Introduction to Information Retrieval and Text Mining

Maximum Entropy Classifier, Feature Selection, Vector Space Classification

Roman Klinger

Institute for Natural Language Processing, University of Stuttgart

2021-01-07

Overview

Outline

Maximum a posteriori class

- Goal in Naive Bayes classification is to find the "best" class
- The best class is the most likely class "maximum a posteriori" (MAP) c_{map}:

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \ \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c)$$

- Classification rule:

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} [\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k|c)]$$

To avoid zeros: Add-one smoothing

- Before:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

- Now: Add one to each count to avoid zeros:

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B}$$

- B is the number of bins – the number of different words or the size of the vocabulary $|V| = M$

Time complexity of Naive Bayes

mode	time complexity
training	$\Theta(\lvert\mathbb{D}\rvert L_{\text{ave}} + \lvert\mathbb{C}\rvert\lvert V\rvert)$
testing	$\Theta(L_{\text{a}} + \lvert\mathbb{C}\rvert M_{\text{a}}) = \Theta(\lvert\mathbb{C}\rvert M_{\text{a}})$

- L_{ave}: average length of a training doc, L_{a}: length of the test doc, M_{a}: number of distinct terms in the test doc, $\mathbb{D}$: training set, V: vocabulary, $\mathbb{C}$: set of classes
- $\Theta(|\mathbb{D}|L_{\text{ave}})$ is the time it takes to compute all counts.
- $\Theta(|\mathbb{C}||V|)$ is the time it takes to compute the parameters from the counts.
- Generally: $|\mathbb{C}||V| < |\mathbb{D}|L_{\text{ave}}$
- Test time is also linear (in the length of the test document).
- Thus: Naive Bayes is linear in the size of the training set (training) and the test document (testing). This is optimal.

Violation of Naive Bayes independence assumptions

- Conditional independence:

$$P(\langle t_1, \ldots, t_{n_d} \rangle | c) = \prod_{1 \leq k \leq n_d} P(X_k = t_k | c)$$

- Positional independence:

$$\hat{P}(X_{k_1} = t | c) = \hat{P}(X_{k_2} = t | c)$$

- The independence assumptions do not really hold!
- How can Naive Bayes work if it makes such inappropriate assumptions?

Naive Bayes is not so naive

- Naive Bayes models have won some shared tasks
- More robust to nonrelevant features than some more complex learning methods
- More robust to concept drift (changing of definition of class over time) than some more complex learning methods
- Better than methods like decision trees when we have many equally important features
- A good dependable baseline for text classification (but not the best)
- Optimal if independence assumptions hold (never true for text, but true for some domains)
- Very fast
- Low storage requirements

Evaluation of Classification: Macro and Micro Average

Calculate Micro and Macro F Measure

DocId	True Class	Predicted Class
1	Europe	Europe
2	Europe	Europe
3	Europe	Asia
4	Asia	Europe
5	Asia	Asia
6	Europe	Europe
7	Europe	Europe

- $P = TP/(TP + FP)$
- $R = TP/(TP + FN)$
- $F = 2PR/(P + R)$

	is Eur.	is not Eur.
pred Eur.	TP=4	FP=1
not pred Eur.	FN=1	TN=1

	is Asia	is not Asia
pred Asia	TP=1	FP=1
not pred Asia	FN=1	TN=4

Micro F= 0.7

Macro F= 0.65

Take-away today

- The problem of overcounting in Naive Bayes
- Maximum Entropy Classifier
- Overfitting and the Bias-Variance Dilemma
- Feature Selection
- Vector space classification

Outline

Problems with correlated features (I)

Document Classification Example

(example adapted from Chris Manning's slides)

Europe				Asia			
Monaco Monaco	Monaco	Monaco Monaco	Monaco Hong Kong	Monaco Hong Kong	Monaco	Hong Kong	Hong Kong

Model Parameters

- $p(\text{Europe}) = \frac{1}{2}$; $p(\text{Asia}) = \frac{1}{2}$
- $p(\text{Monaco} \mid \text{Europe}) = \frac{6}{8}$; $p(\text{Hong} \mid \text{Europe}) = \frac{1}{8}$; $p(\text{Kong} \mid \text{Europe}) = \frac{1}{8}$
- $p(\text{Monaco} \mid \text{Asia}) = \frac{2}{8} = \frac{1}{4}$; $p(\text{Hong} \mid \text{Asia}) = \frac{3}{8}$; $p(\text{Kong} \mid \text{Asia}) = \frac{3}{8}$

Inference

Given a document with only the term “Monaco”:

$p(\text{Europe}) \cdot p(\text{Monaco} \mid \text{Europe}) = \frac{1}{2} \cdot \frac{3}{4} = \frac{3}{8}$

$p(\text{Asia}) \cdot p(\text{Monaco} \mid \text{Asia}) = \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8}$

Problems with correlated features (II)

Document Classification Example

(example adapted from Chris Manning's slides)

Europe				Asia			
Monaco Monaco	Monaco	Monaco Monaco	Monaco Hong Kong	Monaco Hong Kong	Monaco	Hong Kong	Hong Kong

Model Parameters

- $p(\text{Europe}) = \frac{1}{2}$; $p(\text{Asia}) = \frac{1}{2}$
- $p(\text{Monaco} \mid \text{Europe}) = \frac{3}{4}$; $p(\text{Hong} \mid \text{Europe}) = \frac{1}{8}$; $p(\text{Kong} \mid \text{Europe}) = \frac{1}{8}$
- $p(\text{Monaco} \mid \text{Asia}) = \frac{2}{8} = \frac{1}{4}$; $p(\text{Hong} \mid \text{Asia}) = \frac{3}{8}$; $p(\text{Kong} \mid \text{Asia}) = \frac{3}{8}$

Inference

Given a document with terms "Hong" and "Kong":

$p(\text{Europe}) \cdot p(\text{Hong} \mid \text{Europe}) \cdot p(\text{Kong} \mid \text{Europe}) = \frac{1}{2} \cdot \frac{1}{8} \cdot \frac{1}{8} = \frac{1}{128}$

$p(\text{Asia}) \cdot p(\text{Hong} \mid \text{Asia}) \cdot p(\text{Kong} \mid \text{Asia}) = \frac{1}{2} \cdot \frac{3}{8} \cdot \frac{3}{8} = \frac{9}{128}$

Problems with correlated features (III)

Document Classification Example

(example adapted from Chris Manning's slides)

Europe				Asia			
Monaco Monaco	Monaco	Monaco Monaco	Monaco Hong Kong	Monaco Hong Kong	Monaco	Hong Kong	Hong Kong

Inference

Given a document with terms "Hong" and "Kong":

$p(\text{Europe}) \cdot p(\text{Hong} \mid \text{Europe}) \cdot p(\text{Kong} \mid \text{Europe}) = \frac{1}{2} \cdot \frac{1}{8} \cdot \frac{1}{8} = \frac{1}{128}$

$p(\text{Asia}) \cdot p(\text{Hong} \mid \text{Asia}) \cdot p(\text{Kong} \mid \text{Asia}) = \frac{1}{2} \cdot \frac{3}{8} \cdot \frac{3}{8} = \frac{9}{128}$

- Given Hong and Kong, Asia is 9 times more probably than Europe!
- This is overcounting and can lead to wrong predictions
- What about a document d with Monaco, Hong, and Kong?
- $p(\text{Europe}|d) \propto \frac{1}{2}\frac{3}{4}\frac{1}{8}\frac{1}{8} = \frac{3}{512}$ vs. $p(\text{Asia}|d) \propto \frac{1}{2}\frac{1}{4}\frac{3}{8}\frac{3}{8} = \frac{9}{512}$

Outline

Generative vs. Discriminative Model

Idea: A simple classifier which does not have such problems.

- Naïve Bayes: $p(y, x_1, \dots x_n)$
  (joint probability)
- Maximum Entropy Classifier: $p(y \mid x_1, \dots x_n)$
  (conditional probability)

Joint

- Weights: just count

Conditional

- Maximize conditional likelihood
- Optimization process!

Maximum Entropy Classifier

positive weights

features $\in \{0,1\}$

$$p_\lambda(y \mid \mathbf{x}) = \frac{\exp \sum_i \lambda_i f_i(y, \mathbf{x})}{}$$

remember NB;

Words: money, buy

c: spam, ham

$p(m|spam)$ $p(buy|spam)$

$p(m|H)$ $p(b|H)$

Maximum Entropy Classifier

score

$$p_\lambda(y \mid \mathbf{x}) = \frac{\exp \sum_i \lambda_i f_i(y, \mathbf{x})}{\sum_{y'} \exp \sum_i \lambda_i f_i(y', \mathbf{x})}$$

sum of scores for all classes

Maximum Entropy Classifier

$$
\begin{aligned}
p_\lambda(y \mid \mathbf{x}) &= \frac{\exp \sum_i \lambda_i f_i(y, \mathbf{x})}{\sum_{y'} \exp \sum_i \lambda_i f_i(y', \mathbf{x})} \\
&= \frac{1}{Z(\mathbf{x})} \exp \sum_i \lambda_i f_i(y, \mathbf{x})
\end{aligned}
$$

where $Z(\mathbf{x}) = \sum_{y'} \exp \sum_i \lambda_i f_i(y', \mathbf{x})$

- $\mathbf{x}$: Evidence, given data
- y: Class variable to be predicted
- $f_i(y, \mathbf{x})$ Features (here: words with class)
- λ_i Parameters to be learned
- $Z(\mathbf{x})$ normalization, partition function

Feature Extraction example (I)

Example texts

- Europe: “Monaco”
- Asia “Hong Kong”
- Europe “Monaco Hong Kong”
- Asia “Hong Kong Monaco”

- Features: Occurrence of words “Monaco”, “Hong”, “Kong”

- Remember:
$$p_\lambda(y \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_i \lambda_i f_i(y, \mathbf{x})$$

- Features in model:
$$f_1(y, \mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Monaco}] \quad \lambda_1 = 7.44$$
$$f_2(y, \mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Monaco}] \quad \lambda_2 = -7.44$$
$$f_3(y, \mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Hong}] \quad \lambda_3 = -3.72$$
$$f_4(y, \mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Hong}] \quad \lambda_4 = 3.72$$
$$f_5(y, \mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Kong}] \quad \lambda_5 = -3.72$$
$$f_6(y, \mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Kong}] \quad \lambda_6 = 3.72$$

Feature Extraction example (II)

- $p_\lambda(y \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_i \lambda_i f_i(y, \mathbf{x})$

- Features in model:

$$
\begin{aligned}
f_1(y,\mathbf{x}) &= [y = \text{Europe} \wedge \mathbf{x} \ni \text{Monaco}] & \lambda_1 &= 7.44\\
f_2(y,\mathbf{x}) &= [y = \text{Asia} \wedge \mathbf{x} \ni \text{Monaco}] & \lambda_2 &= -7.44\\
f_3(y,\mathbf{x}) &= [y = \text{Europe} \wedge \mathbf{x} \ni \text{Hong}] & \lambda_3 &= -3.72\\
f_4(y,\mathbf{x}) &= [y = \text{Asia} \wedge \mathbf{x} \ni \text{Hong}] & \lambda_4 &= 3.72\\
f_5(y,\mathbf{x}) &= [y = \text{Europe} \wedge \mathbf{x} \ni \text{Kong}] & \lambda_5 &= -3.72\\
f_6(y,\mathbf{x}) &= [y = \text{Asia} \wedge \mathbf{x} \ni \text{Kong}] & \lambda_6 &= 3.72
\end{aligned}
$$

- Predict class for: $\mathbf{x} =$ “Let’s make a boat trip in Hong Kong.”
- Sum up relevant features, assuming it is Europe:

$$
\begin{aligned}
\sum_i \lambda_i f_i(\text{Europe}, \mathbf{x}) &= \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4 + \lambda_5 f_5 + \lambda_6 f_6\\
&= 7.44 \cdot 0 + 7.44 \cdot 0 - 3.72 \cdot 1 + 3.72 \cdot 0 - 3.72 \cdot 1 + 3.72 \cdot 0\\
&= -3.72 - 3.72\\
&= -7.44
\end{aligned}
$$

Feature Extraction example (III)

- $p_\lambda(y \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_i \lambda_i f_i(y, \mathbf{x})$

- Features in model:

$$
\begin{array}{ll}
f_1(y,\mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Monaco}] & \lambda_1 = 7.44 \\
f_2(y,\mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Monaco}] & \lambda_2 = -7.44 \\
f_3(y,\mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Hong}] & \lambda_3 = -3.72 \\
f_4(y,\mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Hong}] & \lambda_4 = 3.72 \\
f_5(y,\mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Kong}] & \lambda_5 = -3.72 \\
f_6(y,\mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Kong}] & \lambda_6 = 3.72
\end{array}
$$

- Predict class for: $\mathbf{x} =$ "Let's make a boat trip in Hong Kong."
- Sum up relevant features, assuming it is Europe:
  $\sum_i \lambda_i f_i(\text{Europe}, \mathbf{x}) = -7.44$
- Assume it is Asia:

$$
\begin{aligned}
\sum_i \lambda_i f_i(\text{Asia}, \mathbf{x}) &= \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4 + \lambda_5 f_5 + \lambda_6 f_6 \\
&= 7.44 \cdot 0 + 7.44 \cdot 0 - 3.72 \cdot 0 + 3.72 \cdot 1 - 3.72 \cdot 0 + 3.72 \cdot 1 \\
&= 7.44
\end{aligned}
$$

Feature Extraction example (IV)

- $p_\lambda(y \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_i \lambda_i f_i(y, \mathbf{x})$
- $Z(\mathbf{x}) = \sum_{y'} \exp \sum_i \lambda_i f_i(y', \mathbf{x})$
- $\sum_i \lambda_i f_i(\text{Europe}, \mathbf{x}) = -7.44$
- $\sum_i \lambda_i f_i(\text{Asia}, \mathbf{x}) = 7.44$

  - $\exp \sum_i \lambda_i f_i(\text{Europe}, \mathbf{x}) \approx 0.0005872852$
  - $\exp \sum_i \lambda_i f_i(\text{Asia}, \mathbf{x}) \approx 1702.75$

  - $p_\lambda(\text{Europe} \mid \mathbf{x}) \approx \frac{0.00059}{0.00059 + 1702.75} = \frac{0.00059}{1702.75059} \approx 0$
  - $p_\lambda(\text{Asia} \mid \mathbf{x}) \approx \frac{1702.75}{0.00059 + 1702.75} = \frac{1702.75}{1702.75059} \approx 1$

Short Exercise

$$f_1(y, \mathbf{x}) = [y = \text{Europe} \land \mathbf{x} \ni \text{Monaco}] \qquad \lambda_1 = 7.44$$
$$f_2(y, \mathbf{x}) = [y = \text{Asia} \land \mathbf{x} \ni \text{Monaco}] \qquad \lambda_2 = -7.44$$
$$f_3(y, \mathbf{x}) = [y = \text{Europe} \land \mathbf{x} \ni \text{Hong}] \qquad \lambda_3 = -3.72$$
$$f_4(y, \mathbf{x}) = [y = \text{Asia} \land \mathbf{x} \ni \text{Hong}] \qquad \lambda_4 = 3.72$$
$$f_5(y, \mathbf{x}) = [y = \text{Europe} \land \mathbf{x} \ni \text{Kong}] \qquad \lambda_5 = -3.72$$
$$f_6(y, \mathbf{x}) = [y = \text{Asia} \land \mathbf{x} \ni \text{Kong}] \qquad \lambda_6 = 3.72$$

Training instances

- Europe: “Monaco”
- Asia “Hong Kong”
- Europe “Monaco Hong Kong”
- Asia “Hong Kong Monaco”

- Predict class for: $\mathbf{x} =$“Monaco Hong Kong.” with ME and NB (without smoothing)

Short Exercise: Solution for Maximum Entropy

$$f_1(y, \mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Monaco}] \qquad \lambda_1 = 7.44$$
$$f_2(y, \mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Monaco}] \qquad \lambda_2 = -7.44$$
$$f_3(y, \mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Hong}] \qquad \lambda_3 = -3.72$$
$$f_4(y, \mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Hong}] \qquad \lambda_4 = 3.72$$
$$f_5(y, \mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Kong}] \qquad \lambda_5 = -3.72$$
$$f_6(y, \mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Kong}] \qquad \lambda_6 = 3.72$$

- Europe: $\exp(7.44 - 3.72 - 3.72) = \exp(0) = 1$
- Asia: $\exp(7.44 - 3.72 - 3.72) = \exp(0) = 1$
- $p(\text{Europe} \mid \mathbf{x}) = \frac{1}{2}$
- $p(\text{Asia} \mid \mathbf{x}) = \frac{1}{2}$

Short Exercise: Solution for Naive Bayes

Example texts

- Europe: “Monaco”
- Asia “Hong Kong”
- Europe “Monaco Hong Kong”
- Asia “Hong Kong Monaco”

- Priors: $p(\text{Europe}) = p(\text{Asia}) = 0.5$
- Term propabilities:
  - $p(\text{Monaco}|\text{Europe}) = \frac{1}{2}$ $p(\text{Hong}|\text{Europe}) = p(\text{Kong}|\text{Europe}) = \frac{1}{4}$
  - $p(\text{Monaco}|\text{Asia}) = \frac{1}{5}$ $p(\text{Hong}|\text{Asia}) = p(\text{Kong}|\text{Asia}) = \frac{2}{5}$
- Prediction for “Monaco Hong Kong”:
  - $p(\text{Europe}|\mathbf{x}) = \frac{1}{2}\frac{1}{2}\frac{1}{4}\frac{1}{4} = \frac{1}{64} = 0.015625$
  - $p(\text{Asia}|\mathbf{x}) = \frac{1}{2}\frac{1}{5}\frac{2}{5}\frac{2}{5} = \frac{4}{250} = 0.016$

$\Rightarrow$ Asia is more likely due to overcounting w/ NB, but not w/ ME!

Features

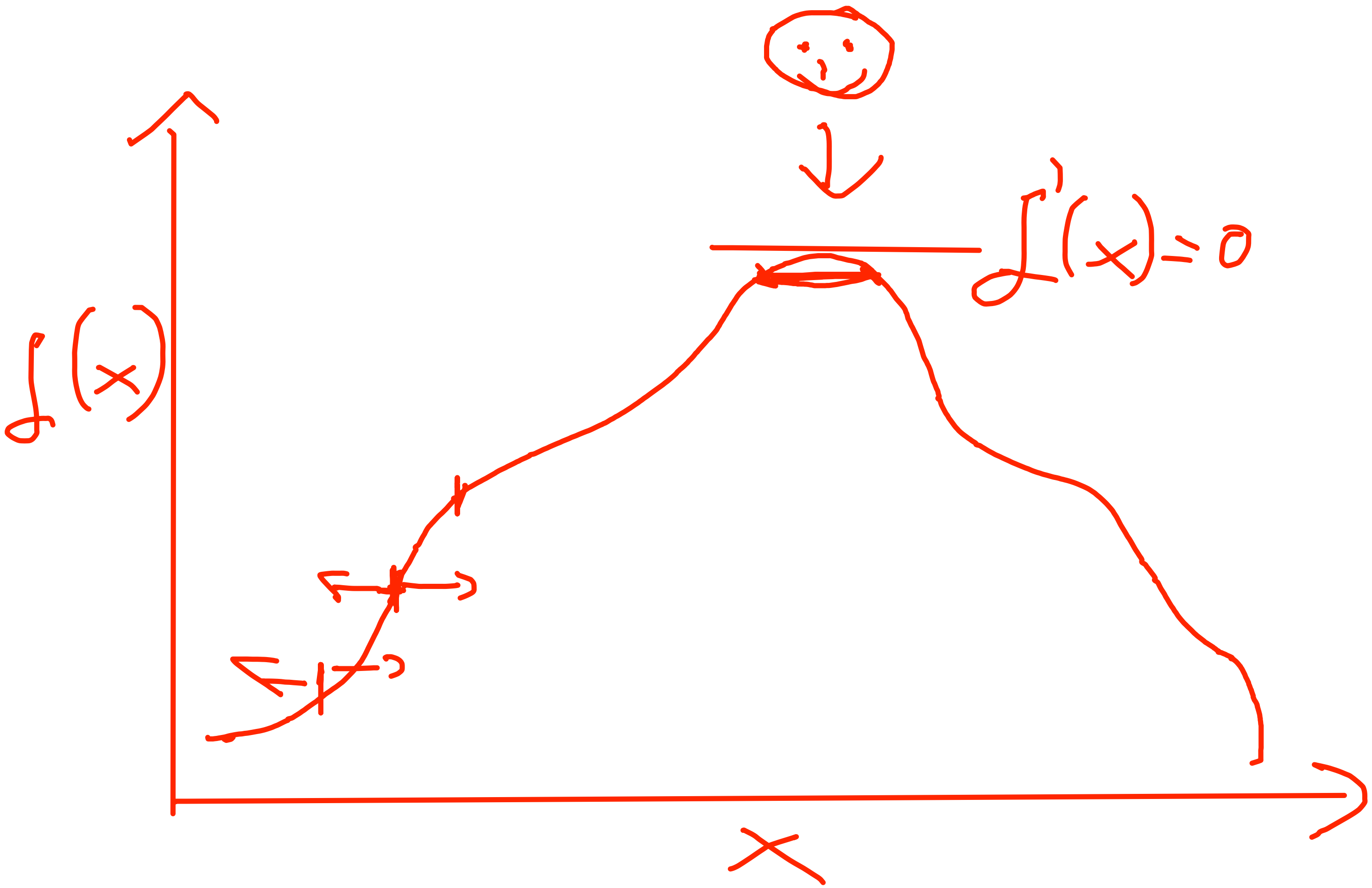
- Features are typically $f : Y \times X \rightarrow \mathbb{R}^{>0}$ or $f : Y \times X \rightarrow \{0, 1\}$
- Weights represent the importance of a class-feature combination
- Measure compatibility!
- Weights for correlated features are lower than for independent features (of same importance)
- When designing features, you can make use of correlations!

Features in text classification

- Words
- Occurrence of words in a dictionary
- Number of specific word class
- Bigrams, trigrams,...
- Number of sentences
- Meta data
- ...
- $\Rightarrow$ Good choice is application/data specific.

Outline

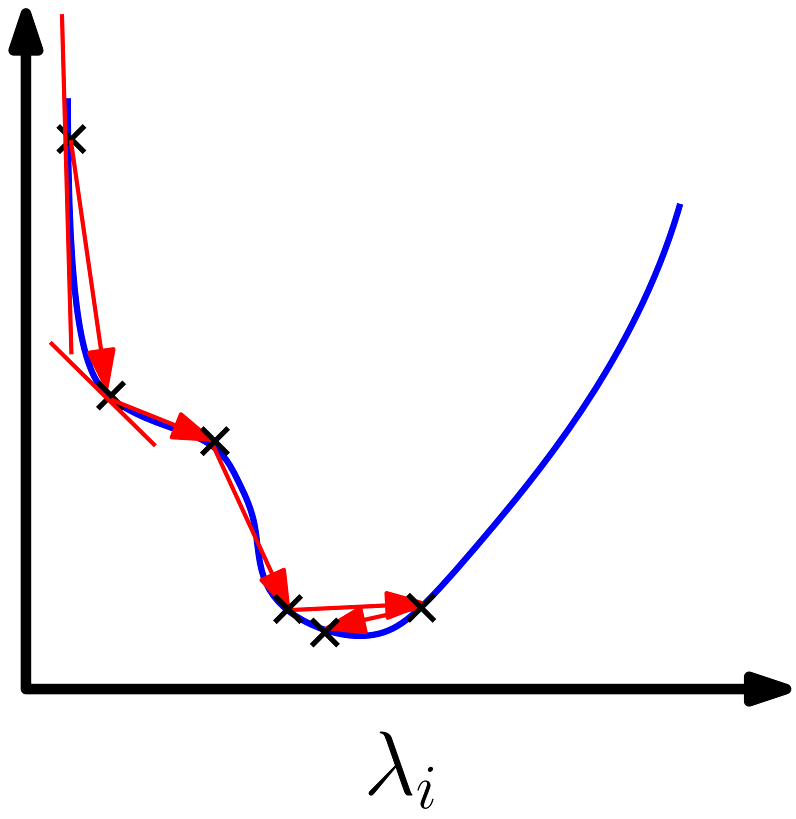
Iterative Optimization (very briefly)



Iterative Optimization (very briefly)

We need an iterative optimization method: gradient descent

- Initialize parameters λ randomly.
- Iterate:
  - Test how good performance is on training set.
    - If satisfied (e.g. improvement between iterations smaller than a predefined threshold): exit
  - Improve each parameter: $\lambda_i^{t+1} = \lambda_i^t - \nabla F(\lambda_i^t)$ with $\nabla F(\lambda_i^t)$ being the derivative of the objective F at λ_i^t.



Parameter Estimation (I)

- How to learn the parameters λ_i?
- What is the objective function to optimize?

$\Rightarrow$ Maximize the conditional log likelihood of the data, given the model.

$$\max_{\lambda}^{!} \log p_\lambda(Y \mid X) = \sum_{(y,\mathbf{x}) \in (Y,X)} \log p_\lambda(y \mid \mathbf{x})$$

$$= \sum_{(y,\mathbf{x}) \in (Y,X)} \log \frac{\exp \sum_i \lambda_i f_i(y,\mathbf{x})}{\sum_{y'} \exp \sum_i \lambda_i f_i(y',\mathbf{x})}$$

Parameter Estimation (II)

Reformulate a bit...

$$
\begin{aligned}
& \sum_{(y,x)\in(Y,X)} \log \frac{\exp \sum_i \lambda_i f_i(y,\mathbf{x})}{\sum_{y'} \exp \sum_i \lambda_i f_i(y',\mathbf{x})} \\
= & \sum_{(y,x)\in(Y,X)} \left[\log \exp \sum_i \lambda_i f_i(y,\mathbf{x}) - \log \sum_{y'} \exp \sum_i \lambda_i f_i(y',\mathbf{x}) \right] \\
= & \underbrace{\sum_{(y,x)\in(Y,X)} \log \exp \sum_i \lambda_i f_i(y,\mathbf{x})}_{\mathcal{A}_\lambda} - \underbrace{\sum_{(y,x)\in(Y,X)} \log \sum_{y'} \exp \sum_i \lambda_i f_i(y',\mathbf{x})}_{\mathcal{B}_\lambda} \\
= & \ \mathcal{A}_\lambda - \mathcal{B}_\lambda
\end{aligned}
$$

Derivative

$$\underbrace{\sum_{(y,x)\in(Y,X)} \log \exp \sum_i \lambda_i f_i(y,\mathbf{x})}_{\mathcal{A}_\lambda} - \underbrace{\sum_{(y,x)\in(Y,X)} \log \sum_{y'} \exp \sum_i \lambda_i f_i(y',\mathbf{x})}_{\mathcal{B}_\lambda}$$

Derivatives:

- $\dfrac{\partial \mathcal{A}_\lambda}{\partial \lambda_i} = \displaystyle\sum_{(y,x)\in(Y,X)} f_i(y,x)$
- $\dfrac{\partial \mathcal{B}_\lambda}{\partial \lambda_i} = \displaystyle\sum_{(y,x)\in(Y,X)} \sum_{y'} p_\lambda(y' \mid \mathbf{x}) f_i(y',x)$
- "Empirical feature count" – "Predicted feature count"
- Optimal: Both values are the same for all features!

Derivative Example (I)

Empirical feature count:

$$\sum_{(y,\mathbf{x})\in(Y,X)} f_i(y,\mathbf{x}) = \frac{\partial\mathcal{A}}{\partial\lambda_i}$$

Predicted feature count:

$$\sum_{(y,\mathbf{x})\in(Y,X)} \sum_{y'} p_\lambda(y' \mid \mathbf{x}) f_i(y',\mathbf{x}) = \frac{\partial\mathcal{B}}{\partial\lambda_i}$$

Calculate the derivative $\frac{\partial\mathcal{A}}{\partial\lambda_i} - \frac{\partial\mathcal{B}}{\partial\lambda_i}$ for

$f_4(y,\mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Hong}]$

$\lambda_4 = 1.5$

Instances

- Europe: "Monaco"
- Asia "Hong Kong"
- Europe "Monaco Hong Kong"
- Asia "Hong Kong Monaco"

$$\frac{\partial\mathcal{A}}{\partial\lambda_4} = 0 + 1 + 0 + 1 = 2$$

$$\begin{aligned}
\frac{\partial\mathcal{B}}{\partial\lambda_4} &= p_\lambda(\text{Europe} \mid \mathbf{x}_1) f_4(\text{Europe},\mathbf{x}_1) + p_\lambda(\text{Asia} \mid \mathbf{x}_1) f_4(\text{Asia},\mathbf{x}_1) + \\
&\quad p_\lambda(\text{Europe} \mid \mathbf{x}_2) f_4(\text{Europe},\mathbf{x}_2) + p_\lambda(\text{Asia} \mid \mathbf{x}_2) f_4(\text{Asia},\mathbf{x}_2) + \\
&\quad p_\lambda(\text{Europe} \mid \mathbf{x}_3) f_4(\text{Europe},\mathbf{x}_3) + p_\lambda(\text{Asia} \mid \mathbf{x}_3) f_4(\text{Asia},\mathbf{x}_3) + \\
&\quad p_\lambda(\text{Europe} \mid \mathbf{x}_4) f_4(\text{Europe},\mathbf{x}_4) + p_\lambda(\text{Asia} \mid \mathbf{x}_4) f_4(\text{Asia},\mathbf{x}_4) \\
&= p_\lambda(\text{Asia} \mid \mathbf{x}_2) f_4(\text{Asia},\mathbf{x}_2) + p_\lambda(\text{Asia} \mid \mathbf{x}_3) f_4(\text{Asia},\mathbf{x}_3) + p_\lambda(\text{Asia} \mid \mathbf{x}_4) f_4(\text{Asia},\mathbf{x}_4) + \\
&= p_\lambda(\text{Asia} \mid \mathbf{x}_2) + p_\lambda(\text{Asia} \mid \mathbf{x}_3) + p_\lambda(\text{Asia} \mid \mathbf{x}_4)
\end{aligned}$$

Derivative Example (II)

Calculate

- $p_\lambda(\text{Asia} \mid \mathbf{x}_2)$
- $p_\lambda(\text{Asia} \mid \mathbf{x}_3)$
- $p_\lambda(\text{Asia} \mid \mathbf{x}_4)$

Features

- $f_1(y, \mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Monaco}]$ $\quad \lambda_1 = 2.5$
- $f_2(y, \mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Monaco}]$ $\quad \lambda_2 = 0.3$
- $f_3(y, \mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Hong}]$ $\quad \lambda_3 = -0.5$
- $f_4(y, \mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Hong}]$ $\quad \lambda_4 = 1.5$
- $f_5(y, \mathbf{x}) = [y = \text{Europe} \wedge \mathbf{x} \ni \text{Kong}]$ $\quad \lambda_5 = -0.5$
- $f_6(y, \mathbf{x}) = [y = \text{Asia} \wedge \mathbf{x} \ni \text{Kong}]$ $\quad \lambda_6 = 1.5$

$$p_\lambda(y \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_i \lambda_i f_i(y, \mathbf{x})$$

Example texts

Europe: "Monaco"
Asia "Hong Kong"
Europe "Monaco Hong Kong"
Asia "Hong Kong Monaco"

- $p_\lambda(\text{Asia} \mid \mathbf{x}_2) = \frac{\exp(3)}{\exp(3)+\exp(-1)} \approx \frac{20.01}{20.45} \approx 0.98$
- $p_\lambda(\text{Asia} \mid \mathbf{x}_3) = \frac{\exp(0.3+1.5+1.5)}{\exp(0.3+1.5+1.5)+\exp(2.5-0.5-0.5)} \approx \frac{27.1}{31.59} \approx 0.86$
- $p_\lambda(\text{Asia} \mid \mathbf{x}_4) = \frac{\exp(0.3+1.5+1.5)}{\exp(0.3+1.5+1.5)+\exp(2.5-0.5-0.5)} \approx \frac{27.1}{31.59} \approx 0.86$
- $\frac{\partial \mathcal{A}}{\partial \lambda_4} - \frac{\partial \mathcal{B}}{\partial \lambda_4} = 2 - (0.98 + 0.86 + 0.86) = -0.7$

Parameter Estimation

- We know now how to calculate the model's value.
- We know how to calculate the gradients.
- Convex optimization function
- Optimize actual feature weights:
  - Apply parameter optimization package
  - Gradient descend methods
  - Generalized Iterative Scaling

Demo (I)

Example Text 1

```
1        Europe  Monaco
2        Asia    Hong
3        Europe  Monaco Hong
4        Asia    Hong Monaco
```

Package maxent in R

```
library(maxent)
data <- read.delim("ex1.data",header=FALSE)
corpus <- Corpus(VectorSource(data$V3))
matrix <- DocumentTermMatrix(corpus)
sparse <- as.compressed.matrix(matrix)
model <- maxent(sparse,data$V2)
#results <- predict(model,sparse[3,])
```

Model

```
Slot "weights":
      Weight      Label Feature
1      -4.52 Europe           1
2       4.52 Asia             1
3       4.52 Europe           2
4      -4.52 Asia             2
```

Demo (II)

Example Text 2

```
1       Europe  Monaco
2       Asia    Hong Kong
3       Europe  Monaco Hong Kong
4       Asia    Hong Kong Monaco
```

Package maxent in R

```
library(maxent)
data <- read.delim("ex2.data",header=FALSE)
corpus <- Corpus(VectorSource(data$V3))
matrix <- DocumentTermMatrix(corpus)
sparse <- as.compressed.matrix(matrix)
model <- maxent(sparse,data$V2)
#results <- predict(model,sparse[3,])
```

Model

```
Slot "weights":
      Weight      Label Feature
1      -3.72 Europe           1
2       3.72 Asia             1
3      -3.72 Europe           2
4       3.72 Asia             2
5       7.44 Europe           3
6      -7.44 Asia             3
```

Outline

Feature Selection: Motivation

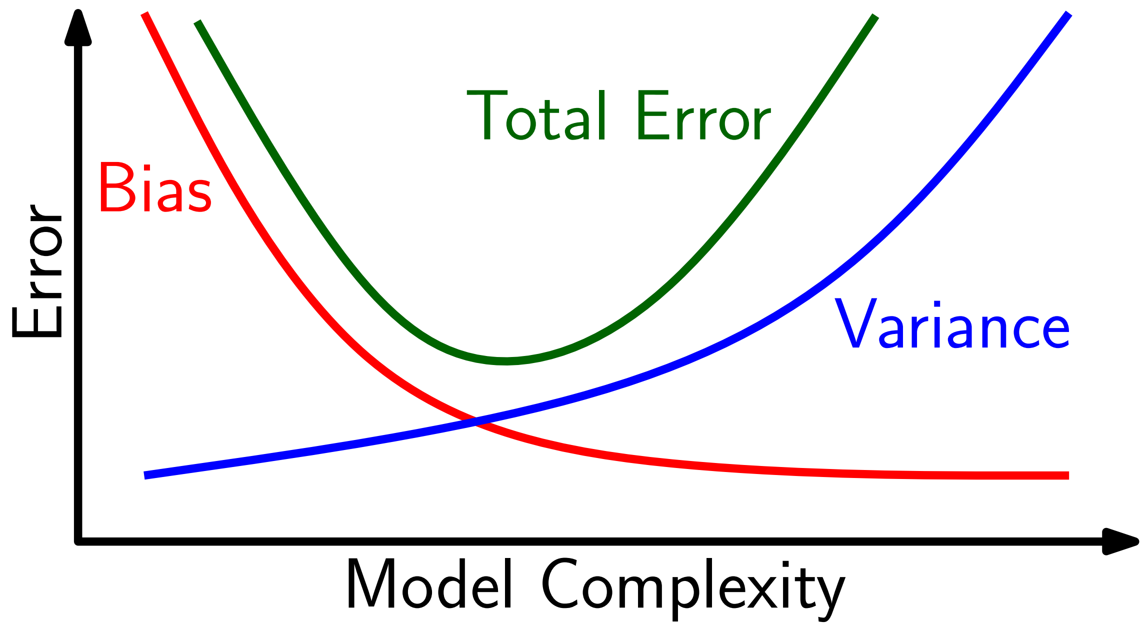
- We have seen that there exist models which can deal with many features
- Features can be correlated
  - Maximum Entropy Classifier can deal with that
  - Naive Bayes is more confused by that
  - Other models...
- Features might be misleading
- Features might lead to overfitting

$\Rightarrow$ Feature selection can help

- Recall: We want to get a good performance on an independent test set!

Bias Variance Dilemma

- Bias:
  Error based on wrong assumptions in the learning algorithm
- Variance: Too sensitive to training data



Filter vs. Wrapper

Wrapper

- Use the learning procedure
- Change feature sets
- Evaluate model using development set/cross validation

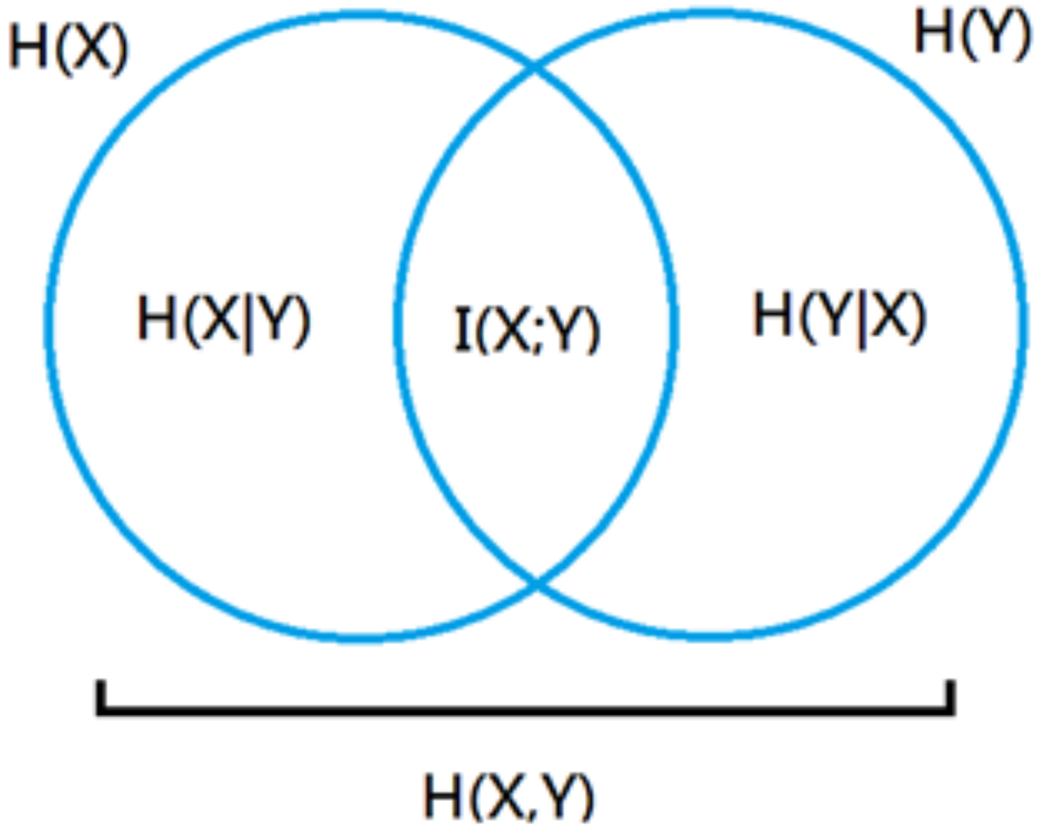
Filter

- Estimate impact of each feature
- Select feature set (based on threshold)

Filter: Measures

- Mutual Information
  (in more detail next slide)
- Ξ^2 (Chi^2)
  Based on a statistical test that two events are independent.
- Frequency-based
  Select features that are most frequent in a class.
- ...

Mutual Information



- Entropy $H(X) = E(I(X))$ – Expectation of Information
- Mutual Information
  Shared Expectation of Information between two variables

Mutual Information

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right)$$

X: Feature values; Y: Classes

Toy example

Y	X_1	X_2
A	1	1
B	2	1
A	1	2
B	2	2

Mutual Information

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$

X: Feature values; Y: Classes

Toy example

Y	X_1	X_2
A	1	1
B	2	1
A	1	2
B	2	2

MI

$$\begin{aligned}I(X_1;Y) &= p(1,A)\log\left(\frac{p(1,A)}{p(1)\,p(A)}\right)\\&+ p(1,B)\log\left(\frac{p(1,B)}{p(1)\,p(B)}\right)\\&+ p(2,A)\log\left(\frac{p(2,A)}{p(2)\,p(A)}\right)\\&+ p(2,B)\log\left(\frac{p(2,B)}{p(2)\,p(B)}\right)\\&= \frac{1}{2}\log\frac{\frac{1}{2}}{\frac{1}{2}\frac{1}{2}} + 0 + 0 + \frac{1}{2}\log\frac{\frac{1}{2}}{\frac{1}{2}\frac{1}{2}} = 1\end{aligned}$$

Mutual Information

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right)$$

X: Feature values; Y: Classes

Toy example

Y	X_1	X_2
A	1	1
B	2	1
A	1	2
B	2	2

MI

$$I(X_1;Y) = \frac{1}{2} \log \frac{\frac{1}{2}}{\frac{1}{2}\frac{1}{2}} + 0 + 0 + \frac{1}{2} \log \frac{\frac{1}{2}}{\frac{1}{2}\frac{1}{2}} = 1$$

$$\begin{aligned} I(X_2;Y) &= p(1,A) \log \left(\frac{p(1,A)}{p(1)\,p(A)} \right) + p(1,B) \log \left(\frac{p(1,B)}{p(1)\,p(B)} \right) \\ &+ p(2,A) \log \left(\frac{p(2,A)}{p(2)\,p(A)} \right) + p(2,B) \log \left(\frac{p(2,B)}{p(2)\,p(B)} \right) \\ &= \frac{1}{4} \log \frac{\frac{1}{4}}{\frac{1}{2}\frac{1}{2}} + \frac{1}{4} \log \frac{\frac{1}{4}}{\frac{1}{2}\frac{1}{2}} + \frac{1}{4} \log \frac{\frac{1}{4}}{\frac{1}{2}\frac{1}{2}} + \frac{1}{4} \log \frac{\frac{1}{4}}{\frac{1}{2}\frac{1}{2}} = 0 \end{aligned}$$

Pointwise mutual information (PMI) on Reuters

UK

london	0.1925
uk	0.0755
british	0.0596
stg	0.0555
britain	0.0469
plc	0.0357
england	0.0238
pence	0.0212
pounds	0.0149
english	0.0126

China

china	0.0997
chinese	0.0523
beijing	0.0444
yuan	0.0344
shanghai	0.0292
hong	0.0198
kong	0.0195
xinhua	0.0155
province	0.0117
taiwan	0.0108

poultry

poultry	0.0013
meat	0.0008
chicken	0.0006
agriculture	0.0005
avian	0.0004
broiler	0.0003
veterinary	0.0003
birds	0.0003
inspection	0.0003
pathogenic	0.0003

coffee

coffee	0.0111
bags	0.0042
growers	0.0025
kg	0.0019
colombia	0.0018
brazil	0.0016
export	0.0014
exporters	0.0013
exports	0.0013
crop	0.0012

elections

election	0.0519
elections	0.0342
polls	0.0339
voters	0.0315
party	0.0303
vote	0.0299
poll	0.0225
candidate	0.0202
campaign	0.0202
democratic	0.0198

sports

soccer	0.0681
cup	0.0515
match	0.0441
matches	0.0408
played	0.0388
league	0.0386
beat	0.0301
game	0.0299
games	0.0284
team	0.0264

Filter vs. Wrapper

- Filter:
  - Pro: Preprocessing, efficient
  - Con: Not necessarily the best result as classifier-agnostic
- Wrapper:
  - Pro: Can lead to very good results
  - Con: Slow, depending on feature space not feasible, might lead to overfitting
- Combinations are possible!
  (forward search, backward search)

Outline

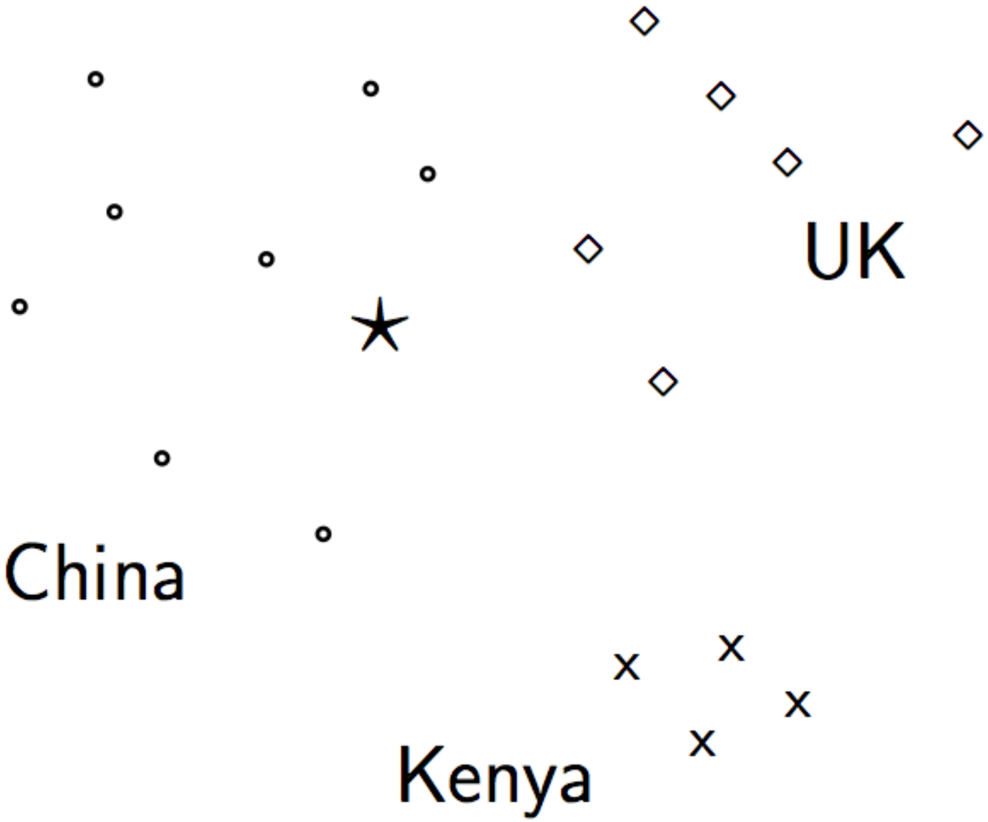
Recall vector space representation

- Each document is a vector, one component for each term.
- Terms (features, in general) are axes.
- High dimensionality: 100,000s of dimensions
- Normalize vectors (documents) to unit length
- How can we do classification in this space?

Vector space classification

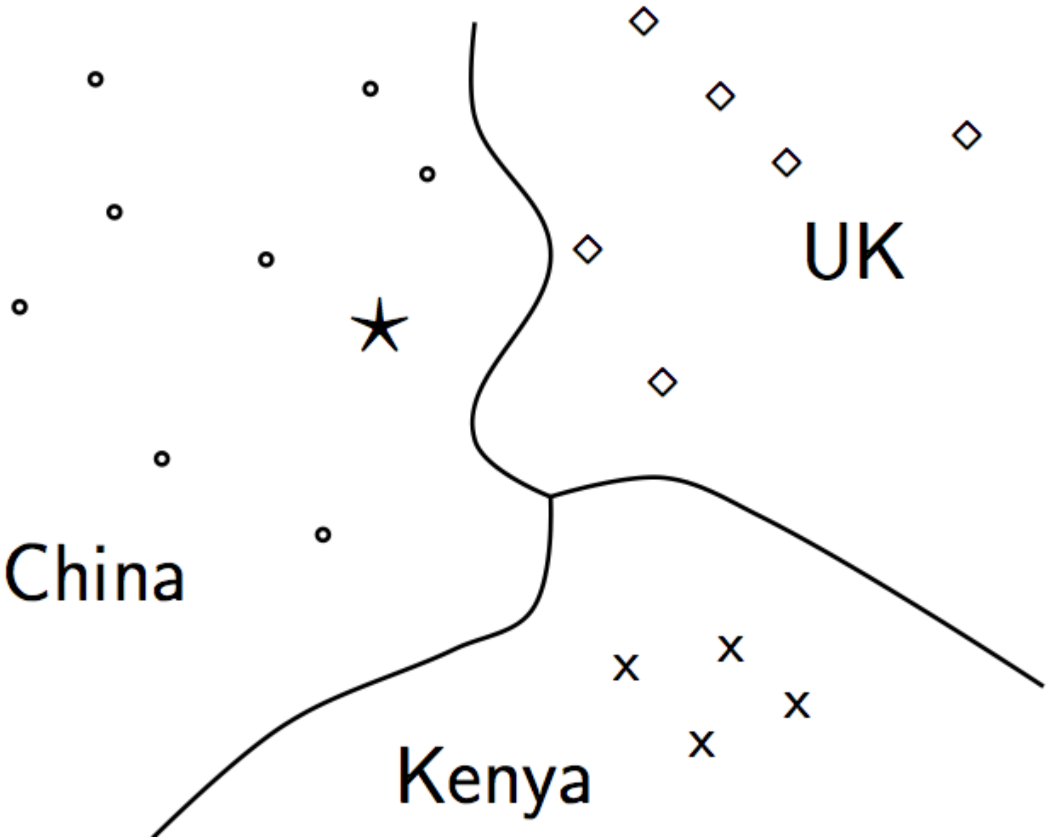
- As before, training set is set of documents, each labeled with its class.
- In vector space classification, set corresponds to a labeled set of points or vectors in the vector space.
- Premise 1:
  Documents in the same class form a contiguous region.
- Premise 2:
  Documents from different classes don't overlap.
- We define lines, surfaces, hypersurfaces to divide regions.

Classes in the vector space



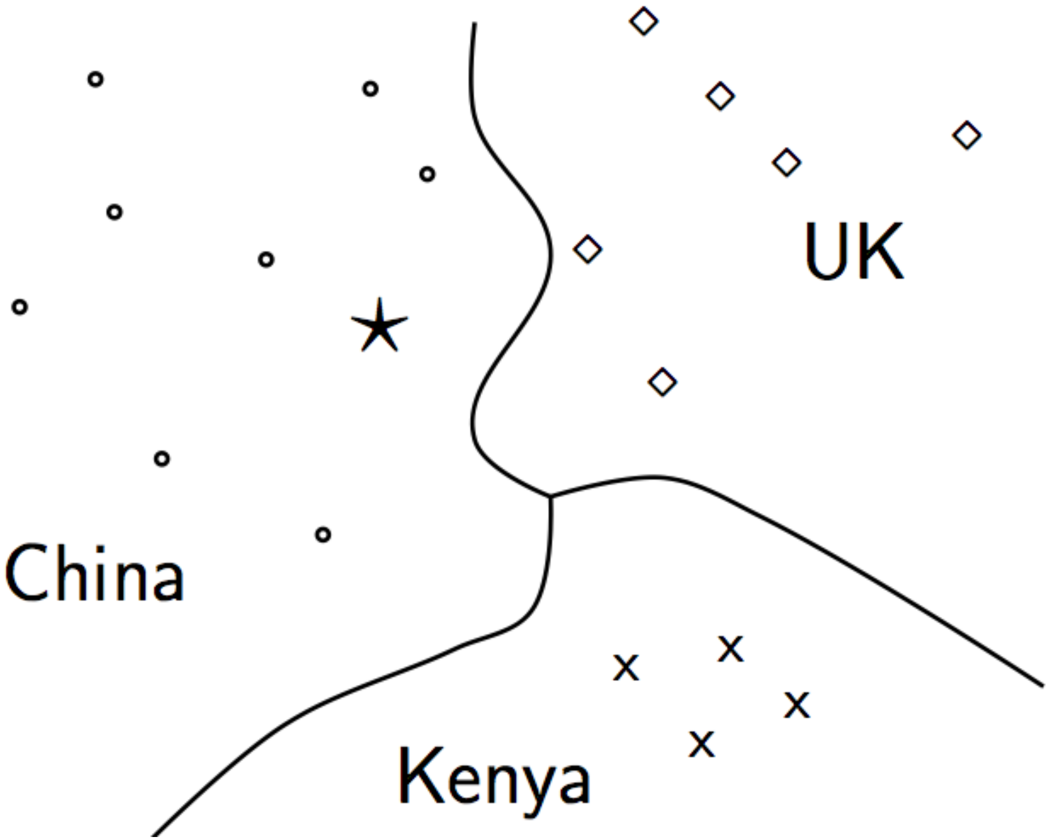
Should the document $\star$ be assigned to *China*, *UK* or *Kenya*?

Classes in the vector space



Find separators between the classes

Classes in the vector space



How do we find separators that do a good job at classifying new documents like $\star$? – Main topic of today

Outline

kNN classification

- kNN classification is a lazy approach to vector space classification
  - Lazy: Generalization during prediction
  - Eager: Generalization during learning
- It is very simple and easy to implement.
- kNN is often more accurate than Naive Bayes
- If you need to get a pretty accurate classifier up and running in a short time . . .
  - . . . and you don't care about runtime that much . . .
  - . . . use kNN.

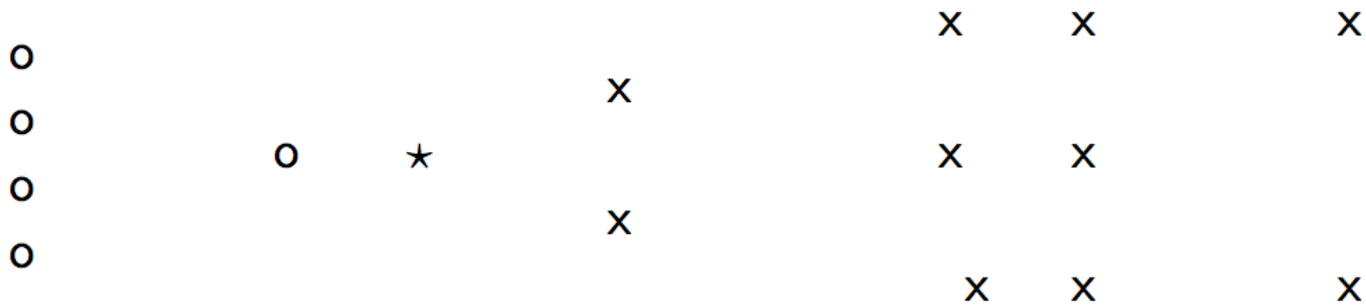
kNN classification

- kNN $=$ k nearest neighbors
- kNN classification rule for $k = 1$ (1NN): Assign each test document to the class of its nearest neighbor in the training set.
  - 1NN is not very robust:
    one document can be mislabeled or atypical.
- kNN classification rule for $k > 1$ (kNN): Assign each test document to the majority class of its k nearest neighbors in the training set.
- Rationale of kNN: contiguity hypothesis
  - We expect a test document d to have the same label as the training documents located in the local region surrounding d.

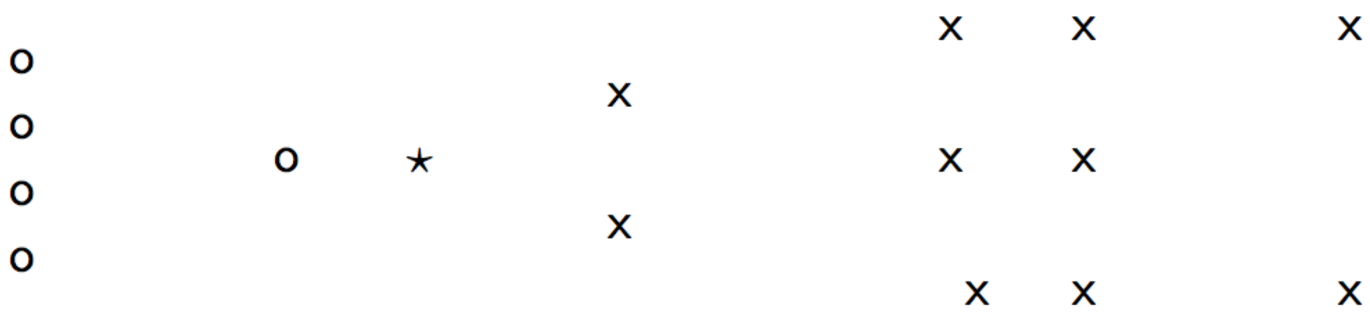
Probabilistic kNN

- Probabilistic interpretation of kNN:
  $P(c|d) =$ fraction of k neighbors of d that are in c
- kNN classification rule for probabilistic kNN:
  Assign d to class c with highest $P(c|d)$

Exercise



Exercise



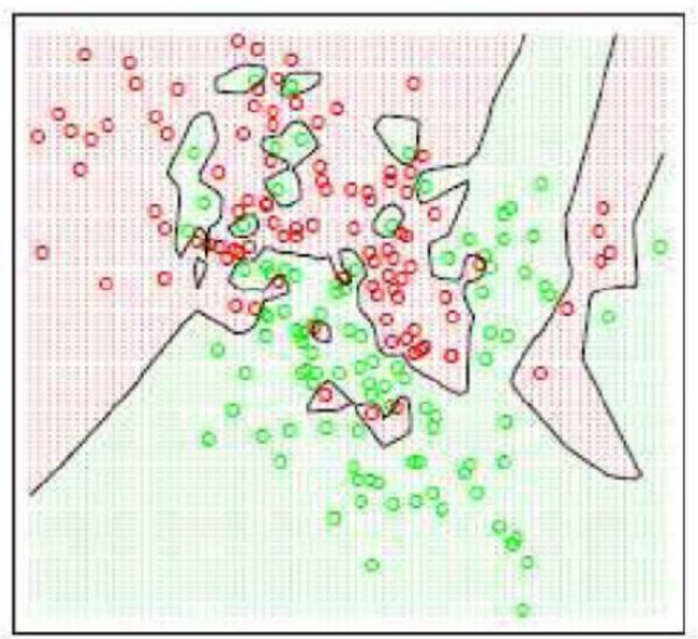
How is star classified by:
(i) 1-NN (ii) 3-NN (iii) 9-NN (iv) 15-NN?

o x o x

$p(x \mid \star) = \frac{2}{3}$

Influence of k in kNN

K=1 K=15



Distance Weighted kNN

- Use all instances instead of just k
- Weight distance to the query instance
- "Shepard's Method"
- Metric:
  - Euclidean
  - Manhatten
  - ...
- Active research area:
  Learn that metric such that prediction error is minimized

Time complexity of kNN

kNN with preprocessing of training set

training	$\Theta(\lvert\mathbb{D}\rvert L_{\mathrm{ave}})$
testing	$\Theta(L_{\mathrm{a}} + \lvert\mathbb{D}\rvert M_{\mathrm{ave}} M_{\mathrm{a}}) = \Theta(\lvert\mathbb{D}\rvert M_{\mathrm{ave}} M_{\mathrm{a}})$

- kNN test time proportional to the size of the training set!
- The larger the training set, the longer it takes to classify a test document.
- kNN is inefficient for very large training sets.

kNN with inverted index

- Naive: find nearest neighbors requires a linear search through all documents

  Any idea how to make use of our inverted index?

d_5 — c

d_4 — c

d_6 — c

d_{10} — c

⋮

kNN with inverted index

- Naive: find nearest neighbors requires a linear search through all documents
  Any idea how to make use of our inverted index?
- Use test document as query: finding k nearest neighbors is the same as determining the k best retrievals
- Use standard vector space inverted index methods to find the k nearest neighbors.

kNN: Discussion

- No training necessary
  - But linear preprocessing of documents is as expensive as training Naive Bayes.
  - We always preprocess the training set, so in reality training time of kNN is linear.
- kNN is very accurate if training set is large.
- kNN can be very inaccurate if training set is small.
- Test time proportional to training set size

Take-away today

- The problem of overcounting in Naive Bayes
- Maximum Entropy Classifier
- Overfitting and the Bias-Variance Dilemma
- Feature Selection
- Vector space classification