# Simple Sensor Models

Noel Welsh

14 October 2010

# Outline

- Swipe card access

- In class assessment

# Recap

Last lecture we looked at a the maximum likelihood principle, which states that if we have to choose between a set of hypotheses we should choose the one that makes the data most likely. We then came up with some models to use for classifying colours (RGB values) and briefly discussed generative and discriminative models.

Today we're going to look at some models from the literature that are appropriate for use with the simple sensors found in your robotics kit.

# Models

- Point plus noise
- Naive Bayes
- Linear Regression
- Logisitic Regression
- All *linear*

# Linear is Not Linear

- All the models we're looking at are *linear*.
- Linearity does *not* mean they model lines.
- Linearity means *superposition* holds
  - $f(a+b) = f(a) + f(b)$
- Linear models are fast to learn and use.

# Classification and Regression

- We're looking at classification and regression models.
- Classifiers choose one of a discrete set of choices. E.g. "Is this a red can?"
- Regressers predict a continuous output from a set of input variables. E.g. "How far away is this object given these sensor readings?"

# Point Plus Noise

- The data is described by a point plus noise centered at that point.
- E.g. $c + \mathcal{N}(0, 1)$. A point $c$ plus normally distributed noise.
- All the models we consider today are extensions of this model.

# Point Plus Noise Classifier

- If we have multiple classes we can learn a point plus noise model for each class.
- This is the model a lot of you came up with for the colour classifier last lecture.
- Classification is given by which class has the highest probability of generating the data (i.e. maximum likelihood again).

# Naive Bayes

- Extends the point plus noise model to multiple attributes.
- For example, combining two sensor readings.
- We want to compute $P(C|A_1, A_2)$ where $C$ is the class and $A_1$ and $A_2$ are the attributes (sensor readings).
- Apply Bayes theorem

$$P(C|A_1, A_2) \propto P(A_1, A_2|C)P(C)$$

- *Assume* attributes are independent. $P(A_1, A_2|C)$ factors into $P(A_1|C)P(A_2|C)$.

# Terminology

- In regression we attempt to predict a *dependent variable* from the value of one or more *independent variables*.
- We assume we have an equation that describes the relationship.
- This equation has several *parameters* we can change.
- In linear regression the equation must be linear in the parameters.
- Example: $y = \beta e^x$ is linear in the parameter $\beta$ but not in the independent variable $x$.

# Linear Regression

- Instead of a point plus noise we have a line (or any other function that is linear in the parameters) plus noise.
- The standard model assumes normally distributed noise.
- Simplest model $y = \alpha + \beta x$.

# Learning a Linear Regression Model

- The method of *least squares* is typically used.
- This sets the parameters to minimise the sum of the squared error between the observations and the predicted value.
- Minimising squared error implies normally distributed noise.
- Can solve through simple calculus (take the derivative of the error, set to zero, and solve).
- For $y = \alpha + \beta x$ the solution is $\hat{\beta} = \frac{Cov(x,y)}{Var(x)}$ and $\hat{\alpha} = Mean(y) + \hat{\beta} Mean(x)$.

# Assumptions and Assessment

- Form of the equation is known.
- Noise is constant with zero mean.
- Number of observations greater than number of parameters.
- The sum of squared error (which is what we minimise) provides a simple way to estimate the quality of fit.

# Logistic Regression

- A form of regression suitable for estimating probabilities.
- Model has form $y = f(\beta_0 + \beta_1 x_1 + \ldots)$.
- $f$ is the logistic function $f(x) = \frac{1}{1+e^{-x}}$.
- Output (dependent variable) is always between 0 and 1. Training data should be binary.
- Thus can use as a discrminiative binary classifier giving probability of observing a 1.

# The Logistic Function

- The logistic function is the inverse of the logit function.
- The logit function $logit(p) = \frac{p}{1-p}$ is the log odds of a binary random variable.
- So logit maps probability to odds and thus logisitic maps odds to probabilities.

# Training Logistic Regression

- No closed form solution.
- Gradient ascent is simple to implement and commonly used.

# Notes

- We haven't looked at a large and important area of linear discriminative classifiers.
- Many software packages (e.g. spreadsheets) will do linear regression. All others are easy to code, though you might have a bit of trouble with logistic regression if you aren't familiar with calculus. Ask for help if you need it.
- There is a lot of active work on these models for large scale data mining. E.g. random naive Bayes, regularised logisitc regression.