Thumbs Up? Sentiment Classification using Machine Learning Techniques

Pang, Lee, Vaithyanathan - EMNLP 2002

Slides by: Dor Cohen, Itai Gat

IE @ Technion

October 16, 2018

Agenda

Introduction

Topic classification

- Recent (2002) works sort documents according to their **subject**
  - e.g., sports vs. politics

Topic classification

- Recent (2002) works sort documents according to their **subject**
  - e.g., sports vs. politics
- Yet crucial part of online posted articles is their **sentiment**
  - provide useful insights for readers automatically
  - e.g., product review is negative or positive

Sentiment analysis

- This work: apply topic classification techniques on sentiment analysis
  - Q: What are our expected challenges?

Sentiment analysis

- This work: apply topic classification techniques on sentiment analysis
  - Q: What are our expected challenges?
  - A: Topics are identifiable by key words alone, while sentiment requires more **understanding**

Sentiment analysis

- This work: apply topic classification techniques on sentiment analysis
  - Q: What are our expected challenges?
  - A: Topics are identifiable by key words alone, while sentiment requires more **understanding**
- e.g., "How could anyone sit through this movie?"
  - Can you mark any negative word?

Sentiment analysis
Motivation

- Should we watch this movie?

Sentiment analysis

Motivation

A Star Is Born (2018) ★ 8.4/10 46,238

R | 2h 16min | Drama , Music , Romance | 5 October 2018 (USA)

- Should we watch this movie?
- Ideally: read each review and decide

★ 10/10

Amazing!

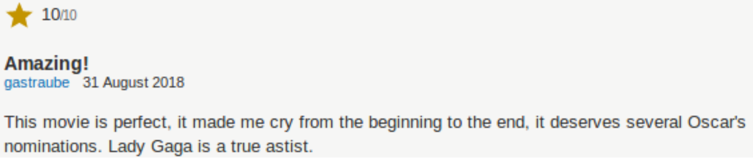
gastraube 31 August 2018

This movie is perfect, it made me cry from the beginning to the end, it deserves several Oscar's nominations. Lady Gaga is a true astist.

Sentiment analysis

Motivation

- Should we watch this movie?
- Ideally: read each review and decide



Problem

Problem definition

- Find mapping from text to binary label

Problem definition

- Find mapping from text to binary label
  - Supervised learning

Problem definition

- Find mapping from text to binary label
  - Supervised learning
- For m numeric features we define:

Problem definition

- Find mapping from text to binary label
  - Supervised learning
- For m numeric features we define:

Definition (Binary classifier)

$$f : X \to y$$

where $X \in \mathbb{R}^m$, $y \in \{0, 1\}$

Problem definition

- Find mapping from text to binary label
  - Supervised learning
- For m numeric features we define:

Definition (Binary classifier)

$$f : X \rightarrow y$$
$$\text{where } X \in \mathbb{R}^m,\ y \in \{0, 1\}$$

- Evaluate by loss function

Problem definition

- Find mapping from text to binary label
  - Supervised learning
- For m numeric features we define:

Definition (Binary classifier)

$$f : X \to y$$
$$\text{where } X \in \mathbb{R}^m,\ y \in \{0, 1\}$$

- Evaluate by loss function
- e.g., Zero-one loss: $L(x, y, f_w) = \mathbf{1}\{f_w(x) \neq y\}$
  - w denotes learned parameters

Data: IMDB Movie Reviews

- Lucky for us: user rating provides **supervised** learning
- Converted into 3 categories:
  - *Positive*, *negative*, (*neutral* - not used)
- Avoid bias issues:
  - 20 reviews per author per sentiment
  - 752 negative vs 1301 positive
  - total of 144 reviewers

Human based sentiment classifiers

- In contrast to topics, detecting sentiment is easier for us (why?)

Human based sentiment classifiers

- In contrast to topics, detecting sentiment is easier for us (why?)
  - People tend to express strong feelings, topics can be related.

Human based sentiment classifiers

- In contrast to topics, detecting sentiment is easier for us (why?)
  - People tend to express strong feelings, topics can be related.
- **Hypothesis:** certain words indicate on sentiment type

Human based sentiment classifiers

- In contrast to topics, detecting sentiment is easier for us (why?)
  - People tend to express strong feelings, topics can be related.
- **Hypothesis:** certain words indicate on sentiment type
- **Test:** count positive vs. negative words

Human based sentiment classifiers

- In contrast to topics, detecting sentiment is easier for us (why?)
  - People tend to express strong feelings, topics can be related.
- **Hypothesis:** certain words indicate on sentiment type
- **Test:** count positive vs. negative words

Human	Proposed words	Accuracy	Ties [1]
1	positive (5): dazzling, brilliant.. negative (5): suck, terrible..	58%	75%
2	positive (11): gripping, spectacular.. negative (6): cliched, boring..	64%	39%

Table: Baseline results for human word lists, data is balanced (700 vs. 700)

[1] Documents percentage where sentiments rated equally

Human based sentiment classifiers

Should we worry about high rate of ties?

- Proposed list is relatively short (usually effect is 0 vs. 0)

Human based sentiment classifiers

Should we worry about high rate of ties?

- Proposed list is relatively short (usually effect is 0 vs. 0)
  - Not necessarily the reason for low accuracy!

Human based sentiment classifiers

Should we worry about high rate of ties?

- Proposed list is relatively short (usually effect is 0 vs. 0)
    - Not necessarily the reason for low accuracy!
- Authors propose their list
    - Backed up with statistics

Human based sentiment classifiers

Should we worry about high rate of ties?

- Proposed list is relatively short (usually effect is 0 vs. 0)
  - Not necessarily the reason for low accuracy!
- Authors propose their list
  - Backed up with statistics

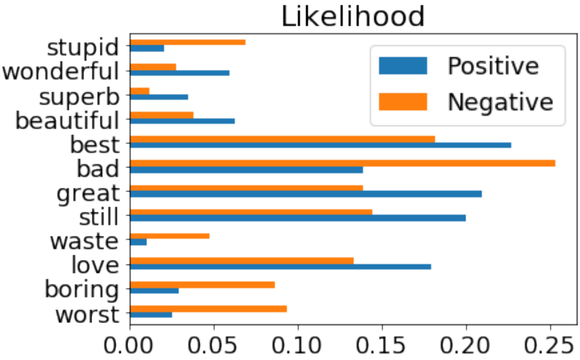
Human	Proposed words	Accuracy	Ties
3+Stats	positive (7): love, wonderful.. negative (7): bad, worst, '?', '!',..	69%	16%

Table: Results where words (total 14) were chosen based on data statistics

Human based sentiment classifiers

(2018) Data analysis

We reproduced the analysis, following are example estimates



- Note words occurrences were binarized

Methods

Bag of words

Framework details

Theorem (1)

Let $\{f_1, ..f_m\}$ *denote set of m features that can appear in document.*

Bag of words

Framework details

Theorem (1)

Let $\{f_1, ..f_m\}$ denote set of m features that can appear in document.

- d: "Audio quality rocks"

Bag of words

Framework details

Theorem (1)

Let $\{f_1, ..f_m\}$ denote set of m features that can appear in document.

- d: "Audio quality rocks"
- Features:
  - Unigram: {'audio', 'rocks',.. }
  - Bigram: {'audio quality',.. }
  - N-gram !

Bag of words

Framework details

Theorem (1)

Let $\{f_1, ..f_m\}$ denote set of m features that can appear in document.

- d: "Audio quality rocks"
- Features:
  - Unigram: {'audio', 'rocks',.. }
  - Bigram: {'audio quality',.. }
  - N-gram !

Theorem (2)

Let $n_i(d)$ be the number of times f_i occurs in document d.

Bag of words

Framework details

Theorem (1)

Let $\{f_1, ..f_m\}$ denote set of m features that can appear in document.

- d: "Audio quality rocks"
- Features:
  - Unigram: {'audio', 'rocks',.. }
  - Bigram: {'audio quality',.. }
  - N-gram !

Theorem (2)

Let $n_i(d)$ be the number of times f_i occurs in document d.

Definition (BOW)

Then each document d is represented by $d^{bow} := (n_1(d), ..., n_m(d))$.

Bag of words

Example

- d_1: "Audio rocks"
- d_2: "Act boring"

Bag of words

Example

- d_1: "Audio rocks"
- d_2: "Act boring"

	act	audio	boring	rocks
mapping	0	1	2	3
d1	0	1	0	1
d2	1	0	1	0

Bag of words

Example

- d_1: "Audio rocks"
- d_2: "Act boring"

	act	audio	boring	rocks
mapping	0	1	2	3
d1	0	1	0	1
d2	1	0	1	0

- d_3: "Boring effects", $d_3^{bow} = ?$

Bag of words

Example

- d_1: "Audio rocks"
- d_2: "Act boring"

	act	audio	boring	rocks
mapping	0	1	2	3
d1	0	1	0	1
d2	1	0	1	0

- d_3: "Boring effects", $d_3^{bow} = ?$
- $d_3^{bow} = (0,0,1,0)$

Naive Bayes classifier

- Assign class which maximizes: $c^* = argmax_c P(c|d)$

Naive Bayes classifier

- Assign class which maximizes: $c^* = argmax_c P(c|d)$
- Recap:

Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

Naive Bayes classifier

- Assign class which maximizes: $c^* = argmax_c P(c|d)$
- Recap:

Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- To estimate $P(d|c)$ we **naively** assume f_i's are independent.

Naive Bayes classifier

- Assign class which maximizes: $c^* = argmax_c P(c|d)$
- Recap:

Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- To estimate $P(d|c)$ we **naively** assume f_i's are independent.
  - Hence $\widehat{P(d|c)} = \prod_{i=1}^{m} P(f_i|c)^{n_i(d)}$

Naive Bayes classifier

- Assign class which maximizes: $c^* = argmax_c P(c|d)$
- Recap:

Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- To estimate $P(d|c)$ we **naively** assume f_i's are independent.
  - Hence $\widehat{P(d|c)} = \prod_{i=1}^{m} P(f_i|c)^{n_i(d)}$
- Q: Any numeric issues you notice?

Naive Bayes classifier

- Assign class which maximizes: $c^* = argmax_c P(c|d)$
- Recap:

Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- To estimate $P(d|c)$ we **naively** assume f_i's are independent.
  - Hence $\widehat{P(d|c)} = \prod_{i=1}^{m} P(f_i|c)^{n_i(d)}$
- Q: Any numeric issues you notice?
- A_1: Estimates could be zero

Naive Bayes classifier

- Assign class which maximizes: $c^* = argmax_c P(c|d)$
- Recap:

Definition (Bayes theorem)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

- To estimate $P(d|c)$ we **naively** assume f_i's are independent.
  - Hence $\widehat{P(d|c)} = \prod_{i=1}^{m} P(f_i|c)^{n_i(d)}$
- Q: Any numeric issues you notice?
- A_1: Estimates could be zero
- A_2: Short vs. long documents

Naive bayes example

Assume the following Bow model with 4 documents for each class:

	act	audio	boring	rocks
mapping	0	1	2	3
positive	1	3	1	3
negative	3	1	3	1

Naive bayes example

Assume the following Bow model with 4 documents for each class:

	act	audio	boring	rocks
mapping	0	1	2	3
positive	1	3	1	3
negative	3	1	3	1

- $d_3 =$ "Boring effects" , $d_3^{bow} = (0, 0, 1, 0)$

Naive bayes example

Assume the following Bow model with 4 documents for each class:

	act	audio	boring	rocks
mapping	0	1	2	3
positive	1	3	1	3
negative	3	1	3	1

- $d_3 =$ "Boring effects" , $d_3^{bow} = (0, 0, 1, 0)$
- Recall $\widehat{P(d|c)} = \prod_{i=1}^{m} P(f_i|c)^{n_i(d)}$

Naive bayes example

Assume the following Bow model with 4 documents for each class:

	act	audio	boring	rocks
mapping	0	1	2	3
positive	1	3	1	3
negative	3	1	3	1

- $d_3 =$ "Boring effects" , $d_3^{bow} = (0, 0, 1, 0)$
- Recall $\widehat{P(d|c)} = \prod_{i=1}^{m} P(f_i|c)^{n_i(d)}$
- $P(d|pos) = \frac{1}{4}^0 * \frac{1}{4}^0 * \frac{1}{4}^1 * \frac{1}{4}^0 = \frac{1}{4}$

Naive bayes example

Assume the following Bow model with 4 documents for each class:

	act	audio	boring	rocks
mapping	0	1	2	3
positive	1	3	1	3
negative	3	1	3	1

- $d_3 =$ "Boring effects" , $d_3^{bow} = (0, 0, 1, 0)$
- Recall $\widehat{P(d|c)} = \prod_{i=1}^{m} P(f_i|c)^{n_i(d)}$
- $P(d|pos) = \frac{1}{4}^0 * \frac{1}{4}^0 * \frac{1}{4}^1 * \frac{1}{4}^0 = \frac{1}{4}$
- $P(d|neg) = \frac{1}{4}^0 * \frac{1}{4}^0 * \frac{3}{4}^1 * \frac{1}{4}^0 = \frac{3}{4}$

Maximum entropy classifier

aka logistic regression

Definition (MaxEnt estimator)

$P(c|d) = \frac{1}{Z(d)} exp(\sum_i \lambda_{i,c} F_{i,c}(d, c))$

Maximum entropy classifier

aka logistic regression

Definition (MaxEnt estimator)

$P(c|d) = \frac{1}{Z(d)} exp(\sum_i \lambda_{i,c} F_{i,c}(d,c))$

- $Z(d)$ - normalization function

Maximum entropy classifier

aka logistic regression

Definition (MaxEnt estimator)

$$P(c|d) = \frac{1}{Z(d)} exp(\sum_i \lambda_{i,c} F_{i,c}(d, c))$$

- $Z(d)$ - normalization function
- $F_{i,c}$ is a *feature/class function*

Maximum entropy classifier

aka logistic regression

Definition (MaxEnt estimator)

$P(c|d) = \frac{1}{Z(d)} exp(\sum_i \lambda_{i,c} F_{i,c}(d,c))$

- $Z(d)$ - normalization function
- $F_{i,c}$ is a *feature/class function*
  - Defined: $F_{i,c}(d,c') = 1$ if feature i **appears** on document d and its estimated class is c, o.w. $F_{i,c}(d,c) = 0$

Maximum entropy classifier

aka logistic regression

Definition (MaxEnt estimator)

$P(c|d) = \frac{1}{Z(d)} exp(\sum_i \lambda_{i,c} F_{i,c}(d,c))$

- $Z(d)$ - normalization function
- $F_{i,c}$ is a *feature/class function*
  - Defined: $F_{i,c}(d,c') = 1$ if feature i **appears** on document d and its estimated class is c, o.w. $F_{i,c}(d,c) = 0$
- $\lambda_{i,c}$ - feature-weight parameters
  - large values imply f_i is a strong indicator for class c

Maximum entropy classifier

aka logistic regression

Definition (MaxEnt estimator)

$P(c|d) = \frac{1}{Z(d)} exp(\sum_i \lambda_{i,c} F_{i,c}(d,c))$

- $Z(d)$ - normalization function
- $F_{i,c}$ is a *feature/class function*
  - Defined: $F_{i,c}(d,c') = 1$ if feature i **appears** on document d and its estimated class is c, o.w. $F_{i,c}(d,c) = 0$
- $\lambda_{i,c}$ - feature-weight parameters
  - large values imply f_i is a strong indicator for class c

Fit procedure

- Training data used to estimate distribution F
- λ's are set to maximize entropy of induced distribution

Support vector machines

- Goal: Find hyperplane w which separates classes with margin large as possible

Support vector machines

- Goal: Find hyperplane w which separates classes with margin large as possible
- In this setting we define w as:

Definition (SVM hyperplane)

Let $c_j \in \{1, -1\}$ be the class of document d_j then:

$$w := \sum_j \alpha_j c_j \overrightarrow{d_j},\ \alpha_j \geq 0$$

- α_j are obtained by solving dual optimization problem.

Support vector machines

- Goal: Find hyperplane w which separates classes with margin large as possible
- In this setting we define w as:

Definition (SVM hyperplane)

Let $c_j \in \{1, -1\}$ be the class of document d_j then:

$$w := \sum_j \alpha_j c_j \overrightarrow{d_j},\ \alpha_j \geq 0$$

- α_j are obtained by solving dual optimization problem.

<u>Alternative fitting procedure:</u>

Definition (Gradient descent)

$$w = w - \alpha * \frac{\partial L(X,w)}{\partial w} \text{ , Update till convergence}$$

Results

Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	**82.9%**

Table: 3-fold average accuracies, unigrams appear at least 4 times on corpus.

Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	**82.9%**

Table: 3-fold average accuracies, unigrams appear at least 4 times on corpus.

- Recall human baseline ranges between 50% – 69%

Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	**82.9%**

Table: 3-fold average accuracies, unigrams appear at least 4 times on corpus.

- Recall human baseline ranges between $50\% - 69\%$
- Topic-based classification reached $90\%+$ accuracy

Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	**82.9%**

Table: 3-fold average accuracies, unigrams appear at least 4 times on corpus.

- Recall human baseline ranges between $50\% - 69\%$
- Topic-based classification reached $90\%+$ accuracy
  - Settings were multi-class
  - We conclude that sentiment analysis is harder

Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	**82.9%**

Table: 3-fold average accuracies, unigrams appear at least 4 times on corpus.

- Recall human baseline ranges between $50\% - 69\%$
- Topic-based classification reached $90\%+$ accuracy
  - Settings were multi-class
  - We conclude that sentiment analysis is harder
- The frequency vs. presence of features seems to make the difference

Results and discussion

ID	Features	count	freq/pres	NB	ME	SVM
1	unigrams	16165	freq	78.7%	NA	72.8%
2	unigrams	16165	pres	81.0%	80.4%	**82.9%**

Table: 3-fold average accuracies, unigrams appear at least 4 times on corpus.

- Recall human baseline ranges between $50\% - 69\%$
- Topic-based classification reached $90\%+$ accuracy
  - Settings were multi-class
  - We conclude that sentiment analysis is harder
- The frequency vs. presence of features seems to make the difference
  - Hence from this point authors use presence (**binarized occurrences**)

Results and discussion

Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	**82.9%**
3	uni+bigrams	32330	80.6%	80.8	**82.7%**
4	bigrams	16165	77.3%	77.4%	77.1%

Results and discussion

Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	**82.9%**
3	uni+bigrams	32330	80.6%	80.8	**82.7%**
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	**81.9%**

Results and discussion

Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	**82.9%**
3	uni+bigrams	32330	80.6%	80.8	**82.7%**
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	**81.9%**
6	adjectives	2633	77.0%	77.7%	75.1%

Results and discussion

Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	**82.9%**
3	uni+bigrams	32330	80.6%	80.8	**82.7%**
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	**81.9%**
6	adjectives	2633	77.0%	77.7%	75.1%
7	top 2633 unigrams	2633	80.3%	81.0%	81.4%

Results and discussion

Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	**82.9%**
3	uni+bigrams	32330	80.6%	80.8	**82.7%**
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	**81.9%**
6	adjectives	2633	77.0%	77.7%	75.1%
7	top 2633 unigrams	2633	80.3%	81.0%	81.4%
8	unigram+position	22430	81.0%	80.1%	**81.6%**

Table: 3-fold average accuracies, bigrams appear at least 7 times on corpus.

Results and discussion

Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	**82.9%**
3	uni+bigrams	32330	80.6%	80.8	**82.7%**
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	**81.9%**
6	adjectives	2633	77.0%	77.7%	75.1%
7	top 2633 unigrams	2633	80.3%	81.0%	81.4%
8	unigram+position	22430	81.0%	80.1%	**81.6%**

Table: 3-fold average accuracies, bigrams appear at least 7 times on corpus.

- Adding bigrams doesn't improve results; Bigrams alone is worse

Results and discussion

Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	**82.9%**
3	uni+bigrams	32330	80.6%	80.8	**82.7%**
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	**81.9%**
6	adjectives	2633	77.0%	77.7%	75.1%
7	top 2633 unigrams	2633	80.3%	81.0%	81.4%
8	unigram+position	22430	81.0%	80.1%	**81.6%**

Table: 3-fold average accuracies, bigrams appear at least 7 times on corpus.

- Adding bigrams doesn't improve results; Bigrams alone is worse
- Part-of-speech: "I love this movie" vs. "This is a love story"

Results and discussion

Using feature presence

ID	Features	count	NB	ME	SVM
2	unigrams	16165	81.0%	80.4%	**82.9%**
3	uni+bigrams	32330	80.6%	80.8	**82.7%**
4	bigrams	16165	77.3%	77.4%	77.1%
5	unigrams+POS	16695	81.5%	80.4%	**81.9%**
6	adjectives	2633	77.0%	77.7%	75.1%
7	top 2633 unigrams	2633	80.3%	81.0%	81.4%
8	unigram+position	22430	81.0%	80.1%	**81.6%**

Table: 3-fold average accuracies, bigrams appear at least 7 times on corpus.

- Adding bigrams doesn't improve results; Bigrams alone is worse
- Part-of-speech: "I love this movie" vs. "This is a love story"
- Position based on dividing text into quarters.

Reproduce results

Reproduce results
(2018)

- We have tried to reproduce the experiment for the best setting reported

Features	count	NB	ME	SVM	MLP
unigrams	16165	81.0%	80.4%	**82.9%**	NA

Reproduce results
(2018)

- We have tried to reproduce the experiment for the best setting reported

Features	count	NB	ME	SVM	MLP
unigrams	16165	81.0%	80.4%	**82.9%**	NA
unigrams	16165	77.48%	**81.52%**	80.66%	**82.75%**

Table: Original vs. our results

Reproduce results
(2018)

- We have tried to reproduce the experiment for the best setting reported

Features	count	NB	ME	SVM	MLP
unigrams	16165	81.0%	80.4%	**82.9%**	NA
unigrams	16165	77.48%	**81.52%**	80.66%	**82.75%**

Table: Original vs. our results

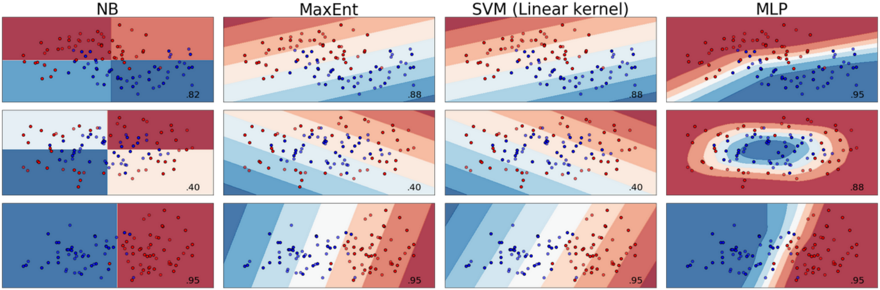
- MLP is 2-layer neural network with 100 Relu neurons
- **No tuning** was used (sklearn 0.19.2 default parameters)
  - Plus not all described processing steps applied
- Notebook is available here

Classifier comparison
(2018)

- Let's observe our classifiers decision boundaries for some toy datasets

Classifier comparison
(2018)

- Let's observe our classifiers decision boundaries for some toy datasets



- Accuracy is reported

Conclusions

Conclusions

- Unigrams presence setting achieves the best performance
  - Apply feature selection algorithms

Conclusions

- Unigrams presence setting achieves the best performance
  - Apply feature selection algorithms
- Contrarily, performance isn't comparable to topic classification

Conclusions

- Unigrams presence setting achieves the best performance
  - Apply feature selection algorithms
- Contrarily, performance isn't comparable to topic classification

Review example

"This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up."

Conclusions

- Unigrams presence setting achieves the best performance
  - Apply feature selection algorithms
- Contrarily, performance isn't comparable to topic classification

Review example

"This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up."

- Difficult for bag-of-words classifiers.

Conclusions

- Unigrams presence setting achieves the best performance
  - Apply feature selection algorithms
- Contrarily, performance isn't comparable to topic classification

Review example

"This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up."

- Difficult for bag-of-words classifiers.
- Authors suggest determining the **focus** of each sentence, if is on/off topic.

Thank you for participating!
Questions?