Stat 406: Algorithms for classification and prediction

Lecture 1: Introduction

Kevin Murphy

Mon 7 January, 2008[1]

[1]Slides last updated on January 7, 2008

Outline

- Administrivia
- Some basic definitions.
- Simple examples of regression.
- Real-world applications of regression.
- Simple examples of classification.
- Real-world applications of classification.

Administrivia

- Web page http://www.cs.ubc.ca/~murphyk/Teaching/Stat406/Spring08/index.html Check the 'news' section before every class!
- Optional Lab Wed 4-5, LSK 310.
- The TA is Aline Tabet
- My office hours are Fri 4-5pm LSK 308d. Please send me email ahead of time if you plan to show up!

Grading

- There will be weekly homework assignments worth 20%.
  Out on Mondays, return on Mondays (in class).
- The homeworks will involve theory and programming; you can get help during the lab session.
- The midterm will be Feb 25th and is worth 35%.
- The final will be in April (15-29) and is worth 45%.

Pre-requisites

- Math: multivariate calculus, linear algebra, probability theory.
- Stats: stats 306 or CS 340 or equivalent.
- CS: some experience with programming (eg in R) is required.

Matlab

- There will be weekly programming assignments.
- We will use Matlab.
- Matlab is very similar to R, but is somewhat faster. Matlab is widely used in the machine learning and Bayesian statistics community.
- Unfortunately Matlab is not free (unlike R). You can buy a copy from the bookstore for $150, or you can use the copy installed in the lab machines.
- In the first lab (this Wed), Aline will give an introduction to Matlab. More info on the class web page.

Textbook

- I am writing my own textbook, but it is not yet finished. You should buy a photocopy of the current draft (433 pages) at Copiesmart in the village (near MacDonalds) for $30 (available on Wednesday).
- The following books are recommended additional reading, but not required
  - *Pattern recognition and machine learning*, Chris Bishop, 2006
  - *Elements of statistical learning*, Hastie, Friedman and Tibshirani, 2001.

Learning objectives

- Understand basic principles of machine learning and its connections to other field
- Derive, in a precise and concise fashion, the relevant mathematical equations needed for familiar and novel models/ algorithms
- Implement, in reasonably efficient Matlab, various familiar and novel ML model/ algorithms
- Know how to choose an appropriate method and apply it to various kinds of data/ problem domains

Syllabus

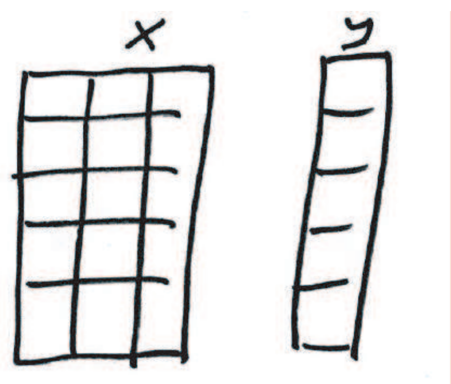
- We will closely follow my book.
- Since people have different backgrounds (cs 340, stat 306, multiple versions), the exact syllabus may change as we go.
- See the web page for details.
- You will get a good feeling for the class during today's lecture.

Outline

- Administrivia $\surd$
- Machine learning: some basic definitions.
- Simple examples of regression.
- Real-world applications of regression.
- Simple examples of classification.
- Real-world applications of classification.

Learning to predict

- This class is basically about machine learning.
- We will initially focus on supervised approaches.
- Given a training set of n input-output pairs $D = (\vec{x}_i, \vec{y}_i)_{i=1}^n$, we attempt to construct a function f which will accurately predict $f(\vec{x}_*)$ on future, test examples $\vec{x}_*$.
- Each input $\vec{x}_i$ is a vector of d features or covariates. Each output $\vec{y}_i$ is a target variable. The training data is stored in an $n \times d$ design matrix $X = [\vec{x}_i^T]$. The training outputs are stored in a $n \times q$ matrix $Y = [\vec{y}_i^T]$.

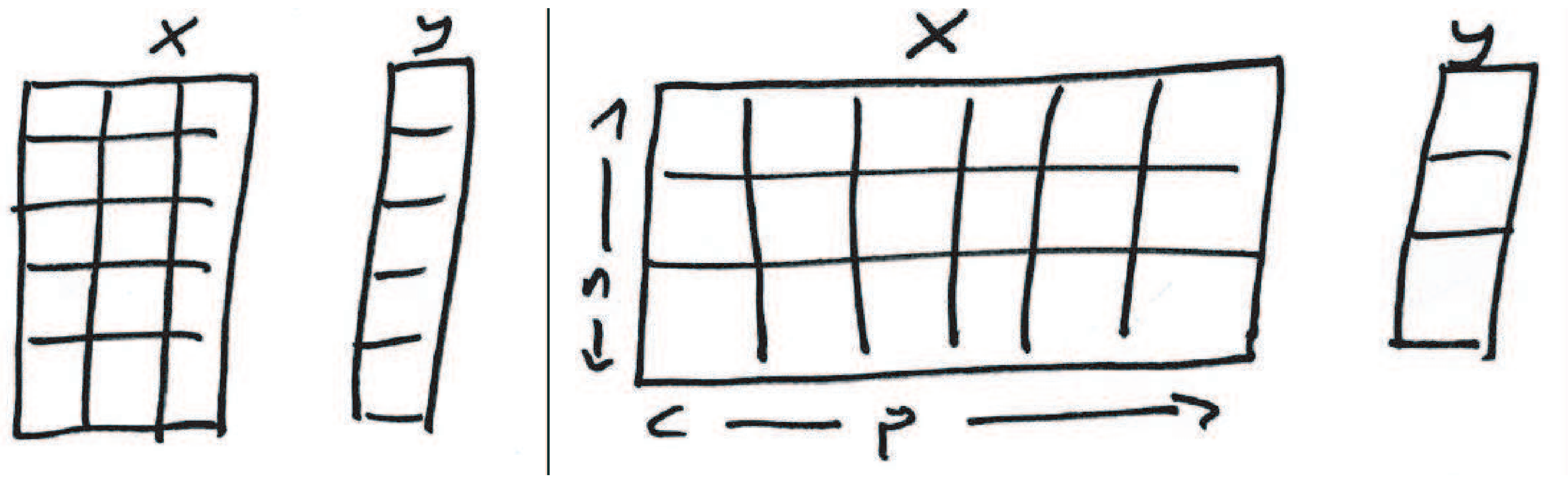


Classification vs regression

- If $\vec{y} \in \mathbb{R}^q$ is a continuous-valued output, this is called **regression**. Often we will assume $q = 1$, i.e., scalar output.
- If $y \in \{1, \ldots, C\}$ is a discrete label, this is called **classification** or **pattern recognition**. The labels can be ordered (eg. low, medium, high) or unordered (e.g., male, female). N_Y is the number of classes. If $C = 2$, this is called binary (dichotomous) classification.

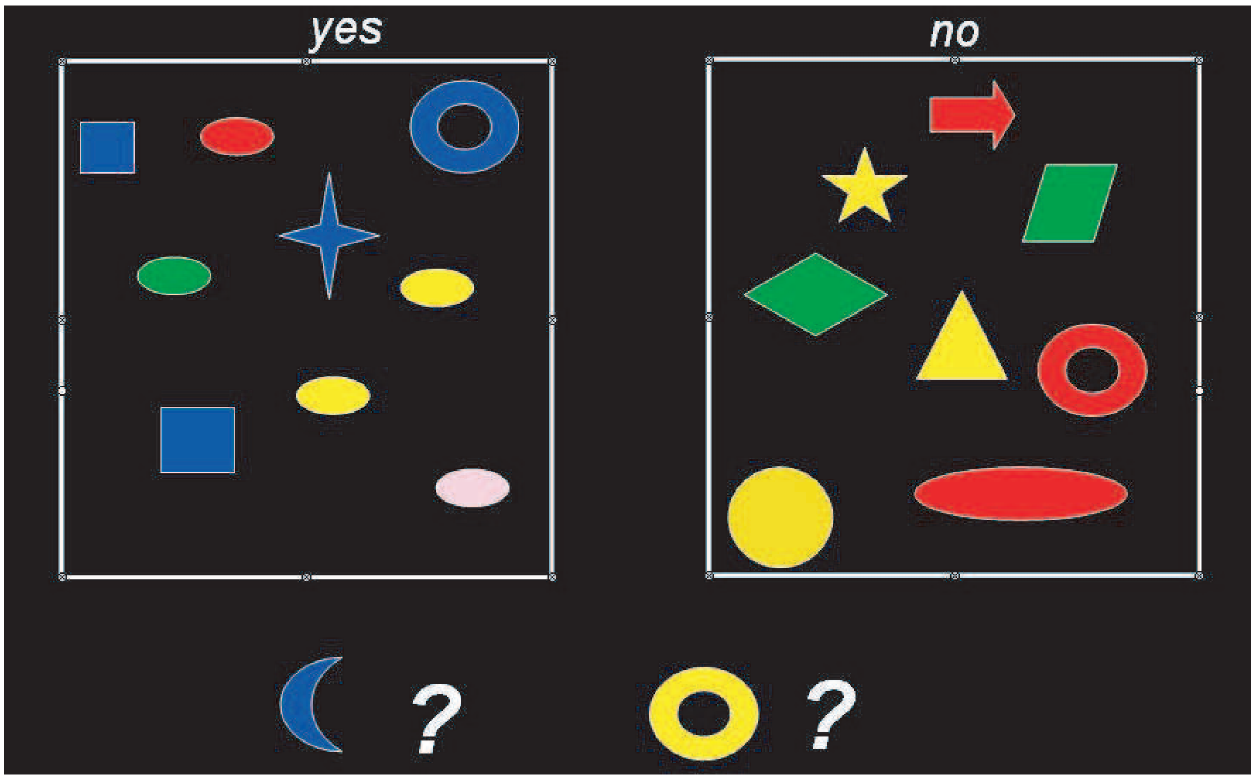
Short/fat vs tall/skinny data

- In traditional applications, the design matrix is tall and skinny ($n \gg p$), i.e., there are many more training examples than inputs.
- In more recent applications (eg. bio-informatics or text analysis), the design matrix is short and fat ($n \ll p$), so we will need to perform feature selection and/or dimensionality reduction.



Generalization performance

We care about performance on examples that are different from the training examples (so we can't just look up the answer).



No free lunch theorem

- The *no free lunch theorem* says (roughly) that there is no single method that is better at predicting across all possible data sets than any other method.
- Different learning algorithms implicitly make different assumptions about the nature of the data, and if they work well, it is because the assumptions are reasonable in a particular domain.

Supervised vs unsupervised learning

- In supervised learning, we are given $(\vec{x}_i, \vec{y}_i)$ pairs and try to learn how to predict $\vec{y}_*$ given $\vec{x}_*$.
- In unsupervised learning, we are just given $\vec{x}_i$ vectors.
- The goal in unsupervised learning is to learn a model that "explains" the data well. There are two main kinds:
  - Dimensionality reduction (eg PCA)
  - Clustering (eg K-means)

Outline

- Administrivia $\surd$
- Machine learning: some basic definitions. $\surd$
- Simple examples of regression.
- Real-world applications of regression.
- Simple examples of classification.
- Real-world applications of classification.

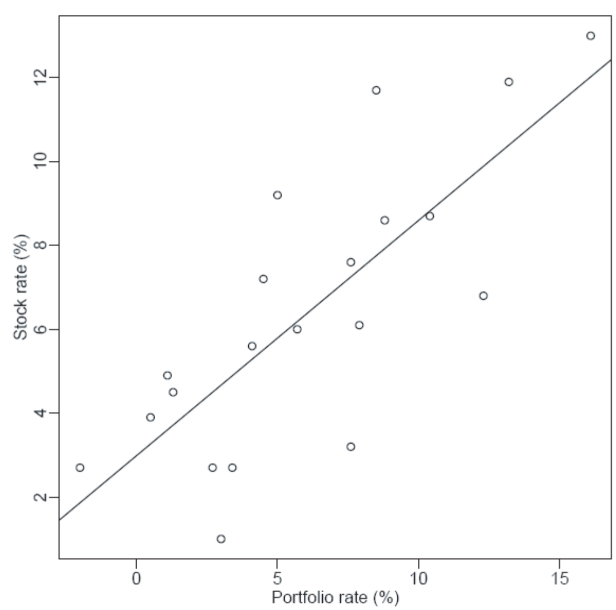
Linear regression

The output density is a 1D Gaussian (Normal) conditional on x:

$$p(y|\vec{x}) = \mathcal{N}(y; \vec{\beta}^T \vec{x}, \sigma) = \mathcal{N}(y; \beta_0 + \beta_1 x_1 + \cdots \beta_p x_p, \sigma)$$

$$\mathcal{N}(y|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{1}{2\sigma^2}(y-\mu)^T(y-\mu))$$

For example, $y = ax_1 + b$ is represented as $\vec{x} = (1, x_1)$ and $\vec{\beta} = (b, a)$.



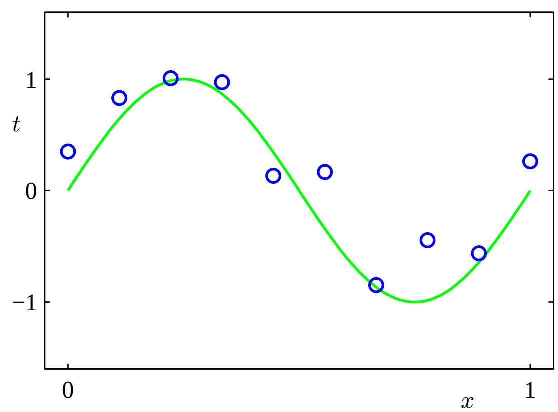
Polynomial regression

If we use linear regression with non-linear basis functions

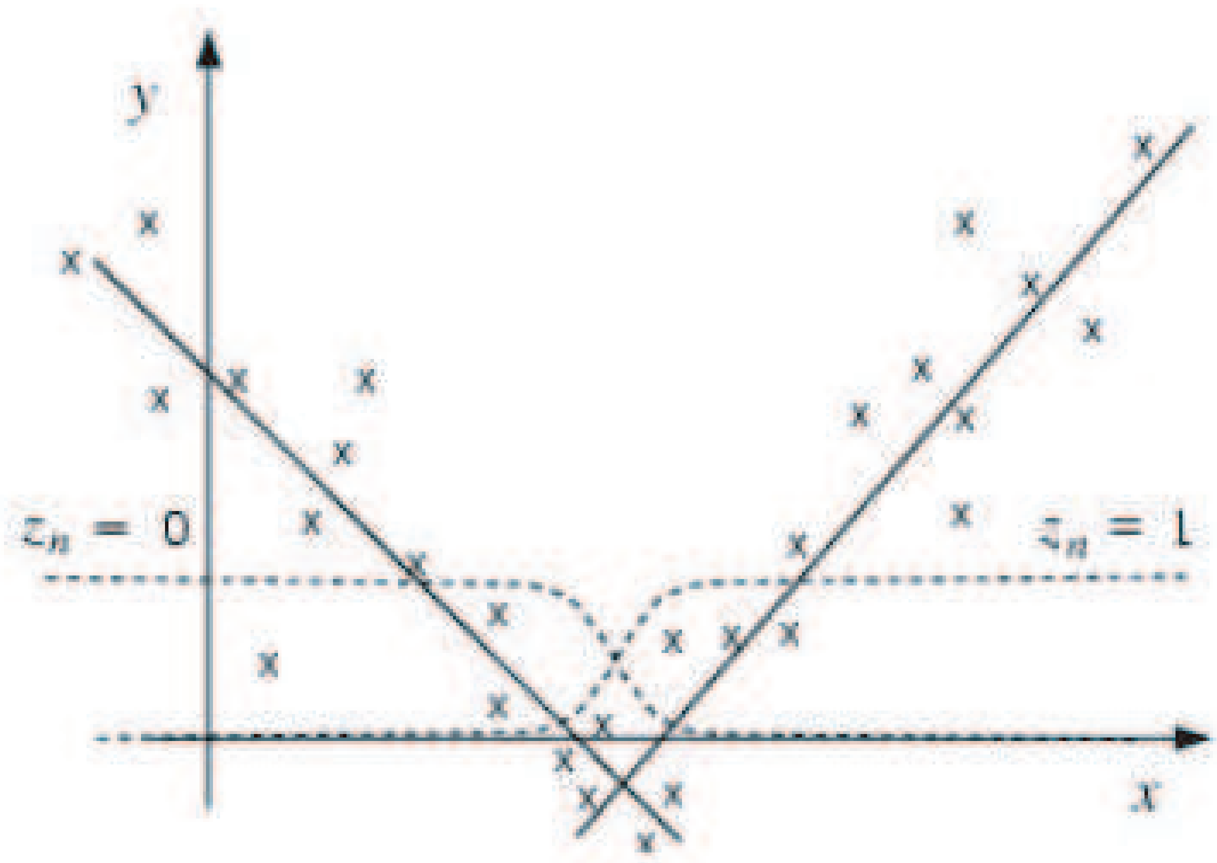
$$p(y|x_1) = \mathcal{N}(y|\beta^T\ [1, x_1, x_1^2, \ldots, x_1^k], \sigma)$$

we can produce curves like the one below.

Note: In this class, we will often use $\vec{w}$ instead of $\vec{\beta}$ to denote the weight vector.



Piecewise linear regression



How many pieces? — Model selection problem.
Where to put them? — Segmentation problem.

2D linear regression

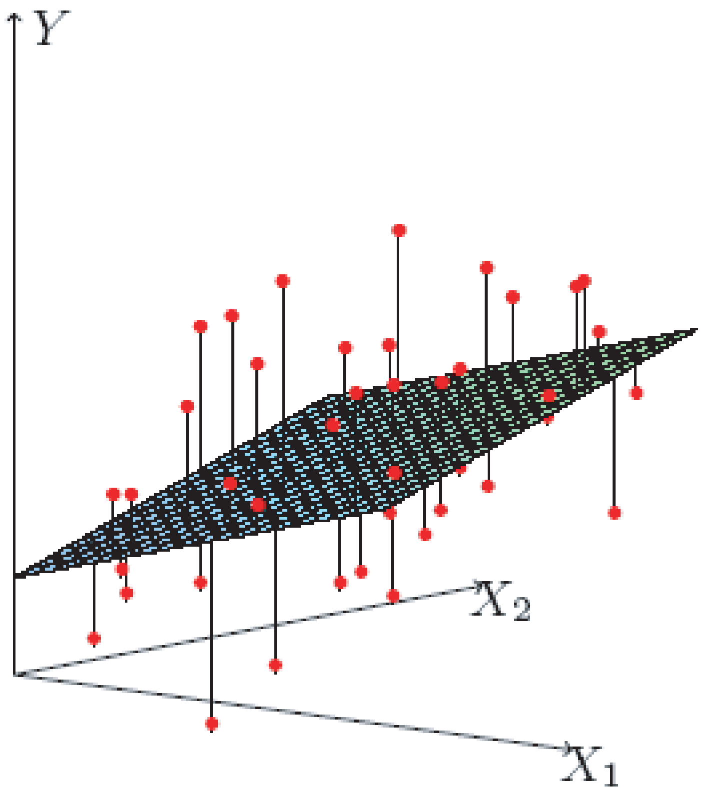


Figure 3.1: *Linear least squares fitting with* $X \in \mathbb{R}^2$. *We seek the linear function of* X *that minimizes the sum of squared residuals from* Y.

Piecewise linear 2D regression

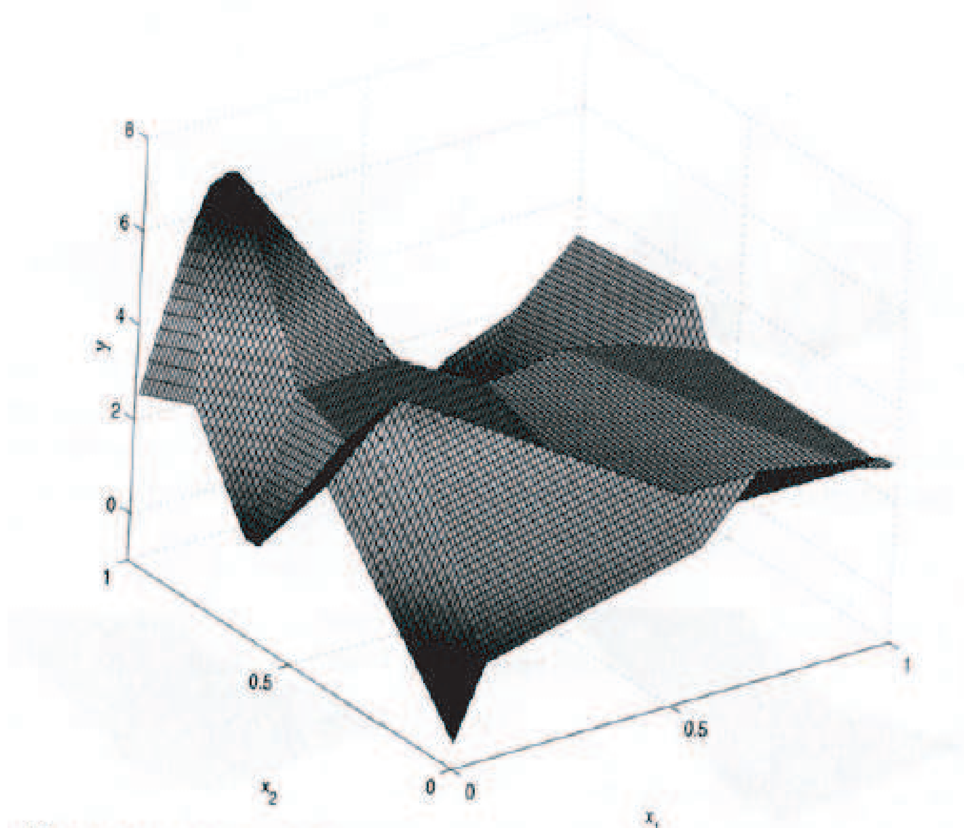


Figure 4.7 An example of a single realisation from a piecewise linear surface in two dimensions. (Reproduced by permission of the Royal Statistical Society.)

How many pieces? — Model selection problem.
Where to put them? — Segmentation problem.

Outline

- Administrivia $\surd$
- Machine learning: some basic definitions. $\surd$
- Simple examples of regression. $\surd$
- Real-world applications of regression.
- Simple examples of classification.
- Real-world applications of classification.

Real-world applications of regression

- $\vec{x}$ = amount of various chemicals in my factory, y = amount of product produced.
- $\vec{x}$ = properties of a house (eg location, size), y = sales price.
- $\vec{x}$ = joint angles of my robot arm, $\vec{y}$ = location of arm in 3-space.
- $\vec{x}$ = stock prices today, $\vec{y}$ = stock prices tomorrow. (Time series data is not iid, and is beyond the scope of this course.)

Collaborative filtering

- A very interesting **ordinal regression** problem is to build a system that can predict what ranking (from 1 to 5) you would give to a new movie.
- The input might just be the name of the movie, plus your past voting patterns, and those of other users.
- The **collaborative filtering** approach says you will give the same score as those people who have similar movie tastes to you, which can you infer by looking at past voting patterns.
- For each movie and each user, you can infer a set of **latent traits** and use these to predict (related to SVD of a matrix).

Netflix prize

- The netflix prize `http://netflixprize.com/` is an award of $1M USD for a system that can predict your movie preferences 10% more accurately than their current system (called Cinematch).
- A large training data set is provided: a sparse 18k $\times$ 480k matrix (movies $\times$ users) containing about 100M rankings (on the scale 1:5) of various movies.
- The test (probe) set is 2.8M (movie,user) pairs, for which the ranking is known but withheld from the training set.
- The performance measure is root mean square error:

$$rmse = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(R(u_i, m_i) - \hat{R}(u_i, m_i))^2} \tag{1}$$

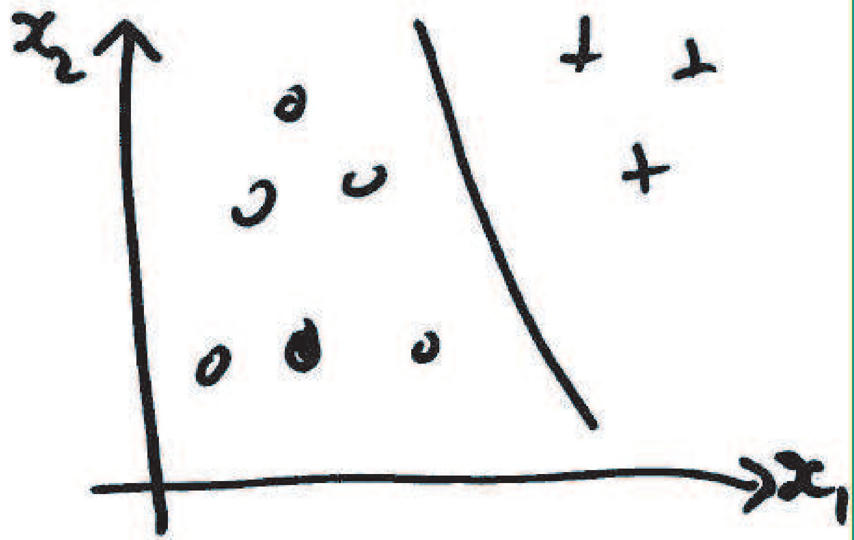
where $R(u_i, m_i)$ is the true rating of user u_i on movie m_i, and $\hat{R}(u_i, m_i)$ is the prediction.

Outline

- Administrivia $\surd$
- Machine learning: some basic definitions. $\surd$
- Simple examples of regression. $\surd$
- Real-world applications of regression. $\surd$
- Simple examples of classification.
- Real-world applications of classification.

LINEARLY SEPARABLE 2D DATA

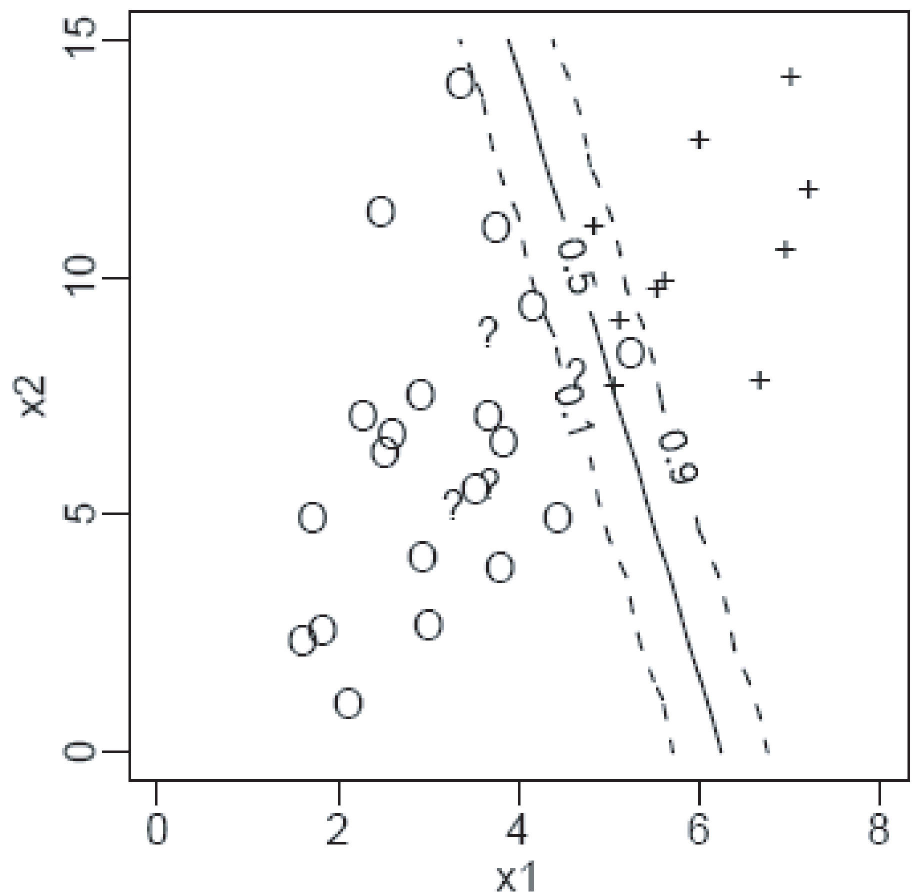
2D inputs $\vec{x}_i \in \mathbb{R}^2$, binary outputs $y \in \{0, 1\}$.
The line is called a *decision boundary*.
Points to the right are classified as $y = 1$, points to the left as $y = 0$.



Logistic regression

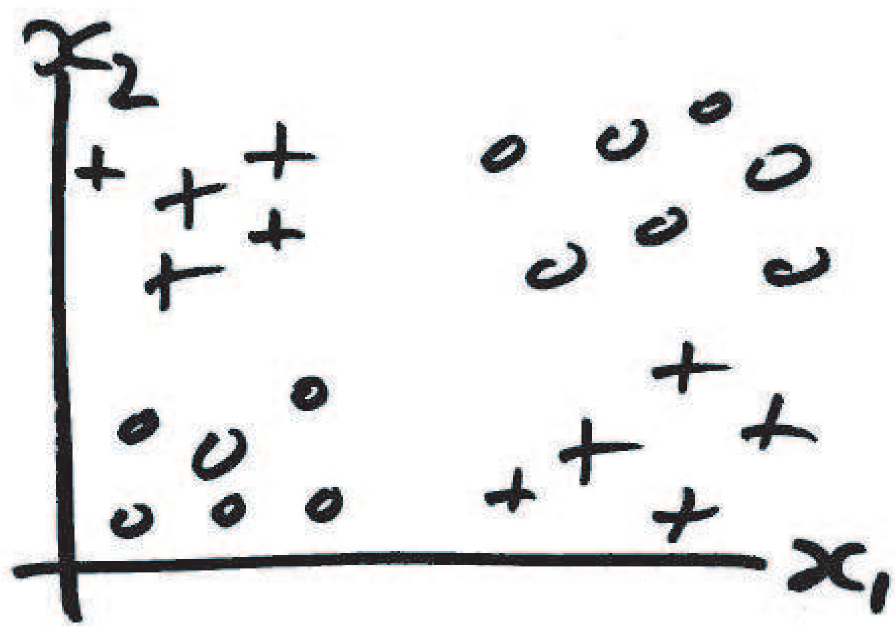
- A simple approach to binary classification is logistic regression (briefly studied in 306).
- The output density is Bernoulli conditional on x:
$$p(y|x) = \pi(x)^y \, (1 - \pi(x))^{1-y}$$
where $y \in \{0, 1\}$ and
$$\pi(x) = \sigma(\vec{w}^T \, [1, x_1, x_2])$$
where
$$\sigma(u) = \frac{1}{1 + e^{-u}}$$
is the sigmoid (logistic) function that maps $\mathbb{R}$ to $[0, 1]$. Hence
$$P(Y = 1|\vec{x}) = \frac{1}{1 + e^{-w_0 + w_1 x_1 + w_2 x_2}}$$
where w_0 is the bias (offset) term corresponding to the dummy column of 1s added to the design matrix.

2D Logistic regression



Non-linearly separable 2D data

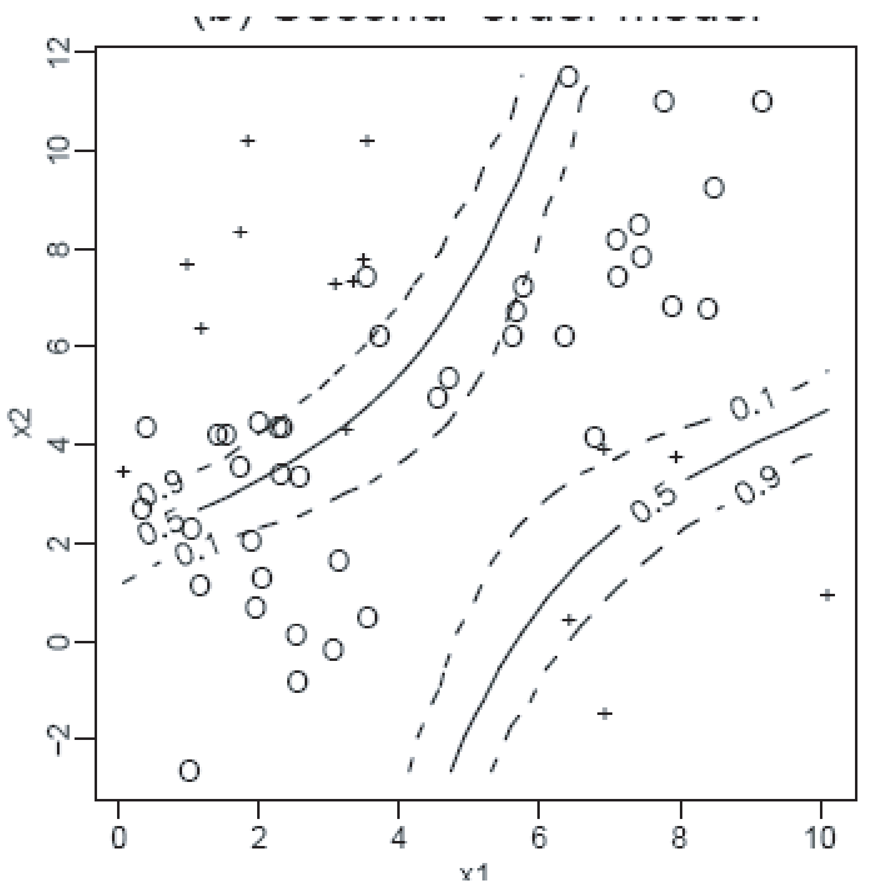
In 306, this is called “checkerboard” data.
In machine learning, this is called the “xor” problem.
The “true” function is $y = x_1 \oplus x_2$.
The decision boundary is non-linear.



Logistic regression with quadratic features

We can separate the classes using

$$P(Y = 1|x_1, x_2) = \sigma(w^T\ [1, x_1, x_2, x_1^2, x_2^2, x_1, x_2])$$



Outline

- Administrivia $\surd$
- Machine learning: some basic definitions. $\surd$
- Simple examples of regression. $\surd$
- Real-world applications of regression. $\surd$
- Simple examples of classification. $\surd$
- Real-world applications of classification.

Handwritten digit recognition

Multi-class classification.

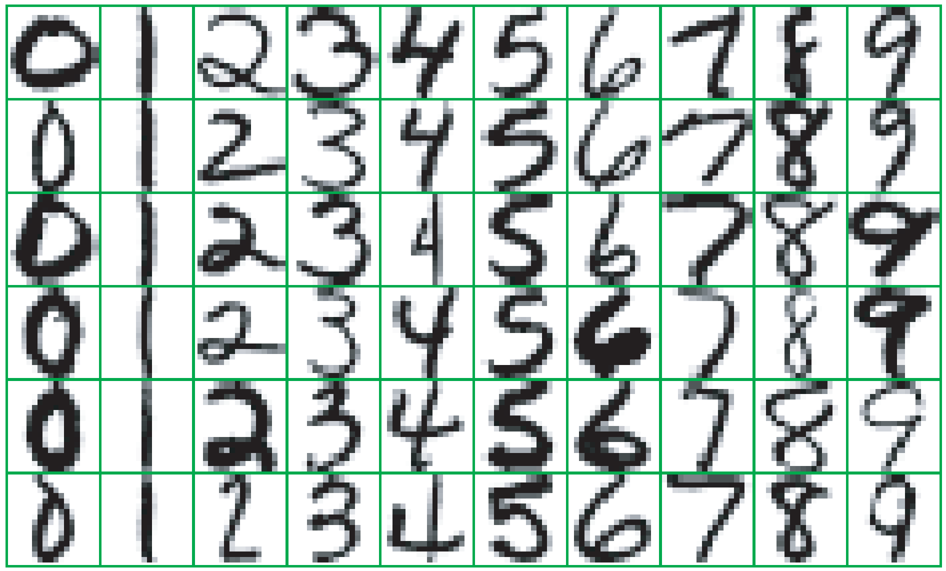
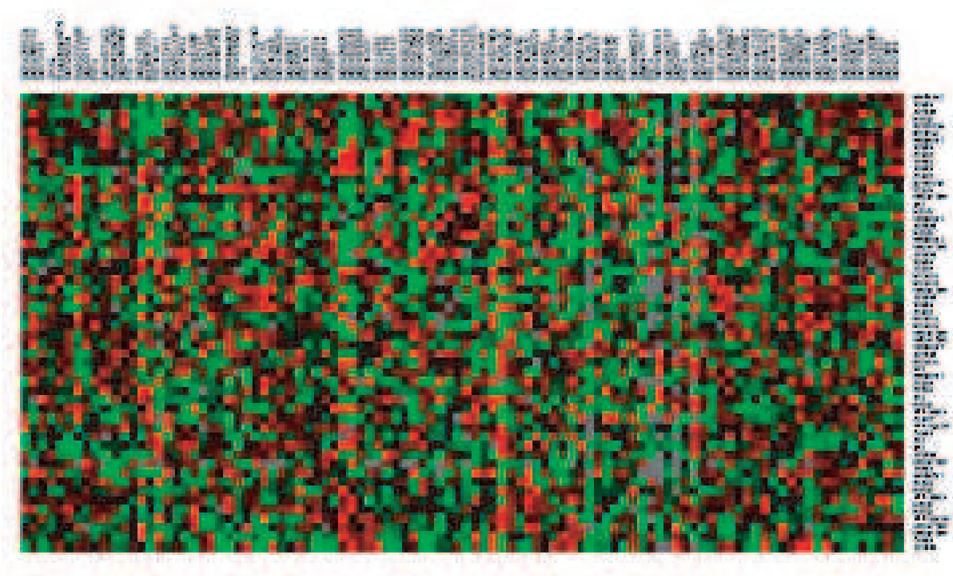


Figure 1.2: *Examples of handwritten digits from U.S. postal envelopes.*

Gene microarray expression data

Rows = examples, columns = features (genes).
Short, fat data ($p \gg n$).
Might need to perform feature selection.



Other examples of classification

- Email spam filtering (spam vs not spam)
- Detecting credit card fraud (fraudulent or legitimate)
- Face detection in images (face or background)
- Web page classification (sports vs politics vs entertainment etc)
- Steering an autonomous car across the US (turn left, right, or go straight)