Classification II: Decision Trees and SVMs

Digging into Data

March 3, 2014

COLLEGE OF INFORMATION STUDIES

Slides adapted from Tom Mitchell, Eric Xing, and Lauren Hannah

Roadmap

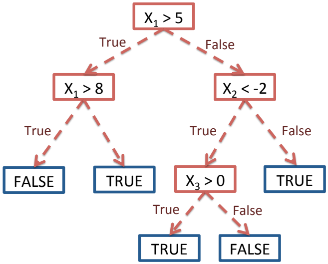
- Classification: machines labeling data for us
- Last time: naïve Bayes and logistic regression
- This time:
  - Decision Trees
    - Simple, nonlinear, interpretable
  - SVMs
    - (another) example of linear classifier
    - State-of-the-art classification
  - Examples in Rattle (Logistic, SVM, Trees)
  - **Discussion:** Which classifier should I use for my problem?

Outline

Trees

Suppose that we want to construct a set of rules to represent the data

- can represent data as a series of if-then statements
- here, "if" splits inputs into two categories
- "then" assigns value
- when "if" statements are nested, structure is called a tree

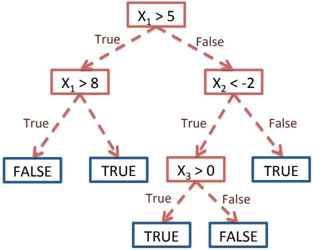


Trees

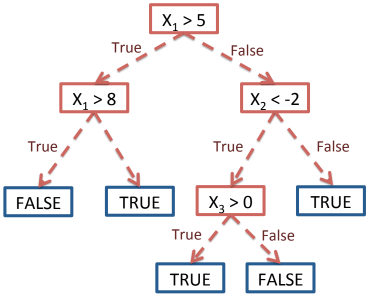
Ex: data (X_1, X_2, X_3, Y) with X_1, X_2, X_3 are real, Y Boolean

First, see if $X_1 > 5$:

- if TRUE, see if $X_1 > 8$
  - if TRUE, return FALSE
  - if FALSE, return TRUE
- if FALSE, see if $X_2 < -2$
  - if TRUE, see if $X_3 > 0$
    - if TRUE, return TRUE
    - if FALSE, return FALSE
  - if FALSE, return TRUE



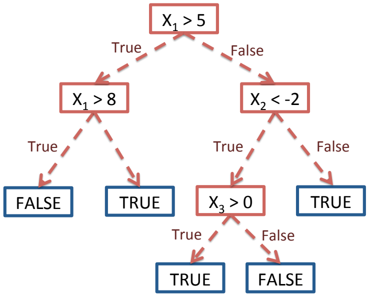
Trees



Example 1: $(X_1, X_2, X_3) = (1, 1, 1)$

Example 2: $(X_1, X_2, X_3) = (10, -3, 0)$

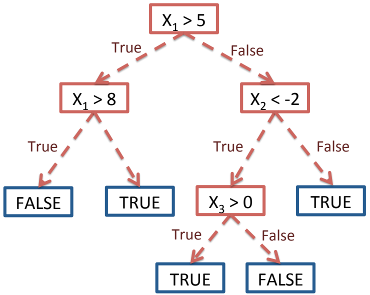
Trees



Example 1: $(X_1, X_2, X_3) = (1, 1, 1) \rightarrow$ TRUE

Example 2: $(X_1, X_2, X_3) = (10, -3, 0)$

Trees



Example 1: $(X_1, X_2, X_3) = (1, 1, 1) \rightarrow$ TRUE

Example 2: $(X_1, X_2, X_3) = (10, -3, 0) \rightarrow$ FALSE

Trees

Terminology:

- branches: one side of a split
- leaves: terminal nodes that return values

Why trees?

- trees can be used for regression or classification
  - regression: returned value is a real number
  - classification: returned value is a class
- unlike linear regression, SVMs, naive Bayes, etc, trees fit *local models*
  - in large spaces, global models may be hard to fit
  - results may be hard to interpret
- fast, interpretable predictions

Example: Predicting Electoral Results

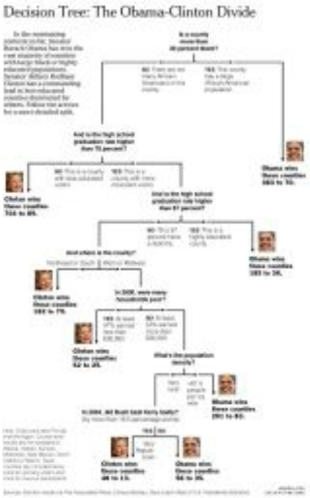
2008 Democratic primary:

- Hillary Clinton
- Barack Obama

Given historical data, how will a county (small administrative unit inside an American state) vote?

- can extrapolate to state level data
- might give regions to focus on increasing voter turnout
- would like to know how variables interact

Example: Predicting Electoral Results



Example: Predicting Electoral Results

Decision Trees

Decision tree representation:

- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

How would we represent as a function of X, Y:

- X AND Y (both must be true)
- X OR Y (either can be true)
- X XOR Y (one and only one is true)

When to Consider Decision Trees

- Instances describable by attribute-value pairs
- Target function is discrete valued
- Disjunctive hypothesis may be required
- Possibly noisy training data

Examples:

- Equipment or medical diagnosis
- Credit risk analysis
- Modeling calendar scheduling preferences

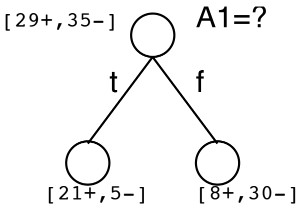
Outline

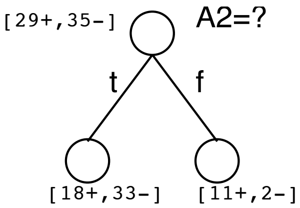
Top-Down Induction of Decision Trees

Main loop:

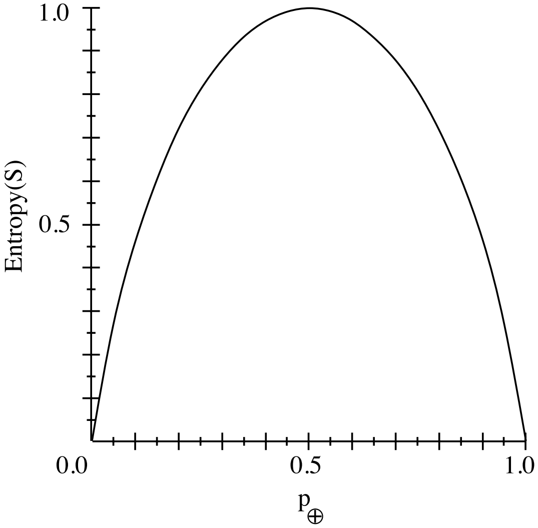
1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?





Entropy: Reminder



- *S* is a sample of training examples

Entropy

How spread out is the distribution of S:

$$p_{\oplus}(-\log_2 p_{\oplus}) + p_{\ominus}(-\log_2 p_{\ominus})$$

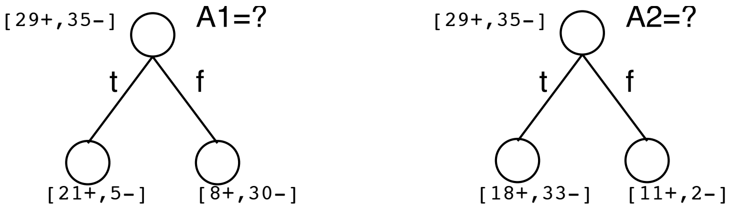
$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Information Gain

Which feature A would be a more useful rule in our decision tree?

$Gain(S, A)$ = expected reduction in entropy due to sorting on A

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



$$
\begin{aligned}
H(S) &= -\frac{29}{54}\lg\left(\frac{29}{54}\right) - \frac{35}{64}\lg\left(\frac{35}{64}\right) \\
&=
\end{aligned}
$$

$$
\begin{aligned}
H(S) &= -\frac{29}{54}\lg\left(\frac{29}{54}\right) - \frac{35}{64}\lg\left(\frac{35}{64}\right) \\
&= 0.96
\end{aligned}
$$

$$H(S) = -\frac{29}{54}\lg\left(\frac{29}{54}\right) - \frac{35}{64}\lg\left(\frac{35}{64}\right)$$
$$= 0.96$$

$$\text{Gain}(S, A_1) = 0.96 - \frac{26}{64}\left[-\frac{5}{26}\lg\left(\frac{5}{26}\right) - \frac{21}{26}\lg\left(\frac{21}{26}\right)\right]$$
$$- \frac{38}{64}\left[-\frac{8}{38}\lg\left(\frac{8}{38}\right) - \frac{30}{38}\lg\left(\frac{30}{38}\right)\right]$$
$$=$$

$$
\begin{aligned}
H(S) &= -\frac{29}{54}\lg\left(\frac{29}{54}\right) - \frac{35}{64}\lg\left(\frac{35}{64}\right) \\
&= 0.96
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Gain}(S, A_1) &= 0.96 - \frac{26}{64}\left[-\frac{5}{26}\lg\left(\frac{5}{26}\right) - \frac{21}{26}\lg\left(\frac{21}{26}\right)\right] \\
&\quad - \frac{38}{64}\left[-\frac{8}{38}\lg\left(\frac{8}{38}\right) - \frac{30}{38}\lg\left(\frac{30}{38}\right)\right] \\
&= 0.96 - 0.28 - 0.44 = 0.24
\end{aligned}
$$

$$H(S) = -\frac{29}{54}\lg\left(\frac{29}{54}\right) - \frac{35}{64}\lg\left(\frac{35}{64}\right)$$
$$= 0.96$$

$$\text{Gain}(S, A_1) = 0.96 - \frac{26}{64}\left[-\frac{5}{26}\lg\left(\frac{5}{26}\right) - \frac{21}{26}\lg\left(\frac{21}{26}\right)\right]$$
$$- \frac{38}{64}\left[-\frac{8}{38}\lg\left(\frac{8}{38}\right) - \frac{30}{38}\lg\left(\frac{30}{38}\right)\right]$$
$$= 0.96 - 0.28 - 0.44 = 0.24$$

$$\text{Gain}(S, A_2) = 0.96 - \frac{51}{64}\left[-\frac{18}{51}\lg\left(\frac{18}{51}\right) - \frac{33}{51}\lg\left(\frac{33}{51}\right)\right]$$
$$- \frac{13}{64}\left[-\frac{11}{13}\lg\left(\frac{11}{13}\right) - \frac{2}{13}\lg\left(\frac{2}{13}\right)\right]$$
$$=$$

$$
\begin{aligned}
H(S) &= -\frac{29}{54}\lg\left(\frac{29}{54}\right) - \frac{35}{64}\lg\left(\frac{35}{64}\right) \\
&= 0.96
\end{aligned}
$$

$$
\begin{aligned}
\text{Gain}(S, A_1) &= 0.96 - \frac{26}{64}\left[-\frac{5}{26}\lg\left(\frac{5}{26}\right) - \frac{21}{26}\lg\left(\frac{21}{26}\right)\right] \\
&- \frac{38}{64}\left[-\frac{8}{38}\lg\left(\frac{8}{38}\right) - \frac{30}{38}\lg\left(\frac{30}{38}\right)\right] \\
&= 0.96 - 0.28 - 0.44 = 0.24
\end{aligned}
$$

$$
\begin{aligned}
\text{Gain}(S, A_2) &= 0.96 - \frac{51}{64}\left[-\frac{18}{51}\lg\left(\frac{18}{51}\right) - \frac{33}{51}\lg\left(\frac{33}{51}\right)\right] \\
&- \frac{13}{64}\left[-\frac{11}{13}\lg\left(\frac{11}{13}\right) - \frac{2}{13}\lg\left(\frac{2}{13}\right)\right] \\
&= 0.96 - 0.75 - 0.13 = 0.08
\end{aligned}
$$

ID3 Algorithm

- Start at root, look for best attribute
- Repeat for subtrees at each attribute outcome
- Stop when information gain is below a threshold
- Bias: prefers shorter trees (Occam's Razor)
  - $\longrightarrow$ a short hyp that fits data unlikely to be coincidence
  - $\longrightarrow$ a long hyp that fits data might be coincidence
  - Prevents overfitting (more later)

Outline

Thinking Geometrically

- Suppose you have two classes: vacations and sports
- Suppose you have four documents

Sports

Doc_1: {ball, ball, ball, travel}

Doc_2: {ball, ball}

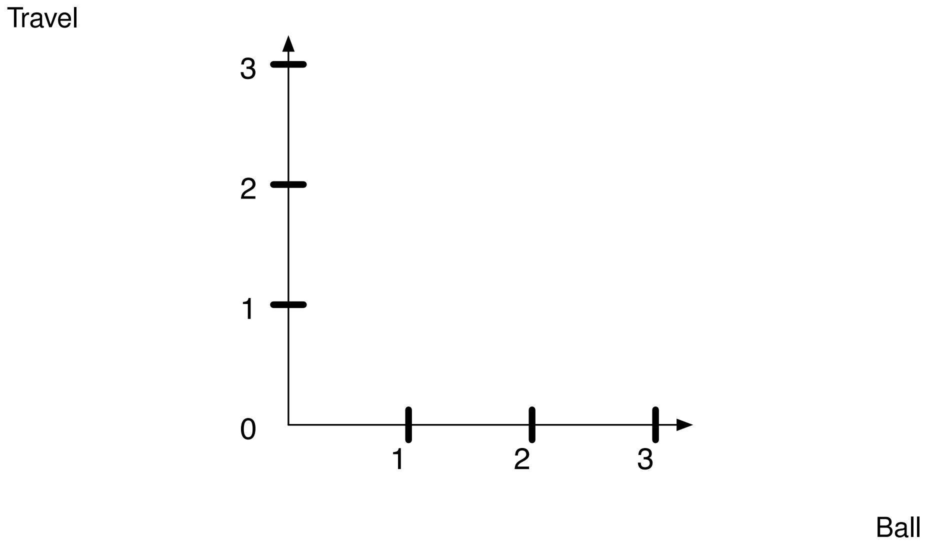
Vacations

Doc_3: {travel, ball, travel}

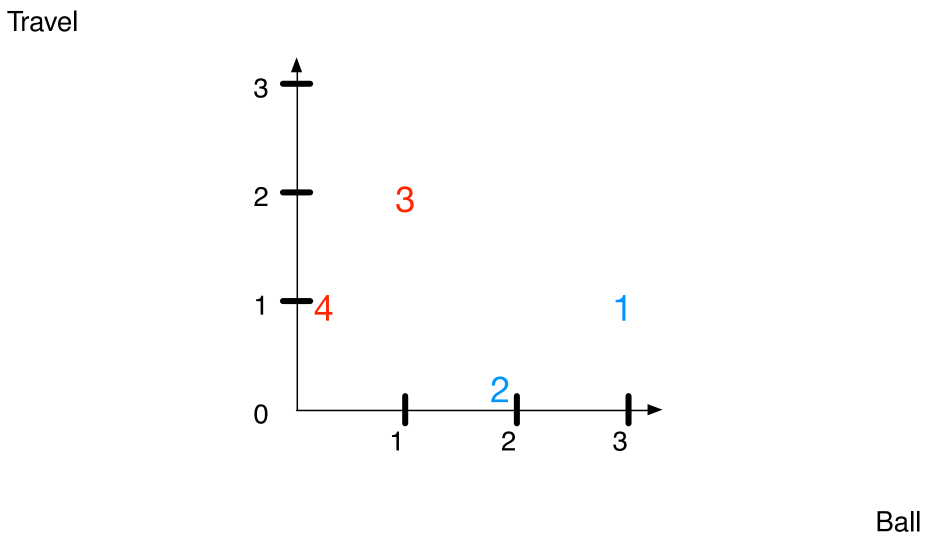
Doc_4: {travel}

- What does this look like in vector space?

Put the documents in vector space



Put the documents in vector space



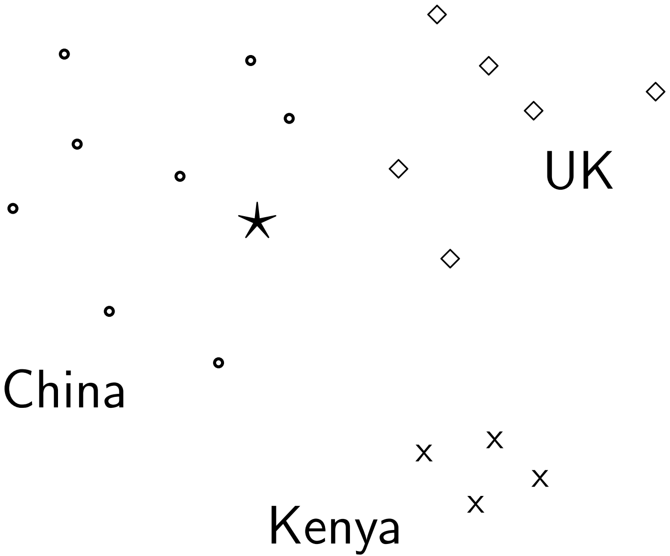
Vector space representation of documents

- Each document is a vector, one component for each term.
- Terms are axes.
- High dimensionality: 10,000s of dimensions and more
- How can we do classification in this space?

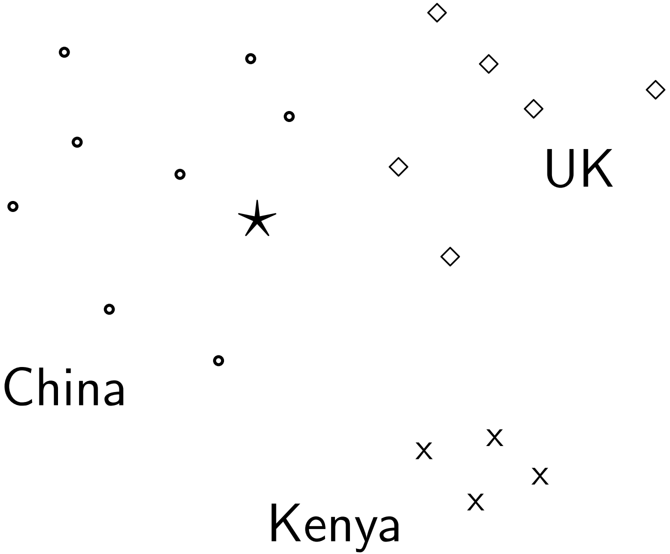
Vector space classification

- As before, the training set is a set of documents, each labeled with its class.
- In vector space classification, this set corresponds to a labeled set of points or vectors in the vector space.
- Premise 1: Documents in the same class form a **contiguous region**.
- Premise 2: Documents from different classes **don't overlap**.
- We define lines, surfaces, hypersurfaces to divide regions.

Classes in the vector space

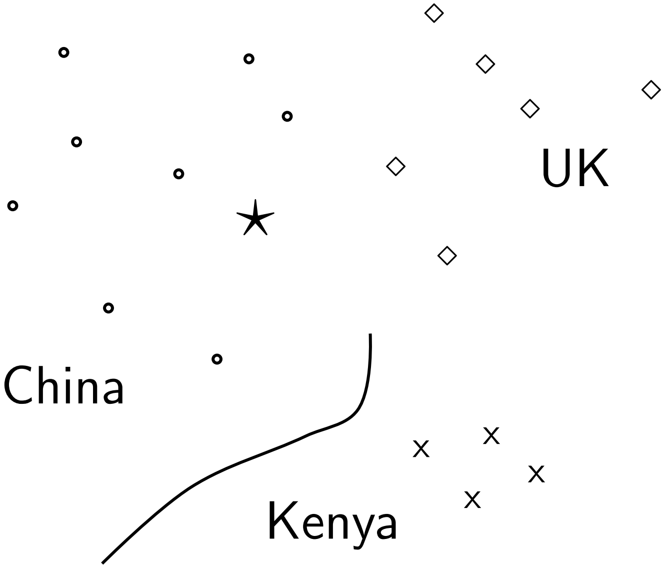


Classes in the vector space



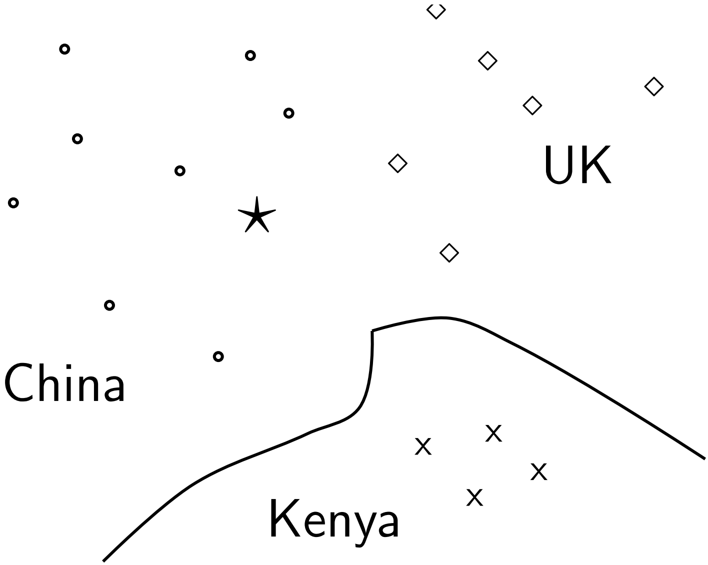
Should the document ⋆ be assigned to `China`, `UK` or `Kenya`?

Classes in the vector space



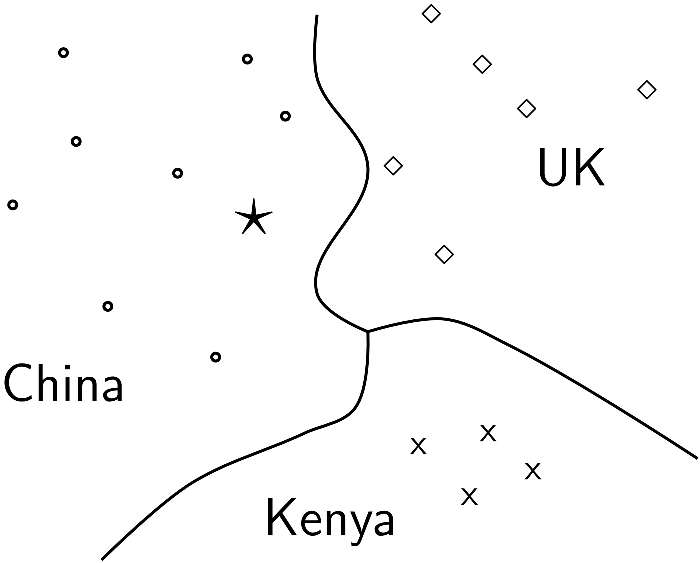
Find separators between the classes

Classes in the vector space



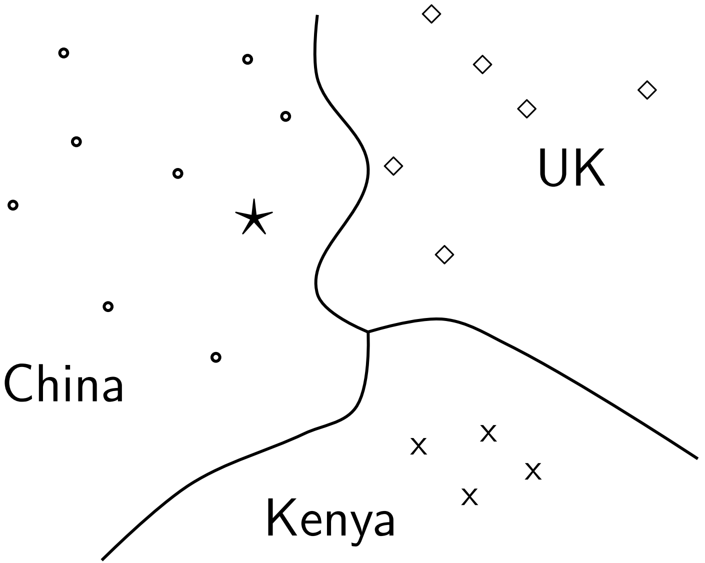
Find separators between the classes

Classes in the vector space



Based on these separators: ⋆ should be assigned to `China`

Classes in the vector space



How do we find separators that do a good job at classifying new documents like $\star$?

Outline

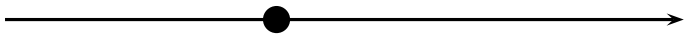
Linear classifiers

- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i \beta_i x_i$ of the feature values.
  - Classification decision: $\sum_i \beta_i x_i > \theta$?
  - ...where θ (the threshold) is a parameter.
- (First, we only consider binary classifiers.)
- Geometrically, this corresponds to a line (2D), a plane (3D) or a hyperplane (higher dimensionalities).
- We call this the **separator** or **decision boundary**.
- We find the separator based on training set.
- Methods for finding separator: logistic regression, naïve Bayes, linear SVM
- Assumption: The classes are **linearly separable**.

Linear classifiers

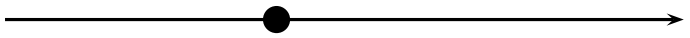
- Definition:
  - A linear classifier computes a linear combination or weighted sum $\sum_i \beta_i x_i$ of the feature values.
  - Classification decision: $\sum_i \beta_i x_i > \theta$?
  - ...where θ (the threshold) is a parameter.
- (First, we only consider binary classifiers.)
- Geometrically, this corresponds to a line (2D), a plane (3D) or a hyperplane (higher dimensionalities).
- We call this the **separator** or **decision boundary**.
- We find the separator based on training set.
- Methods for finding separator: logistic regression, naïve Bayes, linear SVM
- Assumption: The classes are **linearly separable**.
- Before, we just talked about equations. What's the geometric intuition?

A linear classifier in 1D



- A linear classifier in 1D is a point x described by the equation $\beta_1 d_1 = \theta$

A linear classifier in 1D

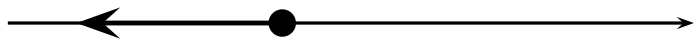


- A linear classifier in 1D is a point x described by the equation $\beta_1 d_1 = \theta$
- $x = \theta / \beta_1$

A linear classifier in 1D

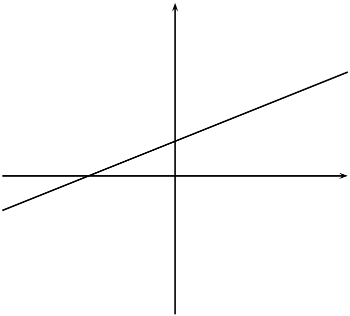
- A linear classifier in 1D is a point x described by the equation $\beta_1 d_1 = \theta$
- $x = \theta / \beta_1$
- Points (d_1) with $\beta_1 d_1 \geq \theta$ are in the class c.

A linear classifier in 1D



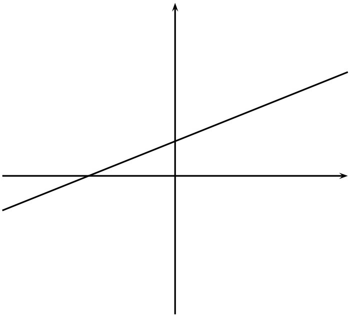
- A linear classifier in 1D is a point x described by the equation $\beta_1 d_1 = \theta$
- $x = \theta / \beta_1$
- Points (d_1) with $\beta_1 d_1 \geq \theta$ are in the class c.
- Points (d_1) with $\beta_1 d_1 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 2D



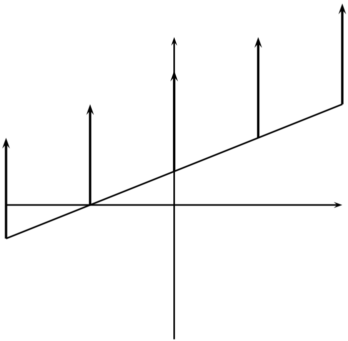
- A linear classifier in 2D is a line described by the equation $\beta_1 d_1 + \beta_2 d_2 = \theta$

A linear classifier in 2D



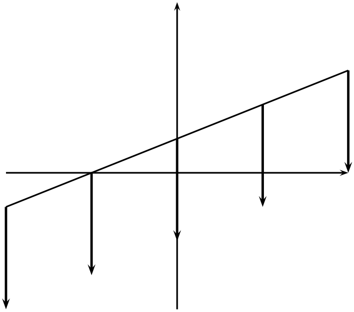
- A linear classifier in 2D is a line described by the equation $\beta_1 d_1 + \beta_2 d_2 = \theta$
- Example for a 2D linear classifier

A linear classifier in 2D



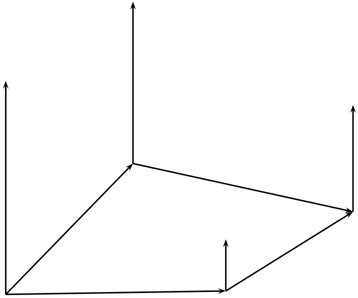
- A linear classifier in 2D is a line described by the equation $\beta_1 d_1 + \beta_2 d_2 = \theta$
- Example for a 2D linear classifier
- Points $(d_1 \ d_2)$ with $\beta_1 d_1 + \beta_2 d_2 \geq \theta$ are in the class c.

A linear classifier in 2D



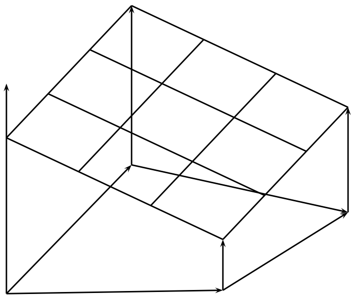
- A linear classifier in 2D is a line described by the equation $\beta_1 d_1 + \beta_2 d_2 = \theta$
- Example for a 2D linear classifier
- Points $(d_1 \; d_2)$ with $\beta_1 d_1 + \beta_2 d_2 \geq \theta$ are in the class c.
- Points $(d_1 \; d_2)$ with $\beta_1 d_1 + \beta_2 d_2 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 3D



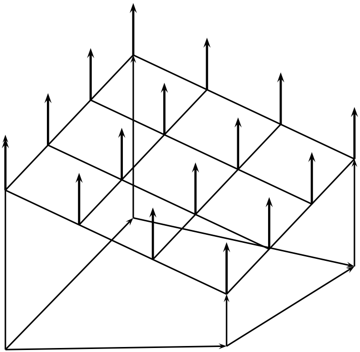
- A linear classifier in 3D is a plane described by the equation $\beta_1 d_1 + \beta_2 d_2 + \beta_3 d_3 = \theta$

A linear classifier in 3D



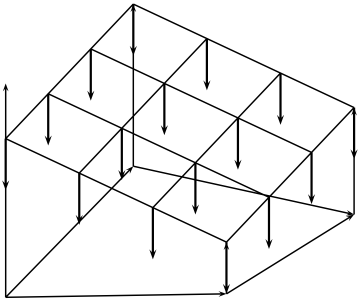
- A linear classifier in 3D is a plane described by the equation $\beta_1 d_1 + \beta_2 d_2 + \beta_3 d_3 = \theta$
- Example for a 3D linear classifier

A linear classifier in 3D



- A linear classifier in 3D is a plane described by the equation $\beta_1 d_1 + \beta_2 d_2 + \beta_3 d_3 = \theta$
- Example for a 3D linear classifier
- Points $(d_1 \; d_2 \; d_3)$ with $\beta_1 d_1 + \beta_2 d_2 + \beta_3 d_3 \geq \theta$ are in the class c.

A linear classifier in 3D



- A linear classifier in 3D is a plane described by the equation $\beta_1 d_1 + \beta_2 d_2 + \beta_3 d_3 = \theta$
- Example for a 3D linear classifier
- Points $(d_1 \; d_2 \; d_3)$ with $\beta_1 d_1 + \beta_2 d_2 + \beta_3 d_3 \geq \theta$ are in the class c.
- Points $(d_1 \; d_2 \; d_3)$ with $\beta_1 d_1 + \beta_2 d_2 + \beta_3 d_3 < \theta$ are in the complement class $\overline{c}$.

Naive Bayes and Logistic Regression as linear classifiers

Multinomial Naive Bayes is a linear classifier (in log space) defined by:

$$\sum_{i=1}^{M} \beta_i d_i = \theta$$

where $\beta_i = \log[\hat{P}(t_i|c)/\hat{P}(t_i|\bar{c})]$, $d_i =$ number of occurrences of t_i in d, and $\theta = -\log[\hat{P}(c)/\hat{P}(\bar{c})]$. Here, the index i, $1 \leq i \leq M$, refers to terms of the vocabulary.

Logistic regression is the same (we only put it into the logistic function to turn it into a probability).

Naive Bayes and Logistic Regression as linear classifiers

Multinomial Naive Bayes is a linear classifier (in log space) defined by:

$$\sum_{i=1}^{M} \beta_i d_i = \theta$$

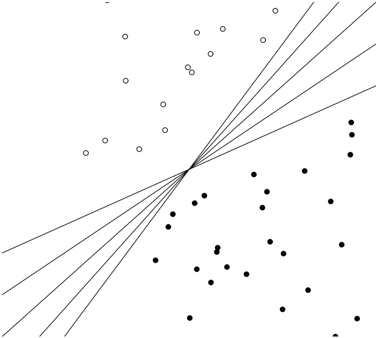
where $\beta_i = \log[\hat{P}(t_i|c)/\hat{P}(t_i|\bar{c})]$, $d_i =$ number of occurrences of t_i in d, and $\theta = -\log[\hat{P}(c)/\hat{P}(\bar{c})]$. Here, the index i, $1 \leq i \leq M$, refers to terms of the vocabulary.

Logistic regression is the same (we only put it into the logistic function to turn it into a probability).

Takeway

Naïve Bayes, logistic regression and SVM (which we'll get to in a second) are all linear methods. They choose their hyperplanes based on different objectives: joint likelihood (NB), conditional likelihood (LR), and the margin (SVM).

Which hyperplane?



Which hyperplane?

- For linearly separable training sets: there are **infinitely** many separating hyperplanes.
- They all separate the training set perfectly . . .
- . . . but they behave differently on test data.
- Error rates on new data are low for some, high for others.
- How do we find a low-error separator?

Outline

Support vector machines

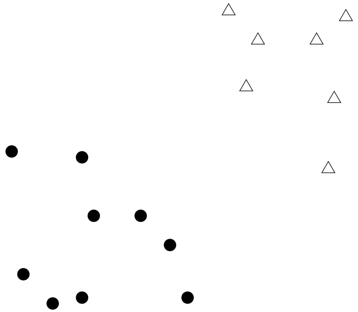
- Machine-learning research in the last two decades has improved classifier effectiveness.
- New generation of state-of-the-art classifiers: support vector machines (SVMs), boosted decision trees, regularized logistic regression, neural networks, and random forests
- Applications to IR problems, particularly text classification

SVMs: A kind of large-margin classifier

Vector space based machine-learning method aiming to find a decision boundary between two classes that is maximally far from any point in the training data (possibly discounting some points as outliers or noise)

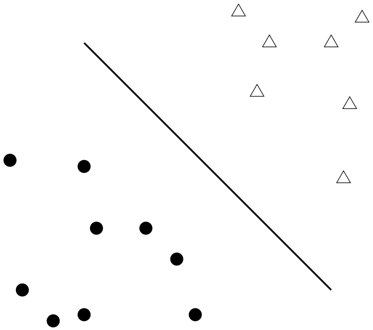
Support Vector Machines

- 2-class training data



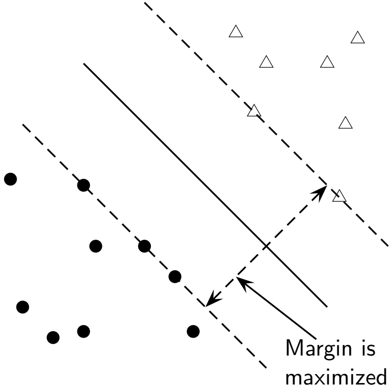
Support Vector Machines

- 2-class training data
- decision boundary → **linear separator**



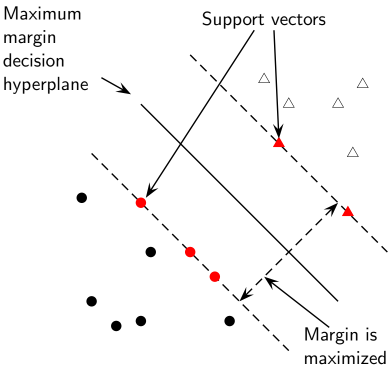
Support Vector Machines

- 2-class training data
- decision boundary ⟶ **linear separator**
- criterion: being maximally far away from any data point ⟶ determines classifier **margin**



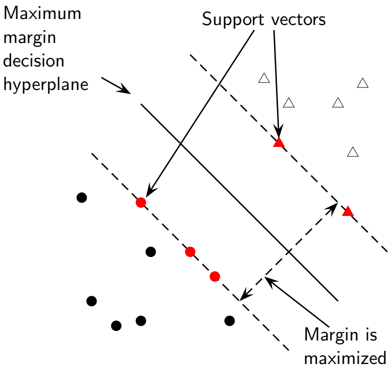
Support Vector Machines

- 2-class training data
- decision boundary ⟶ **linear separator**
- criterion: being maximally far away from any data point ⟶ determines classifier **margin**
- linear separator position defined by **support vectors**



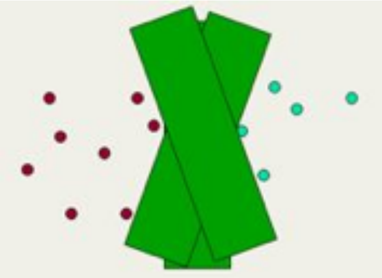
Why maximize the margin?

- Points near decision surface → uncertain classification decisions (50% either way).
- A classifier with a large margin makes no low certainty classification decisions.
- Gives classification safety margin w.r.t slight errors in measurement or documents variation



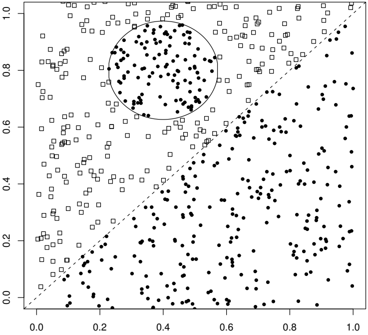
Why maximize the margin?

- SVM classifier: large margin around decision boundary
- compare to decision hyperplane: place fat separator between classes
  - unique solution
- decreased memory capacity
- increased ability to correctly generalize to test data



SVM extensions

- Slack variables: not perfect line
- Kernels: different geometries



- Loss functions: Different penalties for getting the answer wrong

Outline

Classification

- Many commercial applications
- There are many applications of text classification for corporate Intranets, government departments, and Internet publishers.
- Often greater performance gains from exploiting domain-specific features than from changing from one machine learning method to another. (Homework 3)

Choosing what kind of classifier to use

When building a text classifier, first question: **how much training data is there currently available?**

- None?
- Very little?
- A fair amount?
- A huge amount

Choosing what kind of classifier to use

When building a text classifier, first question: **how much training data is there currently available?**

- None? **Hand write rules or use active learning**
- Very little?
- A fair amount?
- A huge amount

Choosing what kind of classifier to use

When building a text classifier, first question: **how much training data is there currently available?**

- None? **Hand write rules or use active learning**
- Very little? **Naïve Bayes**
- A fair amount?
- A huge amount

Choosing what kind of classifier to use

When building a text classifier, first question: **how much training data is there currently available?**

- None? **Hand write rules or use active learning**
- Very little? **Naïve Bayes**
- A fair amount? **SVM**
- A huge amount

Choosing what kind of classifier to use

When building a text classifier, first question: **how much training data is there currently available?**

- None? **Hand write rules or use active learning**
- Very little? **Naïve Bayes**
- A fair amount? **SVM**
- A huge amount **Doesn't matter, use whatever works**

Choosing what kind of classifier to use

When building a text classifier, first question: **how much training data is there currently available?**

- None? **Hand write rules or use active learning**
- Very little? **Naïve Bayes**
- A fair amount? **SVM**
- A huge amount **Doesn't matter, use whatever works**

Interpretable?

Choosing what kind of classifier to use

When building a text classifier, first question: **how much training data is there currently available?**

- None? **Hand write rules or use active learning**
- Very little? **Naïve Bayes**
- A fair amount? **SVM**
- A huge amount **Doesn't matter, use whatever works**

Interpretable? **Decision trees**

Recap

- Is there a learning method that is optimal for all text classification problems?
- No, because there is a tradeoff between bias and variance.
- Factors to take into account:
  - How much training data is available?
  - How simple/complex is the problem? (linear vs. nonlinear decision boundary)
  - How noisy is the problem?
  - How stable is the problem over time?
    - For an unstable problem, it's better to use a simple and robust classifier.
    - You'll be investigating the role of features in HW3!