Text Classification with Naive Bayes

Tanel Alumäe

Is this spam?

Saatja: Cindy Pinda <membership@mrjbf.org>
Reply-to: sweetcindyj68@gmail.com
Kellele: tanel.alumae@phon.ioc.ee
Teema: I am Cindy
Kuupäev: Fri, 17 Nov 2017 06:59:27 +0200 (EET)

Hi,

I am Cindy and my surname is Pinda.I am 18 years old. My father was a very wealthy oil merchant and businessman in Turkey. My father was poisoned to death by his brothers because they want to take over his properties and wealth.My Mother died after giving birth to me.I am currently in my secondary school where i stay in their boarding house for students.

My father left family valuables worth the sum of Two Million United States Dollars in a security company for me which i want you to stand as my trustee to claim this from the security company. They contacted me and informed me that the deposit is due for claim and demand that i should come forward to claim or send a trustee. I cannot handle this so i am contacting you to help me as i am too young and cannot follow the process. I will authorize you to them and they will release the fund to you.i wait to read from you.Write back to me in my email

sweetcindyj68@gmail.com

Thanks and God bless.

Cindy Pinda

Probably male or female author?

- *By 1925 present-day Vietnam was divided into three parts under French colonial rule. The southern region embracing Saigon and the Mekong delta was the colony of Cochin-China; the central area with its imperial capital at Hue was the protectorate of Annam…*
- Clara never failed to be astonished by the extraordinary felicity of her own name. She found it hard to trust herself to the mercy of fate, which had managed over the years to convert her greatest shame into one of her greatest assets…

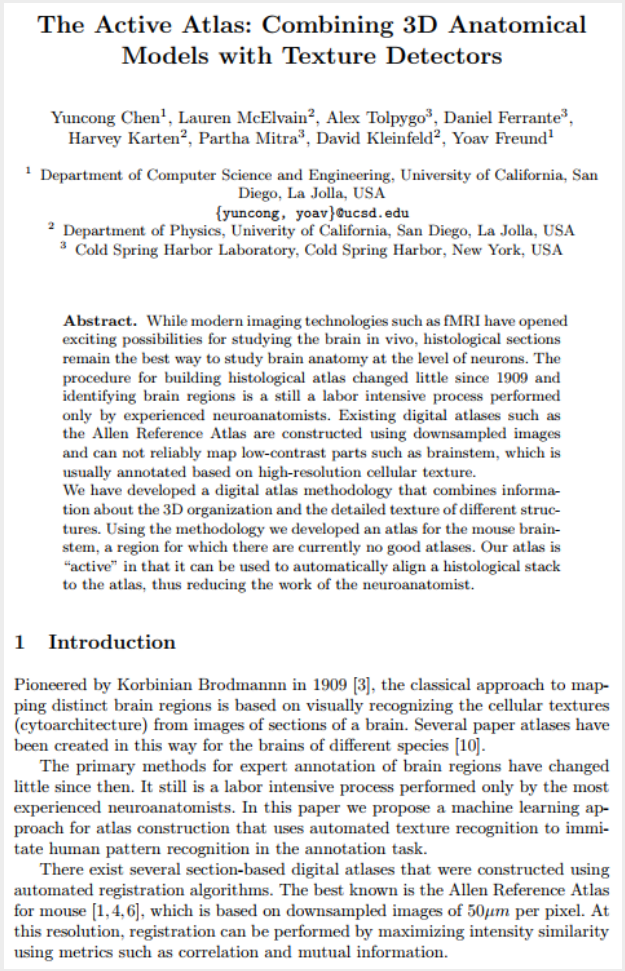
Positive or negative review?

- unbelievably disappointing
- Full of zany characters and richly applied satire, and some great plot twists
- this is the greatest screwball comedy ever filmed
- It was pathetic. The worst part about it was the boxing scenes.

How many stars?

- *Väga kiired tegijad. Ei uskunud, et nii kiiresti saab kaup kohale, kuid juba hommikul oli sõnum , et kaup kohal ja võib järgi minna see siis posti teel.* (*****)
- *Tellimus esitatud ja makstud, nädal hiljem siiski selgub, et antud toodet ei saa enam tellida. Arvan, et on mainimist väärt.* (**)
- *Parim hind, kiire teenindus ja vastutulelikud. Soovitan!* (*****)
- *Vaatamata sellele, et 99% kaubast tuuakse õigeaegselt kohale, peaksin ma ennast meelitatuna tundma, et sattusin sinna 1% sisse, kuid kuna tegemist pole lotovõiduga, siis eriti rahul ma sellega pole. Lubatud kuupäeval kaup ei saabunud kuigi väidetavalt anti kullerile üle juba kaks päeva tagasi. Soovitan neil siis kullerfirmat vahetada, kui viimased oma ülesannete kõrgusel ei ole.* (***)
-

Document categorization



The Active Atlas: Combining 3D Anatomical Models with Texture Detectors

Yuncong Chen[1], Lauren McElvain[2], Alex Tolpygo[3], Daniel Ferrante[3], Harvey Karten[2], Partha Mitra[3], David Kleinfeld[2], Yoav Freund[1]

[1] Department of Computer Science and Engineering, University of California, San Diego, La Jolla, USA
{yuncong, yoav}@ucsd.edu
[2] Department of Physics, Univerity of California, San Diego, La Jolla, USA
[3] Cold Spring Harbor Laboratory, Cold Spring Harbor, New York, USA

Abstract. While modern imaging technologies such as fMRI have opened exciting possibilities for studying the brain in vivo, histological sections remain the best way to study brain anatomy at the level of neurons. The procedure for building histological atlas changed little since 1909 and identifying brain regions is a still a labor intensive process performed only by experienced neuroanatomists. Existing digital atlases such as the Allen Reference Atlas are constructed using downsampled images and can not reliably map low-contrast parts such as brainstem, which is usually annotated based on high-resolution cellular texture.
We have developed a digital atlas methodology that combines information about the 3D organization and the detailed texture of different structures. Using the methodology we developed an atlas for the mouse brainstem, a region for which there are currently no good atlases. Our atlas is "active" in that it can be used to automatically align a histological stack to the atlas, thus reducing the work of the neuroanatomist.

1 Introduction

Pioneered by Korbinian Brodmannn in 1909 [3], the classical approach to mapping distinct brain regions is based on visually recognizing the cellular textures (cytoarchitecture) from images of sections of a brain. Several paper atlases have been created in this way for the brains of different species [10].

The primary methods for expert annotation of brain regions have changed little since then. It still is a labor intensive process performed only by the most experienced neuroanatomists. In this paper we propose a machine learning approach for atlas construction that uses automated texture recognition to immitate human pattern recognition in the annotation task.

There exist several section-based digital atlases that were constructed using automated registration algorithms. The best known is the Allen Reference Atlas for mouse [1, 4, 6], which is based on downsampled images of $50\mu m$ per pixel. At this resolution, registration can be performed by maximizing intensity similarity using metrics such as correlation and mutual information.

?

- Medicine
- Computer science
- Mathematics
- Physics
- Chemistry
- Social science
- Linguistics
- ...

Text Classification

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis
- …

Formal definition

- Input:
  - Document (text) d
  - Finite set of classes $C=\{c_1, c_2, \ldots, c_j\}$
- Output: $c \in C$

Hand-crafted rules

- Rules based on combinations of words or other features
  - spam: black-listed-address OR (“*dollars*” AND “*have been selected*”)
- Accuracy can be high
  - If rules carefully refined by expert
- But building and maintaining these rules is expensive

Classification using machine learning

- Input:
  - Document (text) d
  - Finite set of classes $C=\{c_1, c_2, \ldots, c_j\}$
  - A training set of m hand-labeled documents $(d_1,c_1),\ldots,(d_m,c_m)$
- Output: a classifier learned from the training corpus $f: d \rightarrow c$

Advantages of using machine learning

- Advantages:
  - No need to create and manage hand-crafted rules
  - Simple, relatively easy to achieve good accuracy
  - No deep knowledge of the domain or language required
- Disadvantages
  - A large training corpus needed
  - Need to keep the training corpus up-to date and retrain models
  - Difficult to guarantee very high accuracy
  - Difficult to fix a classifier to avoid certain mistakes (black box)

Machine learning methods

- Possible to use many different machine learning methods for text classification:
  - Naive Bayes
  - Logistic regression (other names: log-linear model, maximum entropy model)
  - Support vector machine
  - Decision trees, random forests
  - K-Nearest neighbor
  - (Deep) neural networks
  - Recurrent neural networks
  - Ensembles of the above

Naive Bayes method

- Simple (although formally naive) classifier
- Uses Bayes rule (probability theory)
- Naive, as it assumes that the words occuring in the document are independent (e.g., occurrence of the word “*economy*” should not have any effect on the probability of seeing “*money*”)
- In reality, the words are not independent
- However, the method works very well for text classification

Bag or words

Kaks nädalat tagasi tellisin sülearvuti. Ütlesid, et läheb nädal aega. Kui lubatud nädal oli möödas, saatsin kirja ja küsisin, kas on kohal, nad ütlesid, et läheb nädal veel. Siis saatsid kirja umbes tund hiljem, et siiski läheb nädal ja üks päev veel. Täna pidi asi kohal olema. [...] ja ma ei taha enam seda korrata, mitte kunagi enam ei osta neilt mitte midagi. Täielik pettumus. Hinnang: Lauspask

```
nädal: 3  veel: 2      kunagi: 1
läheb: 3  ütlesid: 2   pettumus: 1
ja: 3     ei: 2        tagasi: 1
...
```

Theory behind Naive Bayes

- Document *d* and class *c*

$$c_{max} = argmax_c P(c|d)$$

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

$$= argmax_c P(d|c) \times P(c)$$

Probability that document *d* *belongs to class c*

Bayes rule

Probability that class *c* generated such document

Prior probability of class *c*

Dropping the denominator: we don't care about the actual probability, rather than the class for which it is the largest. Therefore, we can just drop the denominator, as it is the same for all classes: the largest probability will still result in the largest pseudo-probability

Theory continues

- Document consists of words $w_1 w_2 .. w_n$

$$c_{max} = argmax_c P(d|c) \times P(c)$$
$$= argmax_c P(w_1 w_2 .. w_n|c) \times P(c)$$

The second term, *P(c)* is the prior probability of *c*: what's the probability that the next document that we see will belong to class *c*?

Theory, continues

- In reality, document consists of a sequence of words
- Naive Bayes makes two simplifications (not true in reality):
  - *Bag-of-words:* the order of words is not important
  - *Naive Bayes:* the words are independent
- This allows us to break the probability calculation into subcomponents

$$P(w_1, w_2, ..., w_n|c) = P(w_1|c) \times P(w_2|c) \times ... \times P(w_2|c)$$

$$= \prod_w P(w_i|c)$$

How to find word-given class probabilities?

- How to calculate $P(w_i|c)$?
- Maximum likelihood estimate:

$$\hat{P}(w_i|c_j) = \frac{Count(w_i, c_j)}{\sum_{w \in V} Count(w_i, c_j)}$$

  - Fraction of times word w_i appears among all words in documents of topic c_j
  - Create mega-document for class j by concatenating all docs in this topic
  - Use frequency of w_i in the mega-document

Problems with maximum likelihood

- What if we have seen no training documents with the word *fantastic* and classified in the topic *positive*?

$$P(fantastic|positive)=\frac{Count(fantastic,positive)}{\sum_{w\in V} Count(w_i,positive)}=\frac{0}{\sum_{w\in V} Count(w_i,positive)}=0$$

- Since the probability of a P(fantastic|positive)=0, the whole probability of the document being positive becomes zero

$$P(positive|w_{1,\dots},w_n)\propto P(positive)\prod_i P(w_i|positive)=0$$

Laplace (add-1) smoothing

- Add one to every word-category pair, to avoid zeros

$$\hat{P}(w_i|c_j) = \frac{Count(w_i, c_j)+1}{\sum_{w \in V} (Count(w_i, c_j)+1)} = \frac{Count(w_i, c_j)+1}{(\sum_{w \in V} Count(w_i, c_j))+|V|}$$

Number of words in the model's vocabulary

Summary: training Naive Bayes model

Given: documents d_i, each belonging to a class c_i

- Extract vocabulary V (e.g., all words)
- Calculate $P(c_j)$ terms: for each c_j do:

$$P(c_j) = \frac{N_{d \in c_j}}{N_d}$$

- Calculate $P(w_k|c\_j)$ terms

  $n \leftarrow$ # of tokens in the whole corpus

  - For each word w_k in vocabulary:

    $n_k \leftarrow$ # of occurrences of w_k in the whole corpus

$$P(w_k|c_j) = \frac{n_k + \alpha}{n + \alpha|Vocabulary|}$$

Example

- Training corpus
  - “jalgpall eelarve värav” - S (sport)
  - “korvpall võit” - S
  - “jalgpall võit raha” - S
  - “eelarve defitsiit raha” - M (majandus)
- Test document
  - “eelarve raha võit” - ?

$$P(c_j) = \frac{Count(c_j)}{N_{doc}} \qquad P(w_i|c_j) = \frac{Count(w_i, c_j) + 1}{\sum_w Count(w_i, c_j) + |V|}$$

$$\begin{aligned} c_{max} &= argmax_c P(d|c) \times P(c) \\ &= argmax_c P(w_1 w_2 .. w_n|c) \times P(c) \end{aligned}$$

$$\begin{aligned} P(w_1, w_2, ..., w_n|c) &= P(w_1|c) \times P(w_2|c) \times ... \times P(w_2|c) \\ &= \prod_w P(w_i|c) \end{aligned}$$

Solution

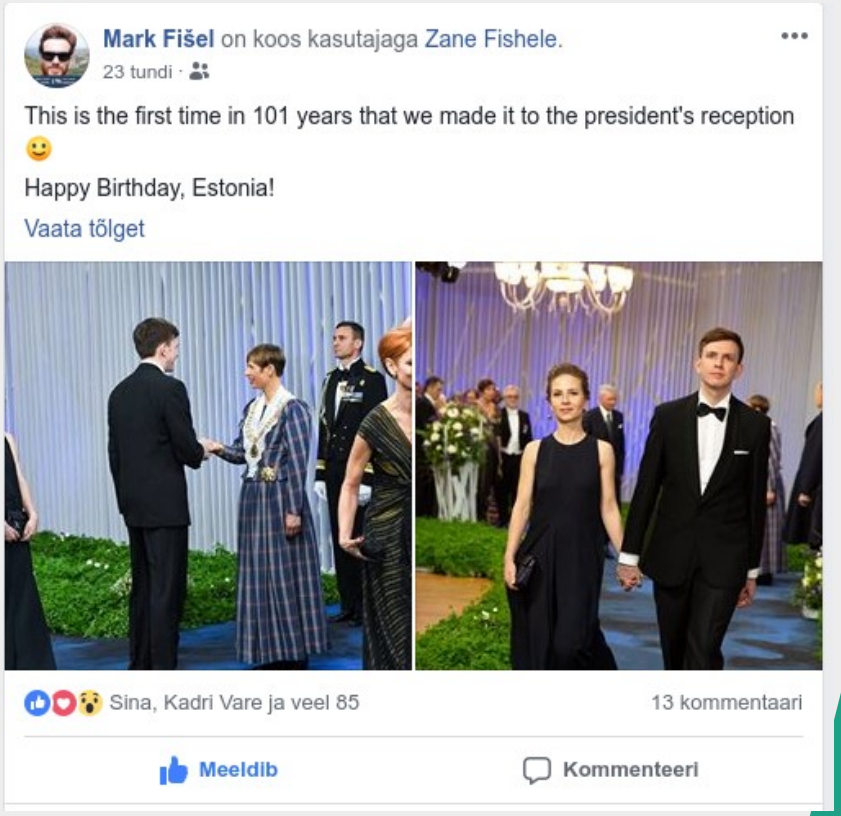
$$
\begin{aligned}
P(c = S) &= 3/4 = 0.75 \\
P(c = M) &= 1/4 = 0.25 \\
P(eelarve|S) &= (1+1)/(8+7) = 2/15 \\
P(raha|S) &= (1+1)/(8+7) = 2/15 \\
P(voit|S) &= (2+1)/(8+7) = 3/15 \\
P(eelarve|S) &= (1+1)/(3+7) = 0.2 \\
P(raha|S) &= (1+1)/(3+7) = 0.2 \\
P(voit|S) &= (0+1)/(3+7) = 0.1 \\
P(S|D_{test}) &\propto 0.75 * 2/15 * 2/15 * 3/15 = 0.00267 \\
P(M|D_{test}) &\propto 0.25 * 0.2 * 0.2 * 0.1 = 0.001
\end{aligned}
$$

Naive Bayes and features

- Using the bag-of-words as a document representation is **feature extraction**
- We don’t have to use word occurrence features
  - e.g., for Estonian, using **lemmas** (*spordi* → *sport*) might be useful
- But features can be “anything” that can be derived from the document and that could be helpful for classification
- Other features possibly used by a spam filter:
  - Is *Subject* written in ALL CAPITAL LETTERS?
  - Does the e-mail originate from a known spam host?
  - Is the sender in recipients address book?
  - Does the e-mail contain a link to a known malicious web site?

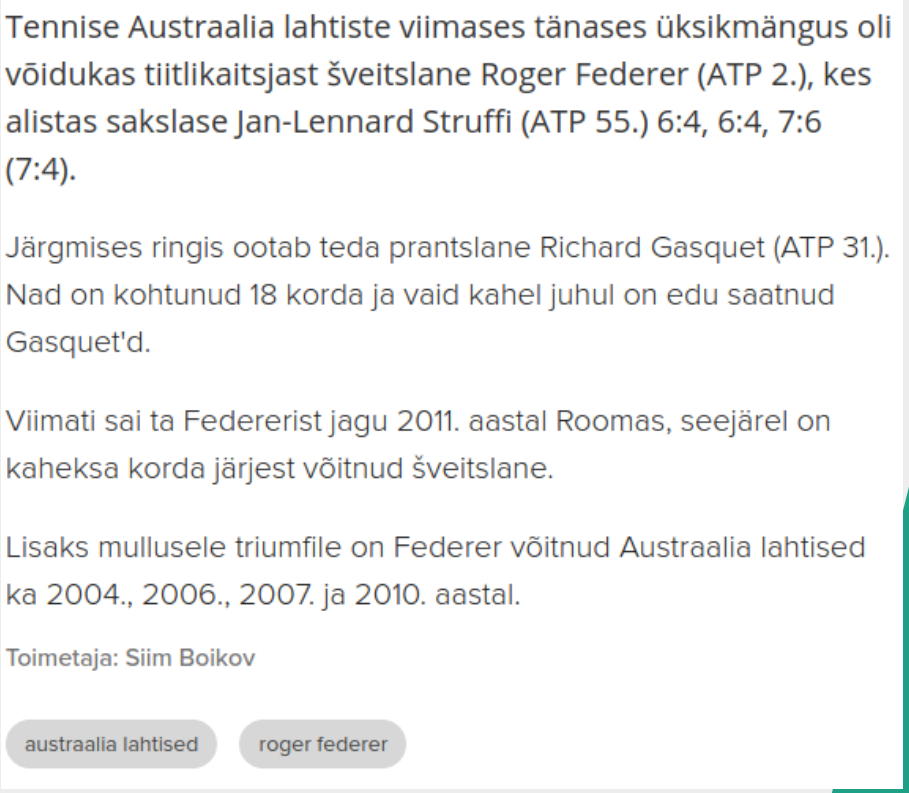
Feature extraction exercise

- You decide to train a classifier to predict whether you will '**like**' a Facebook post (by your friends)
- What features will you try?



Multi-class classification

- What if a document can belong to many categories?
- Solution: train a one-against-all classifier for all classes
- Given a test doc:
  - Evaluate it for membership in each class using each classifier



Tennise Austraalia lahtiste viimases tänases üksikmängus oli võidukas tiitlikaitsjast šveitslane Roger Federer (ATP 2.), kes alistas sakslase Jan-Lennard Struffi (ATP 55.) 6:4, 6:4, 7:6 (7:4).

Järgmises ringis ootab teda prantslane Richard Gasquet (ATP 31.). Nad on kohtunud 18 korda ja vaid kahel juhul on edu saatnud Gasquet'd.

Viimati sai ta Federerist jagu 2011. aastal Roomas, seejärel on kaheksa korda järjest võitnud šveitslane.

Lisaks mullusele triumfile on Federer võitnud Austraalia lahtised ka 2004., 2006., 2007. ja 2010. aastal.

Toimetaja: Siim Boikov

austraalia lahtised roger federer

Quality metrics for classification

- How good is our classifier?
- Test on data that was not used for training
- Compare predicted classes to the real (human-annotated) classes
- **Accuracy**: how many documents in the test set were classified correctly?

$$A = \frac{N_{correct}}{N_{total}}$$

Problem with accuracy

- Accuracy metric doesn’t work well for inbalanced classes
- For example, if test corpus contains
  - 990 documents about economy
  - 10 documents about sport
- If the classifer assigns “economy” to all documents, the accuracy is 990/1000 = 99%
- Often, error rate (ratio of errors in all data) is used instead of accuracy: 10/1000= 1%

Precision and recall

Precision and recall are metrics for binary classifiers, where we are mostly interested in finding items of a certain class (e.g., Facebook posts that we like)

- Precision: % of selected items that are correct
- Recall: % of correct items that are selected

	correct	not correct
selected	**tp**	**fp**
not selected	**fn**	**tn**

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

F_1-measure

- Precision and recall give different views to the classifiers performance:
  - Try to increase precision if you are really bothered about false alarms (and you can live with some items being missed)
  - Try to increase recall if you are care about really finding most positive items, even when many of the found items are negative (e.g., cancer test)
- Combined metrics: F_1 measure

$$F_1 = \frac{2 * Precison * Recall}{Precision + Recall}$$

Example

- Example: 100 doping tests, 10 are really positive
- Automatic doping test analyzer classified 7 positive samples as positive, 5 negative samples as positive, and 3 positive tests as negative.
- TP=7 FP=5 TN=85 FN=3
- *Accuracy*: 92/100 = 92%
- *Precision*: 7/(7+5) =58%
- *Recall*: 7/(7+3) =70%
- *F1-measure*: 2*0.58*0.7/(0.58+0.7)=0.63

Confusion matrix

- Nice way to analyze the performance of the classifier
  - Usable if the number of classes is not very big

Predicted \ Truth	Asphalt	Concrete	Grass	Tree	Building	Total
Asphalt	2385	4	0	1	4	2394
Concrete	0	332	0	0	1	333
Grass	0	1	908	8	0	917
Tree	0	0	0	1084	9	1093
Building	12	0	0	6	2053	2071
Total	2397	337	908	1099	2067	6808

Micro-average and macro-average

- If we have more than one class, how do we combine multiple performance measures into one quantity?
- **Macro-averaging**: Compute performance for each class, then average.
- **Micro-averaging**: Collect decisions for all classes, compute contingency table, evaluate.

Micro-average and macro-average: example

Class 1

	Truth: yes	Truth: no
Classifier: yes	10	10
Classifier: no	10	970

Class 2

	Truth: yes	Truth: no
Classifier: yes	90	10
Classifier: no	10	890

Micro Ave. Table

	Truth: yes	Truth: no
Classifier: yes	100	20
Classifier: no	20	1860

- Macro-averaged precision: (0.5 + 0.9)/2 = 0.7
- Micro-averaged precision: 100/120 = .83
- Micro-averaged score is dominated by score on common classes

Training/development/test

- When training a classifier, split the data into training, development (evaluation) and test set
- Typically using a ratio 80% 10% 10%
- Training data if used for training the model
- Development data is used for optimizing the architecture and meta-parameters of the model
  - Which features to use?
  - What kind of smoothing to use?
- Test data is used for estimating the true performance of the classifier
  - Using development data for this is too optimistic, as the model's hyperparameters have been already optimized on it

Questions?