= \boldsymbol{x}^1$
The book was excellent.	0	1	0	0	1	0	0	0	0	1	1	$= \boldsymbol{x}^2$
The movie was good.	0	0	0	0	0	1	0	1	0	1	1	$= \boldsymbol{x}^3$
The book was good.	0	1	0	0	0	1	0	0	0	1	1	$= \boldsymbol{x}^4$
The movie was bad.	1	0	0	0	0	0	0	1	0	1	1	$= \boldsymbol{x}^1$
I disliked the book.	0	1	1	0	0	0	1	0	0	1	0	$= \boldsymbol{x}^2$
The book was poor.	0	1	0	0	0	0	0	0	1	1	1	$= \boldsymbol{x}^3$
The movie was poor.	0	0	0	0	0	0	0	1	1	1	1	$= \boldsymbol{x}^4$
I enjoyed the book.	0	1	0	0	1	0	1	0	0	1	0	$= \boldsymbol{x}^{test}$

Practical connection

0. [If not done] Split data into train/test

On training data:

1. Sort samples according to labels (class)
2. For each class: Compute estimates for class prior as class frequency
3. For each class: Assume distribution of likelihood and compute corresponding $\hat{\theta}_{MLE}$ **(for each feature)**

On test data:

1. For each class: Compute discriminant function
2. Assign samples to class with maximum discriminant function
3. Compare predictions to labels and compute your confusion matrix

Road map

1. Probabilistic thinking and Random vectors
2. Feature types
3. Multivariate Bayes classifier
4. The naïve Bayes assumption

   Break

5. (Multivariate) Bernoulli Naïve Bayes Classifier
6. Practical application: Text classification
7. Practical connection (when implementing)

Conclusion

- Bayes classifier for univariate data (Tuesday)
  - Assume distribution for likelihoods
  - Learn class priors and MLE parameters for each class

- Naïve Bayes classifier for multivariate data
  - Assume conditional independence (Bayes naïve assumption)
  - Assume distribution for likelihoods
  - Learn class priors and MLE parameters for *each feature* for each class

Additional

- Multivariate (continuous) example on blackboard

- Some features are more descriptive than others
  - Very small likelihoods for on feature in a certain class weigh heavily to the discriminant function

- Use cross-validation
  - Cross validation can be used for feature selection (not only model/classifier selection)
  - Choice of features can be considered similar to tuning parameters

- Google Colab Example 1
- Google Colab Example 2