EB5204 : NEW MEDIA AND SENTIMENT MINING

MODULE 2.2: SUPERVISED TRAINING METHODS

Dr Wang Aobo

isswan@nus.edu.sg

Module Objectives

- Identify and evaluate basic methodologies for supervised training algorithms commonly used in sentiment mining

Agenda

- Lexicon-based approach for sentiment mining

- Supervised training method session
  - Naïve Bayes methodology
  - kNN classification method
  - Maximum entropy method
  - Support vector machine (SVM)

1. Lexicon-based semi-supervised approach

Lexicon-based approach

- Requires a **sentiment lexicon** (patterns - words, phrases, idioms), composite expression, rules of opinions
- Essentially compute the sentiment polarity of a target in a sentence by using **sentiment aggregation** taking into account the distances of the sentiment expressions and the target entity/aspect.
- Also consider sentiment shifters, *but*-clauses, and the syntactic relationships of sentiment expressions and opinion targets.

Lexicon semi-supervised approach

- **Opinion Lexicon**: list of words, phrases, expressions used to express sentiment
  - **+ve terms**: good, excellent, cool, wow!, …
  - **-ve terms**: stinks, bad, old-fashioned, yuck!,…
  - **Phrases**: “cost an arm and a leg”
  - **Slangs**: “referee kayu!”, “kiasu”
  - **Idioms**: “soft hearted”, “hard nosed”
  - There are flaws – note due to the context.
    - *“The bag is a little big for her”*
    - *“That’s pretty ugly”*

Creating the Opinion Lexicon

- **Combination** of manual, dictionary and corpus based approaches
  - **Manual**: by inspection
  - **Dictionary**: own source, WordNet, SenitWordNet for words
    - Often cannot get context dependent words/phrases
- The process is **iterative**
  - Knowing some sentiment words helps to find more
    - E.g., "The dress is **elegant** and **refined** but **provocative**"
    - If "elegant" is +ve, then "refined" is +ve; "provocative" is –ve

Apply Opinion Lexicon

"The phone's call quality is not good, but its battery life is long."

- Mark sentiment words and phrases

  The phone's call quality is not good [+1], but its battery life is long.

- Apply sentiment shifters

  The phone's call quality is not good [-1], but its battery life is long.

- Handle but-clauses

  The phone's call quality is not good [-1], ***but*** *its battery life is long [+1].*

- Aggregate opinions
  - *Call quality* – negative, *battery life* - positive

A simple method as example

- Assumption: the target entities and aspects are known (as NER in a sentence, or in a specific review)
- Main steps (Ding, Liu and Yu, 2008):
  - Mark sentiment words and phrases
  - Apply sentiment Rules (such as negation words)
  - Handle *but*-clauses (*but, however, except for, except that, with the exception of*)
    - Indicating contrary, but not always ("*x is great, but y is better*")
  - Aggregate opinions
    - Across sentences and time, weighing by its importance.

Sentiment Rules – Sentiment Reverse

- Sentiment shifters
- Negation words like *not*, *never*, *none*, *nobody*, *nowhere*, *neither*, *cannot*…(identify with the adjective if less than 'n' words?)
  - "*The taste is not good*."
- Modal auxiliary verbs (e.g., *would*, *should*, *could*, *might*, *must*, *ought*) may change sentiment orientation, but not always.
  - "*The service could be improved*."
- Adverbs like *barely*, *hardly*
  - "*It hardly works*."
- Words like *fail*, *omit*, *neglect*, etc.
  - "*The car failed to start*."

POS + shifter => NEG

NEG + shifter => POS

Sentiment reverse- opinionated item

- *Decreased* (or *removal*, *disappearance*) and *increased* quantity of an opinionated item (NEG and POS, often nouns) can change its orientation.

  "*This drug reduced my pain significantly.*"

  "*My pain disappeared after taking the drug.*"

  "*The earphone can isolate noise.*"

POS + decreased => NEG

NEG + decreased => POS

POS + increased => POS (intensification)

NEG + increased => NEG (intensification)

Sentiment composition rules

- The quantity or change of quantity of *potential positive or negative items* (PositivePolarityItem or NegativePolarityItem, like "*battery life*"(1), "*price*"(2), "*memory*"(3), etc.)

  "*The camera memory went up a lot*."

  "*The battery life is short*."

  "*Sony reduced the price of the camera*."

NPI + no/low/less/decreased => POS

NPI + large/larger/increased => NEG

PPI + no/low/less/decreased => NEG

PPI + large/larger/increased => POS

Sentiment conflicts

- When multiple sentiment words occur together
  "*terribly(-) good(+)*", "*pretty (+) ugly (-)*"
- Resolution is to rank the constituents on the basis of relative **weights** (importance)
- E.g. if two sentiment words have the opposite polarity
  - if ADV+ADJ, the first adjective gets higher weight
  - if V+OBJ, the verb polarity prevails
  - If in two clauses connected by "but", the one after the connector is dominant.

Other rules

- ***Intensification*** rule: "*extremely happy*" (for granular scale of sentiment)
- Consume resource and produce waste:

  "*This computer uses a lot of electricity*."
- Add other rule for deviation from the norm or a desired value range

  "*After taking the drug, my blood pressure went up to 200*."

  "I got 45 marks for my score."

Sentiment words in non-opinion contexts

Sometimes need to be careful with such words like…

- Entity names containing sentiment words

  "*Best Denki*"

- Function names containing sentiment words

  "*fast forward*" "*I'm feeling lucky*"

- Greetings and good wishes

  "*Good morning*" "*hope you get well soon*"

- Author's self-description

  "*I know Samsung products very well*."

Senses of sentiment words

Words with different senses can have different sentiment. (SentiWordNet)

- Many sentiment words have multiple meanings or senses and they do not always express sentiment in all contexts.

  "*great grandfather*"

  "*This is clearly a bad phone*."

  "*Well, I do not think that this is a good car*."

- Some express different sentiments in different context

  "This car smells."

  "This perfume smells good."

  "This room smells bad."

  "This room has a smell."

  "This room has a foul smell."

  "This room has a nice smell."

Lexicon semi-supervised approach

- Uses observations from words/ their patterns to construct rules for sentiment scores.
- The sentiment scores from individual ‘patterns’ are modified through rules then aggregated.
- Most used in industry practice actually. Especially finance because of its flexibility.
- Don’t under-estimate its overall effectiveness

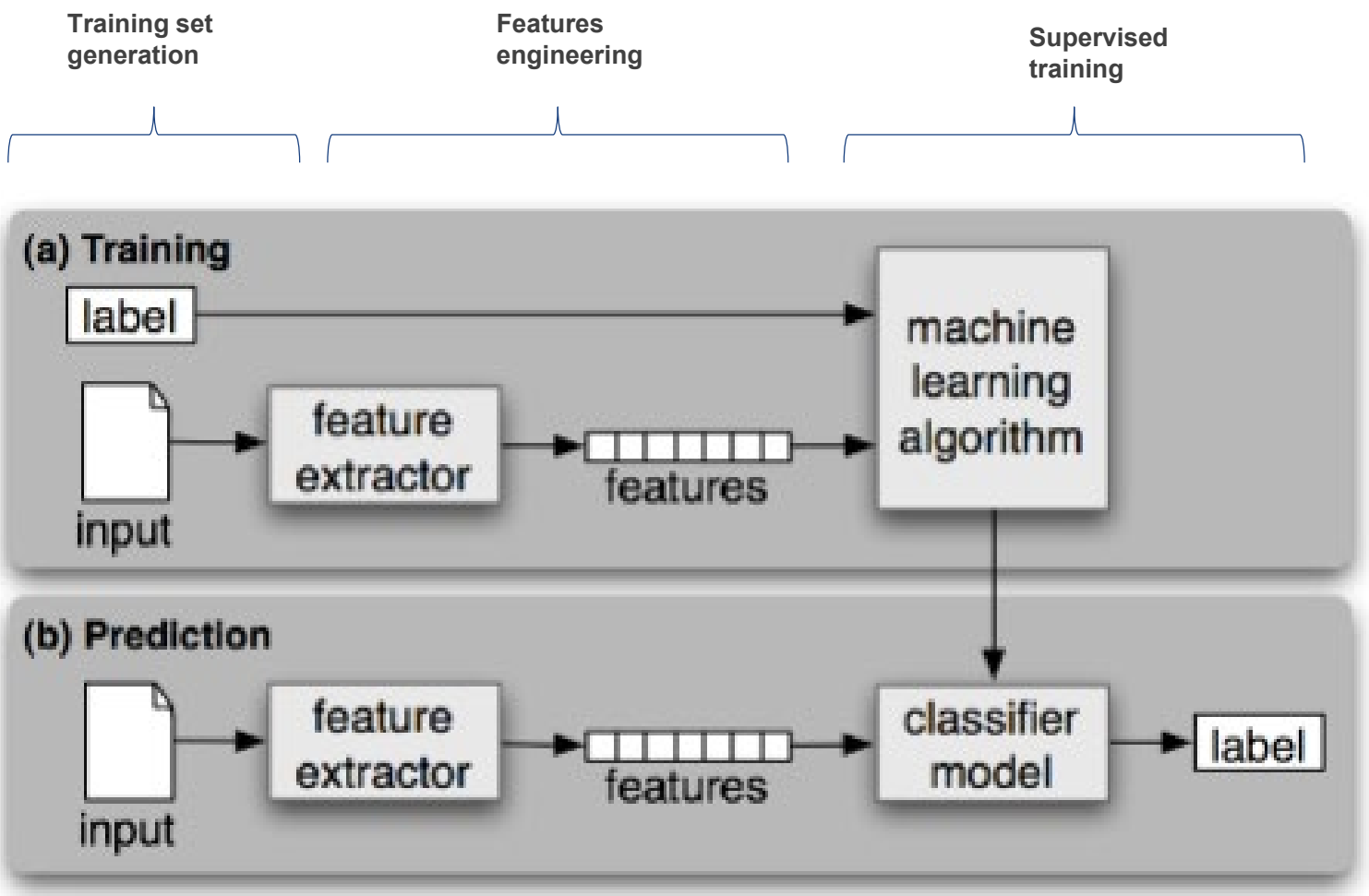
https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2010.01625.x
2011 paper sets off trend in finance.

Pros and Cons of Training Methods

- **Unsupervised**
  - Pros:
    - Domain independence method, shown to perform well in a large number of applications.
    - Flexible, can be easily extended and improved.
  - Cons:
    - Heavy investment in time and effort to build the initial knowledge base of lexicon, patterns and rules
    - Still need to handle domain differences, mainly domain- or context-dependent sentiment expressions.
- **Supervised**
  - Pros: powerful learning of all kinds of features
  - Cons:
    - Dependent on training data which needs to be manually labelled for each domain.
    - Models trained in one domain often perform poorly in another domain.

2. Supervised approach

Overview of Supervised Classifier System



Sentiment Analysis Classifiers:

- **Basic Algorithms:**
  - Naïve Bayes
  - K-Nearest neighbor
  - Max Entropy
  - Support Vector machine
  - Deep learning (detail in another class)
  - Sequential models - hidden markov models, conditional random fields (detail in another class)

Naïve Bayes Classifier

- Based on Bayes Rule - a conditional probability model. Recap on Bayes Rule. Bear in mind that data science is applied statistics.

Bayes Rule:

$$Prob(C_k|x_i) = \frac{Prob(x_i|C_k)Prob(C_k)}{Prob(x_i)}$$

Here, C_k refers to class k. x_i is the feature. So for example, k can mean positive, negative …
x_i can refer to any lexicon features or other feature 'engineered' we earlier discussed.

Bayes Rule question (mini discussion)

- Suppose we have one sentence – “pretty ugly”.
- There are altogether 100 test documents, of which 60 are +ve and 40 are –ve.
  - Of the +ve documents, pretty occurred in 40 of them and ugly in 9 of them,
    Of the -ve documents, pretty occurred in 5 of them and ugly in 30 of them,
    Now from your understanding, write down:

1. $Prob(pretty|+ve)$: _____
2. $Prob(pretty|-ve)$: _____
3. $Prob(ugly|+ve)$: _____
4. $Prob(ugly|-ve)$: _____

From these, what is the polarity for the sentence?

Naïve Bayes Classifier

$$\hat{y} = \max_{k \in \{1,\ldots,K\}} Prob(C_k) \sum_i^n \log Prob(x_i | C_k)$$

- Since taking log won't impact the optimisation, take log and do a sum instead of product to present computing overflow.
- Notice that since all the probabilities of features are 'multiplied', they are regarded as independent.
- In other words, one feature doesn't impact the other. This is not entirely true with semantic dependencies (in language)

Naïve Bayes Classifier

- There are two ways that Bayes Rule can be used in classification. These two ways differ by the ***weighting method***.

$$Prob(x_i|C_k)$$

- The first weighting is binary. That is as long you see the word feature in the document, it is weighted 1. Otherwise 0.

- Another method (also called ***multinomial naïve Bayes***) weighs word frequency and normalized by the total number of words

Naïve Bayes Classifier

- ***multinomial naïve Bayes***

		Feature Words										
	Train Documents	**chicken**	**rice**	**terrible**	**Worst**	**Pissed**	**ok**	**lousy**	**Good**	**Liked**	**Food**	Class
1	Some of the best chicken, the rice was terrible. Liked food.	1	1	1						1	1	1
2	Worst Nandos on the planet. Pissed with lousy food					1	1		1		1	-1
3	Worst nando's Terrible. Too terrible to write reviews on food				2	1						-1
4	Food is ok. Chicken was terrible. Service was lousy.				1			1	1		1	-1
5	Lousy food. But liked music and atmosphere.								1	1	1	1

No of words in Pos Sentiment	9	Eg chicken, rice, terrible, Good, liked...	P(Pos)	0.4 α 1
No of words in Neg Sentiment	=SUM(C9	worst, lousy ...	P(Neg)	0.6

Freq word appears in Pos	1	1	1	0	0	0	1	1	2	2
Freq word appears in Neg	0	0	3	2	1	1	2	0	0	2

P(word|Pos)	0.22	0.22	0.22	0.11	0.11	0.11	0.22	0.22	0.33	0.33
P(word|Neg)	0.09	0.09	0.36	0.27	0.18	0.18	0.27	0.09	0.09	0.27

0.22 = (1+a) / 9 when a = 1
0.09 = (0+a) / 11 when a = 1

Naïve Bayes Classifier

- There are two ways that Bayes Rule can be used in classification. These two ways differ by the ***weighting method***.

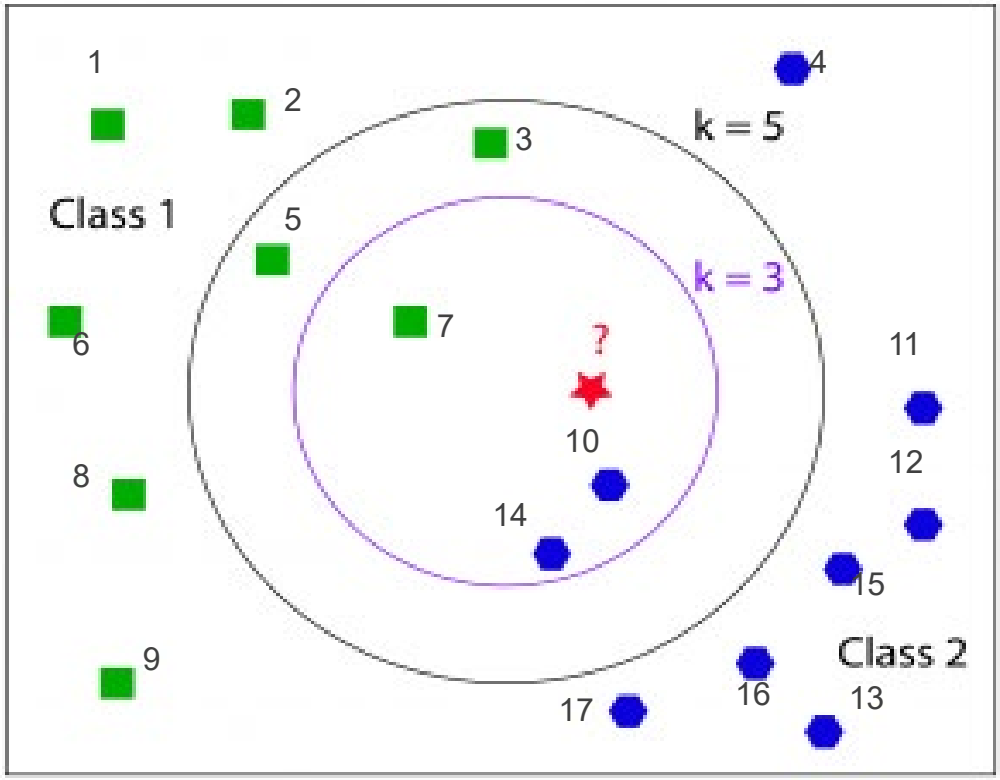
$$Prob(x_i|C_k)$$

- Used with good feature engineering, the Bayes rule classifier actually works very well and efficiently.

Naïve Bayes Classifier is the **golden standard** by which all classifiers are measured.

K-nearest neighbor classifier (an illustration)

- Compare the training text data now against the test text data using a similarity score. We shall use the cosine distance which is the most popular.



Which are the 3 closest points are to the red star? How about 5 closest? If they take on the same importance in influencing the red star, what class will the red star be?

K-nearest neighbor classifier

- Not to be confused with KMeans clustering method, which is an unsupervised method. This is a supervised classification technique that is based on ***similarity scores***.

- Essentially, the reference data (labelled) are measured using a certain 'distance'.

- Then the test data are compared with these reference data, picking out the ***most similar 'K' samples***. The test data class with the 'K' samples labels are weighted through a ***weight scheme*** for its final score.

K-nearest neighbor classifier in sentiment mining (steps II)

- Say for k=1, the test data will just be labelled with the closest (most similar) test data.
- For k=2, the test data can be weighted equally with the training data, or by the similarity scores.
- kNN is not often used as other simple methods as it is not as efficient computationally nor work as well generally.

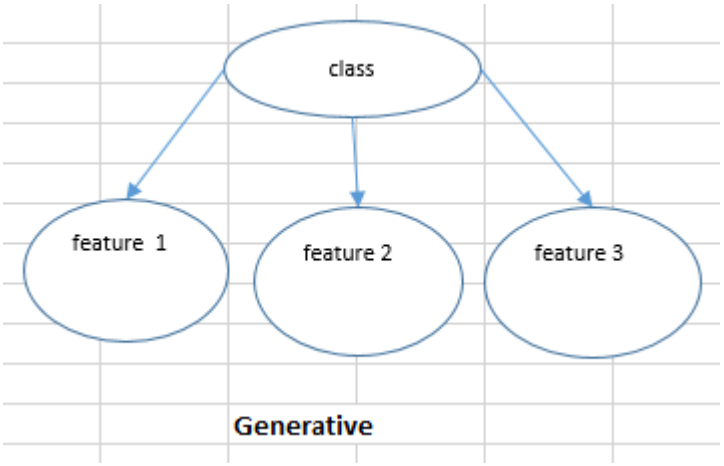
2. Supervised training method - Maximum entropy

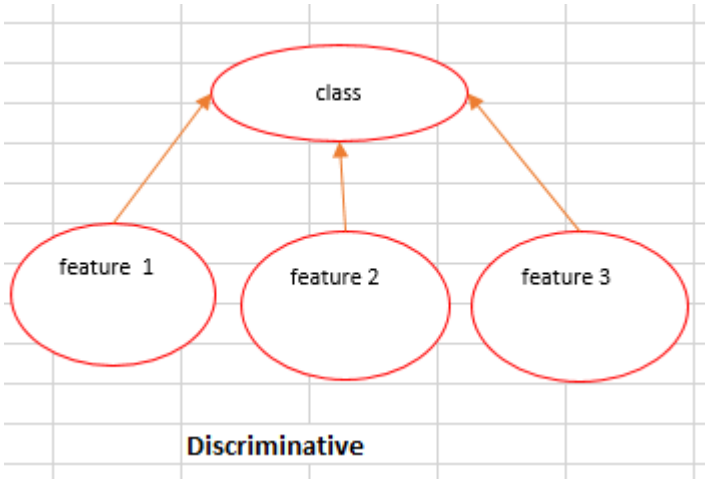
Discriminative vs Generative Classifiers

- Thus far we have studied generative classifiers, which uses joint probabilities, whilst discriminative classifiers use conditional probabilities.

- By this, it means generative classifiers tend to look at the data as it is (empirically) and draws inferences on the class.

- On the other discriminative classifiers, make some 'educated' (use models) to draw inferences.

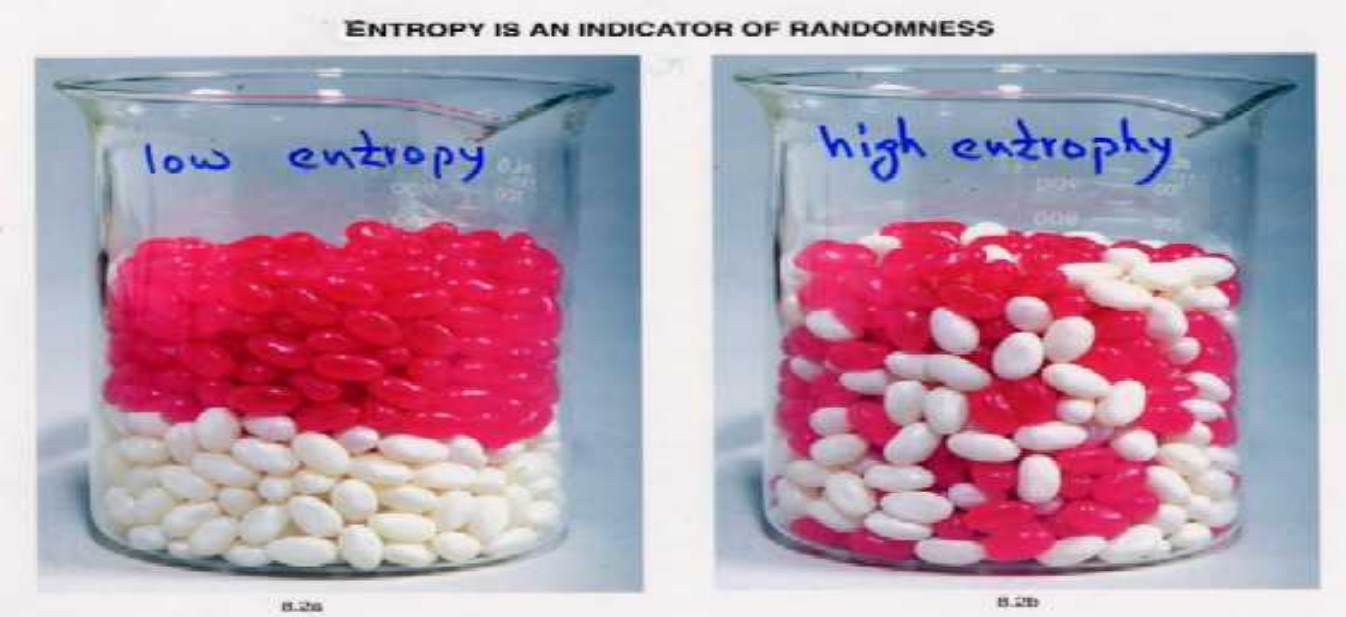




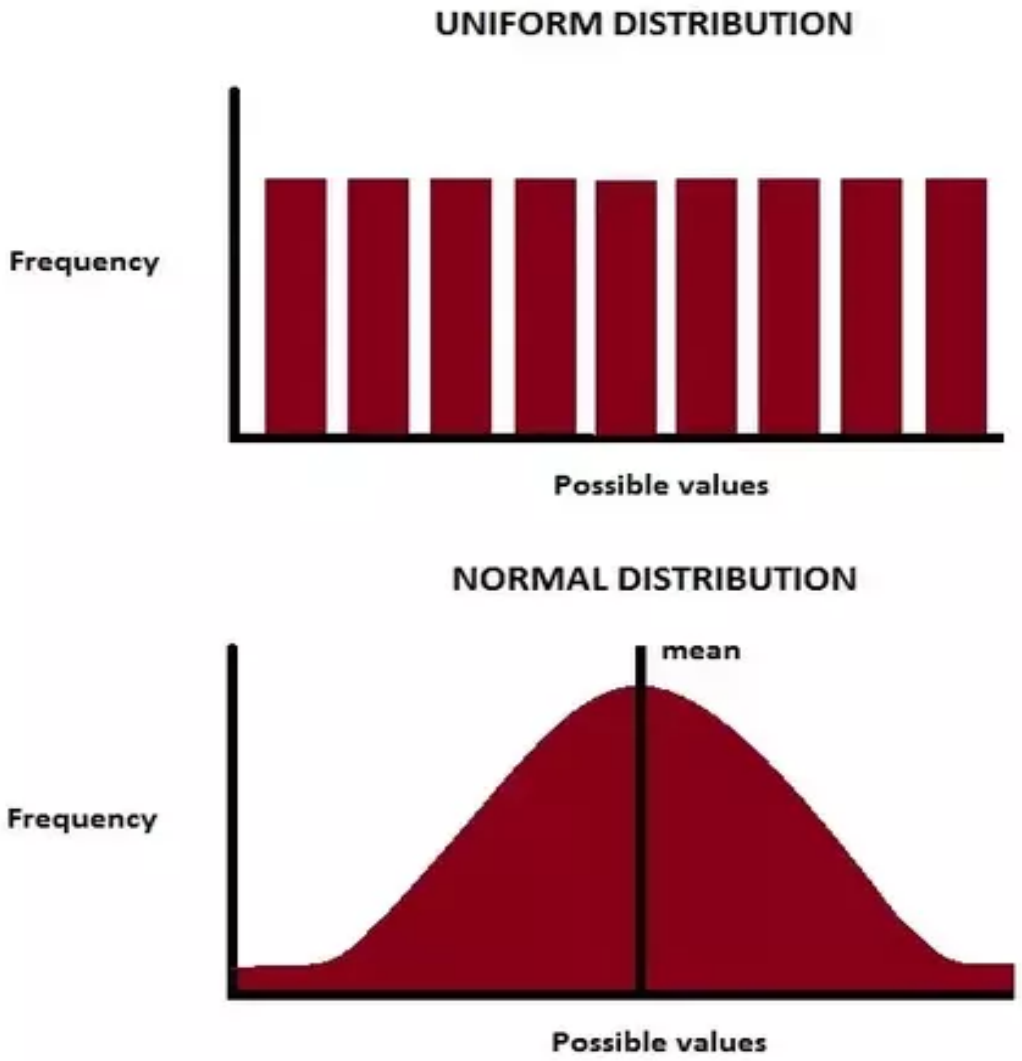
Entropy

- A maximum entropy model classifier needs to be understood from the concept of entropy.

- Entropy refers to how random a system is. A system that is more random has higher entropy.



Maximum Entropy Classifier



Maximum Entropy Classifier (the mathematics)

- Uniform distribution preferred as it has the maximum entropy
- How to apply in classification?

- The entropy of a random variable d is defined as.

$$\mathrm{H(d)} = -p(d)\log(p(d))$$

In text processing context, the d 'refers' to say how often a 'document' appears across all the texts? How random it is?

Maximum Entropy Classifier (the mathematics)

- The entropy of a random variable d is defined as.

$$\mathrm{H(d)} = -p(d)\log(p(d))$$

how often a 'document' appears across all the texts?
How random it is?

- Considering all the possible pairs of (c,d), apply Conditional Entropy

$$H(c|d) = -\sum_{c,d} \tilde{p}(d)p(c|d)log\, p(c|d)$$

how often a 'document' appears across all the texts and categories?
How random it is?

Maximum Entropy Classifier (the mathematics)

- The entropy of a random variable d is defined as.

$$\mathrm{H(d)} = -p(d)\log(p(d))$$

- Considering all the possible pairs of (c,d), apply Conditional Entropy

$$H(c|d) = -\sum_{c,d} \tilde{p}(d)p(c|d)log\, p(c|d)$$

Maximise the objective function such that it has the largest entropy

$$\arg\max_{p\in C}(\mathrm{H(c|d)}) = \arg\max_{p\in C}(-\sum_{c,w} \tilde{p}(d)p(c|d)log\, p(c|d))$$

MaxEnt Intuition

- Maximise the objective function such that it has the largest entropy
- the optimisation provides the weights (λ) to the MaxEnt classifier:
- Contrast this to the Naïve Bayes classifier, which is similar but doesn't have weights.

$$\arg\max_{p \in C}(\mathrm{H}(\mathrm{c}|\mathrm{d})) = \arg\max_{p \in C}(-\sum_{c,w} \tilde{p}(d)p(c|d)log\, p(c|d))$$

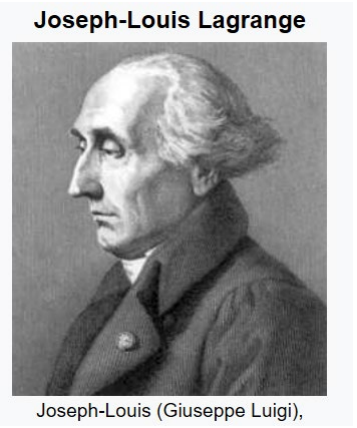
$f_i(c,d) = 1$ for the data when word x appears in class c. 0 otherwise.

$$P(c \mid d, \lambda) = \frac{\exp \sum \lambda_i f_i(c,d)}{\sum \exp \sum \lambda_i f_i(c',d)}$$

For normalisation to 1.0

MaxEnt

- Know-how to optimize (calculate λ) exactly ?
  - **Lagrange** multipliers
  - duality principle
  - Lagrange duality



Joseph-Louis (Giuseppe Luigi),

$$\mathcal{L}(x, y, \lambda) = f(x, y) + \lambda \cdot \Big(g(x, y) - c\Big)$$

$$\nabla_{x,y,\lambda} \mathcal{L}(x, y, \lambda) = 0 \iff \begin{cases} \nabla_{x,y} f(x, y) = \lambda \nabla_{x,y} g(x, y) \\ g(x, y) = 0 \end{cases}$$

$$P(c \mid d, \lambda) = \frac{\exp \sum \lambda_i f_i(c, d)}{\sum \exp \sum \lambda_i f_i(c', d)}$$

Maximum Entropy Classifier (the optimization)

- Once the optimisation is solved over the training data, it gives a set of weights λ_i that corresponds to the features. The no of weights is equal to the

  # of features * # of class.

- Each of the weight determine how important a feature is in deciding if it belongs to that class.

- This is the major advantage over the Naïve Bayes, since the weights are optimised from the data. But it also makes MaxEnt prone to over-fitting.

$$P(c \mid d, \lambda) = \frac{\exp \sum \lambda_i f_i(c, d)}{\sum \exp \sum \lambda_i f_i(c', d)}$$

Maximum Entropy Classifier

- Intuition for MaxEnt classifier:
  - First, define feature(s). A feature i - $f_i(c,d)$ is a any pattern of unigrams, bigrams, words within syntactic contexts etc. in class c.
  - $f_i(c,d) = 1$ for the data when word x appears in class c. 0 otherwise.
  - The distribution of the features $f_i(c,d)$ are then expressed either empirically or via a model.

MaxEnt classifier (example)

- Suppose learnt from training data set:
  - Only Two Unigram features: "Good" , "Bad"
  - Two polarity : +ve and -ve
  - Features weights :$\lambda 1$ $\lambda 2$ $\lambda 3$ $\lambda 4$
  - *f(+ve, good) = 1000; f(-ve, good) = 100*
  - *f(-ve,bad) = 1000; f(+ve,bad) = 100*

# of λ = 2*2	Good	Bad
Positive sentiment	$\lambda 1 = 0.5$	$\lambda 2 = 0.25$
Negative sentiment	$\lambda 3 = 0.1$	$\lambda 4 = 0.75$

- Suppose for a sentence "*The chicken tastes really good whilst the rice was bad.*" How will you classify it?

MaxEnt classifier (example)

- "*The chicken tastes really good whilst the rice was bad.*" Since both features appear in the sentence, the resulting voting scheme is:
  - *f(+ve, good) = 1k;* *f(-ve, good) = 0.1k*
  - *f(-ve,bad) = 1k;* *f(+ve,bad) = 0.1k*

# of λ = 2*2	Good	Bad
Positive sentiment	0.5	0.25
Negative sentiment	0.1	0.75

$$P(c \mid d, \lambda) = \frac{\exp \sum \lambda_i f_i(c,d)}{\sum \exp \sum \lambda_i f_i(c',d)}$$

	Sentence weights
	$e^{0.5*1k+0.25*0.1k} = e^{525}$
P(+ve | d, λ)	$e^{525}/(e^{525}+e^{760})$ = P1
	$e^{0.1*0.1k+0.75*1k} = e^{760}$
P(-ve | d, λ)	$e^{760}/(e^{525}+e^{760})$ = P2

- So which is the correct classification?

Comparing MaxEnt vs Naïve Bayes

- Previous Example
  - one sentence – "***pretty ugly***".
  - There are altogether 100 test documents, of which 60 are +ve and 40 are –ve.
  - Of the +ve documents, pretty occurred in 40 of them and ugly in 9 of them, Of the -ve documents, pretty occurred in 5 of them and ugly in 30 of them,
  - From these, what is the polarity for the sentence?

Prob(+ve) = 40/60 * 9/60 * 60/100 = 3/50

Prob(-ve) = 5/40 * 30/40 * 40/100 = 3/80

- **In this eg, we don't have a lot of information, (sentence is really short)**
- NB would have given the most likely answer (out of this little information) but being likely doesn't mean correct.
- Instead MaxEnt by allocating weights force the answer to a more 'intelligent' answer.

Features in MaxEnt

- Another advantage of using MaxEnt is the use of features.
- The same idea is applied here with MaxEnt. We can use ideas like if two words appear very close to one another, or has accented Latin character etc. The possibilities are limitless.
- Example feature:

$$[c = \text{LOCATION} \wedge \text{hasAccentedLatinChar}(w)]$$

If the word token is a Location and has a letter “w”

f(Waterloo) = 1
f(Singapore) = 0
f(firewall) =0

Maximum Entropy Classifier

- In maximum entropy classifier
  - the independent assumptions are relaxed.
  - The underlying principle of maximum entropy is that without external knowledge, uniform distribution should be preferred as it has the maximum entropy

- It is useful in helping distribute weights among the 'correlated' features.

Support Vector Classifier (SVC)

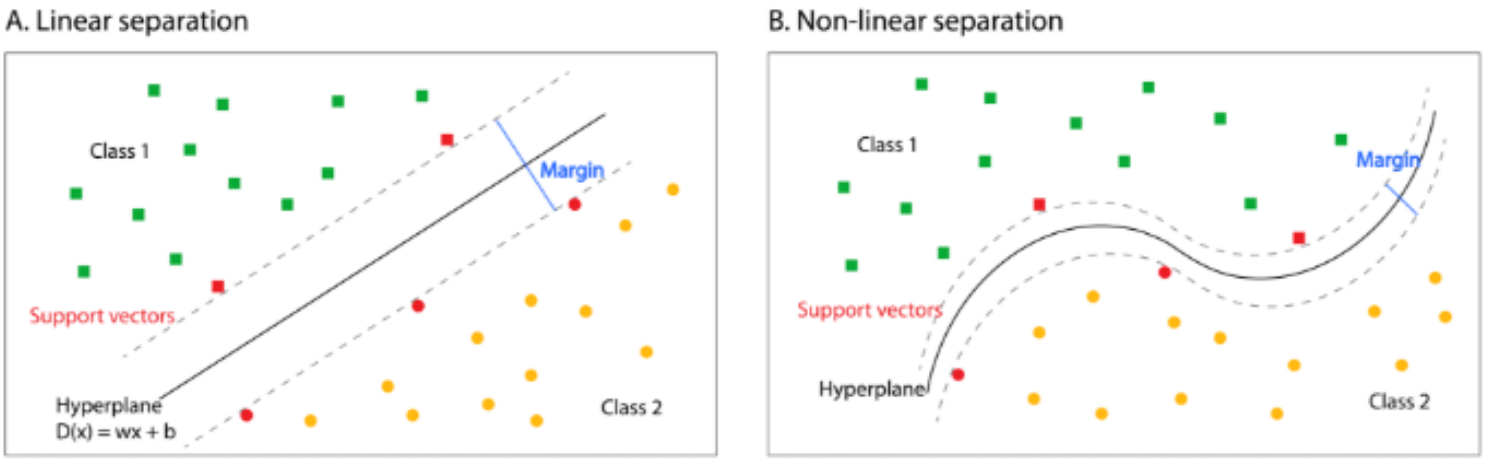
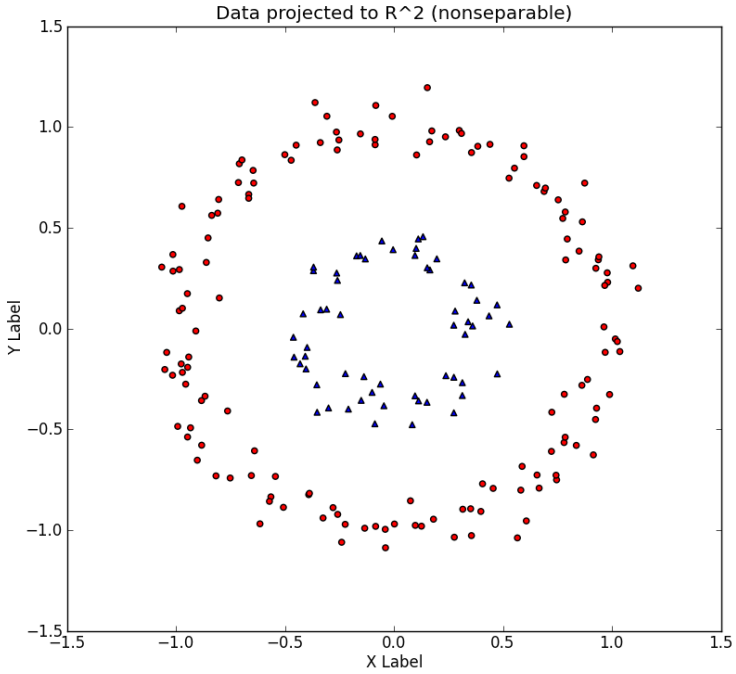


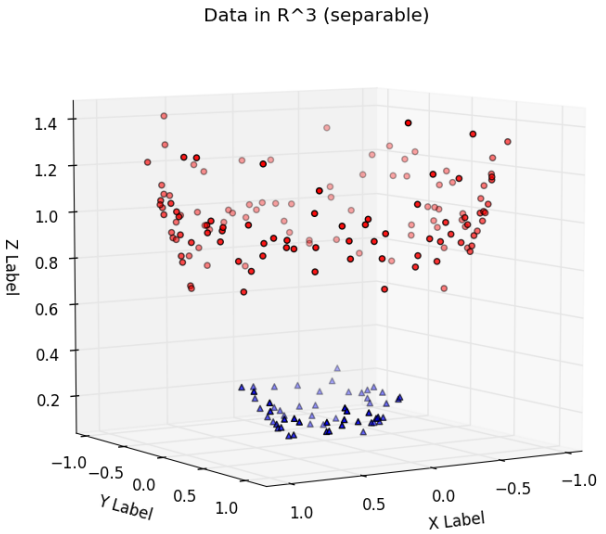
Illustration of separating two classes using SVMs. Linear (A.) and non-linear (B.) perfect separation of two classes (green and orange) with a hyperplane (black) and maximal margin (blue and dotted gray lines). Support vectors defining the hyperplane are in red. No misclassifiactions or margin violations are included.

- In the discussions above, we had touched upon the linearly-separable case (a straight line can pass through the circles and dots). Most cases, it is non-linearly separable as above.

- To resolve this, SVC uses the idea of kernels. Kernels are just functions to transform original data points to 'new' data points, which can then be separated.

Kernel trick for separation



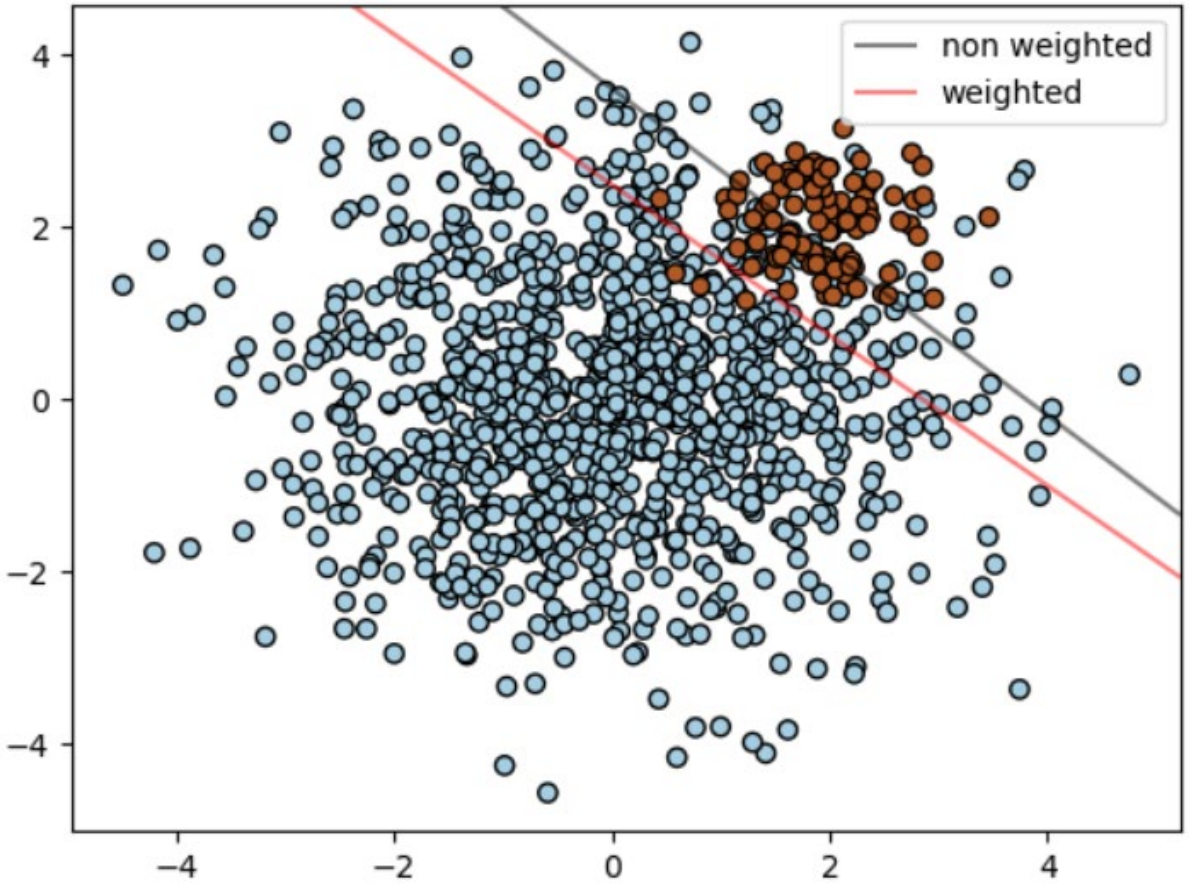


The *kernel function* can be any of the following:

- linear: $\langle x, x' \rangle$.
- polynomial: $(\gamma \langle x, x' \rangle + r)^d$, where d is specified by parameter `degree`, r by `coef0`.
- rbf: $\exp(-\gamma \|x - x'\|^2)$, where γ is specified by parameter `gamma`, must be greater than 0.
- sigmoid $\tanh(\gamma \langle x, x' \rangle + r)$, where r is specified by `coef0`.

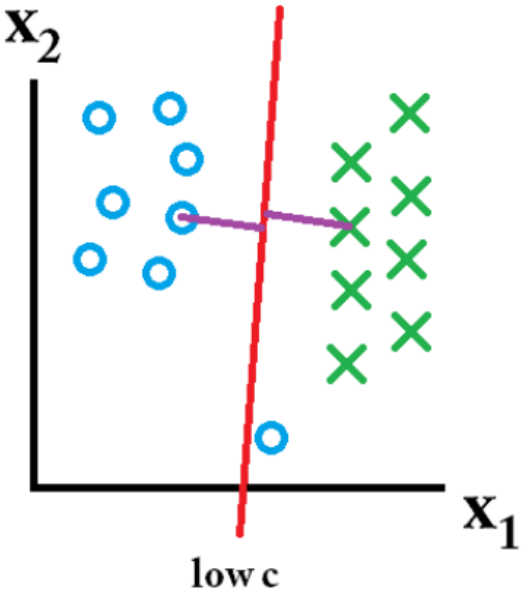
Unbalanced Data

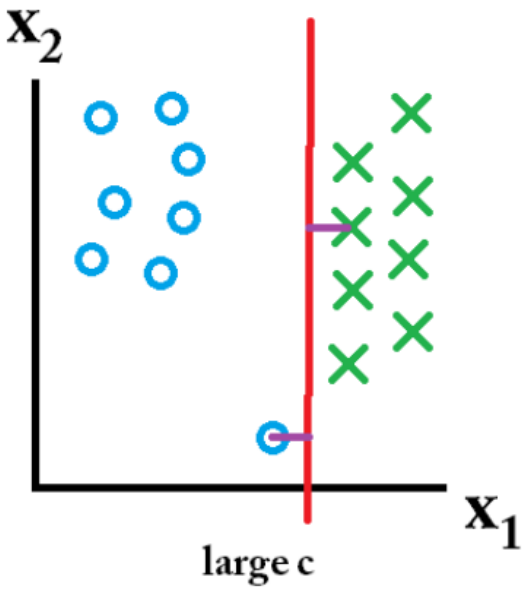
- Weighted Classes



The magic of “C”

- C parameter





Summary

- We have covered SVC and MaxEnt from an intuitive point of view. Going back to NLP and sentiment analysis in general, know that many of the tasks involve supervised classification.
- The SVC and MaxEnt are improvements over the Naïve Bayes classifier.
- The SVC is a margin classifier seeking to classify points with largest distance amongst the support vectors, making it useful for sparse matrices common in NLP.
- The MaxEnt seeks to maximise the entropy of the data set, and uses indicator features function. The expected value of the features are constrained to be equal empirically and conditionally on the model, which in the process produces weights for the features.

Reference

- Support vector machine http://cs229.stanford.edu/notes/cs229-notes3.pdf

- Max entropy classifier:
https://web.stanford.edu/class/cs124/lec/Maximum_Entropy_Classifiers.pdf

- Naives Bayes vs MaxEnt
http://www.denizyuret.com/2010/11/naive-bayes-is-joint-maximum-entropy.html