# Machine Learning for Natural Language Processing

## Seminar Information Extraktion
## Wednesday, 6th November 2013

Robin Tibor Schirrmeister

UNI FREIBURG

# Outline

- Informal definition with example
- Four machine learning algorithms
  - Motivation and main idea
  - Example
  - Training and classification
  - Usage and assumptions
- Types of machine learning
- Improvement of machine learning systems

# Informal Definition

- Machine Learning means a program gets better by automatically learning from data
  - Let's see what this means by looking at an example

# Named Entity Recognition Example

UNI FREIBURG

- How to recognize mentions of persons in a text?
  - Angela Merkel made a decision.
  - Tiger Woods played golf.
  - Tiger ran into the woods.

  Could use handwritten rules, might take a lot of time to define all rules and their interactions… So how to *learn* rules and their interactions?

# Person Name Features

UNI FREIBURG

- Create features for your ML algorithm
  - For a machine to learn if a word is a name, you have to define attributes or features that might indicate a name
  - Word itself
  - Word is a known noun like hammer, boat, tiger
  - Word is capitalized or not
  - Part of speech (verb, noun etc.)
  - …

    Human has to design these features!

# Person Name Features

UNI FREIBURG

- ML algorithm learns typical features
  - What values of these features are typical for a person name?
  - Michael, Ahmed, Ina
  - Word is not a known noun
  - Word is capitalized
  - Proper noun

  Machine should learn these typical values automatically!

# Person Name Training Set

UNI FREIBURG

- Create a training set
  - Contains words that are person names and words that are not person names

| Word | Known Noun | Capital | Parts of Speech | **Part of Name** |
|---|---|---|---|---|
| Tiger | 1 | 1 | Proper Noun | **Yes** |
| Woods | 1 | 1 | Proper Noun | **Yes** |
| played | 0 | 0 | Verb | **No** |
| … | | | | |
| woods | 1 | 0 | Plural Noun | **no** |

# Naive Bayes Training

- For each feature value count how frequent it occurs in person names and other words.
  - $\mathrm{P}(capitalized|name) = \frac{\#capitalized\ person\ names}{\#all\ person\ names}$

This tells you how likely a random person name is capitalized. How could person name not be capitalized? Think noisy text like tweets..

**Heartless** @KingWedge 28 Okt

That **tiger woods** smile is hilarious □□□□

pic.twitter.com/UcWDsGOCY3

Foto anzeigen

# Naive Bayes Training

- Count total frequency of classes
  - Count how frequent person names are in general: How many words are part of person names and how many are not
  - $P(Name) = \frac{\#Person\ Name\ Words}{\#All\ Words}$
  - Do whole training the same way for words that are not person names

# Naive Bayes Classification Example

UNI FREIBURG

- Let's try to classify Tiger in *Tiger Woods played golf*.
  - Assume 5% of words are parts of person names

| | Person Names | Others |
|---|---|---|
| Tiger | 0.01 | 0.01 |
| Capital | 0.85 | 0.05 |
| Proper Noun | 0.8 | 0.1 |
| Known Noun | 0.1 | 0.2 |

$$\mathrm{P}(\mathrm{Tiger}, \mathrm{Name}) = 0.01 \cdot 0.85 \cdot 0.8 \cdot 0.1 \cdot 0.05 = 0.000034$$

$$\mathrm{P}(\mathrm{Tiger}, \mathrm{Other}) = 0.01 \cdot 0.05 \cdot 0.1 \cdot 0.2 \cdot 0.95 = 0.0000095$$

# Naive Bayes Classification

- Classify a word by multiplying feature value probabilities
  - $P(Word, Name) = P(x_1|Name) \cdot P(x_2|Name) \cdots P(Name)$
  - $P(Word, Other) = P(x_1|Other) \cdot P(x_2|Other) \cdots P(Other)$
  - ($x_i$ value of i-th feature)
  - Higher Score wins! ☺

# Naive Bayes Overview

UNI FREIBURG

## Training

1. For all classes
    1. For all features and all possible feature values
        1. Compute $P(feature\ value\ |class)$ (e.g. chance of word is capitalized if it's a person name)
    2. Compute total probability $P(class)$

## Classification

1. For all classes compute classscore as:
    1. $P(class) \cdot \prod P(feature\ value\ |class)$
2. Data point is classified by class with highest score

# Naive Bayes Assumptions

UNI FREIBURG

- Probability of one feature is independent of another feature when we know the class
  - When we know word is part of person name, probability of capitalization is independent of probability that word is a known noun
    This is not completely true
    If the word is tiger, we already know it is a known noun
  - That's why it's called Naive Bayes
    Naive Bayes often classifies well even if assumption is violated!

# Evaluation

UNI FREIBURG

- Evaluate performance on test set
  - Correct classification rate on test set of words that were not used in training
    Correct classification rate not necessarily the most informative…
  - If we classify every word as Others and only have 5% person name words, we get a 95% classification rate!
  - More informative measures exist
  - Words correctly and incorrectly classified as person name(true positives, false positives)
  - and others (true negatives, false negatives)

# Evaluation Metric

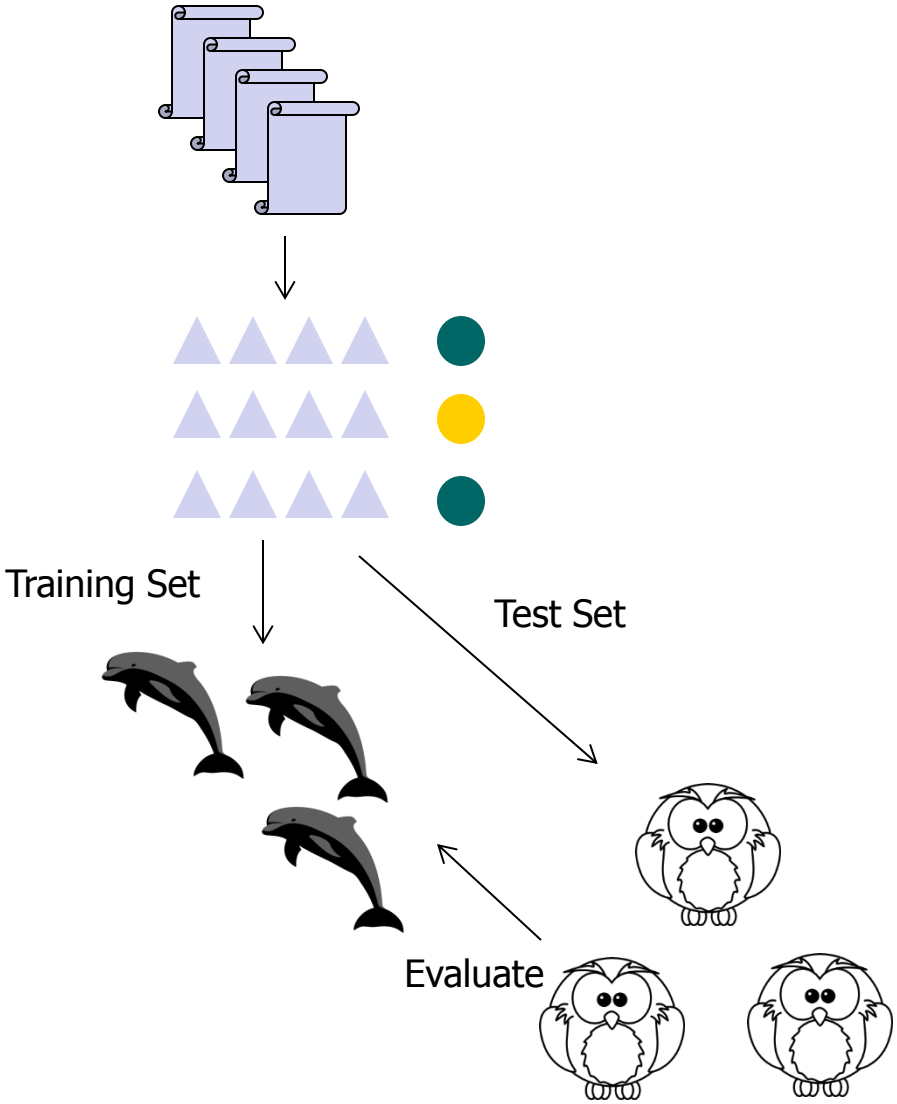
UNI FREIBURG

- Best performance is in part subjective
  - Recall: Maybe want to capture all persons occuring in text even at cost of some non-persons
    E.g. if you want to capture all persons mentioned in connection with a crime
  - Precision: Only want to capture words that are definitely persons
    E.g. if you want to build a reliable list of talked about persons

# Interpretation

- Feature value probabilities show feature contribution to classification
  - Comparing trained values $P(feature\ value|class_1)$ and $P(feature\ value|class_2)$ tells you if this feature value is more likely for class1 or class2
  - $P(capital|Name) = 0.85 > 0.05 = P(capital|Others)$ means capitalized words are more likely to be parts of person names
  - You can look at each feature independently

# Machine Learning System Overview




**Data Acquisition**

Important to get data similar to data you will classify!

**Data Representation as Features**

Important to have the information in the features that allows you to classify

**Machine Learning Algorithm Training**

Important to use algorithm whose assumptions fit the data well enough

**Performance Evaluation**

Important to know what measure of quality you are interested in

# Logistic Regression Motivation

UNI FREIBURG

- Correlated features might disturb our classification
  - Tiger is always a known noun
  - Both features (known noun, word tiger) indicate that it's not a name
  - Since Naive Bayes ignores that the word tiger already determines that it is a known noun, it will underestimate chance of tiger being a name

  Modelling relation from combination of feature values to class more directly might help?

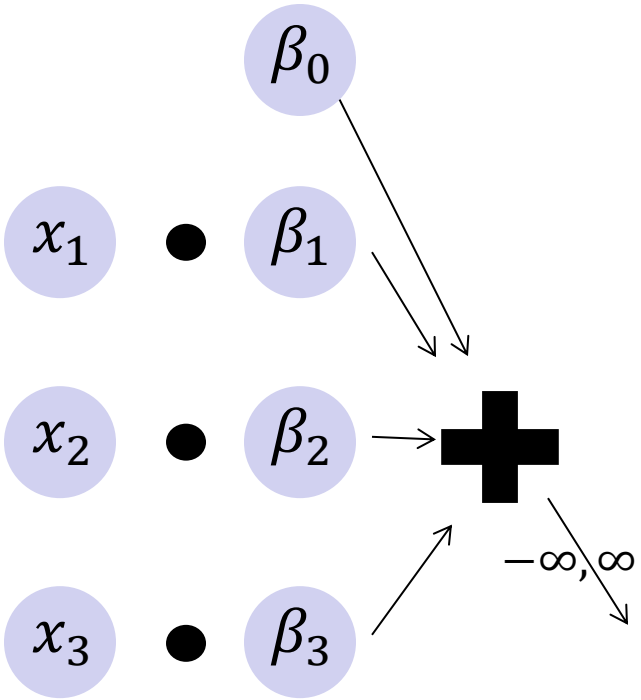
# Logistic Regression Idea

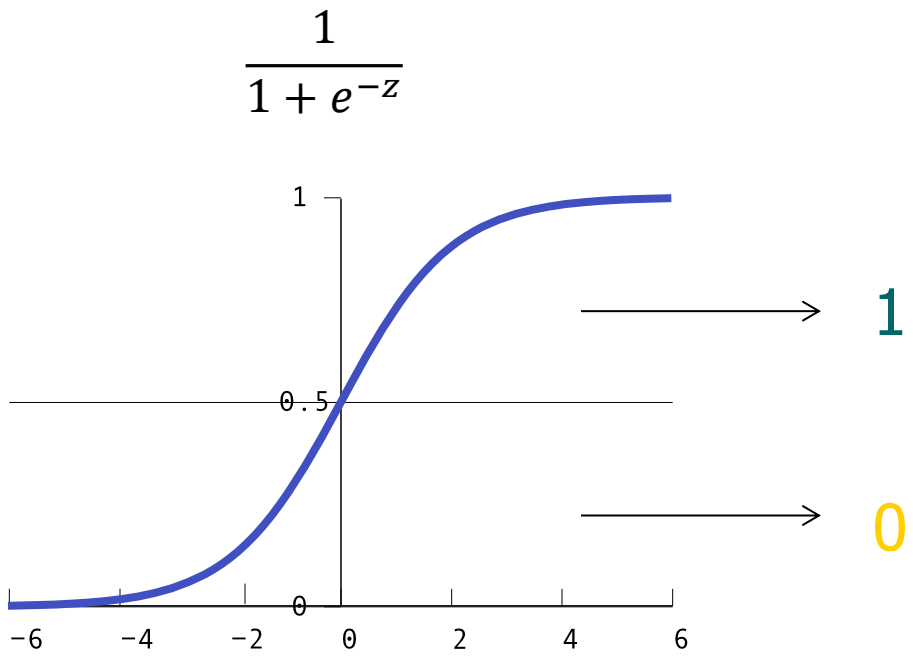
UNI FREIBURG

- Idea: Learn weights together, not separately
  - Make all features numerical
  - Instead of Part Of Speech = verb, noun etc.:
    - One feature for verb which is 0 or 1
    - One feature for noun which is 0 or 1 etc.
  - Then you can make sums of these feature values * weights and learn the weights
  - Sum should be very high for Person Names and very small for non-person names
  - Weights will indicate how strongly a feature value indicates a person name
    Correlated features can get appropriate, not too high weights, because they are learned together!

# Logistic Regression

- Estimate probability for class
  - Use sum of linear function chained to a link function
  - Link function rises sharply around class boundary



# Example

- Let's look at *Tiger Woods played golf.*
  - Assume we learned these weights:
    $\beta_{KnownNoun} = -1$ $\beta_{Tiger} = 3$ $\beta_{ProperNoun} = -0.5$ $\beta_0 = -2$
  - $1 \cdot (-1) + 1 \cdot 3 + 1 \cdot (-0.5) - 2 = 0.5$
  - $\frac{1}{1+e^{-0.5}} \approx 0.62$
  - > 0.5 => looks more like a name☺
  - 0.62 can be interpreted as 62% probability that it's a name

# Training

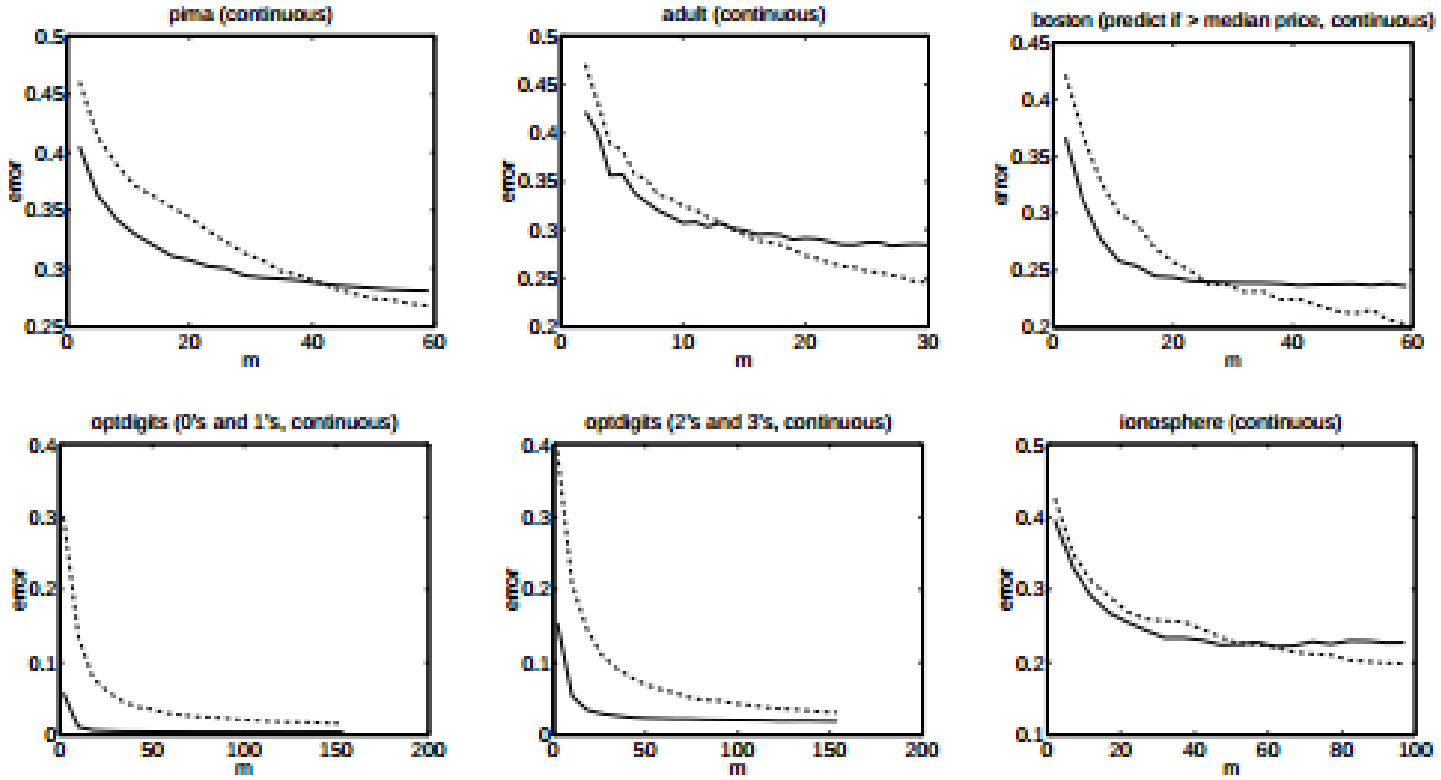
- Solve equation system iteratively
  - From our training examples we get a system of equations. Using $logistic(z) = \frac{1}{1+e^{-z}}$
  - $logistic(\beta_0 + x_{11}\beta_1 + x_{12}\beta_{2+\ldots}) = 0$
  - $logistic(\beta_0 + x_{21}\beta_1 + x_{22}\beta_{2+\ldots}) = 1$
  - …
  - Best Fit cannot be computed directly, is solved by iterative procedures
    Not our topic here ☺
  - Just have to know that weights are estimated together!

# Interpretation

- Higher weights mean probability of yes(1) is increased by the corresponding feature
- Weights have to be interpreted together (no conditional independence assumed)
  - If we have the feature word preceded by Dr. and another feature word preceded by Prof. Dr.
  - But in all our texts there are only Prof. Dr. Both features will always have the same value!
  - Then $\beta_{Dr} = 5$ and $\beta_{ProfDr} = -2$ leads to same predictions as $\beta_{Dr} = -1$ and $\beta_{ProfDr} = 4$
- Also, weights are affected by how big and how small feature value range is

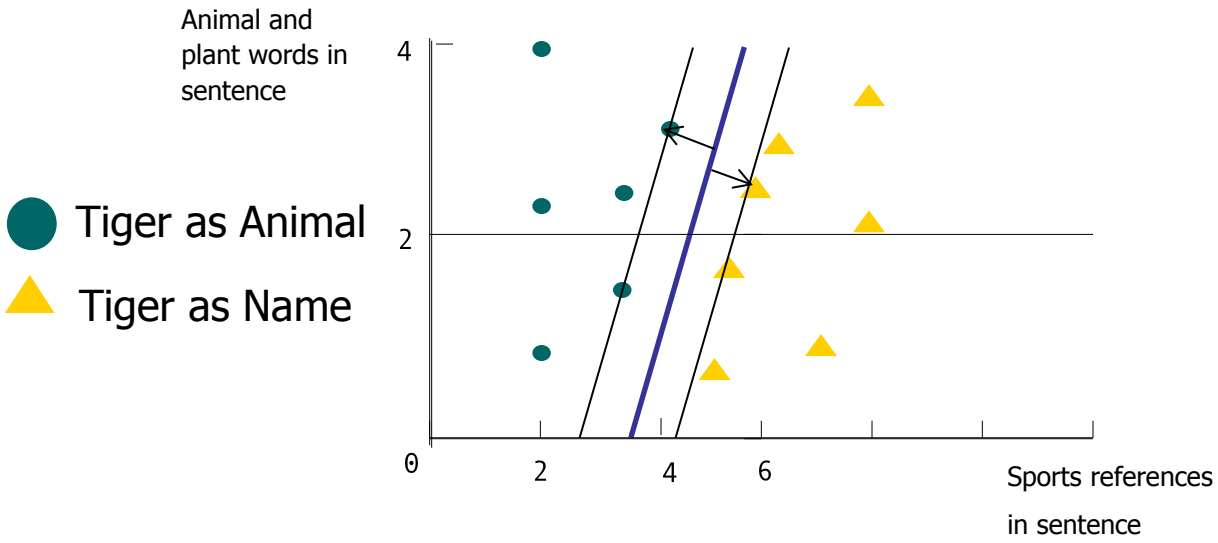
# Naive Bayes vs Logistic Regression

- Logistic Regression better with more data, Naive Bayes better with less data
  - Naive Bayes reaches its optimum faster
  - Logistic Regression has better optimal classification



# Support Vector Machines

- Support Vector Machine tries to separate true and false examples by a big boundary



# Training

UNI FREIBURG

- Soft Margin for inseparable data
  - In practice examples usually not perfectly separable
  - Soft-Margin to allow for wrong classifications
  - Parameter to adjust tradeoff between:
  - Datapoints should be on the correct side of the boundary and outside of the margin
  - Margin should be big
  - Specialized optimization algorithms for SVMs exist

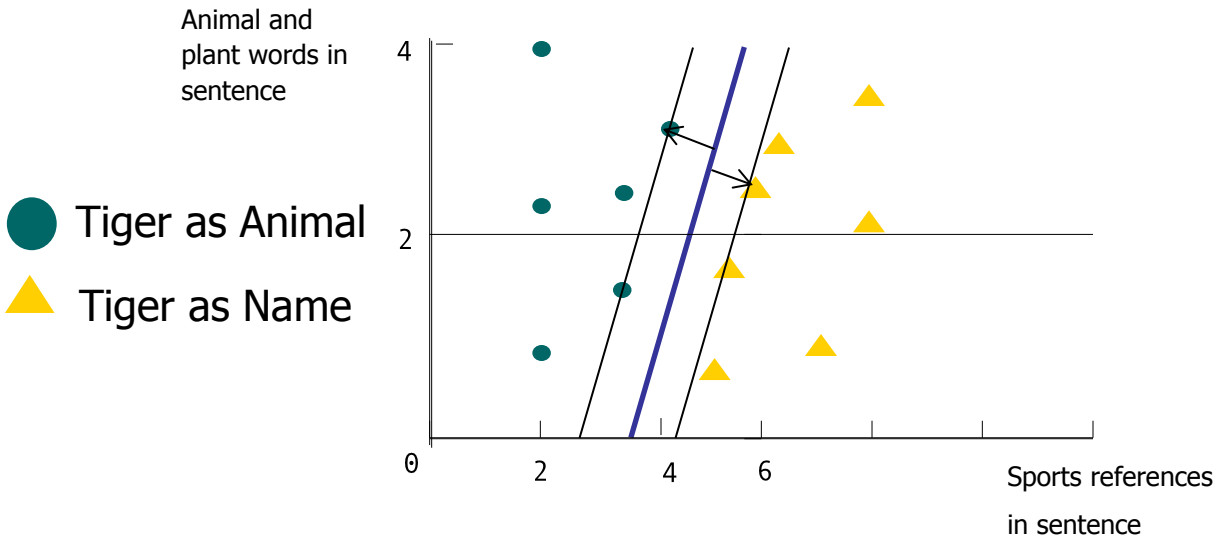
# Usage

UNI FREIBURG

- SVM used often very successfully, very popular
  - Very robust, and fast, only support vectors are needed for the classification
  - => robust against missing data

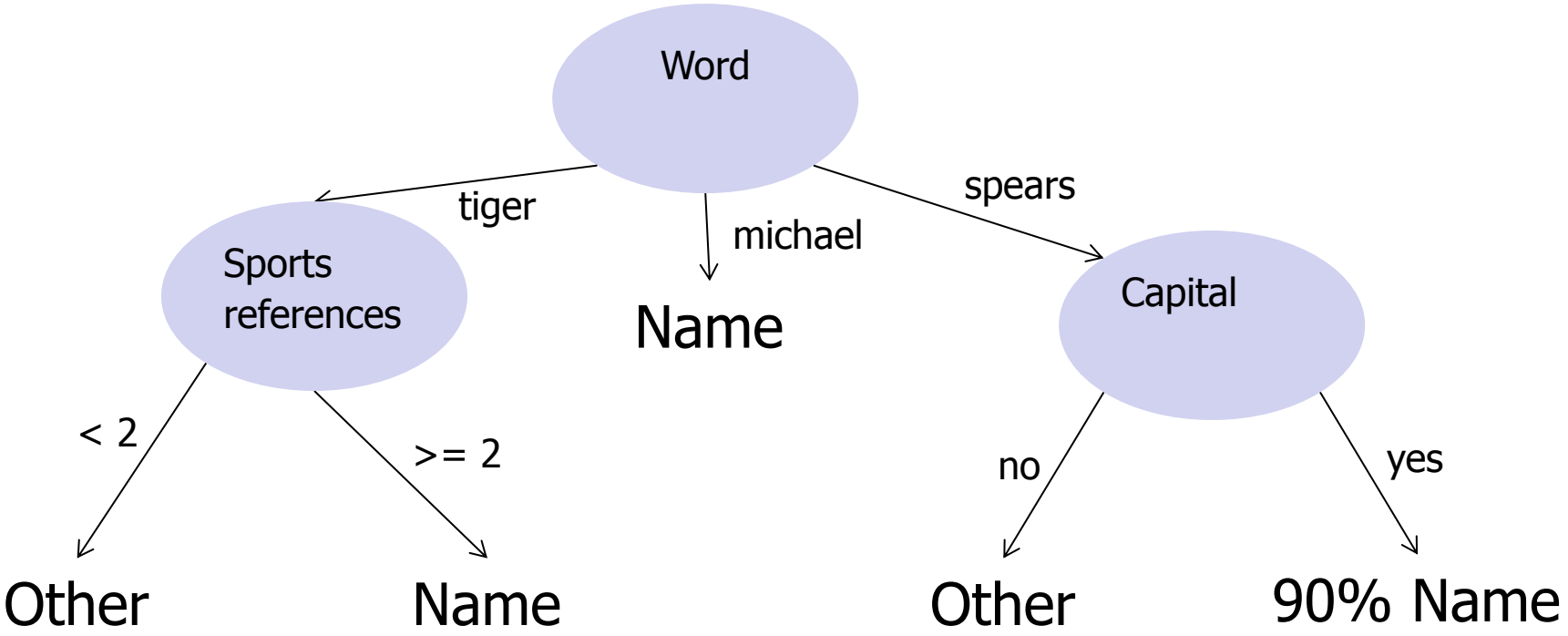
# Interpretation

- Hyperplane can tell you which features matter more for classification



# Decision Trees

- Recursively separate data by features that split well into different classes



# Decision Trees Training

UNI FREIBURG

1. Start with all training examples at root node
2. Pick feature to split training examples into next subtrees
    1. Pick a feature, so that training examples in one subtree are mostly from one class
3. Recursively repeat procedure on subtrees
4. Finished, when subtree only contains examples from one class (convert to leaf, e.g. name)
    - Or most examples from one class (using some predefined threshold)

# Decision Trees Usage

UNI FREIBURG

- Useful especially if you assume some feature interactions
- Also useful for some non-linear relationships of features to classification
  - Word shorter than 3 characters: Unlikely to be a name
  - Word between 3 and 10 charachters: Might be a name
  - Word longer than 10 characters: Unlikely to be a name
- Often many trees are used together as forests (ensemble methods)
- Very clear to interpret learning of single tree
  - For forests, methods exist to determine feature importance

# Conditional Random Fields Motivation

UNI FREIBURG

- Compare
  - Tiger Woods played golf.
  - Tiger ran into the woods.

  We still want to know for both occurences of Tiger if it is a name. There is one helpful characteristic of these sentences we did not use. Can you guess what it is? ☺

  …
- Tiger and Woods could both be names and two parts of a name standing together are more likely than one part of a name by itself.

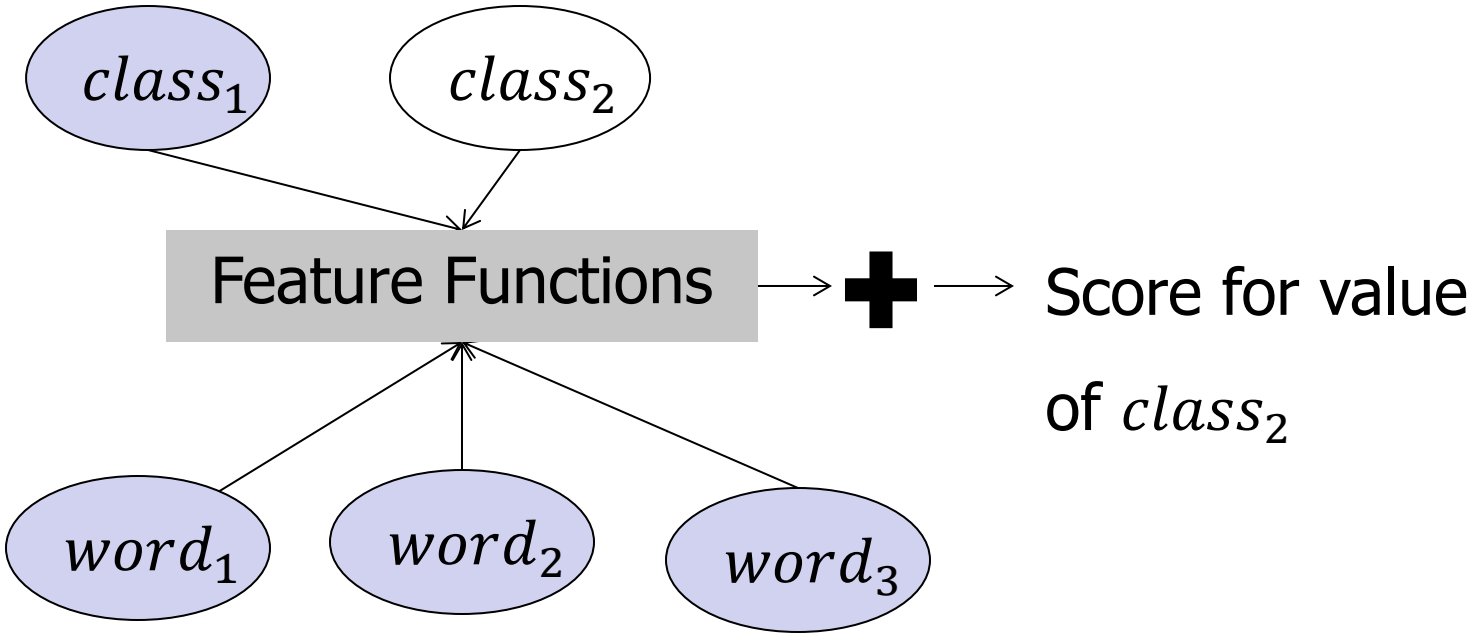
# Conditional Random Fields Idea

UNI FREIBURG

- To classify a datapoint, use the surrounding classifications and datapoints
  - E.g. use the fact that names often stand together
  - (Sequential) input -> sequential output
  - We only use neighbouring classifications (Linear-Chain-CRFs)

# Conditional Random Fields Sketch

- Use feature functions to determine probability of class sequence



# Feature Functions

UNI FREIBURG

- Feature functions for linear-chain CRFs can use
  - the complete sentence
  - Current position (word) in sentence
  - the class of the output node before
  - the class of the current output node
- Return real value
- Each feature function is multiplied by a weight, that needs to be learned

# Examples

- *Washington Post wrote about Mark Post.*
  - City + *Post* usually is a newspaper, First Name + *Post* more likely to be a name
- *Dr. Woods met his client.*
  - Salutation (Mr./Dr. etc) usually followed by name
- Feature function does not have to use all inputs
  - E.g. feature function can just look at is word capitalized, what is part of speech of the next word etc.

# Usage

1. Define feature functions
2. Learn weights for feature functions
3. Classify
    1. Find sequence that maximizes sum of weights * feature functions
    2. Can be done in polynomial time with dynamic programming

- Used a lot for NLP tasks like named entity recognition, parts of speech in noisy text

# Algorithm Characteristics

UNI FREIBURG

- Assumptions on data
  - Linear relationship to output / non-linear
- Interpretability of learning
  - Meaning of feature weights etc.
- Computational time and space
- Type of input and output
  - Categorical, numerical
  - Single datapoints, sequences

# Supervised/Unsupervised

UNI FREIBURG

- Unsupervised algorithms for learning without known classes
  - These algorithms were supervised algorithms
  - We had preclassified training data
  - Sometimes we might need unsupervised algorithms
  - Right now, what kind of topics are covered in news articles?
  - Often work by clustering
  - Similar data gets assigned the same class
    E.g. text with similar words may refer to the same news topic

# Semi-Supervised

UNI FREIBURG

- Create more training data automatically
  - Big amount of training data important for good classification
  - Creating training data by hand time-demanding
  - Your unsupervised algorithm already gives you datapoints with classes
  - Other simple rules can also give you training data
    E.g. Dr. is almost always followed by a name
  - New data you classified with high confidence can also be used as training data

# Improvement ML System

UNI FREIBURG

- It is important to know how and if you can improve your machine learning system
  - Maybe in your overall (NLP) system there are bigger sources of error than your ML system
  - Maybe from current data it is impossible to learn more than your algorithm does
- You can try to:
  - Get more data
  - Use different features
    Also maybe preprocess more
  - Use different algorithms
    Also different combinations

# Machine Learning in NLP

UNI FREIBURG

- Very widely used
  - Make it easier to create systems that deal with new/noisy text
    For example tweets, free text on medical records
  - Can be easier to specify features that maybe important and learn classification automatically than write all rules by hand

# Summary

- Typical machine learning consists of data acquisition, feature design, algorithm training and performance evaluation
- Many algorithms exist with different assumptions on the data
- Important to know whether your assumptions match your data
- Important to know what the goal of your overall system is

# Left out

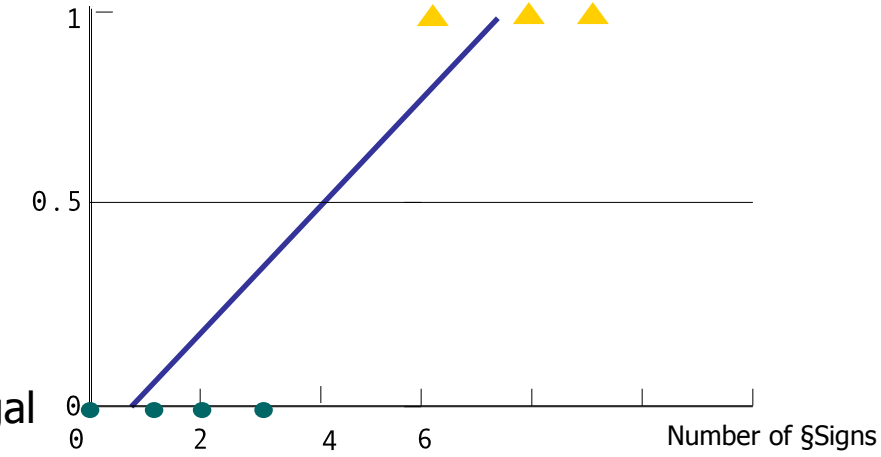
- Explaining log odds ratio Log reg

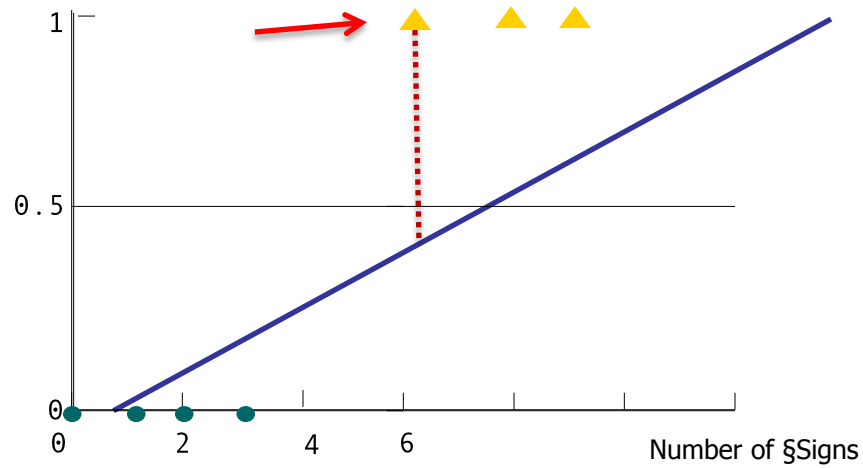
# Comparison to linear regression

- Need for link function like $\frac{1}{1+e^{-z}}$

Legal

Non-Legal



# Machine Learning System Overview

UNI FREIBURG

- Make Wonderful picture out of this ☺ (Still TODO)
- Data Acquisition (Get Spam and Non-Spam-Mails)
  - Important to get training data similar to data you will classify!
  - E.g. people from different countries might get different types of spam mails…
- Data representation as features
  - Feature Design – Create features from your data
  - (Feature Selection) – Keep only features helpful for classification
- Machine Learning Algorithm
- Performance Evaluation