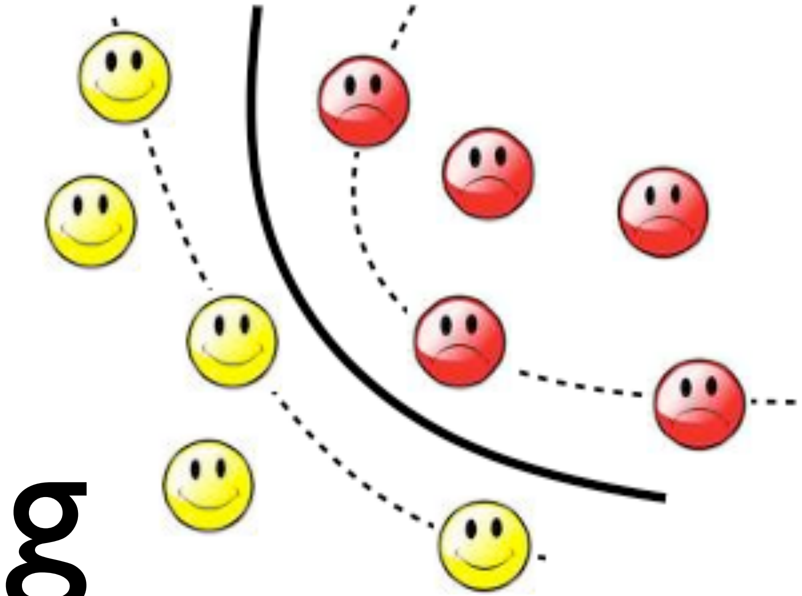
THE UNIVERSITY OF SYDNEY



Machine Learning and Data Mining

(COMP 5318)

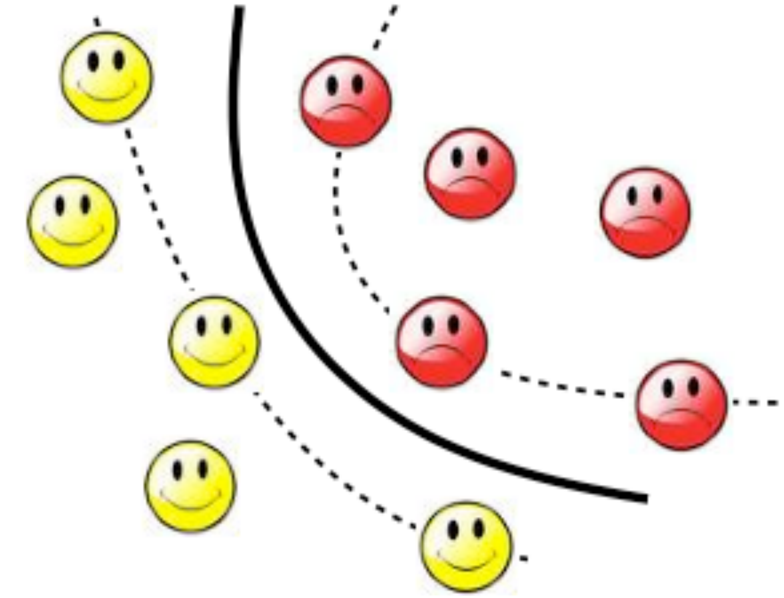
Logistic Regression

Nguyen Hoang Tran

Announcements

THE UNIVERSITY OF SYDNEY

- This lecture is based on:
  - Murphy's book: Chapters 8, 14
  - Ullman's book: Chapter 12



Quick Review

Supervised learning

THE UNIVERSITY OF SYDNEY

- Learn a mapping function *f* from $\mathbf{x}$ to y

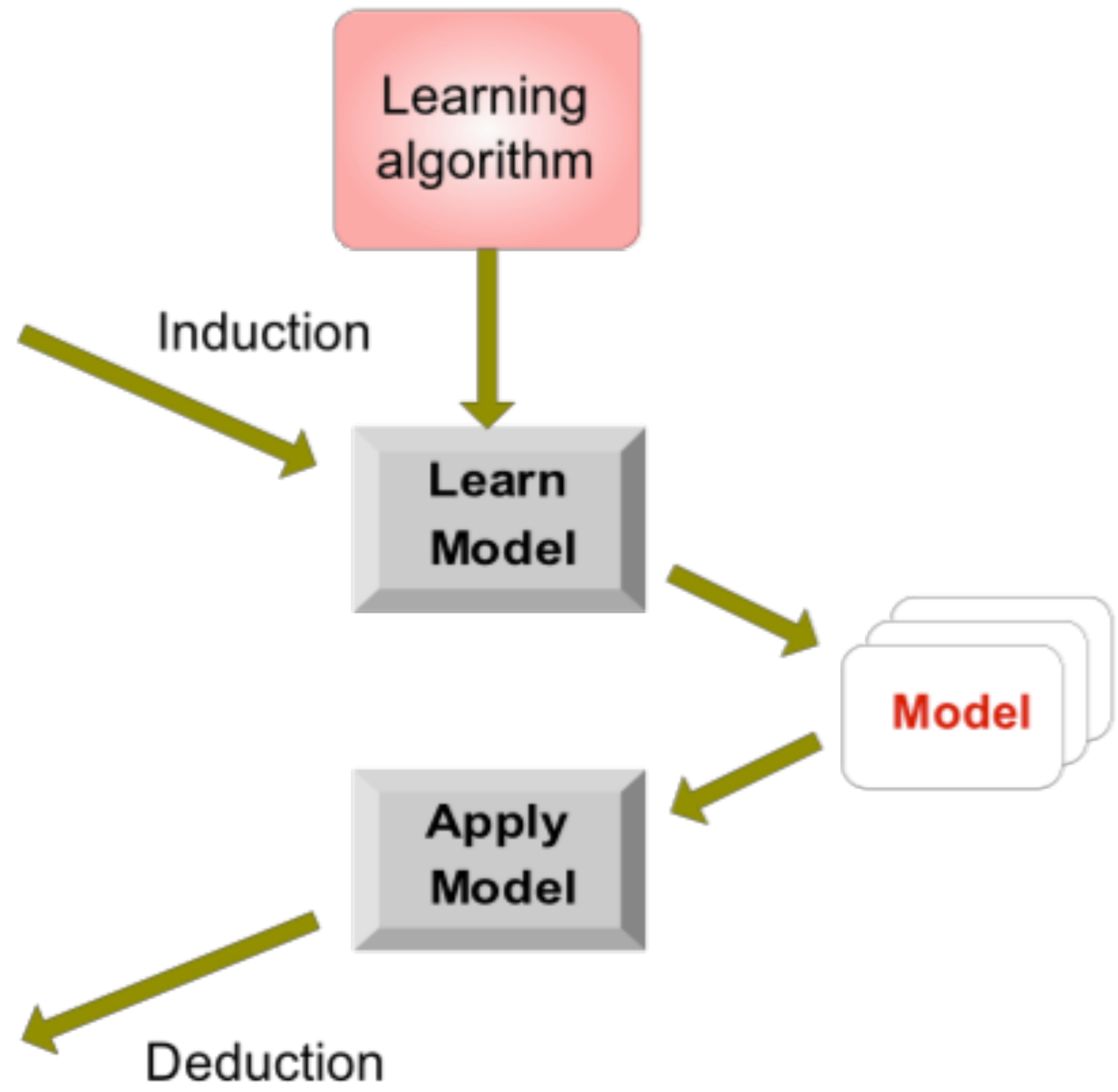
$$y = f(\mathbf{x})$$

- If $y \in \{1, 2, \ldots, C\}$ the problem is called classification
- If $y \in \mathbb{R}$ the problem is called regression

Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

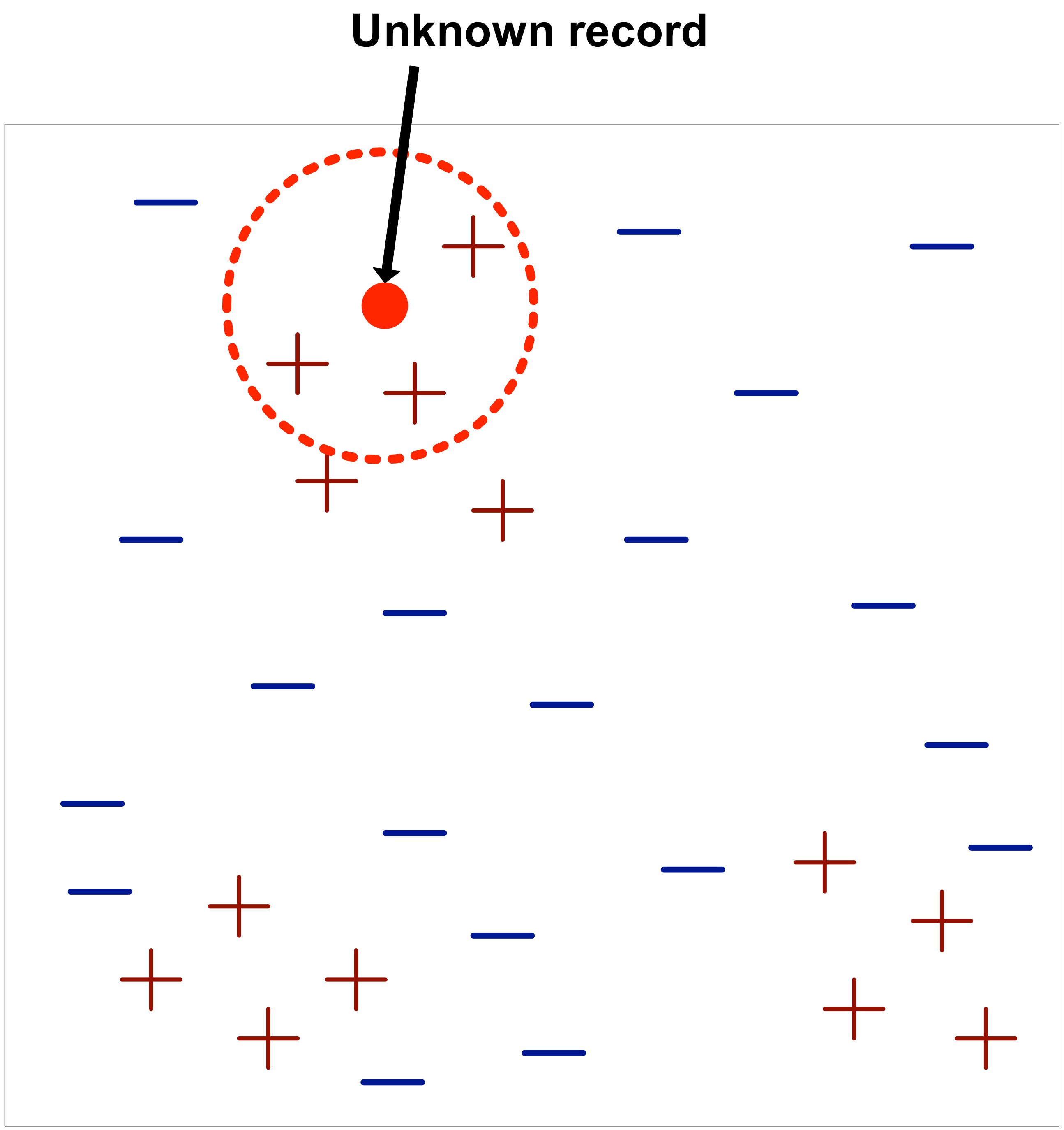
Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

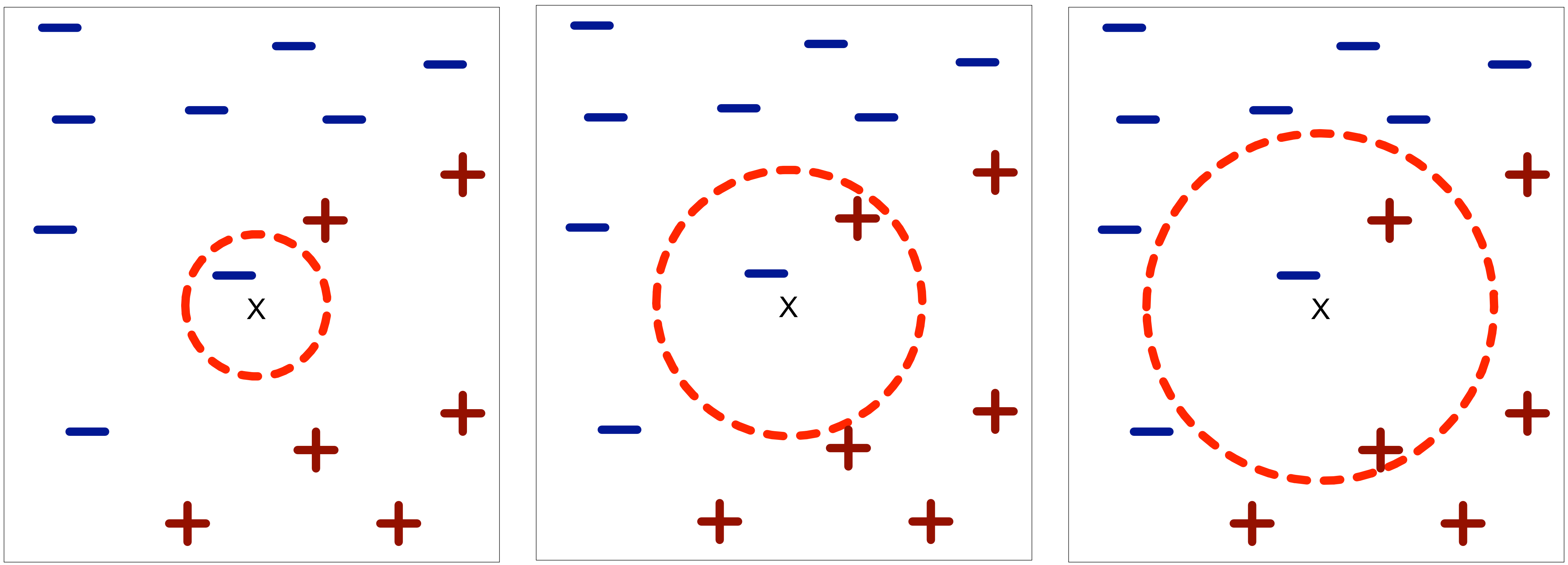
Nearest Neighbour Classifiers



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbours to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbours
  - Use class labels of nearest neighbours to determine the class label of unknown record (e.g., by taking majority vote)

Definition of Nearest Neighbour



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbours of a record x are data points that have the k smallest distance to x

Bayesian Classifiers

- Approach:
  - compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes' theorem

$$\boxed{P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}}$$

  - Choose value of C that maximises
    $P(C \mid A_1, A_2, \ldots, A_n)$
  - Equivalent to choosing value of C that maximises
    $P(A_1, A_2, \ldots, A_n \mid C)\, P(C)$
- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n | C) = P(A_1| C_j)\, P(A_2| C_j) \ldots P(A_n| C_j)$
  - Can estimate $P(A_i| C_j)$ for all A_i and C_j.
  - New point is classified to C_j if $P(C_j) \prod P(A_i| C_j)$ is maximal.

How to Estimate Probabilities from Data?

THE UNIVERSITY OF SYDNEY

#	Refund	Status	Salary	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Class: $P(C) = N_c/N$
  - e.g., $P(No) = 7/10$, $P(Yes) = 3/10$
- For discrete attributes:

$$P(A_i \mid C_k) = |A_{ik}|/ N_{c^k}$$

  - where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k
  - Examples:
    $P(Status=Married|No) = 4/7$
    $P(Refund=Yes|Yes) = 0$

Example of Naïve Bayes Classifier

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

A: attributes

M: mammals

N: non-mammals

$$P(A \mid M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A \mid N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A \mid M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A \mid N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

$$P(A|M)P(M) > P(A|N)P(N)$$

=> Mammals

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

	PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Cost-Sensitive Measures

THE UNIVERSITY OF SYDNEY

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

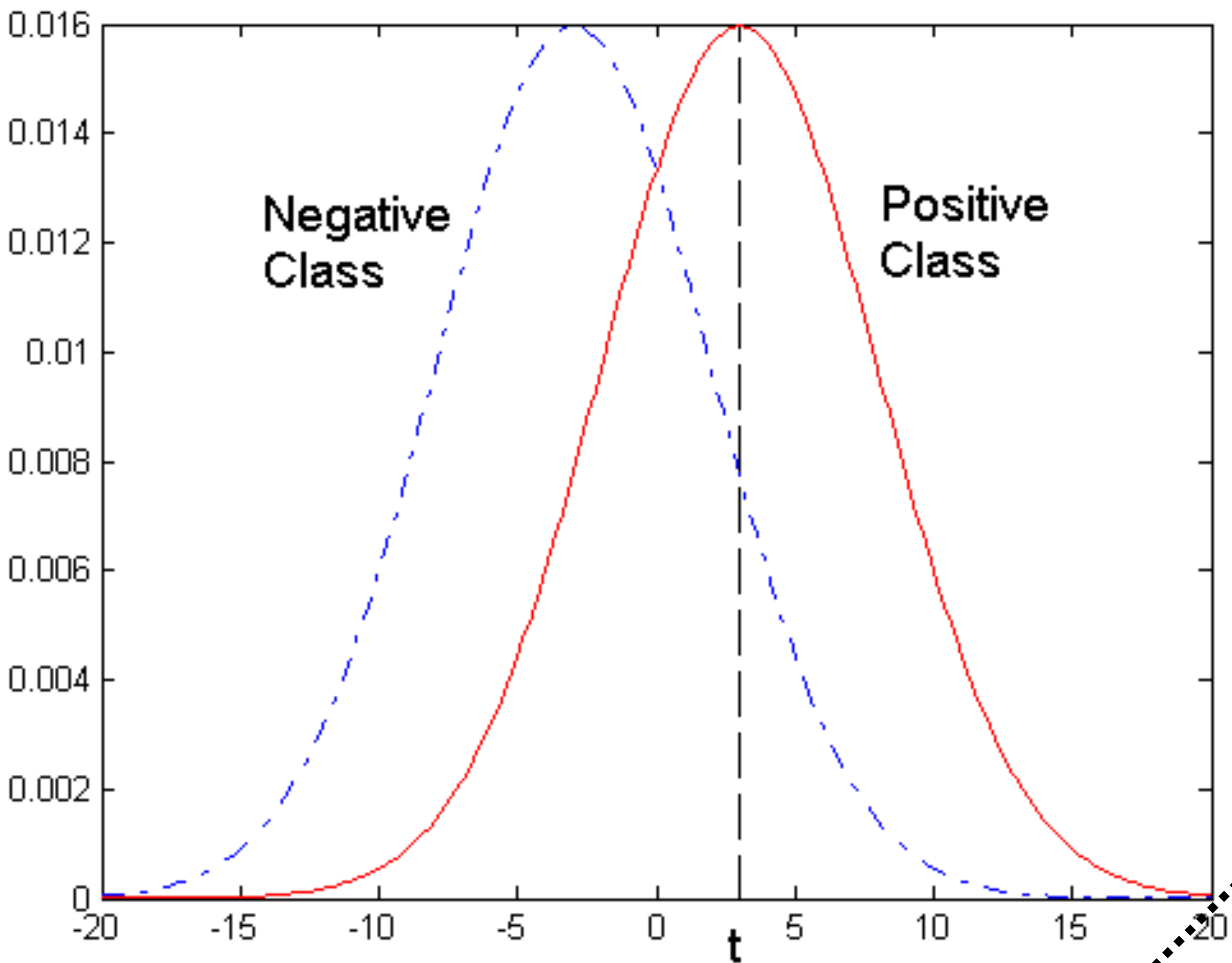
d: TN (true negative)

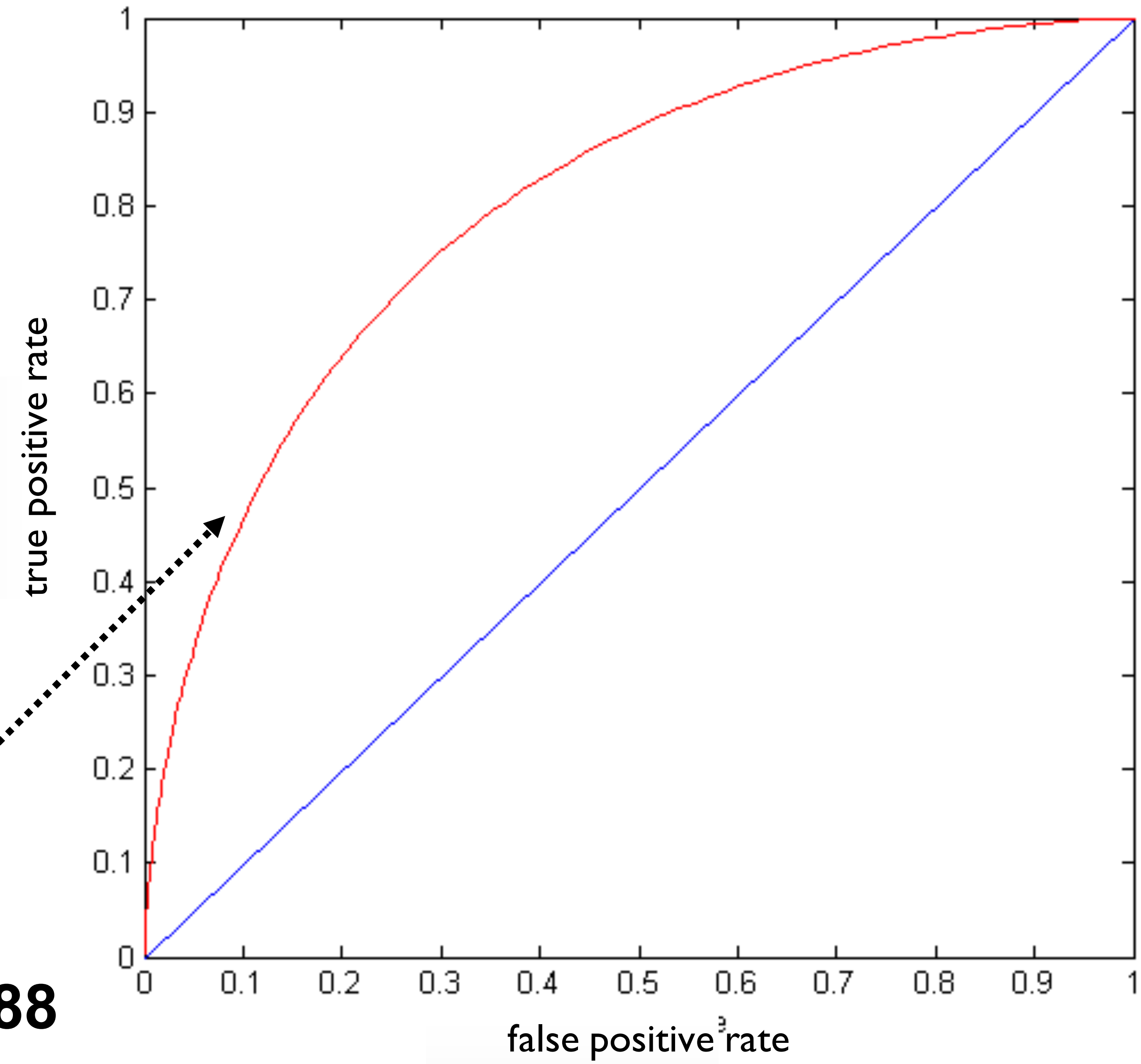
- Precision is biased towards C(Yes | Yes) & C(Yes | No)
- Recall is biased towards C(Yes | Yes) & C(No | Yes)
- F-measure is biased towards all except C(No | No)

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

ROC Curve

- 1-dimensional data set containing 2 classes (positive and negative)
- any points located at x > t is classified as positive





At threshold t:

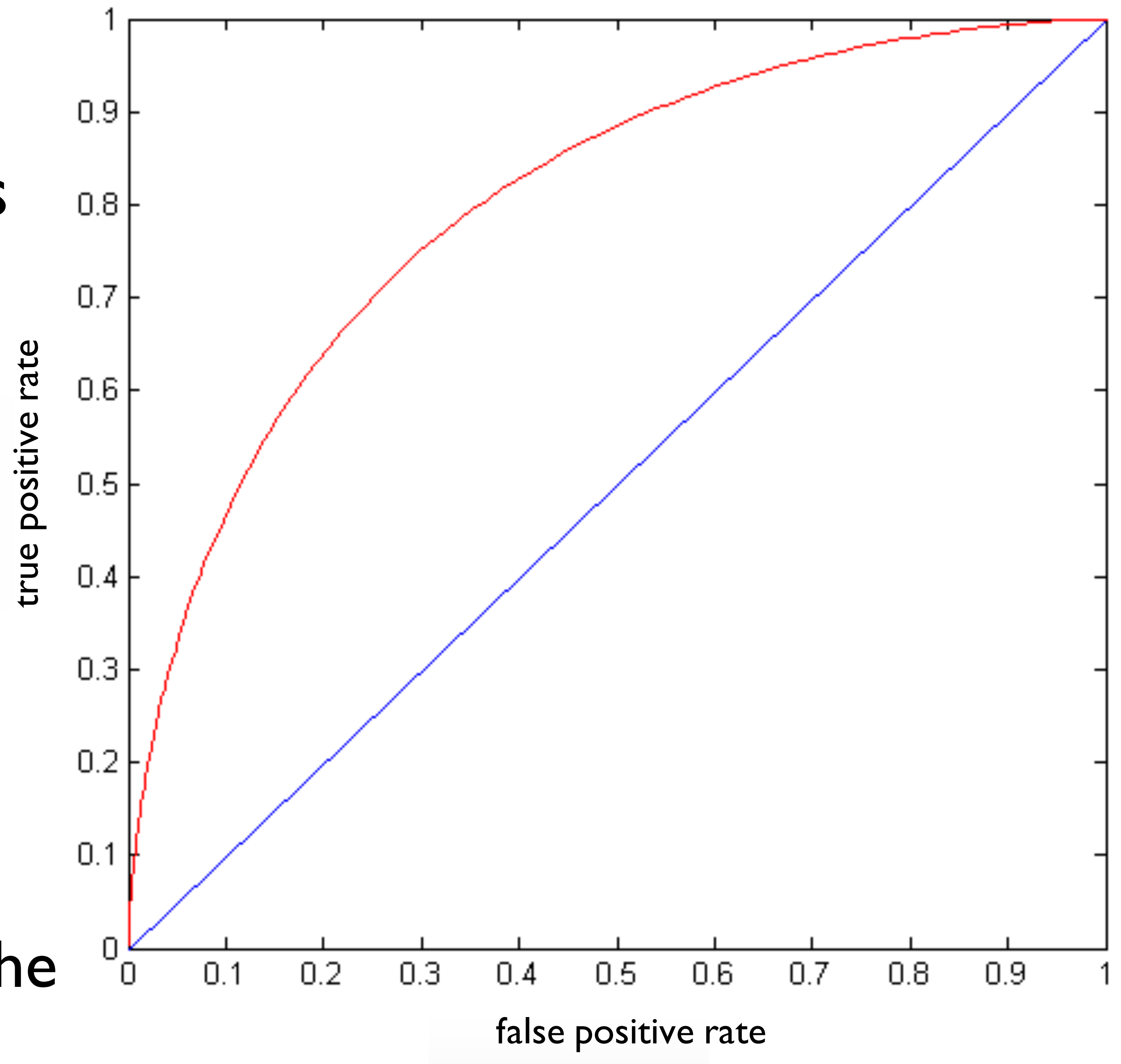
TP=0.5, FN=0.5, FP=0.12, TN=0.88

ROC Curve

(FPR, TPR):

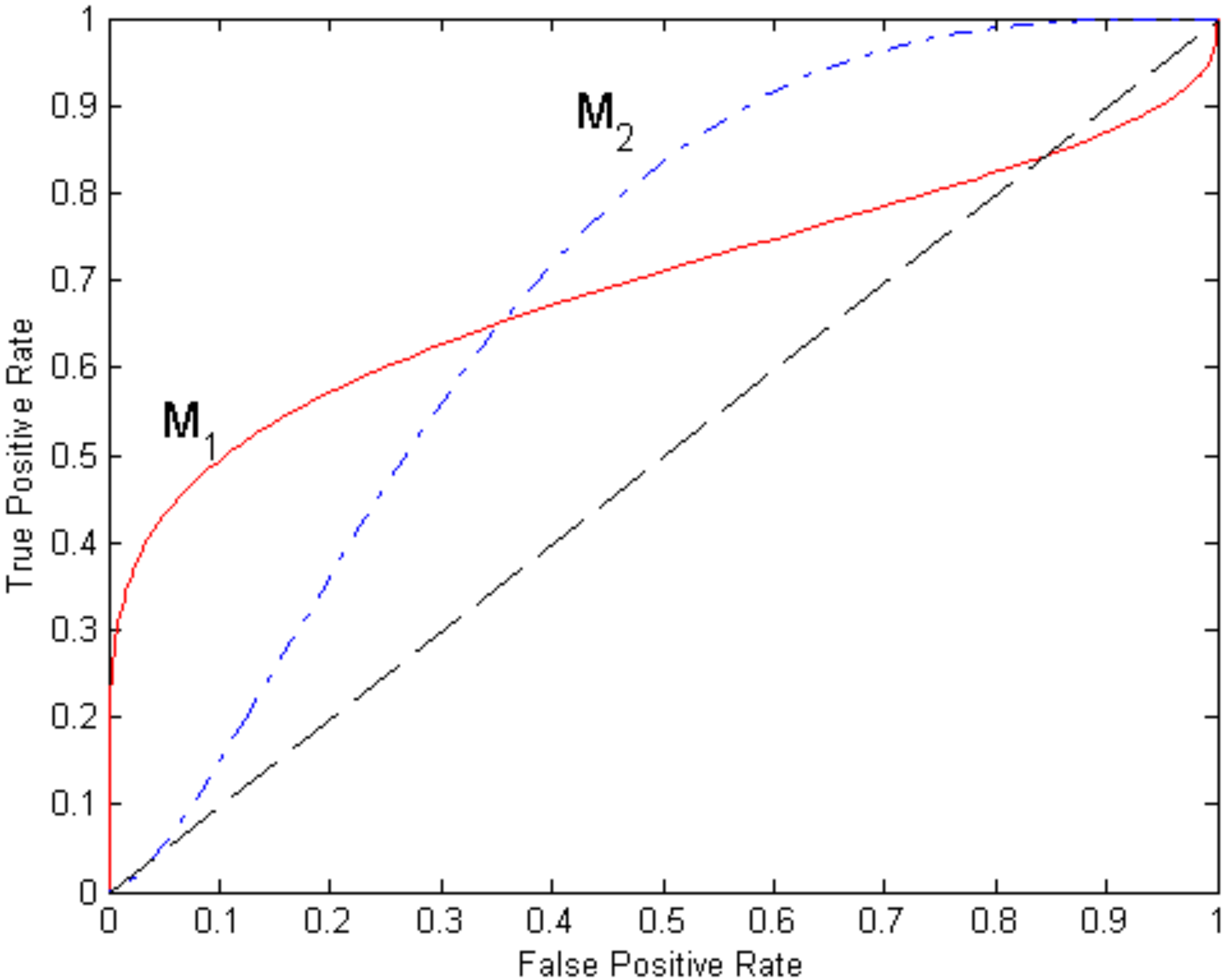
- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (0,1): ideal

- Diagonal line:
  – Random guessing
  – Below diagonal line:
    - prediction is opposite of the true class



Using ROC for Model Comparison

THE UNIVERSITY OF SYDNEY



- No model consistently outperform the other
- M_1 is better for small FPR
- M_2 is better for large FPR
- Area Under the ROC curve
  - Ideal:
    - Area = 1
  - Random guess:
    - Area = 0.5

How to construct a ROC curve

THE UNIVERSITY OF SYDNEY

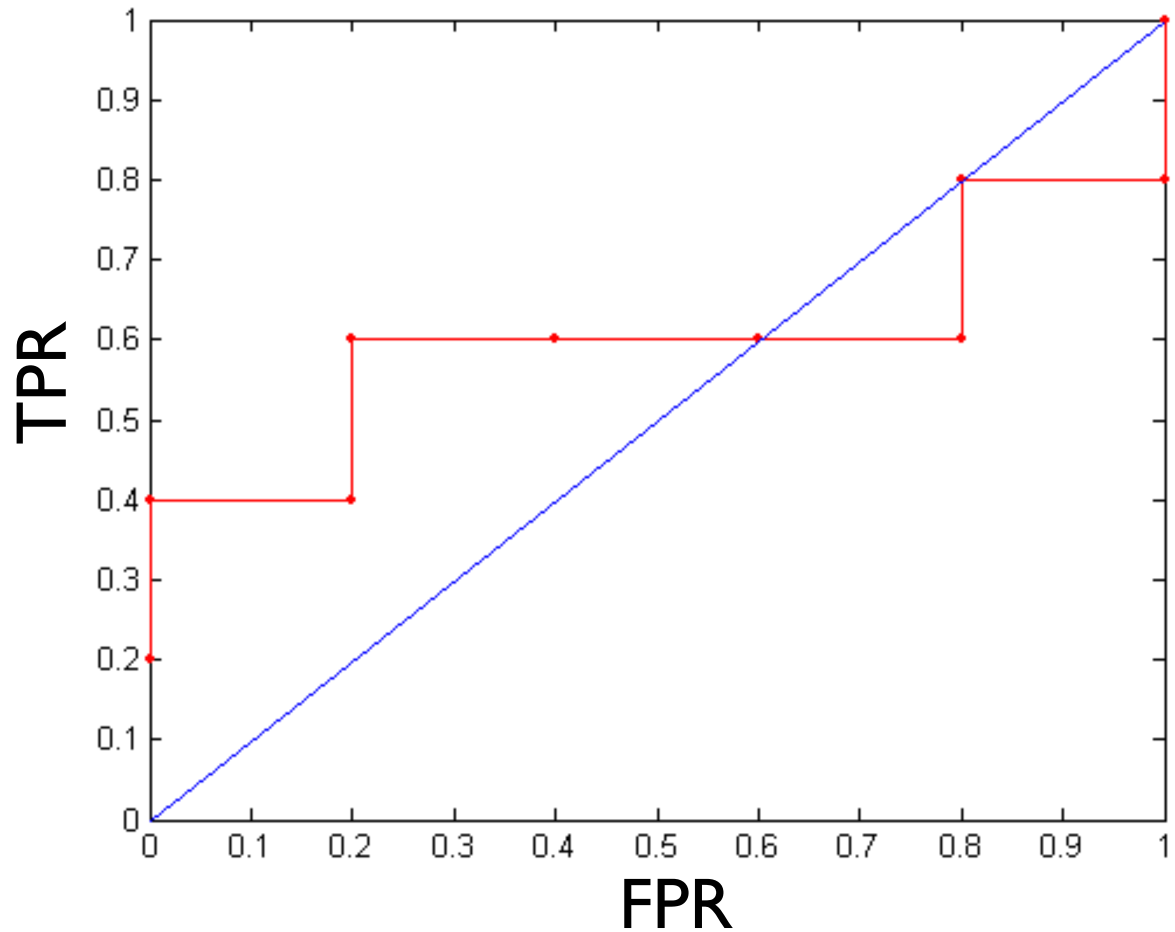
Instance	$P(+\|A)$	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

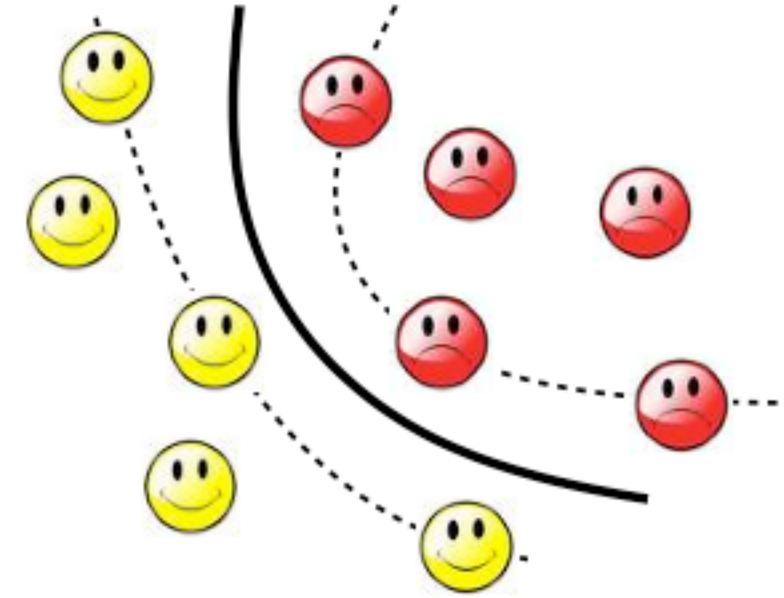
- Use classifier that produces posterior probability for each test instance $P(+ \mid A)$
- Sort the instances according to $P(+ \mid A)$ in decreasing order
- Apply threshold at each unique value of $P(+ \mid A)$
- Count the number of TP, FP, TN, FN at each threshold
- TP rate, TPR = TP/(TP+FN)
- FP rate, FPR = FP/(FP + TN)

How to construct a ROC curve

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC Curve:





Classification II

Loss functions

- Squared error, 0-1 Loss

$$L(y, \hat{y}) = (y - \hat{y})^2$$

$$L(y, \hat{y}) = I(y \neq \hat{y})$$

- Minimise risk, (expected risk, empirical risk)

$$R(\hat{f}) = E_{\mathbf{x},y} L(f(\mathbf{x}), \hat{f}(\mathbf{x}))$$

$$\hat{R}(\hat{f}) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{f}(\mathbf{x}_i))$$

Loss for density estimation

- Suppose output is $\hat{p}(y|\mathbf{x})$, truth is $p(y|\mathbf{x})$
- Use KL (Kullback-Leibler) divergence

$$L(p(y|\mathbf{x}), \hat{p}(y|\mathbf{x})) = KL(p(y|\mathbf{x}), \hat{p}(y|\mathbf{x})) = \sum_y p(y|\mathbf{x}) \log \frac{p(y|\mathbf{x})}{\hat{p}(y|\mathbf{x})}$$

- Risk is expected negative log likelihood

$$R(\hat{p}) = -E_{\mathbf{x}} \sum_y p(y|\mathbf{x}) \log \hat{p}(y|\mathbf{x}) = -E_{\mathbf{x},y} \log \hat{p}(y|\mathbf{x})$$

Estimation vs Inference

- Learning as optimisation (frequentist): Given D, choose $\hat{f}$ to approximate f as closely as possible, so as to minimise (future) expected risk
- Usually compute parameter estimate $\hat{\theta}$
- Learning as inference (Bayesian): Given D, compute posterior over functions $p(f|D)$
- Or posterior over parameters

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

- Decision theory demonstrates that one of the best ways to minimise frequentist risk is to be Bayesian.

Generative vs Discriminative

- Generative approach:
  - Model $p(y, \mathbf{x}) = p(\mathbf{x}|y)p(y)$
  - Use Bayes' theorem $p(y|\mathbf{x}) = \dfrac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$

- Discriminative approach:
  - Model $p(y|\mathbf{x})$ directly

Naïve Bayes Classifier

Generative model:

$$p(\mathbf{x}, \mathcal{C}_k) = p(\mathcal{C}_k)p(\mathbf{x}_n|\mathcal{C}_k) = \pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)$$

class conditional density

class prior

class posterior

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_{\mathcal{C}_j} p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)}$$

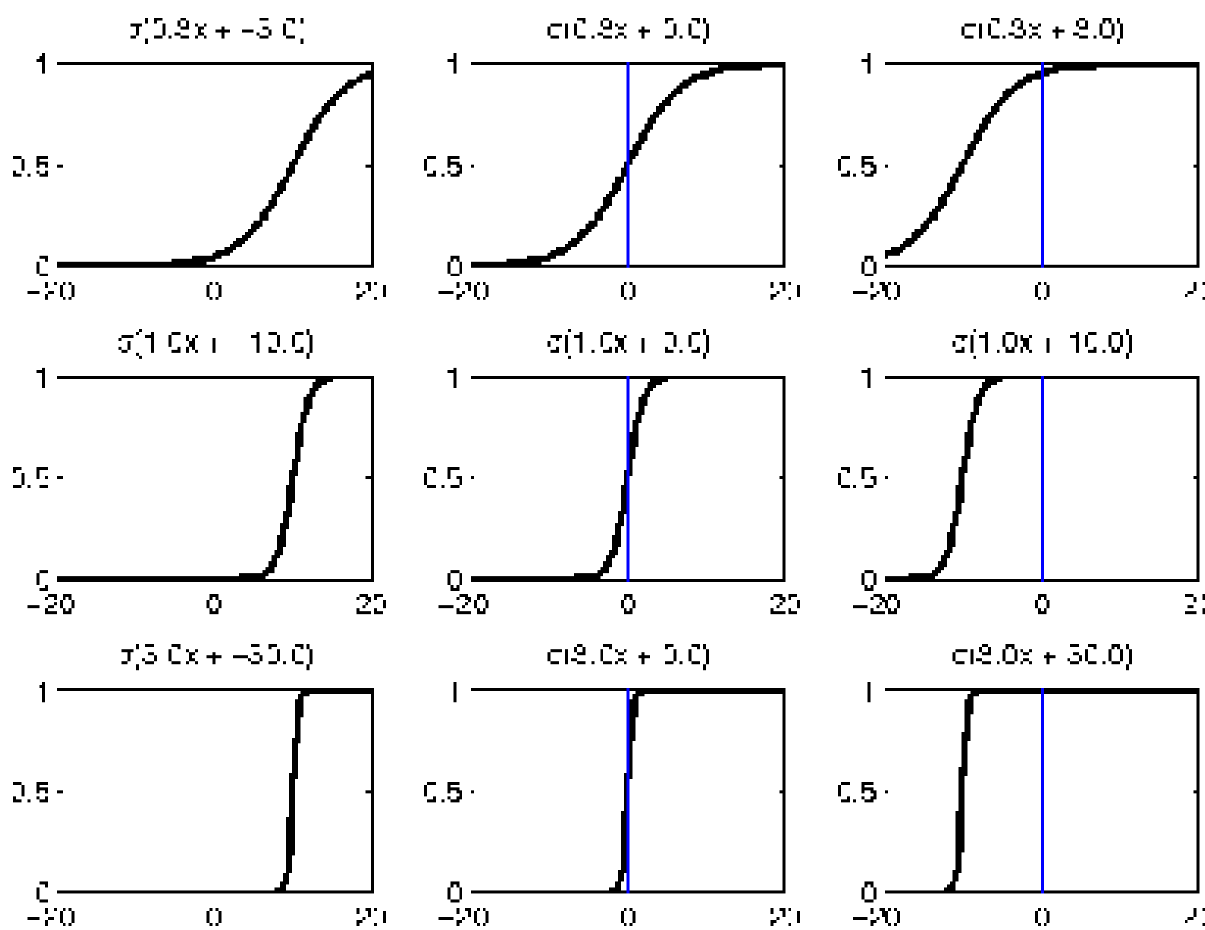
normalising constant

Logistic Regression

Logistic Regression

- Discriminative model for binary classification

$$
\begin{aligned}
p(y|\mathbf{x},\mathbf{w}) &= \mathrm{Ber}(y|\sigma(\eta)) = \sigma(\eta)^y(1-\sigma(\eta))^{1-y} \\
\eta &= \mathbf{w}^T\mathbf{x} \\
\sigma(\eta) &\overset{\text{def}}{=} \frac{1}{1+\exp(-\eta)} = \frac{e^\eta}{e^\eta+1}
\end{aligned}
$$

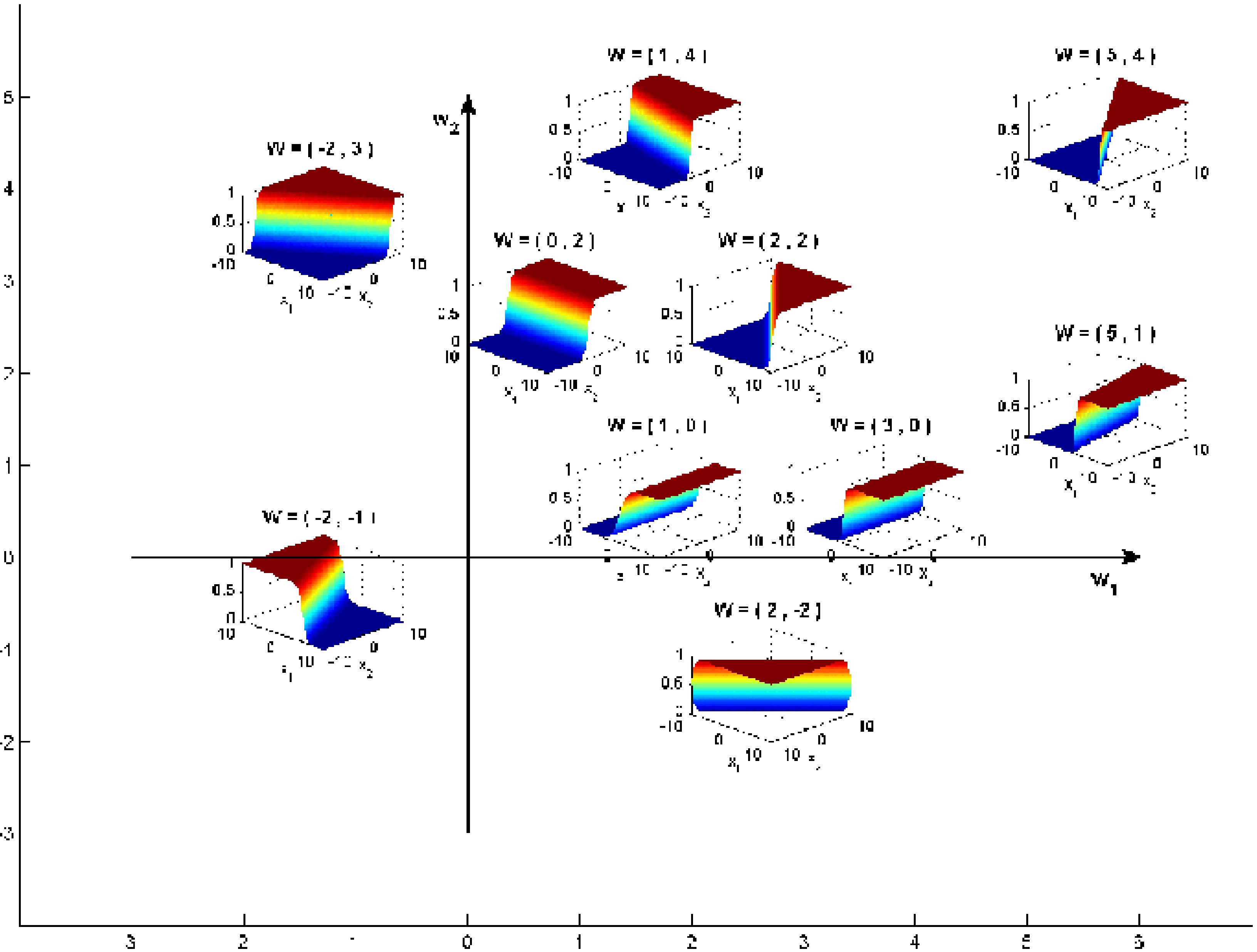


Sigmoid or Logistic function

Decision boundary

- Logistic Regression in 2D

$$p(y = 1|\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T \mathbf{x})$$



Logistic Regression

- Assumes a parametric form for directly estimating P(Y | X). For binary concepts, this is:

$$P(Y=0|X) = \frac{1}{1+\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

$$P(Y=1|X) = 1 - P(Y=0|X)$$

$$= \frac{\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1+\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

- Equivalent to a one-layer backpropagation neural net.
- Logistic regression is the source of the sigmoid function used in backpropagation.
- Objective function for training is somewhat different.

Logistic Regression as a Log-Linear Model

- Logistic regression is basically a linear model, which is demonstrated by taking logs.

$$\text{Assign label } Y = 0 \text{ iff } 1 < \frac{P(Y=0 \mid X)}{P(Y=1 \mid X)}$$

$$1 > \exp(w_0 + \sum_{i=1}^{n} w_i X_i)$$

$$0 > w_0 + \sum_{i=1}^{n} w_i X_i$$

- Also called a **maximum entropy model** (**MaxEnt**) because it can be shown that standard training for logistic regression gives the distribution with maximum entropy that is consistent with the training data.

Logistic Regression Training

- Weights are set during training to maximise the **conditional data likelihood** :

$$W \leftarrow \underset{W}{\operatorname{argmax}} \prod_{d \in D} P(Y^d \mid X^d, W)$$

where D is the set of training examples and Y^d and X^d denote, respectively, the values of Y and X for example d.

- Equivalently viewed as maximising the **conditional log likelihood** (CLL)

$$W \leftarrow \underset{W}{\operatorname{argmax}} \sum_{d \in D} \ln P(Y^d \mid X^d, W)$$

Logistic Regression Training

- Like neural-nets, can use standard gradient descent to find the parameters (weights) that optimise the CLL objective function.
- Many other more advanced training methods are possible to speed up convergence.
  - Conjugate gradient
  - Generalised Iterative Scaling (GIS)
  - Modified Iterative Scaling (MIS)
  - Limited-memory quasi-Newton (L-BFGS)
  - Stochastic gradient descent

Logistic Regression Training

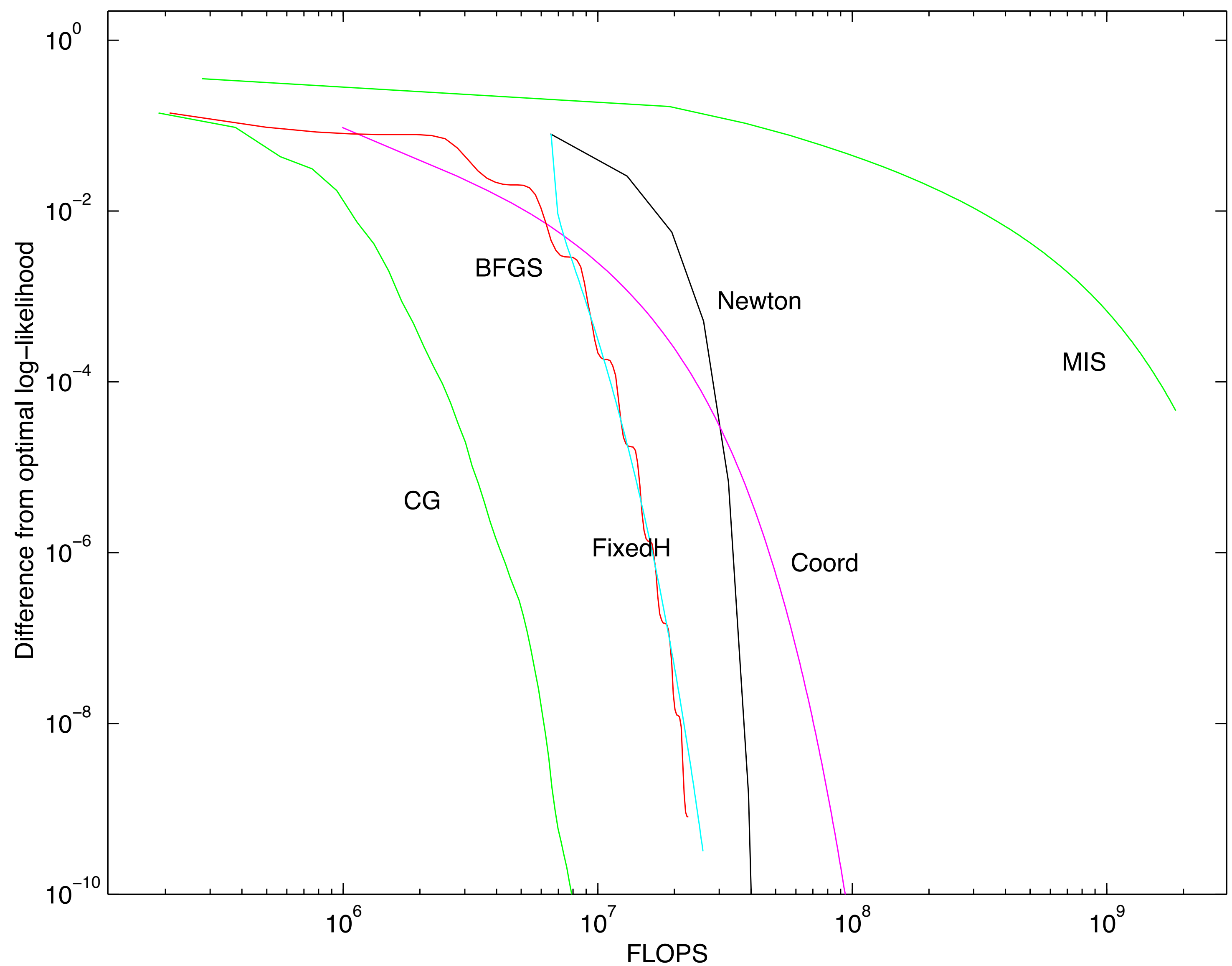


Figure 1: Cost vs. performance of six logistic regression algorithms. The dataset had 300 points in 100 dimensions. "CG" is conjugate gradient (section 4), "Coord" is coordinate-wise Newton, "FixedH" is Fixed-Hessian, and "MIS" is modified iterative scaling. CG also has the lowest actual time in Matlab.

See: "A comparison of numerical optimizers for logistic regression" by Thomas Minka, 2003.

Preventing Overfitting in Logistic Regression

THE UNIVERSITY OF SYDNEY

- To prevent overfitting, one can use **regularisation** (a.k.a. smoothing) by penalising large weights by changing the training objective:

$$W \leftarrow \underset{W}{\operatorname{argmax}} \sum_{d \in D} \ln P(Y^d \mid X^d, W) - \frac{\lambda}{2} \|W\|^2$$

Where λ is a constant that determines the amount of smoothing

- This can be shown to be equivalent to assuming a Gaussian prior for *W* with zero mean and a variance related to $1/\lambda$.

Multinomial Logistic Regression

- Logistic regression can be generalised to multi-class problems (where Y has a multinomial distribution).
- Effectively constructs a linear classifier for each category.

Relation Between NB and Logistic Regression

- Naïve Bayes with Gaussian distributions for features (GNB), can be shown to give the same functional form for the conditional distribution $P(Y \mid X)$.
  - But converse is not true, so Logistic Regression makes a weaker assumption.
- Logistic regression is a **discriminative** rather than generative model, since it models the conditional distribution $P(Y \mid X)$ and directly attempts to fit the training data for predicting Y from X. Does not specify a full joint distribution.

Relation Between NB and Logistic Regression (continued)

- When conditional independence is violated, logistic regression gives better generalisation if it is given sufficient training data.
- GNB converges to accurate parameter estimates faster ($O(\log n)$ examples for *n* features) compared to Logistic Regression ($O(n)$ examples).
- Experimentally, GNB is better when training data is scarce, logistic regression is better when it is plentiful.

Summary

- Logistic Regression
- Binary Classification