Combining Evidence

Module Introduction

Evidence of Relevance

So far, we have tried to determine a document's relevance to a query by comparing document terms to query terms. This works relatively well, but it's far from perfect. We can use many additional forms of evidence to improve our relevance estimates.

- **Document quality scores:** Is a document written and presented well? Is it authoritative, and written by a reputable source? Does it look like spam?
- **Document categories:** Does a document present information about news, sports, some other common category? Is it providing a service, such as a storefront?
- **Internet link structure:** Which pages link to the document, and where does it link to? What does the anchor text of those links say?
- **Document structure:** Does the document have a title? Section headings? A table of contents?
- **User behavior:** click information, duration of page visits, etc.

Types of Evidence

In the last module, we saw how to obtain various forms of evidence of document relevance. Here, we will assume the evidence is provided to us and focus on what to do with it.

Evidence can come in several forms:

- **Binary features:** presence or absence of terms, whether the page is on a well-known domain (wikipedia.org, cnn.com, …), whether the user has previously visited the page…
- **Real-valued features:** probabilities, term counts, page visit durations, product prices…
- **Categorical features:** page categories (sports, news, shopping, reviews…), language, domain categories (business, social site, news, informational)
- **Timestamps:** date crawled, date of last update, date of first appearance on the web, …

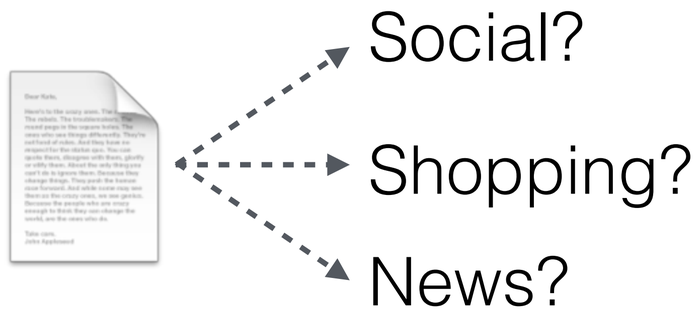
All of these are generally treated as real numbers, after some pre-processing.

Machine Learning Tasks for IR

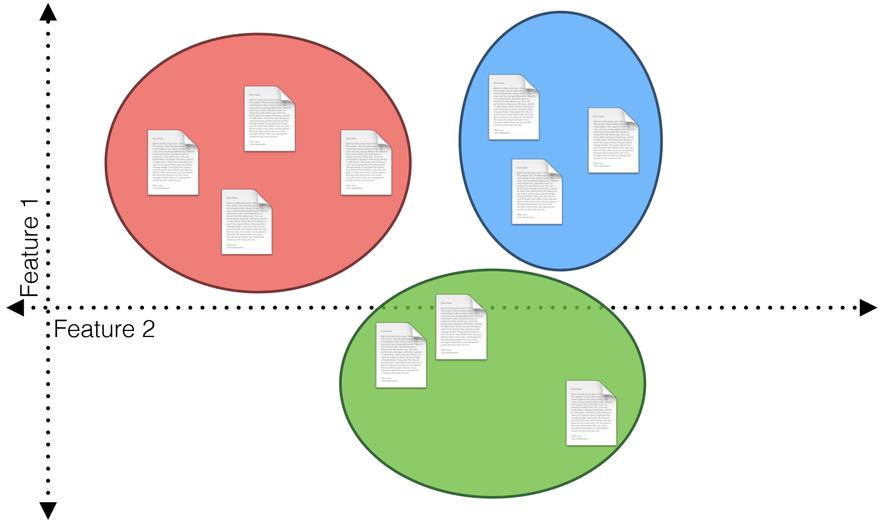
IR is concerned with a few main ML tasks:

- **Document classification:** Which categories does a document fit into?
- **Document ranking:** Which documents are probably more relevant to a query?
- **Document clustering:** Which documents are most similar to each other?

Document Classification



Document Clustering



Module Goals

By the end of this module, you should be able to:

- Classify and rank documents using Support Vector Machines.
- Cluster documents, and use the clusters to produce a more diverse ranking.

Let’s get started!

Supervised Learning

Combining Evidence, session 2

Document Classification

Suppose we know various features of a document, and we want to decide whether it's a news article.

We select a model ("hypothesis space") – a function which determines whether a document is news, based on its features – and want to choose the best model parameters ("hypothesis").

We find the best parameters using *supervised learning* – we use a collection of documents whose true labels are known, and we pick the parameters which best predict those labels.

Goal: Pick θ to predict true labels from features

$$Y = f(X; \vartheta)$$

Document Features = X				**Label = Y**
tf	**tf**	**Known news website?**	**Facebook Likes**	**news**
1	1	0	123	1
0	0	1	54	1
0	0	0	1,213	0
2	0	0	0	0
0	1	1	560	1

Supervised Learning

Supervised Learning is essentially learning by example. A machine learning algorithm takes as input a set of training data:

- An $n \times p$ feature matrix X of n training instances, each with p features.
- An $n \times 1$ label vector Y which provides the correct label for each training instance in X.

Each of the n rows of X represents a distinct training instance. The goal of the learning algorithm is to find a function which outputs the correct Y value for each training instance.

Document Features = X				**Label = Y**
tf	**tf**	**Known news website?**	**Facebook Likes**	**news**
1	1	0	123	1
0	0	1	54	1
0	0	0	1,213	0
2	0	0	0	0
0	1	1	560	1

Training and Test Data

When the machine learning algorithm has chosen a function, we evaluate it by using it to classify a second data set, the *test data*.

- The fraction of correctly-classified instances is called *accuracy*.
- The fraction of incorrectly-classified instances is called *error*.

The test data should be generated by the same process as the training data. Commonly, we will receive a large data set which we randomly split into training and test sets.

Confusion Matrix

	Y = 1	Y = -1
f(X) = 1	TP	FP
f(X) = -1	FN	TN

$$acc = \frac{tp + tn}{tp + fp + fn + tn}$$

$$error = \frac{fp + fn}{tp + fp + fn + tn} = 1 - acc$$

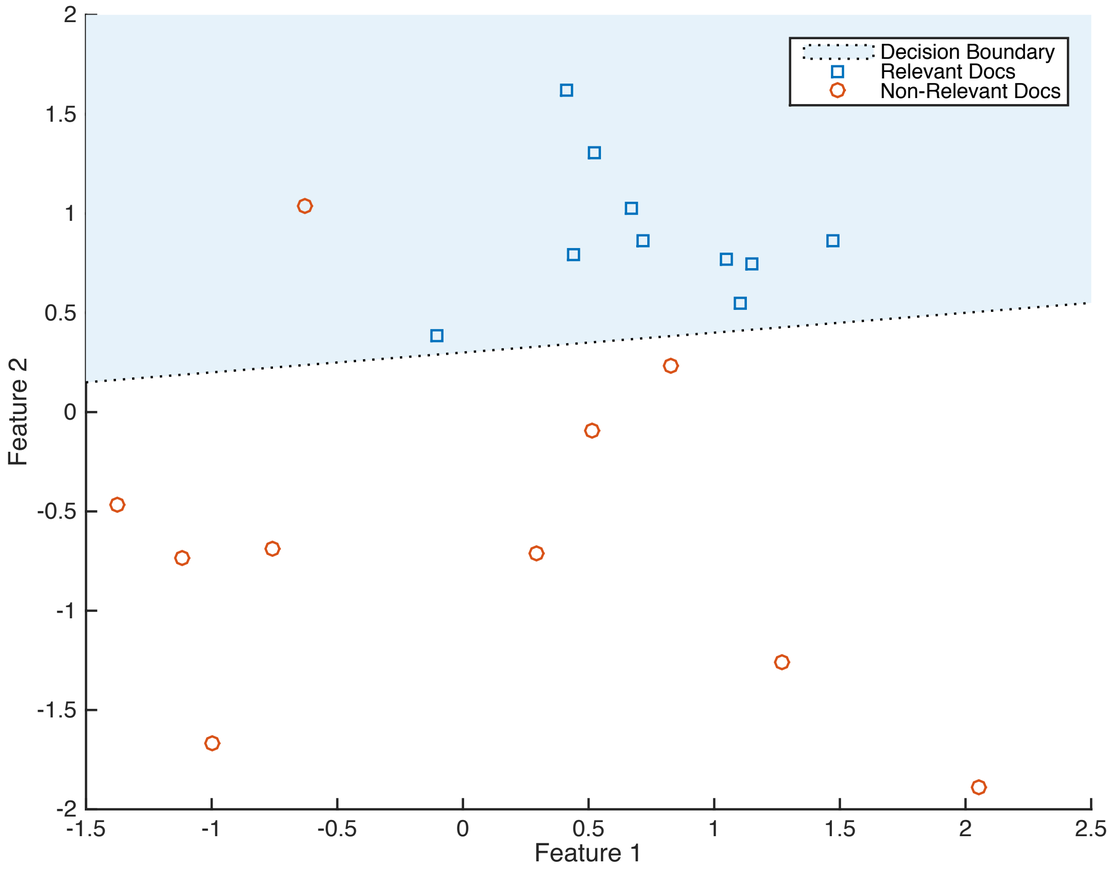
Linear Classifiers

One of the simplest models is the set of lines: everything above a line has one label, and everything below it has the other label. The model for a k-dimensional linear classifier is:

$$Y_i = sign(\vartheta \cdot X_i)$$

$$= \begin{cases} +1 & \text{if } \sum_{j=1}^{k} (\vartheta_j \cdot X_{i,j}) > 0 \\ -1 & \textbf{otherwise} \end{cases}$$

We typically define $X_0 = 1$ so we can use θ_0 as the y-intercept.

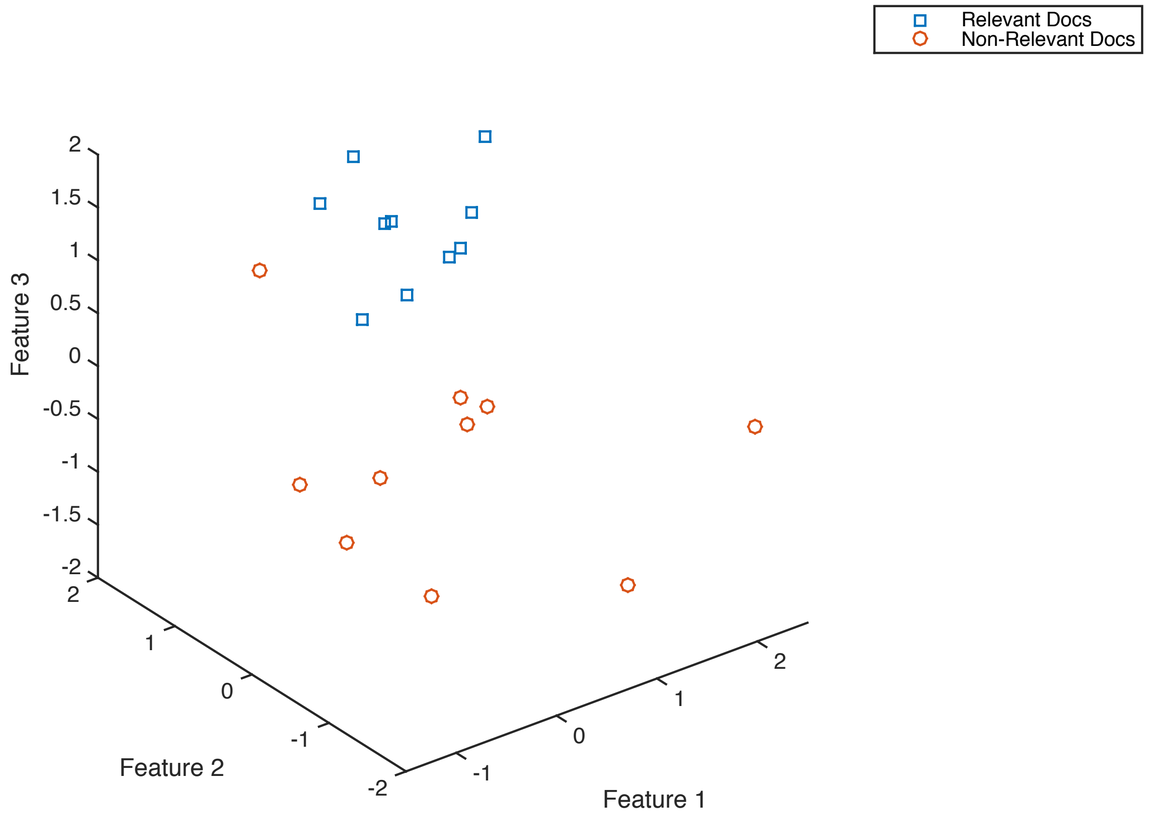


Visualizing Linear Classifiers

A k-dimensional linear classifier is a generalized equation for a line. The decision boundary is always one fewer dimensions less than the feature space.

- For $k = 2$, the boundary is a line.
- For $k = 3$, it is a plane.
- For $k > 3$, it is a $k - 1$ dimensional hyperplane

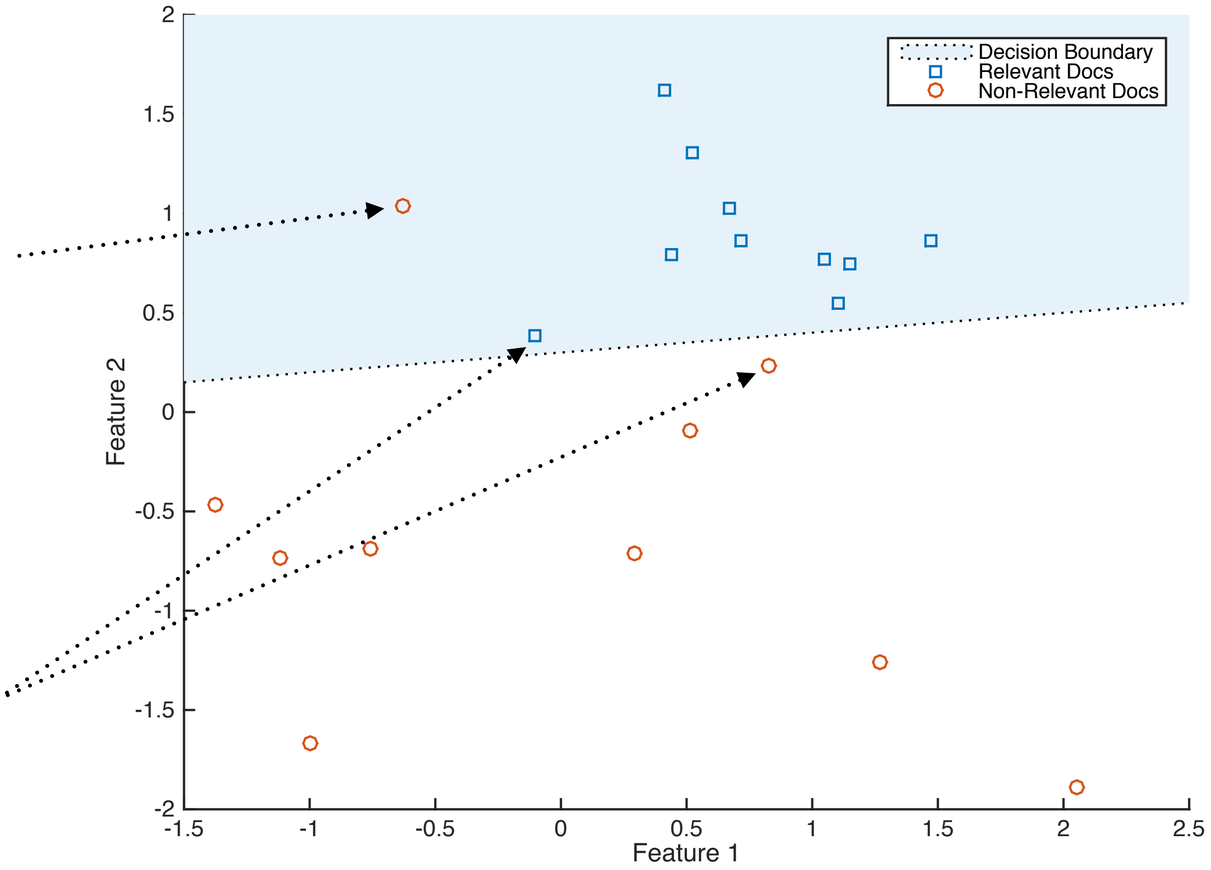
The region on the same side of a linear decision boundary is known as a *half space*.



Linearly Separable Data

If any linear decision surface exists which perfectly divides the training instances, the data is said to be *linearly separable*. Most data sets can't be neatly separated in this way.

- Some data points closely resemble points of the opposite class, and are harder to classify correctly.
- Some points may have incorrect feature or label values, leading the learning algorithm astray.
- Other points are near the decision boundary, and are susceptible to misclassification if a slightly different classification function was chosen.



Wrapping Up

Although most data sets can't be perfectly classified using machine learning techniques, there are many good techniques which can generally achieve high accuracy.

One of the most important techniques uses linear decision surfaces to make a decision: points "above the line" are considered positive instances, and points "below the line" are negative instances.

Next, we'll look at linear classifiers in more depth.