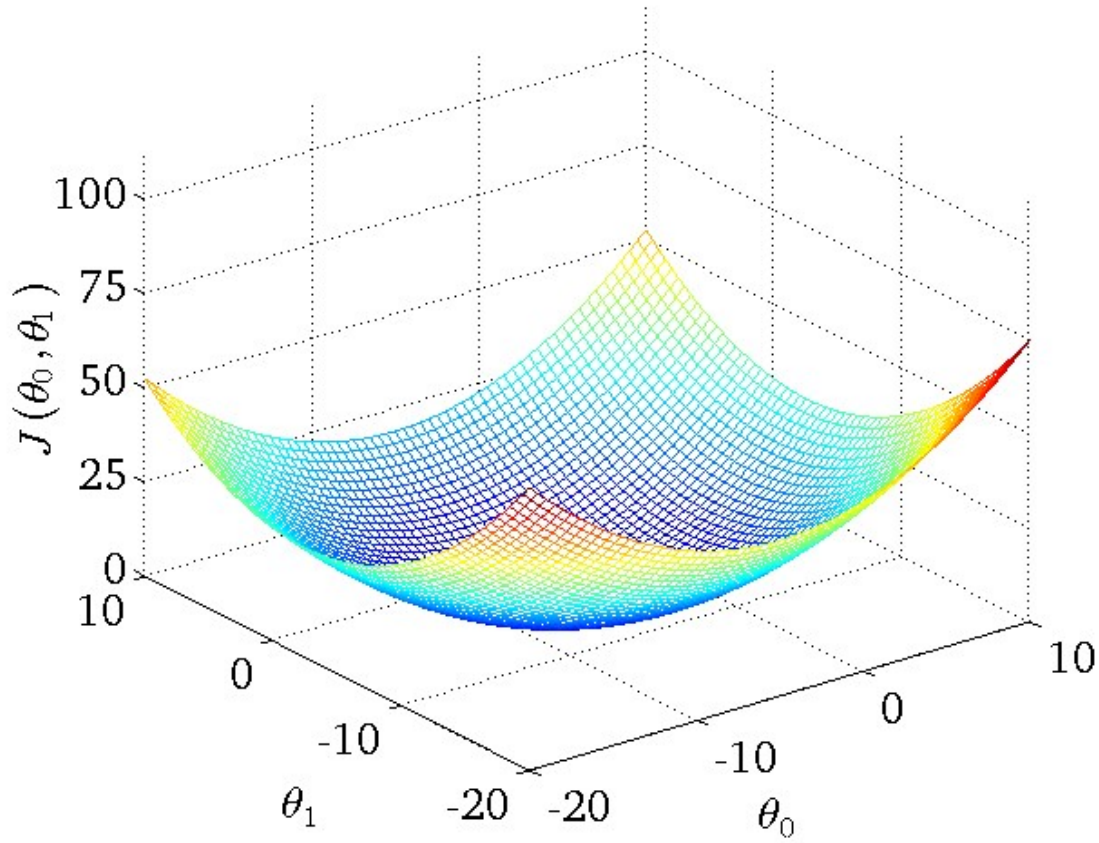
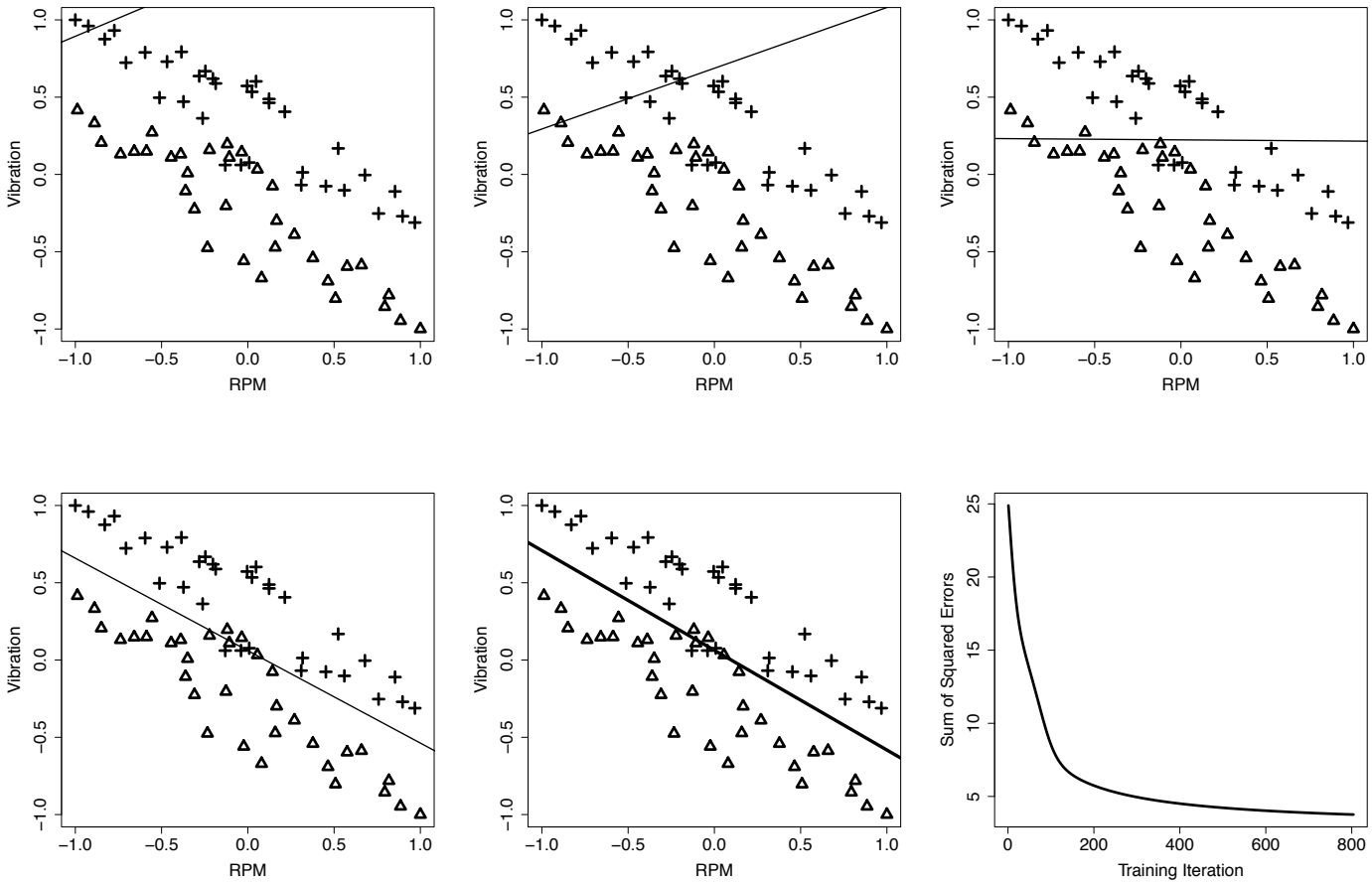
Gradient descend for logistic regression



Fortunately the error surface for logistic regression is convex.

Andrew Ng, Machine Learning, Coursera course

Gradient descend for logistic regression



John Kelleher and Brian Mac Namee and Aoife D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics

Gradient descent variants

- There are three variants of gradient descent.
  - Batch gradient descent
  - Stochastic gradient descent
  - Mini-batch gradient descent

- The difference of these algorithms is **the amount of data.**

Update equation

$$\theta = \theta - \eta * \nabla_\theta J(\theta)$$

This term is different with each method

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Batch gradient descent

This method computes the gradient of the cost function with **the entire training dataset**.

Update equation

$$\theta = \theta - \eta * \nabla_\theta J(\theta)$$

We need to calculate the gradients for the whole dataset to perform **just one update.**

Code

```
for i in range(nb_epochs):
  params_grad = evaluate_gradient(loss_function, data, params)
  params = params - learning_rate * params_grad
```

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Batch gradient descent

- Advantage
  - It is guaranteed to converge **to the global minimum for convex error surfaces and to a local minimum for non-convex surfaces.**

- Disadvantages
  - It can be **very slow**.
  - It is intractable for datasets that **do not fit in memory**.
  - It **does not allow** us to update our model **online.**

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Stochastic gradient descent

This method performs a parameter update for **each** training example $x^{(i)}$ and label $y^{(i)}$.

Update equation

$$\theta = \theta - \eta * \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$$

We need to calculate the gradients for the whole dataset to perform **just one update.**

Code

```
for i in range(nb_epochs):
  np.random.shuffle(data)
  for example in data:
    params_grad = evaluate_gradient(loss_function, example, params)
    params = params - learning_rate * params_grad
```

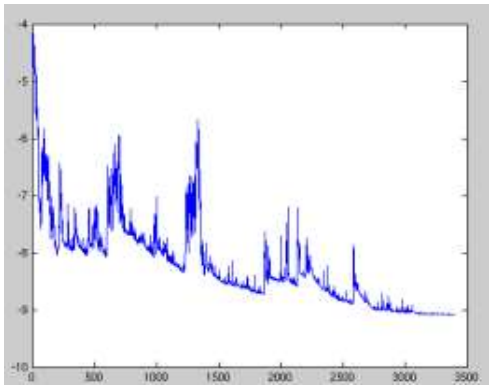
Note : we shuffle the training data at every epoch

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Stochastic gradient descent

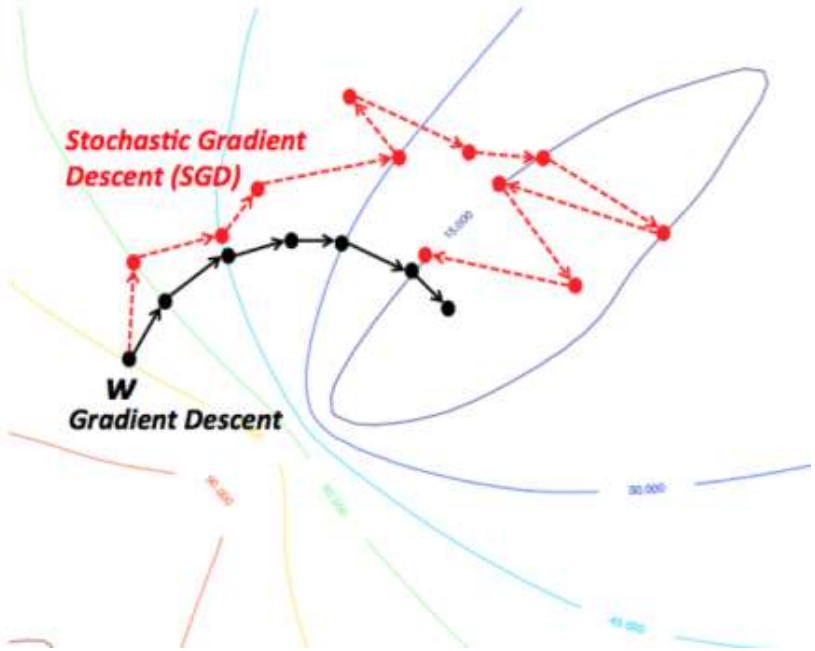
- Advantage
  - It is usually **much faster** than batch gradient descent.
  - It can be **used to learn online.**

- Disadvantages
  - It performs frequent updates with a **high variance** that cause the objective function to fluctuate heavily.



Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

The fluctuation : Batch vs SGD



https://wikidocs.net/3413

- Batch gradient descent converges to the minimum of the basin the parameters are placed in and the **fluctuation is small**.

- **SGD's fluctuation is large but it enables to jump to new and potentially better local minima.**

However, this ultimately complicates convergence to the exact minimum, as SGD will keep overshooting

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Learning rate of SGD

- When **we slowly decrease the learning rate, SGD shows the same convergence behaviour as batch gradient descent**
  - It almost certainly converging to a local or the global minimum for non-convex and convex optimization respectively.

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Mini-batch gradient descent

This method takes the best of both batch and SGD, and performs an update for every mini-batch of n.

Update equation

$$\theta = \theta - \eta * \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)})$$

Code

```
for i in range(nb_epochs):
  np.random.shuffle(data)
  for batch in get_batches(data, batch_size=50):
    params_grad = evaluate_gradient(loss_function, batch, params)
    params = params - learning_rate * params_grad
```

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Mini-batch gradient descent

- Advantage :
  - It **reduces the variance** of the parameter updates.
    - This can lead to more stable convergence.
  - It can make use of highly optimized matrix optimizations common to deep learning libraries that make computing the gradient very efficiently.
- Disadvantage :
  - We have to set mini-batch size.
    - Common mini-batch sizes range between 50 and 256, but can vary for different applications.

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Trade-off

- Depending on the amount of data, they make a trade-off :
  - The **accuracy** of the parameter update
  - The **time** it takes to perform an update.

Method	Accuracy	Time	Memory Usage	Online Learning
Batch gradient descent	○	Slow	High	×
Stochastic gradient descent	△	High	Low	○
Mini-batch gradient descent	○	Midium	Midium	○

Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv:1609.04747 (2016).

Content

Multiclass classification

- Email foldering/tagging: Work, Friends, Family, Hobby

$$y = \begin{cases} 1, & \textit{Work} \\ 2, & \textit{Friends} \\ 3, & \textit{Family} \\ 4, & \textit{Hobby} \end{cases}$$

- Medical diagrams: Not ill, Cold, Flu
- Weather: Sunny, Cloudy, Rain, Snow

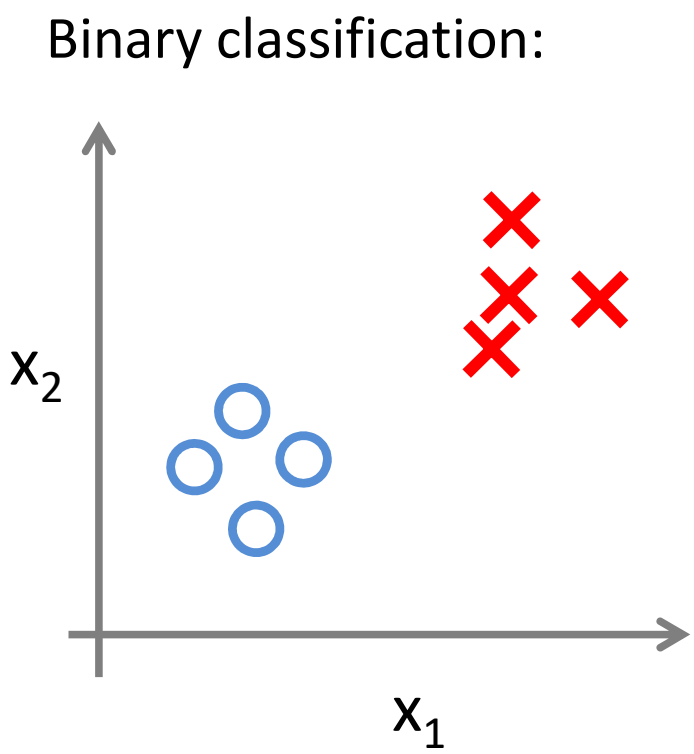
Multiclass classification

- Email foldering/tagging: Work, Friends, Family, Hobby

$$y = \begin{cases} 1, & \textit{Work} \\ 2, & \textit{Friends} \\ 3, & \textit{Family} \\ 4, & \textit{Hobby} \end{cases}$$

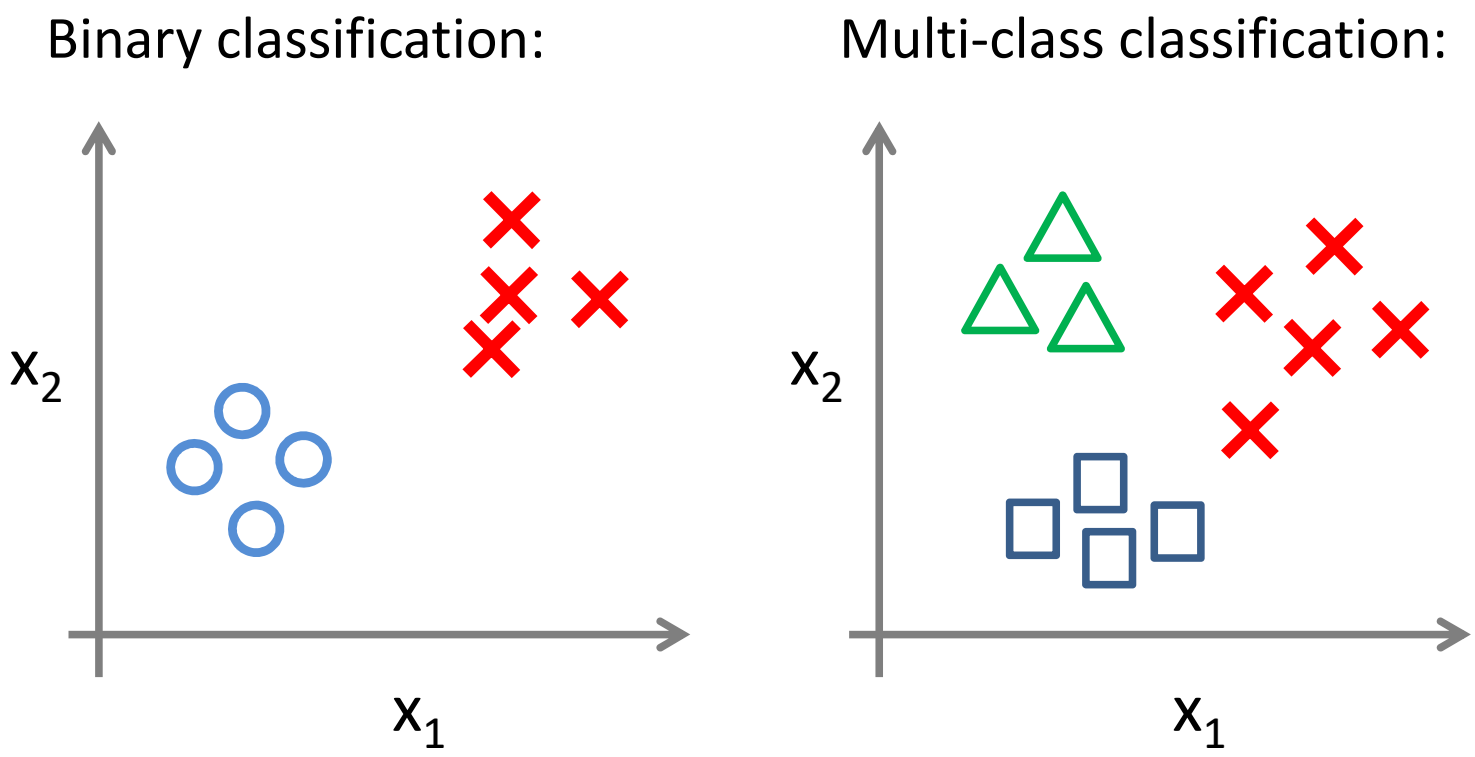
- Medical diagrams: Not ill, Cold, Flu
- Weather: Sunny, Cloudy, Rain, Snow

Multiclass classification



Andrew Ng, Machine Learning, Coursera course

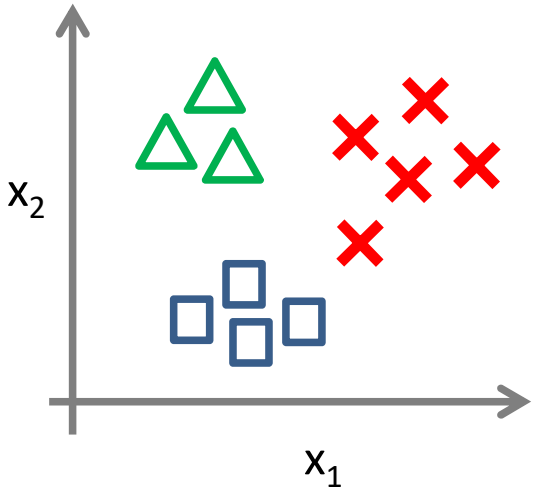
Multiclass classification



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

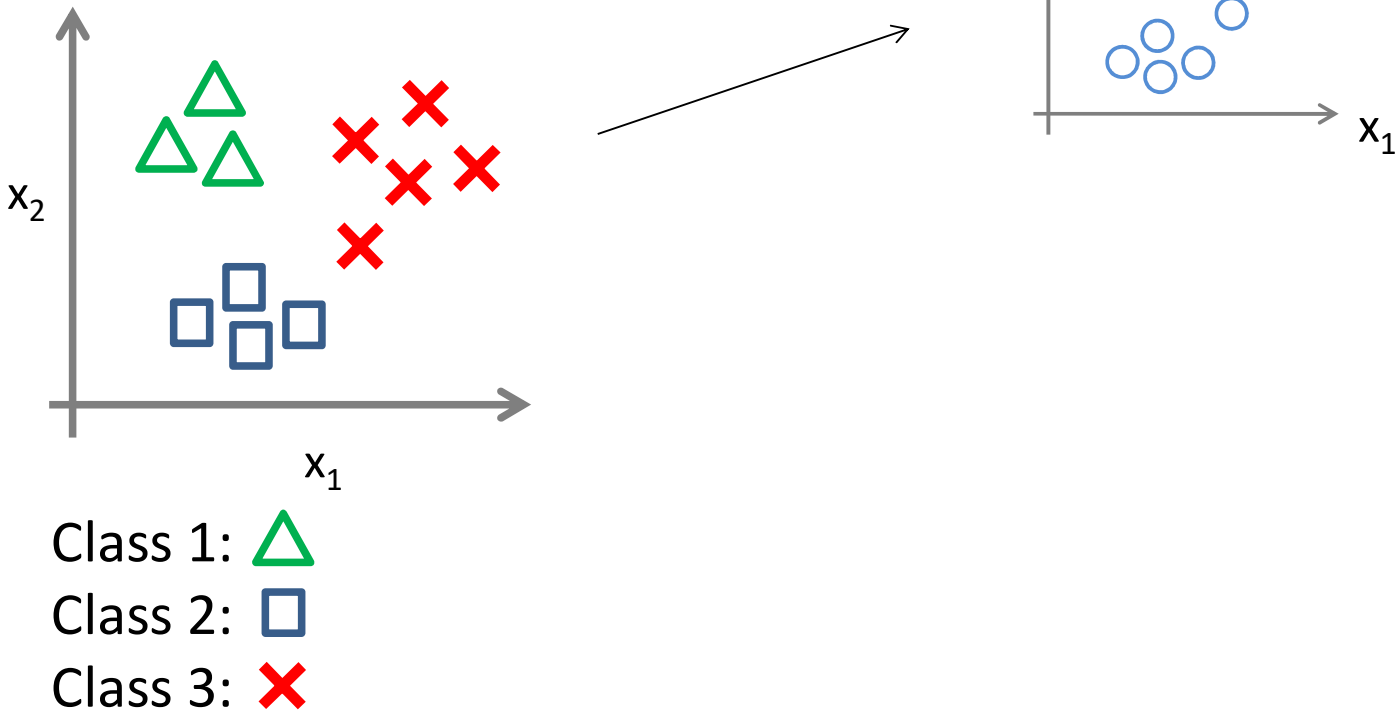
One-vs-all (one-vs-rest):



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

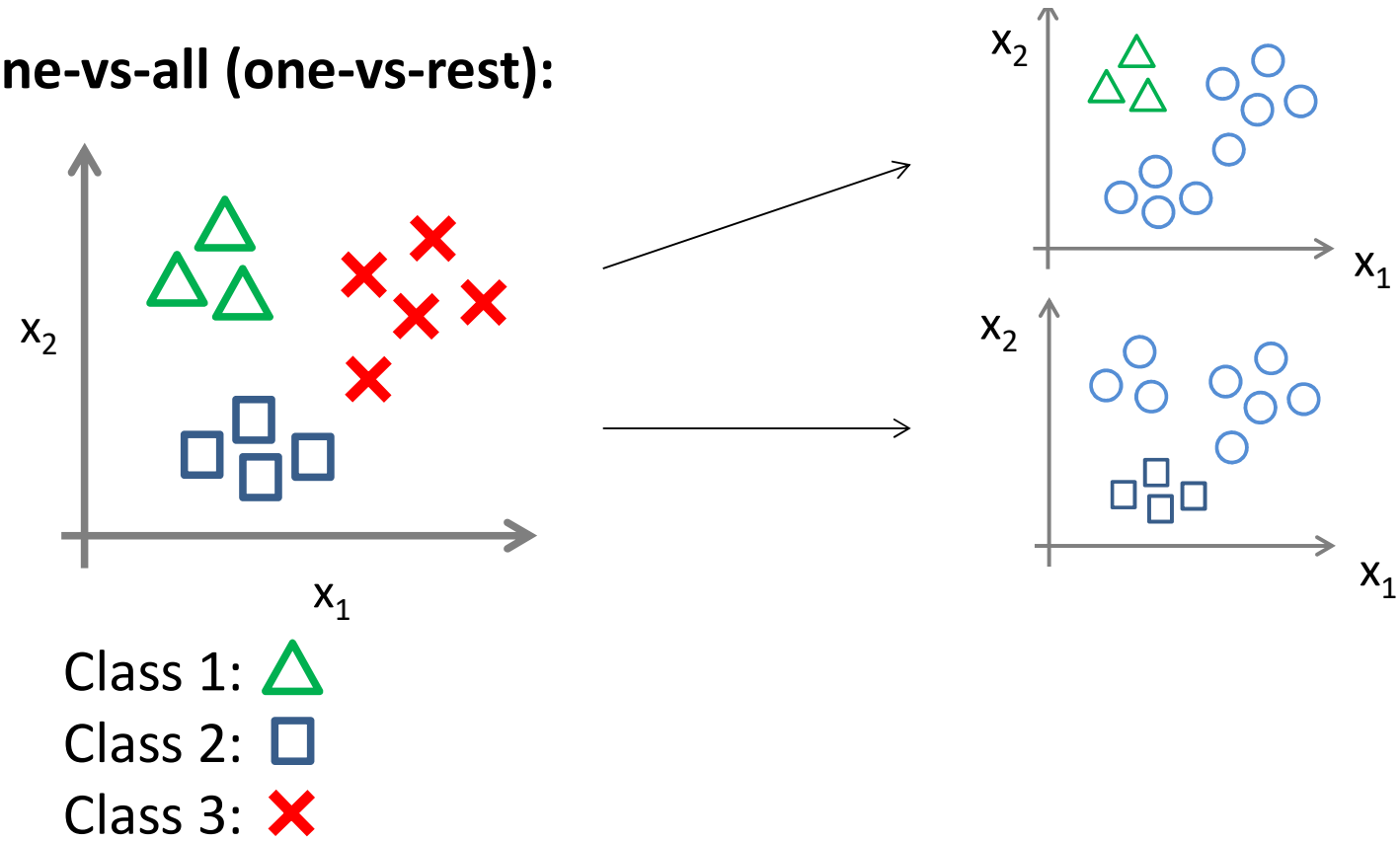
One-vs-all (one-vs-rest):



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

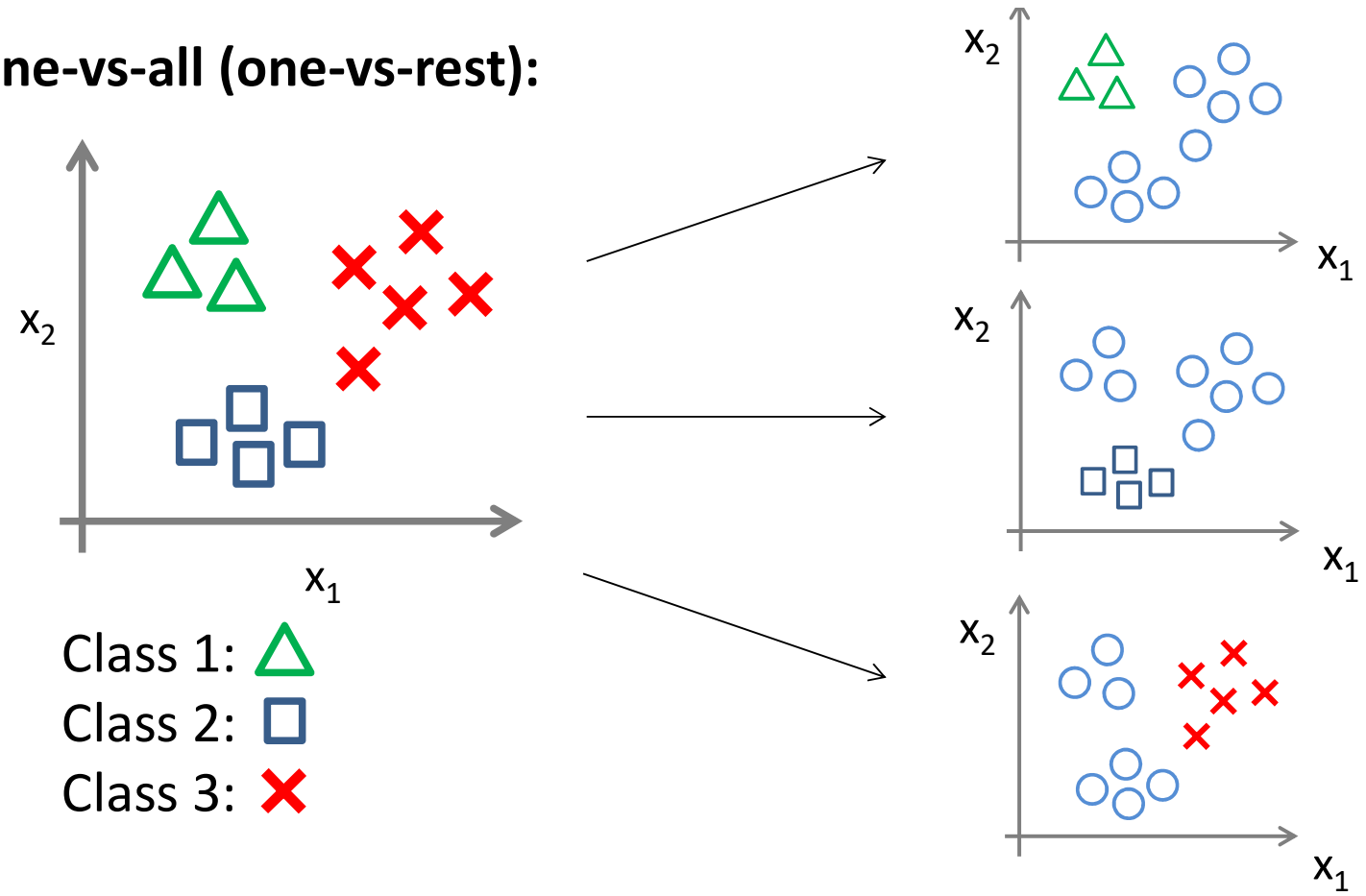
One-vs-all (one-vs-rest):



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

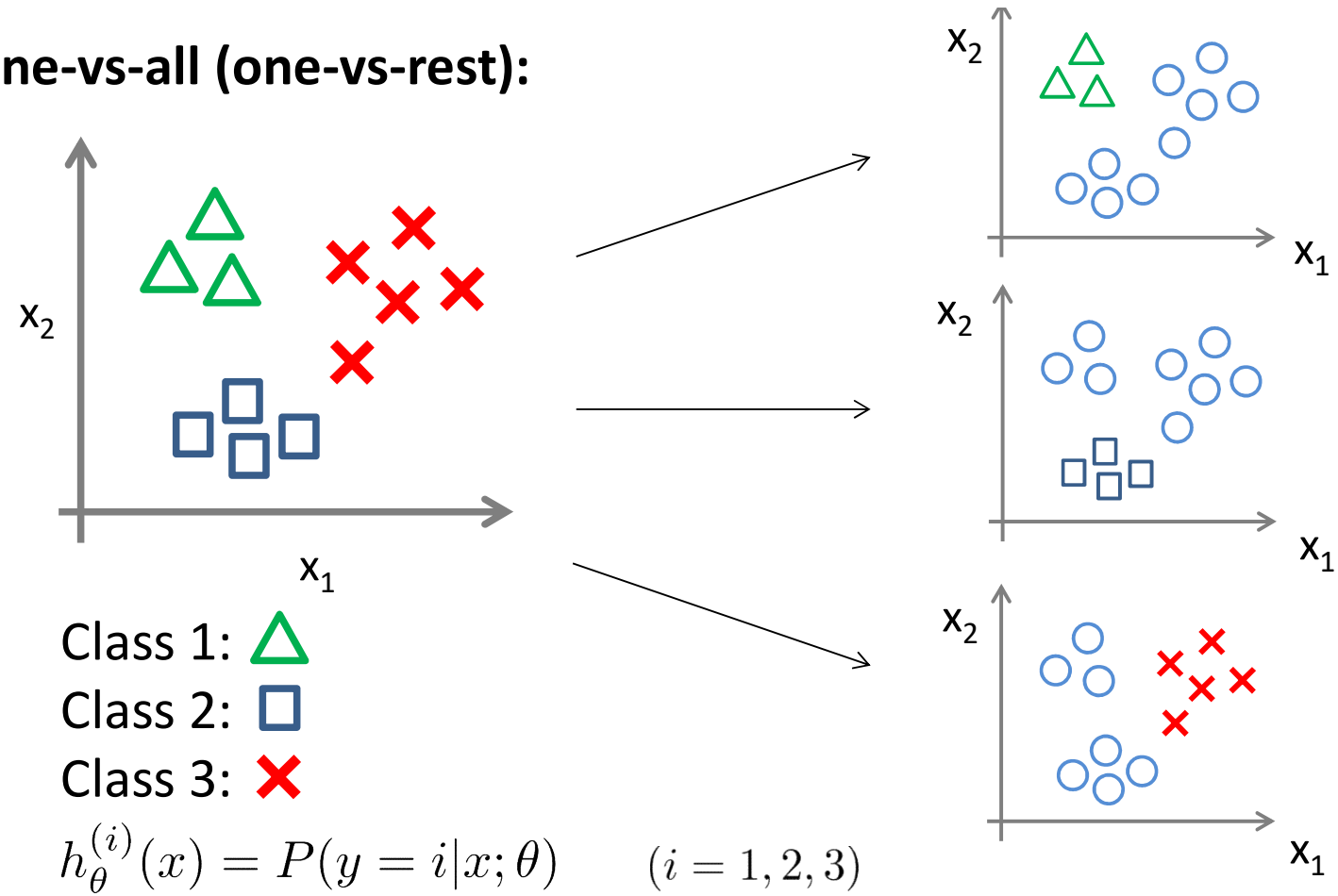
One-vs-all (one-vs-rest):



Andrew Ng, Machine Learning, Coursera course

Multiclass classification

One-vs-all (one-vs-rest):



Class 1: △

Class 2: □

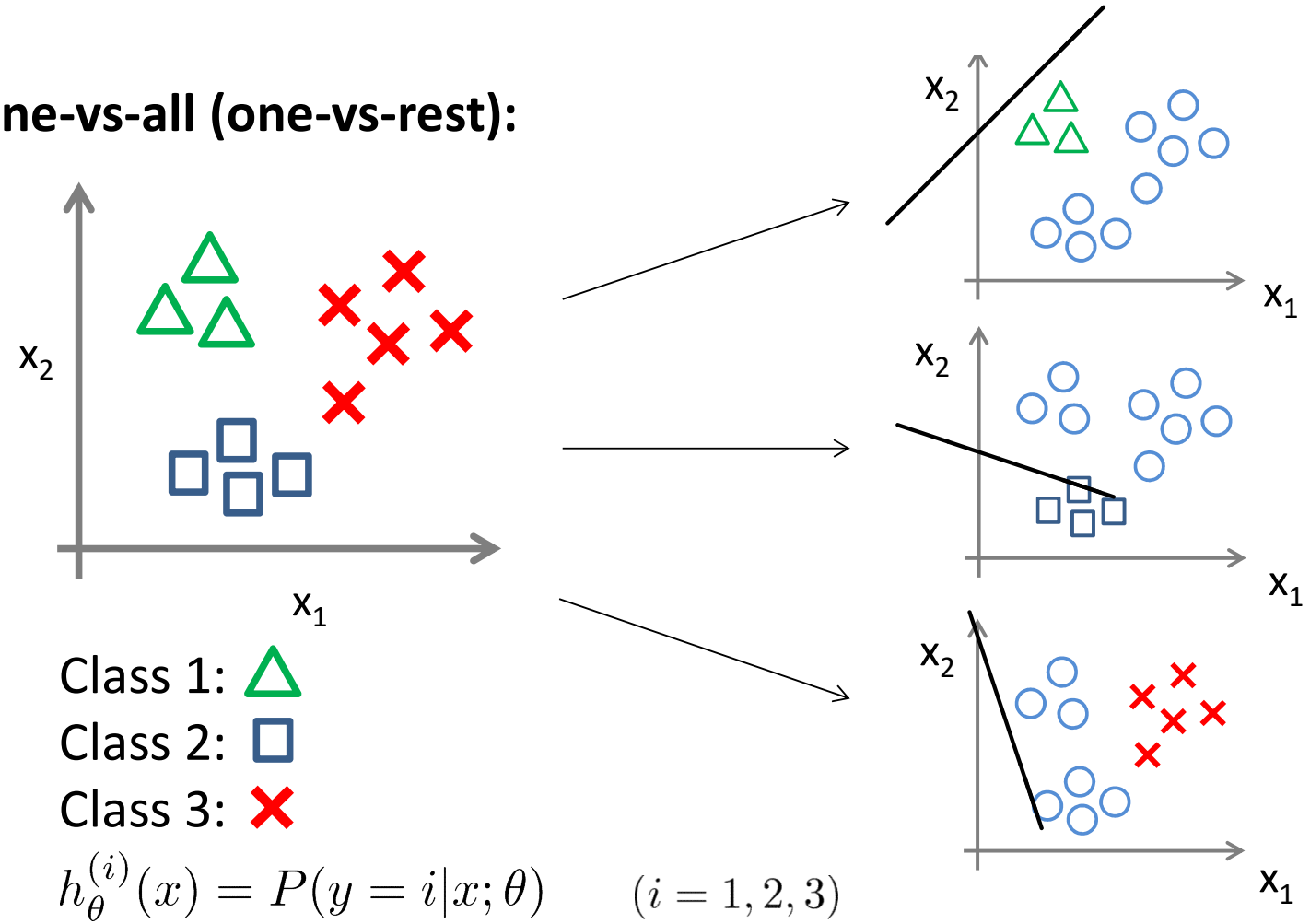
Class 3: ×

$$h_\theta^{(i)}(x) = P(y = i|x; \theta) \qquad (i = 1, 2, 3)$$

Andrew Ng, Machine Learning, Coursera course

Multiclass classification

One-vs-all (one-vs-rest):



Class 1: △

Class 2: □

Class 3: ×

$$h_\theta^{(i)}(x) = P(y = i|x; \theta) \qquad (i = 1, 2, 3)$$

Andrew Ng, Machine Learning, Coursera course

Multiclass classification

One-vs-all

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

Andrew Ng, Machine Learning, Coursera course

Multiclass classification

One-vs-all

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

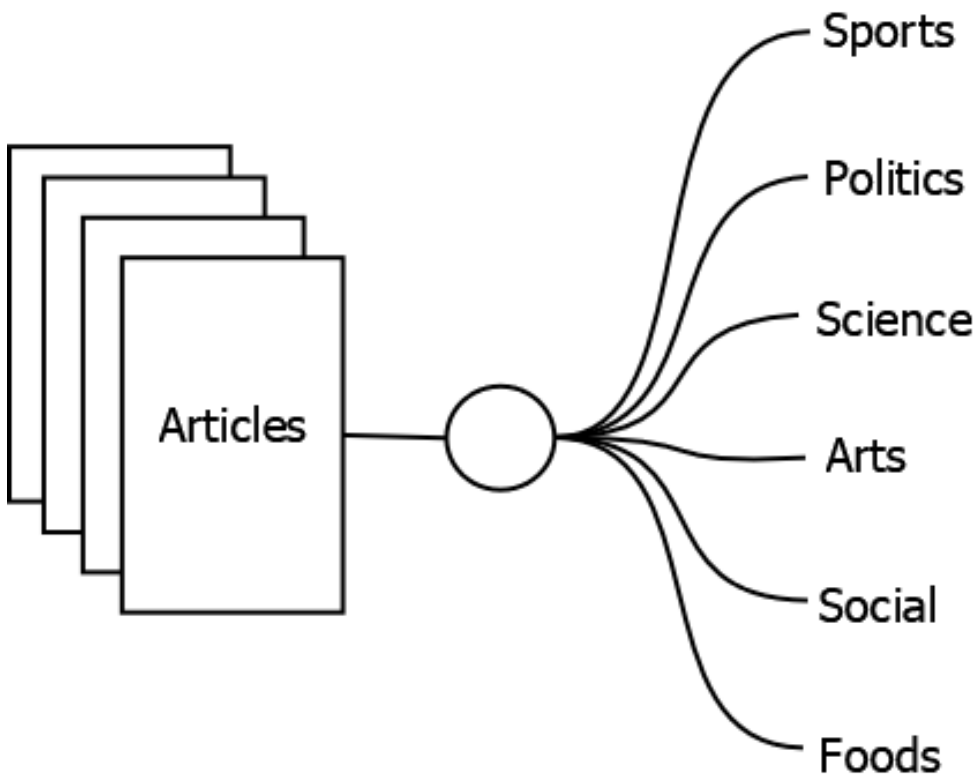
On a new input x, to make a prediction, pick the class i that maximizes

$$\max_i h_\theta^{(i)}(x)$$

Andrew Ng, Machine Learning, Coursera course

Content

Text classification

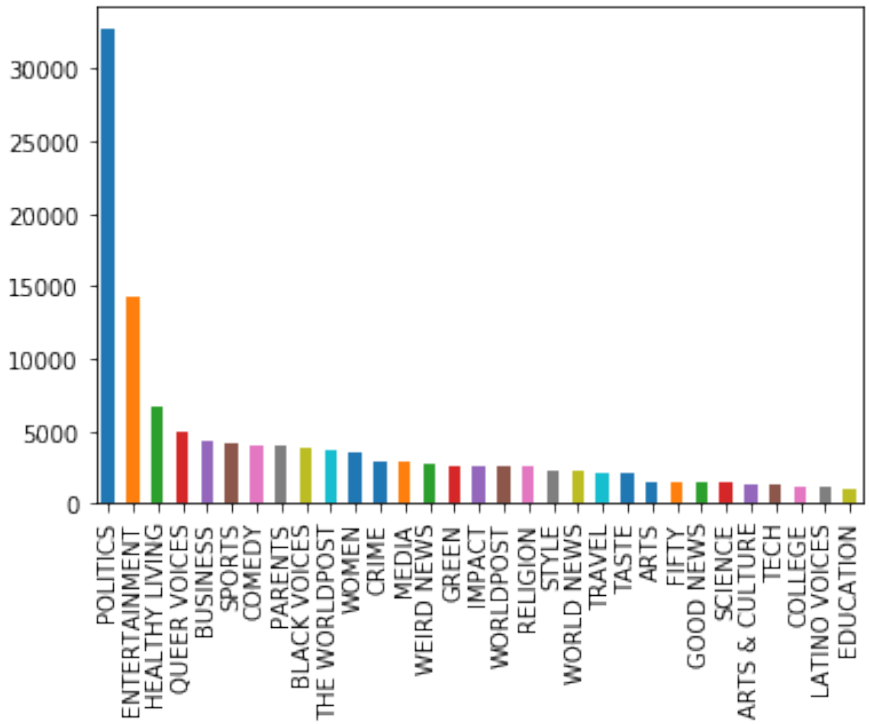


Applications

- Junk email filtering
- News topic classification
- Authorship attribution
- Sentiment analysis
- Genre classification
- Offensive language identification
- Language identification

Huffpost News Category Dataset

This dataset contains around 200k news headlines from the year 2012 to 2018 obtained from HuffPost.



https://www.kaggle.com/rmisra/news-category-dataset

Huffpost News Category Dataset

authors	category	date	headline	link	short_description
	ENTERTAINMENT	2014-10-09	Eek! Mario Lopez Admits He Never Loved Ex-Wife...	https://www.huffingtonpost.com/entry/mario-lop...	
Sarah Ruiz-Grossman	BLACK VOICES	2018-04-06	Protesters Demand Justice For Saheed Vassell, ...	https://www.huffingtonpost.com/entry/protests-...	"They murdered my son, and I want justice," hi...
Lee Moran	CRIME	2016-02-26	Police Hunt For Man Traveling The Midwest And ...	https://www.huffingtonpost.com/entry/rogaine-t...	The suspect, who is bald, reportedly has taken...
Michael Carosone, ContributorWriter, educator,...	QUEER VOICES	2014-09-07	She Inspired Me: My Tribute to Joan Rivers	https://www.huffingtonpost.com/entry/she-inspi...	I never met Joan Rivers; I always wanted to, b...
Amanda Pena	STYLE	2017-11-08	21 Affordable Holiday Gifts That Look Really E...	https://www.huffingtonpost.com/entry/affordabl...	Our idea of luxury won't break the bank.
Zahara Hill	BLACK VOICES	2017-03-06	Viola Davis Gives (Another) Moving Speech As H...	https://www.huffingtonpost.com/entry/viola-dav...	"I want people to be seen. I want them to feel...

Kavita Ganesan, Build Your First Text Classifier in Python with Logistic Regression
https://kavita-ganesan.com/news-classifier-with-logistic-regression-in-python

Procedure

- Text preprocessing
- Feature extraction
- Model training
- Model Application
- Evaluation

Text preprocessing

- Text cleaning (removing HTML/XML tags, figures, formula, etc.)
- Removing stop words
- Tokenization
- Stemming

Stop words

- A stop word is a commonly used word (such as “the”, “a”, “an”, “in”).
- Stop words are not helpful for text classification because they occur in almost all documents,
- Stop words are normally removed before applying a text classification algorithm.

Feature extraction

- Each document is represented as a vector in order to applying a classification algorithm.
- Each dimension of the input vector is called a feature.
- In text classification, the most straightforward idea is to use words as features.

doc_id	book	read	music	go
doc1	3	1	0	5
doc2	2	5	3	0
doc3	0	0	7	2

Weighting of words in document vectors

- **Term** - a word or a collocation.
- **Document** - a sequence of terms.
- **Corpus** - a set of documents.

Weighting of words in document vectors

Boolean weighting: $$w_{ik} = \begin{cases} 1 & \text{if } f_{ik} > 0 \\ 0 & \text{Otherwise} \end{cases}$$

Word frequency weighting: $$w_{ik} = f_{ik}$$

TF-IDF weighting: $$w_{ik} = f_{ik} \times \log \frac{N}{n_i}$$

i: word index

k: document index

f_{ik}: word frequency in a document

N: number of documents in the corpus

n_i: number of documents containing the word

Weighting of words in document vectors

- TF-IDF - short for *term frequency–inverse document frequency*, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
- The assumption behind the use of the inverse document frequency is: the more documents where a word (term) occurs, the less important the word is to that document.
- TF-IDF is proposed in information retrieval but also used in other areas including NLP.
- Which word weighting method is the best for text classification: no universal answer. It empirically depends on the data and the classification algorithm you use.

Algorithms

- Logistic regression
- Nearest neighbor
- Decision trees
- Support vector machines
- Neural networks

Further topics

- Feature selection
- Dimension reduction
- Document embeddings

Content

1. Machine Learning basics
2. Classification and logistic regression
3. Text Classification