COMP90042 Workshop Week 3

Text Classification & N-gram Language Model

Zenan Zhai

The University of Melbourne

16 March 2014

Table of Contents

Table of Contents

Definition

Think of an application of text classification.

Definition

Think of an application of text classification.

- Sentiment analysis
- Author identification
- Fact checking
- ...

What makes text classification challenging?

Definition

Think of an application of text classification.

- Sentiment analysis
- Author identification
- Fact checking
- ...

What makes text classification challenging?

- How to learn document representation?
- How to do feature selection?
- How to deal with data sparsity?

Text classification Models

- K-Nearest Neighbors (KNN)
- Decision Tree
- Naive Bayes
- Logistic Regression
- Support Vector Machine
- ...

K-Nearest Neighbors

Vote by the label of K nearest instances in the training set

Similarity Measure

- Eculidean distance $d(A, B) = \sqrt{\sum (a_i - b_i)^2}$

K-Nearest Neighbors

Vote by the label of K nearest instances in the training set

Similarity Measure

- Eculidean distance $d(A, B) = \sqrt{\sum(a_i - b_i)^2}$

Not ideal, the measure is greatly affected by **document length**.

K-Nearest Neighbors

Vote by the label of K nearest instances in the training set

Similarity Measure

- Cosine similarity $cos(A, B) = \frac{A \cdot B}{||A|| \ ||B|}$

K-Nearest Neighbors

Vote by the label of K nearest instances in the training set

Similarity Measure

- Cosine similarity $cos(A, B) = \frac{A \cdot B}{||A|| \ ||B|}$

Better than Eculidean distance, but suffers from **curse of dimensionality**

Decision Tree

Entropy

$$H(Y) = -\sum_{y \in Y} p(y) log(p(y))$$

Conditional Entropy

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X = x)$$

Information Gain

$$IG(Y|a) = H(Y) - H(Y|a)$$

Decision Tree

Entropy

$$H(Y) = -\sum_{y \in Y} p(y) log(p(y))$$

Conditional Entropy

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X = x)$$

Information Gain

$$IG(Y|a) = H(Y) - H(Y|a)$$

Tends to prefer **rare features** which might only appears in a few documents.

Naive Bayes

Every feature is assumed to be independent

$$
\begin{aligned}
P(y|x_1, x_2, x_3, \cdots, x_n) &\propto P(y, x_1, x_2, x_3, \cdots, x_n) \\
&= P(y) \prod_{i=1}^{n} P(x_i|y)
\end{aligned}
$$

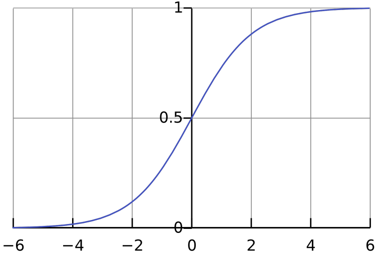
Naive Bayes

Every feature is assumed to be independent

$$\begin{aligned}P(y|x_1, x_2, x_3, \cdots, x_n) &\propto P(y, x_1, x_2, x_3, \cdots, x_n) \\ &= P(y) \prod_{i=1}^{n} P(x_i|y)\end{aligned}$$

Still work on dataset **large set of features**, but suffers from biases caused by **uninformative features**.

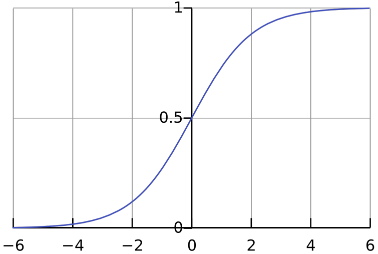
Logistic Regression



Put linear combination of features in logistic function

$$P(y) = \sigma(y) = \frac{1}{1 + \exp(WX + b)}$$

Logistic Regression

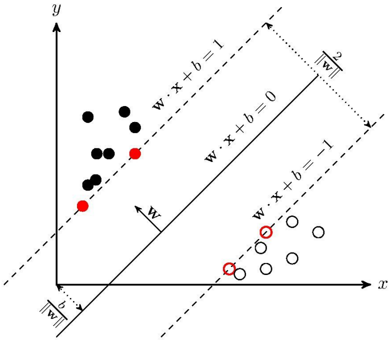


Put linear combination of features in logistic function

$$P(y) = \sigma(y) = \frac{1}{1 + \exp(\,WX + b)}$$

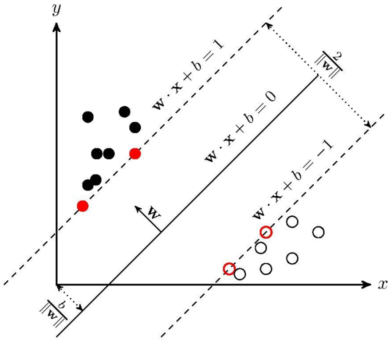
Relaxed the independent assumption and solves the problem caused by uninformative features by taking weighted sum.

Support Vector Machine



Select the decision boundary which maximize the distance to the support vectors.

Support Vector Machine



Select the decision boundary which maximize the distance to the support vectors.

An very effective methods, but no natural support **multi-classification**

Table of Contents

Uni-gram model

Corpus

1. `how much wood would a wood chuck chuck if a wood chuck would chuck wood`
2. `a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood`

Uni-gram model

Corpus

1. how much wood would a wood chuck chuck if a wood chuck would chuck wood
2. a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood

Word counts

a	chuck	could	he	how	if	much	the	wood	would	</s>	Total
4	9	1	1	1	2	1	1	8	4	2	34

Uni-gram model

Corpus

1. how much wood would a wood chuck chuck if a wood chuck would chuck wood
2. a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood

Word counts

a	chuck	could	he	how	if	much	the	wood	would	</s>	Total
4	9	1	1	1	2	1	1	8	4	2	34

Why is <s> left out?

Uni-gram model

Word counts

a	chuck	could	he	how	if	much	the	wood	would	</s>	Total
4	9	1	1	1	2	1	1	8	4	2	34

Sentences

A: a wood could chuck

B: wood would a chuck

Uni-gram model

Word counts

a	chuck	could	he	how	if	much	the	wood	would	</s>	Total
4	9	1	1	1	2	1	1	8	4	2	34

Sentences

A: a wood could chuck

B: wood would a chuck

$$\begin{aligned} P(A) &= P(\texttt{a})P(\texttt{wood})P(\texttt{could})P(\texttt{chuck})P(\texttt{</s>}) \\ &= \frac{4}{34} \times \frac{8}{34} \times \frac{1}{34} \times \frac{9}{34} \times \frac{2}{34} \approx 1.27 \times 10^{-5} \\ P(B) &= P(\texttt{wood})P(\texttt{would})P(\texttt{a})P(\texttt{chuck})P(\texttt{</s>}) \\ &= \frac{8}{34} \times \frac{4}{34} \times \frac{4}{34} \times \frac{9}{34} \times \frac{2}{34} \approx 5.07 \times 10^{-5} \end{aligned}$$

Uni-gram model with Laplacian smoothing

Word counts

a	chuck	could	he	how	if	much	the	wood	would	</s>	Total
4	9	1	1	1	2	1	1	8	4	2	34

Sentences

A: `a wood could chuck`

B: `wood would a chuck`

$$\begin{aligned} P_{\mathrm{L}}(A) &= P_{\mathrm{L}}(\texttt{a})P_{\mathrm{L}}(\texttt{wood})P_{\mathrm{L}}(\texttt{could})P_{\mathrm{L}}(\texttt{chuck})P_{\mathrm{L}}(\texttt{</s>}) \\ &= \frac{5}{45} \times \frac{9}{45} \times \frac{2}{45} \times \frac{10}{45} \times \frac{3}{45} \approx 1.46 \times 10^{-5} \\ P_{\mathrm{L}}(B) &= P_{\mathrm{L}}(\texttt{wood})P_{\mathrm{L}}(\texttt{would})P_{\mathrm{L}}(\texttt{a})P_{\mathrm{L}}(\texttt{chuck})P_{\mathrm{L}}(\texttt{</s>}) \\ &= \frac{9}{45} \times \frac{5}{45} \times \frac{5}{45} \times \frac{10}{45} \times \frac{3}{45} \approx 3.66 \times 10^{-5} \end{aligned}$$

Bi-gram model

Corpus

1. `how much wood would a wood chuck chuck if a wood chuck would chuck wood`
2. `a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood`

Sentences

A: `a wood could chuck`

B: `wood would a chuck`

Bi-gram model

Corpus

1. how much wood would a wood chuck chuck if a wood chuck would chuck wood
2. a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood

Sentences

A: a wood could chuck

B: wood would a chuck

$$
\begin{aligned}
P(A) &= P(\texttt{a}|\texttt{<s>})P(\texttt{wood}|\texttt{a})P(\texttt{could}|\texttt{wood})P(\texttt{chuck}|\texttt{could})P(\texttt{</s>}|\texttt{chuck}) \\
&= \frac{1}{2} \times \frac{4}{4} \times \frac{0}{8} \times \frac{1}{1} \times \frac{0}{9} = 0 \\
P(B) &= P(\texttt{wood}|\texttt{<s>})P(\texttt{would}|\texttt{wood})P(\texttt{a}|\texttt{would})P(\texttt{chuck}|\texttt{a})P(\texttt{</s>}|\texttt{chuck}) \\
&= \frac{0}{2} \times \frac{1}{8} \times \frac{1}{4} \times \frac{0}{4} \times \frac{0}{9} = 0
\end{aligned}
$$

Bi-gram model with Laplacian smoothing

Corpus

1. `how much wood would a wood chuck chuck if a wood chuck would chuck wood`
2. `a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood`

Sentences

A: `a wood could chuck`

B: `wood would a chuck`

$$
\begin{aligned}
P_{\mathrm{L}}(A) &= P_{\mathrm{L}}(\texttt{a}|\texttt{<s>})P_{\mathrm{L}}(\texttt{wood}|\texttt{a})P_{\mathrm{L}}(\texttt{could}|\texttt{wood})P_{\mathrm{L}}(\texttt{chuck}|\texttt{could})P_{\mathrm{L}}(\texttt{</s>}|\texttt{chuck}) \\
&= \frac{2}{13} \times \frac{5}{15} \times \frac{1}{19} \times \frac{2}{12} \times \frac{1}{20} \approx 2.25 \times 10^{-5} \\
P_{\mathrm{L}}(B) &= P_{\mathrm{L}}(\texttt{wood}|\texttt{<s>})P_{\mathrm{L}}(\texttt{would}|\texttt{wood})P_{\mathrm{L}}(\texttt{a}|\texttt{would})P_{\mathrm{L}}(\texttt{chuck}|\texttt{a})P_{\mathrm{L}}(\texttt{</s>}|\texttt{chuck}) \\
&= \frac{1}{13} \times \frac{2}{19} \times \frac{2}{15} \times \frac{1}{15} \times \frac{1}{20} \approx 3.60 \times 10^{-6}
\end{aligned}
$$

Tri-gram model

Corpus

1. `how much wood would a wood chuck chuck if a wood chuck would chuck wood`
2. `a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood`

Sentences

A: `a wood could chuck`

B: `wood would a chuck`

Tri-gram model

Corpus

1. `how much wood would a wood chuck chuck if a wood chuck would chuck wood`
2. `a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood`

Sentences

A: `a wood could chuck`

B: `wood would a chuck`

$$
\begin{aligned}
P(A) &= P(\texttt{a}|\texttt{<s> <s>})P(\texttt{wood}|\texttt{<s> a})\cdots P(\texttt{</s>}|\texttt{could chuck}) \\
&= \frac{1}{2} \times \frac{1}{1} \times \frac{0}{4} \times \frac{0}{0} \times \frac{0}{1} = ? \\
P(B) &= P(\texttt{wood}|\texttt{<s> <s>})P(\texttt{would}|\texttt{<s> wood})\cdots P(\texttt{</s>}|\texttt{a chuck}) \\
&= \frac{0}{2} \times \frac{0}{0} \times \frac{1}{1} \times \frac{0}{1} \times \frac{0}{0} = ?
\end{aligned}
$$

Tri-gram model with Laplacian smoothing

Corpus

1. how much wood would a wood chuck chuck if a wood chuck would chuck wood
2. a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood

Sentences

A: a wood could chuck

B: wood would a chuck

$$
\begin{aligned}
P_{\mathrm{L}}(A) &= P_{\mathrm{L}}(\texttt{a}|\texttt{<s> <s>})P_{\mathrm{L}}(\texttt{wood}|\texttt{<s> a})\cdots P_{\mathrm{L}}(\texttt{</s>}|\texttt{could chuck}) \\
&= \frac{2}{13}\times\frac{2}{12}\times\frac{1}{15}\times\frac{1}{11}\times\frac{1}{12}\approx 1.30\times 10^{-5} \\
P_{\mathrm{L}}(B) &= P_{\mathrm{L}}(\texttt{wood}|\texttt{<s> <s>})P_{\mathrm{L}}(\texttt{would}|\texttt{<s> wood})\cdots P_{\mathrm{L}}(\texttt{</s>}|\texttt{a chuck}) \\
&= \frac{1}{13}\times\frac{1}{11}\times\frac{2}{12}\times\frac{1}{12}\times\frac{1}{11}\approx 8.83\times 10^{-6}
\end{aligned}
$$

Kneser-Ney Smoothing

What is continuation count?

Kneser-Ney Smoothing

What is continuation count?

- number of word types in the vocabulary which appears before a word w
- $|\{w_{i-1} : C(w_{i-1}, w_i) > 0\}|$

What is Kneser-Ney smoothing?

Kneser-Ney Smoothing

What is continuation count?

- number of word types in the vocabulary which appears before a word w
- $|\{w_{i-1} : C(w_{i-1}, w_i) > 0\}|$

What is Kneser-Ney smoothing?

- use continuation probability instead of trivial uni-gram prob
- can be used in either back-off and interpolation

$$P_{cont}(w_i) = \frac{|\{w_{i-1} : C(w_{i-1}, w_i) > 0\}|}{\sum_{w_i \in V} |\{w_{i-1} : C(w_{i-1}, w_i) > 0\}|}$$

KN Smoothing - Example

Corpus

1. how much wood would a wood chuck chuck if a wood chuck would chuck wood
2. a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood

Continuation counts

a =	could =
he =	how =
if =	much =
the =	would =
</s> =	

KN Smoothing - Example

Corpus

1. `how much wood would a wood chuck chuck if a wood chuck would chuck wood`
2. `a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood`

Continuation counts

$\texttt{a} = 2$	$\texttt{could} = 1$
$\texttt{he} = 1$	$\texttt{how} = 0$
$\texttt{if} = 1$	$\texttt{much} = 1$
$\texttt{the} = 1$	$\texttt{would} = 2$
$\texttt{</s>} = 1$	

KN Smoothing - Example

Corpus

1. `how much wood would a wood chuck chuck if a wood chuck would chuck wood`
2. `a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood`

Continuation counts

chuck=	wood=
a = 2	could = 1
he = 1	how = 0
if = 1	much = 1
the = 1	would = 2
</s> = 1	

KN Smoothing - Example

Corpus

1. how much wood would a wood chuck chuck if a wood chuck would chuck wood
2. a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood

Continuation counts

chuck= 4	wood= 4
a = 2	could = 1
he = 1	how = 0
if = 1	much = 1
the = 1	would = 2
</s> = 1	

KN Smoothing - Example

Continuation counts

$\texttt{chuck} = 4$

$\texttt{a} = 2$

$\texttt{he} = 1$

$\texttt{if} = 1$

$\texttt{the} = 1$

$\texttt{</s>} = 1$

$\texttt{wood} = 4$

$\texttt{could} = 1$

$\texttt{how} = 0$

$\texttt{much} = 1$

$\texttt{would} = 2$

Continuation probabilities

$$P_{cont}(\texttt{chuck})$$
$$P_{cont}(\texttt{wood})$$

KN Smoothing - Example

Continuation counts

$\texttt{chuck} = 4$

$\texttt{a} = 2$

$\texttt{he} = 1$

$\texttt{if} = 1$

$\texttt{the} = 1$

$\texttt{</s>} = 1$

$\texttt{wood} = 4$

$\texttt{could} = 1$

$\texttt{how} = 0$

$\texttt{much} = 1$

$\texttt{would} = 2$

Continuation probabilities

$$
\begin{aligned}
P_{cont}(\texttt{chuck}) &= \frac{\#_{cont}(\texttt{chuck})}{\#_{cont}(\texttt{a}) + \ldots + \#_{cont}(\texttt{</s>}) + \#_{cont}(\texttt{chuck}) + \#_{cont}(\texttt{wood})} \\
&= \frac{4}{2+1+1+0+1+1+1+2+1+4+4} \\
P_{cont}(\texttt{wood})
\end{aligned}
$$

KN Smoothing - Example

Continuation counts

`chuck` $= 4$

`a` $= 2$

`he` $= 1$

`if` $= 1$

`the` $=1$

`</s>` $= 1$

`wood` $= 4$

`could` $= 1$

`how` $= 0$

`much` $= 1$

`would` $= 2$

Continuation probabilities

$$P_{cont}(\texttt{chuck}) = \frac{\#_{cont}(\texttt{chuck})}{\#_{cont}(\texttt{a}) + \ldots + \#_{cont}(\texttt{</s>}) + \#_{cont}(\texttt{chuck}) + \#_{cont}(\texttt{wood})}$$

$$= \frac{4}{2+1+1+0+1+1+1+2+1+4+4}$$

$$P_{cont}(\texttt{wood}) = \frac{\#_{cont}(\texttt{wood})}{\#_{cont}(\texttt{a}) + \ldots + \#_{cont}(\texttt{</s>}) + \#_{cont}(\texttt{chuck}) + \#_{cont}(\texttt{wood})}$$

$$= \frac{4}{2+1+1+0+1+1+1+2+1+4+4}$$

Back-off and Interpolation

Back-off

- Use lower-order n-gram model if higher-order is unseen.

$$P(w_i|w_{i-1}) = \begin{cases} \frac{c(w_i, w_{i-1}) - D}{c(w_{i-1})}, & \text{if } c(w_i, w_{i-1}) > 0 \\ \alpha(w_{i-1}) \times \frac{P(w_i)}{\sum_{w_j : C(w_{i-1}, w_j) = 0} P(w_j)}, & \text{otherwise} \end{cases}$$

Interpolation

- Take weighted average sum of all orders

$$P(w_i|w_{i-1}) = \lambda P(w_i|w_{i-1}) + (1 - \lambda) P(w_i)$$

Evaluation

Recall the objective of language model:

- Modeling probability for an arbitrary sequence of m words.

Evaluate based on probability of all sequences in test set

$$PP(w_1, w_2, w_3, ..., w_m) = \sqrt[m]{\frac{1}{P(w_1, w_2, w_3, ..., w_m)}}$$

- Inverted prob. : lower perplexity → better model
- Normalization : take m^{th} root of sequence prob. , $m = length(S)$

Take aways

- Text classification
  - Applications
  - Models (Pros & Cons)
- *n*-gram language model (calculation)
- Smoothing
  - problems they address
  - Pros & Cons