Sentiment Analysis in Ruby

Mateusz Drożdżyński
@matid

UNIVERSITY OF CAMBRIDGE

Object Reload

What is sentiment analysis?

Why is it important?

Examples

- Automated film reviews: www.fflick.com
- Social Data Mining: www.datasift.net
- Customer feedback analysis

Problems?

- Corpus availability
- Crawling and language classification
- Data cleaning
- Multilingual sentiment classification

Crawling

- Where do we get training data from?
- How do we annotate our data set?
- Amazon Mechanical Turk?
- Can we be smarter here?

Data cleaning

- Remove emoticons
- Remove duplicate tweets (inc. retweets)
- Remove Twitter-specific keywords (@names, etc.). Or not?
- Convert into features

Classification

- Baseline
- Naive Bayes
- Support Vector Machines

Baseline

- Bag-of-words dictionary count
- Where do we get dictionaries from?
- Google Translate?
    - Tweets or the dictionary?

Naive Bayes

with Ankusa

https://github.com/livingsocial/ankusa

Assumptions

- Independence of features
- Random distribution of words
- Two exclusive classes

$$p(D|C) = \frac{p(D \cap C)}{p(C)}$$

$$p(C|D) = \frac{p(D \cap C)}{p(D)}$$

$$p(C|D) = \frac{p(C)}{p(D)}\, p(D|C)$$

$$p(D|S) = \prod_i p(w_i|S)$$

$$p(D|\neg S) = \prod_i p(w_i|\neg S)$$

$$p(S|D) = \frac{p(S)}{p(D)} \prod_i p(w_i|S)$$

$$p(\neg S|D) = \frac{p(\neg S)}{p(D)} \prod_i p(w_i|\neg S)$$

$$\ln \frac{p(S|D)}{p(\neg S|D)} = \ln \frac{p(S)}{p(\neg S)} + \sum_i \ln \frac{p(w_i|S)}{p(w_i|\neg S)}$$

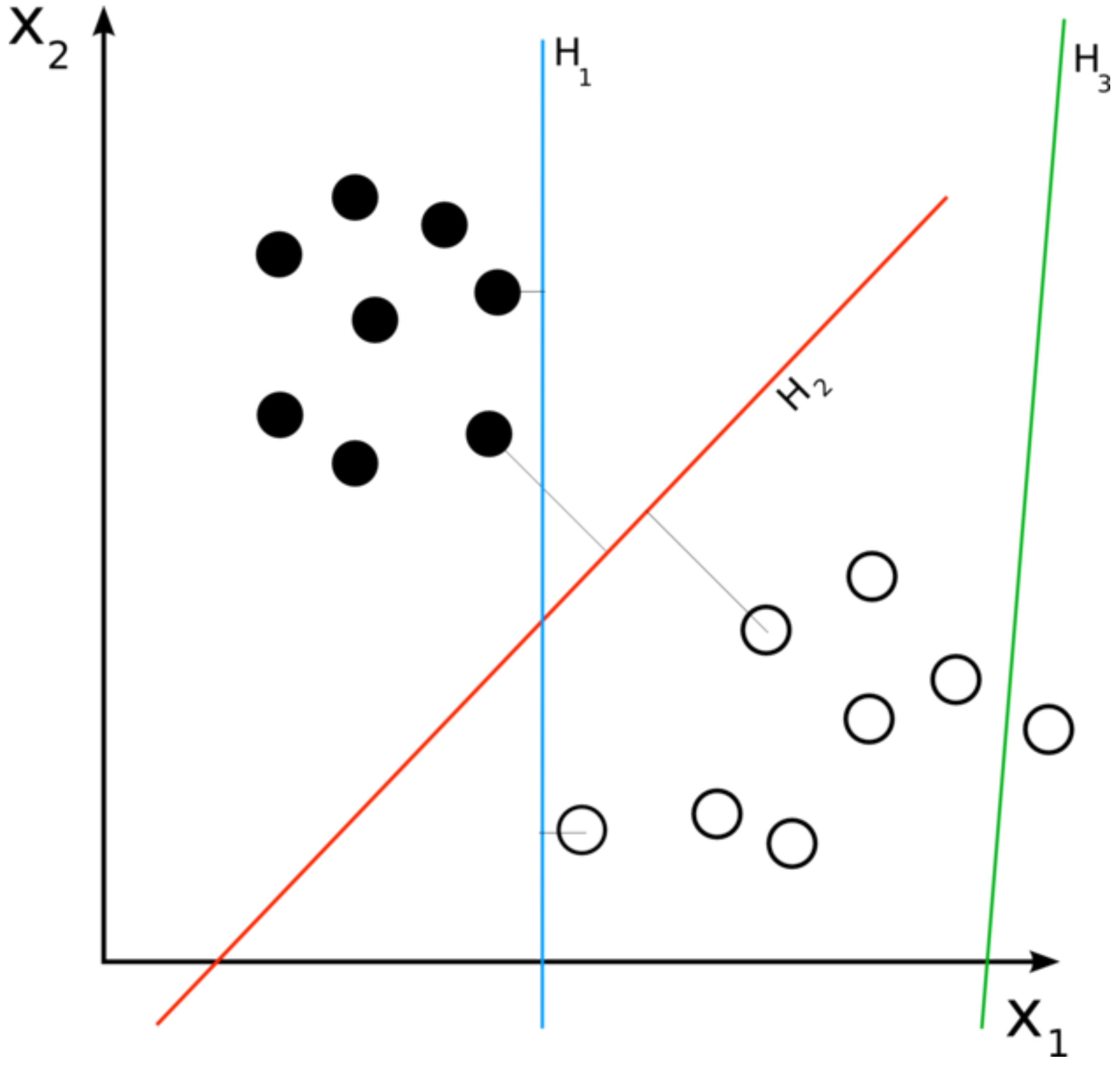
```ruby
require "ankusa"
require "ankusa/memory_storage"

storage = Ankusa::MemoryStorage.new
classifier = Ankusa::NaiveBayesClassifier.new(storage)

training.each do |tweet|
  classifier.train tweet.sentiment, tweet.to_s
end

sentiment = classifier.classify tweet.to_s
```

Support Vector Machines

with Eluka

https://github.com/arrac/eluka

```ruby
class Tweet
  def features
    # Maps "This is a tweet" to:
    # {"this" => 1, "is" => 1, "a" => 1, "tweet" => 1}
    Hash[*self.to_s.downcase.split.map { |feature|
      [feature, 1]
    }.flatten]
  end
end
```


X_2
H_1
H_3
H_2
X_1

```ruby
classifier = Eluka::Model.new

training.each do |tweet|
  classifier.add(tweet.features, tweet.sentiment)
end

classifier.build

sentiment = classifier.classify tweet.features
```

How accurate can we get?

- 80% is realistic, anything above is hard
- Simple models work best
  - 25% improvement with Naive Bayes over Baseline
  - 1% improvement with SVM over Naive Bayes

Have you ever wondered…

- … what time of the day are you most happy?
- … if Americans are happier than Brits?
- … or whether that last gadget you bought actually made you a happier person?

Thank you!