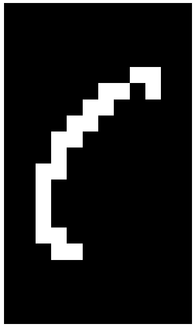
Multi-Class Classification

Basic Classification in ML

	Input	Output
	$\mathbf{x} \in \mathcal{X}$	$y \in \mathcal{Y}$
Spam filtering		**Binary** 
Character recognition		**Multi-Class** **C**

[thanks to Ben Taskar for slide!]

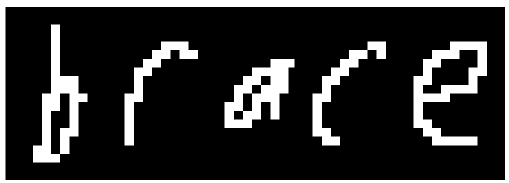
Structured Classification

Input $\mathbf{x} \in \mathcal{X}$

Output $\mathbf{y} \in \mathcal{Y}$

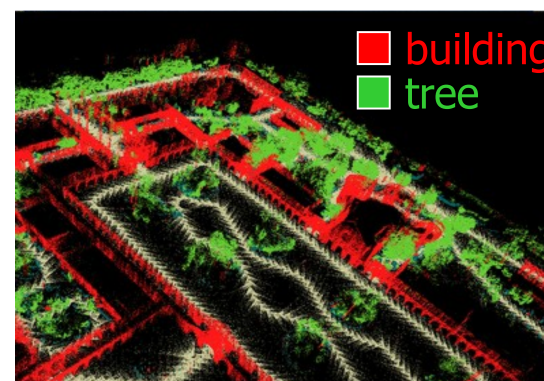
Handwriting recognition

Structured output



brace

3D object recognition



[thanks to Ben Taskar for slide!]

Multi-Class Classification

- Multi-class classification : direct approaches
  - Nearest Neighbor
  - Generative approach & Naïve Bayes
  - Linear classification:
    - geometry
    - Perceptron
    - K-class (polychotomous) logistic regression
    - K-class SVM
- Multi-class classification through binary classification
  - One-vs-All (OVA)
  - One-vs-One (OVO)
  - Others
  - Calibration

Multi-label classification

- Is it eatable?
- Is it sweet?
- Is it a fruit?
- Is it a banana?

Is it a banana?

Is it an apple?

Is it an orange?

Is it a pineapple?

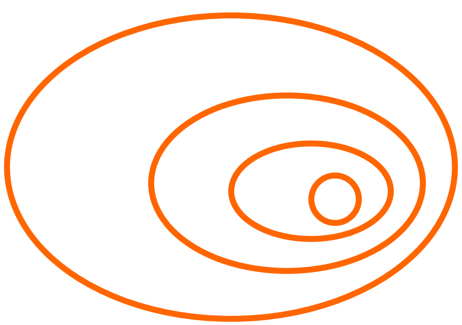
Is it a banana?

Is it yellow?

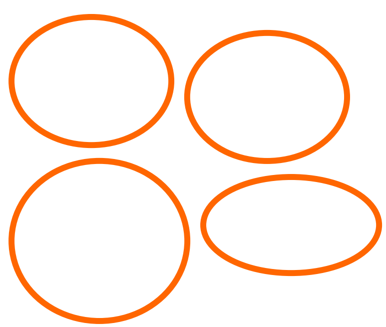
Is it sweet?

Is it round?

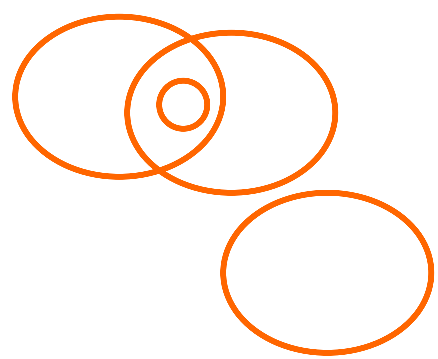
Different structures



Nested/ Hierarchical



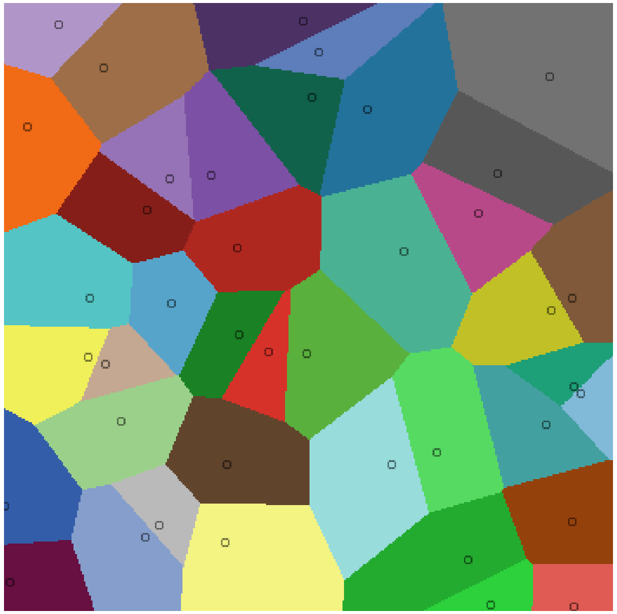
Exclusive/ Multi-class



General/Structured

Nearest Neighbor, Decision Trees

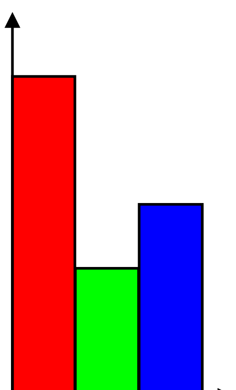
- k-NN is already phrased in a multi-class framework
- For decision tree, want purity of leaves depending on the proportion of each class (want one class to be clearly dominant)



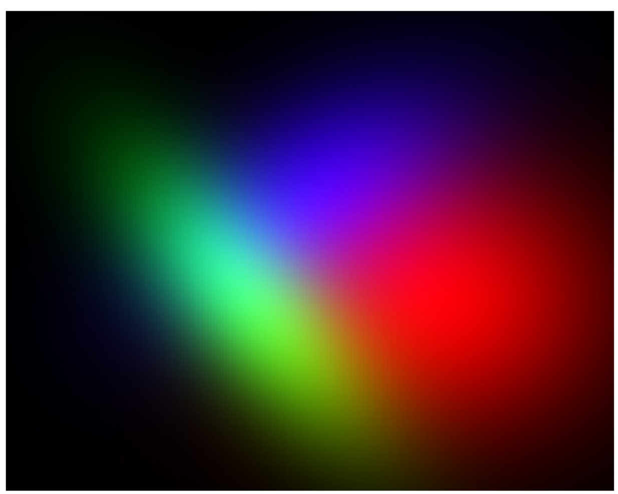
Generative models

As in the binary case:

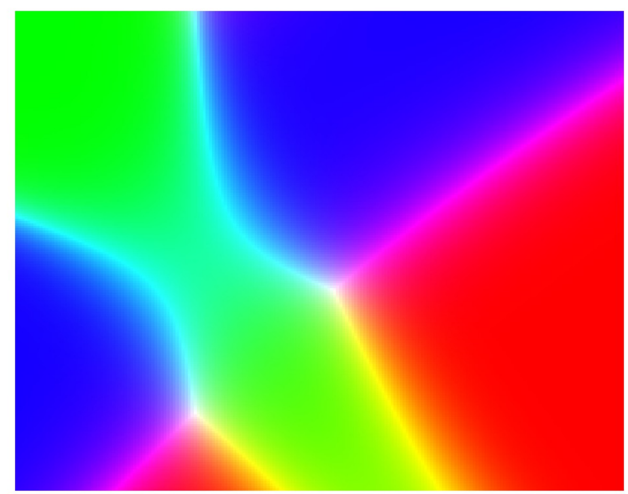
1. Learn p(y) and p(y|x)
2. Use Bayes rule: $p(y=k|x) = \frac{p(x|y=k)p(y=k)}{p(x)}$
3. Classify as $\hat{y}(x) = \mathrm{argmax}_y\, p(y|x)$



p(y)



p(x|y)

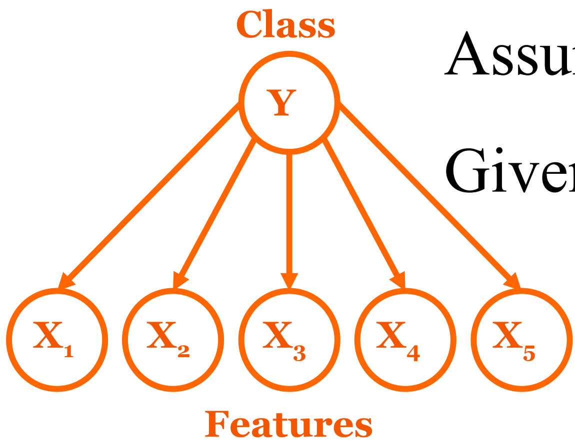


p(y|x)

Generative models

- Advantages:
  - Fast to train: only the data from class k is needed to learn the k^{th} model (reduction by a factor k compared with other method)
  - Works well with little data provided the model is reasonable
- Drawbacks:
  - Depends on the quality of the model
  - Doesn't model $p(y|x)$ directly
  - With a lot of datapoints doesn't perform as well as discriminative methods

Naïve Bayes



Assumption:

Given the class the features are independent

$$p(x|y=k) = \prod_i p(x_i|y=k)$$

$\Rightarrow$ **Bag-of-words models**

$$\log p(y=k|x) = \sum_i \log p(x_i|y=k) + \log p(y=k) - \log p(x)$$

If the features are discrete:

$$\log p(y=k|x) = \sum_i \sum_{u_i} \log p(u_i|y=k) \mathbf{1}\{x_i = u_i\} + \log p(y=k) - \log p(x)$$

$$\log p(y=k|x) = w_k^\top \Phi(x) + \log p(y=k) - \log p(x)$$

$$\log \frac{p(y=k|x)}{p(y=j|x)} = (w_k - w_j)^\top \Phi(x) + \log \frac{p(y=k)}{p(y=j)}$$

Linear classification

- Each class has a parameter vector (w_k, b_k)
- x is assigned to class k iff $w_k^\top x + b_k \geq \max_j w_j^\top x + b_j$
- Note that we can break the symmetry and choose $(w_1, b_1) = 0$
- For simplicity set $b_k = 0$ (add a dimension and include it in w_k)
- So learning goal given separable data: choose w_k s.t.

$$\forall (x^i, y^i), \quad \underbrace{{w_{y^i}}^\top x^i}_{\text{score of truth}} \geq \max_j \underbrace{w_j^\top x^i}_{\text{score of competitor.}}$$

Three discriminative algorithms

Perceptron: $\max_W \sum_i \left[{w_{y^i}}^\top x^i - \max_k w_k^\top x^i \right]$

[mistake driven]

K-class logistic regression: $\max_W \sum_i \left[{w_{y^i}}^\top x^i - \text{softmax}_k\, w_k^\top x^i \right]$

[max conditional likelihood]

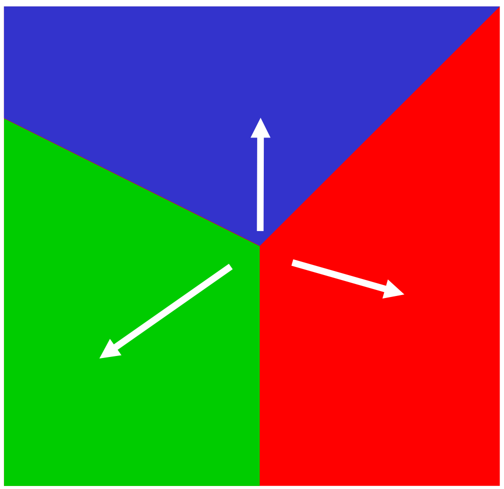
$$\boxed{\mathcal{L}(\mathbf{W}) = \max\{0, 1 + \max_{k \neq y_n} \boldsymbol{w}_k^\top \boldsymbol{x}_n - \boldsymbol{w}_{y_n}^\top \boldsymbol{x}_n\}}$$

(Crammer-Singer multiclass SVM)

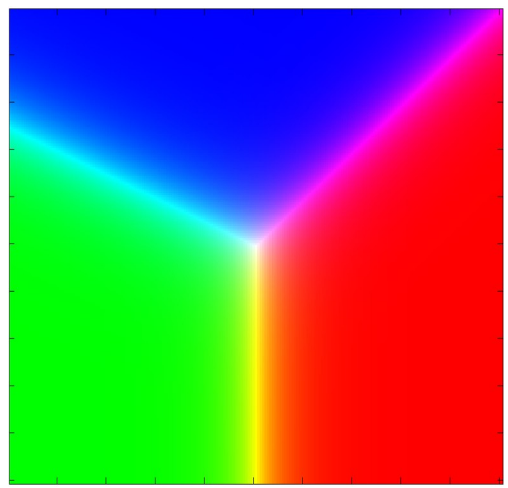
- Loss $= 0$ if score on correct class is at least 1 more than score on next best scoring class
- Can optimize these similar to how we did it for binary SVM [large margin method]

Geometry of Linear classification

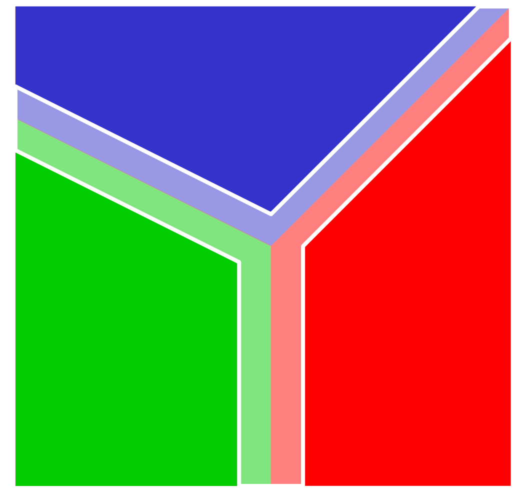
Perceptron



K-class logistic regression



K-class SVM



Multiclass Perceptron

Online: for each datapoint

Predict: $\widehat{y}_i = \arg\max_y \; w_y^\top x^i$

Update: if $\widehat{y}_i \neq y^i$ then

$$\begin{cases} w_{y^i,t+1} = w_{y^i,t} + \alpha x^i \\ w_{\widehat{y}_i,t+1} = w_{\widehat{y}_i,t} - \alpha x^i \end{cases}$$

$$\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$$

- Advantages :
  - Extremely simple updates (no gradient to calculate)
  - No need to have all the data in memory (some point stay classified correctly after a while)
- Drawbacks
  - If the data is not separable decrease α slowly…

Polychotomous logistic regression

$$p(y=k|x) = \frac{\exp w_k^\top x}{\sum_j \exp w_j^\top x}$$

distribution in exponential form

$$\log p(y=k|x) = w_k^\top x - \log \sum_j \exp w_j^\top x$$

Online: for each datapoint

"soft mistake update"

$$w_j \leftarrow w_j + \alpha x^i (\mathbf{1}\{j = y^i\} - p(y=j|x=x^i))$$

Batch: all descent methods

Especially in large dimension, use regularization $\left\{ \begin{array}{l} \|w\|_2^2, \ \|w\|_1 \\ \text{small flip label probability} \\ (0,0,1) \longrightarrow (.1,.1,.8) \end{array} \right.$

Advantages:

- Smooth function
- Get probability estimates

Drawbacks:

- Non sparse

Multi-class SVM

Intuitive formulation: without regularization / for the separable case

$$\max_{W} \left[\sum_i {w_{y^i}}^\top x^i - \max_j (\mathbf{1}\{j \neq y^i\} + w_j^\top x^i) \right]$$

Primal problem: QP

$$\min_{w_1,\ldots,w_K} \quad \frac{1}{2}\|(w_1, \ldots, w_K)\|^2 + C \sum_{ik} \xi_{ik}$$

$$\text{s.t.} \quad \forall (i,k), \quad {w_{y^i}}^\top x^i - w_k^\top x^i \geq \mathbf{1}\{k \neq y^i\} - \xi_{ik}$$

Solved in the dual formulation, also Quadratic Program

Main advantage: Sparsity (but not systematic)

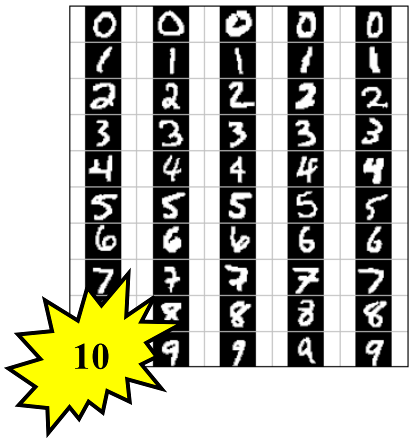
- Speed with SMO (heuristic use of sparsity)
- Sparse solutions

Drawbacks:

- Need to recalculate or store $x_i^T x_j$
- Outputs not probabilities

Real world classification problems

Digit recognition



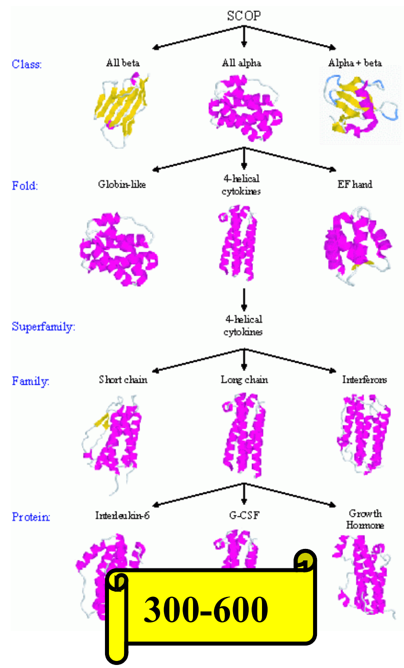
Phoneme recognition

50

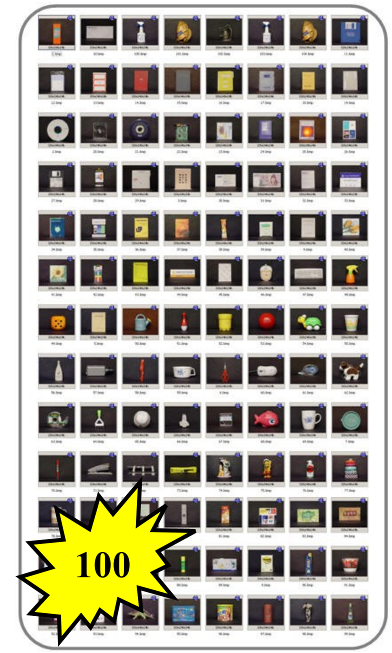
iː READ	ɪ SIT	ʊ BOOK	uː TOO	ɪə HERE	eɪ DAY		
e MEN	ə AMERICA	ɜː WORD	ɔː SORT	ʊə TOUR	ɔɪ BOY	əʊ GO	
æ CAT	ʌ BUT	ɑː PART	ɒ NOT	eə WEAR	aɪ MY	aʊ HOW	
p PIG	b BED	t TIME	d DO	tʃ CHURCH	dʒ JUDGE	k KILO	g GO
[illegible]	[illegible]	θ THINK	ð THE	s SIX	z ZOO	ʃ SHORT	ʒ CASUAL
[illegible]	[illegible]	ŋ SING	h HELLO	l LIVE	r READ	w WINDOW	j YES

[Waibel, Hanzawa, Hinton,Shikano, Lang 1989]

Automated protein classification



Object recognition



http://www.glue.umd.edu/~zhelin/recog.html

- The number of classes is sometimes big
- The multi-class algorithm can be heavy

Combining binary classifiers

One-vs-All (OVA) For each class build a classifier for that class vs the rest

- Often very imbalanced classifiers (use asymmetric regularization)

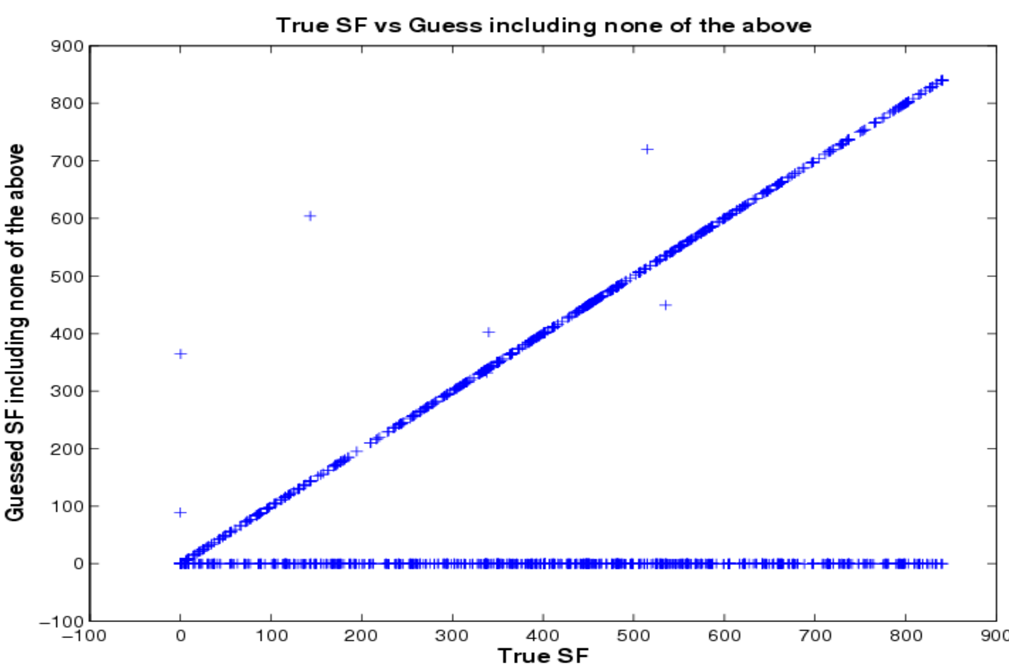
One-vs-One (OVO) We compare all possible pairs of classifiers

- A priori a large number of classifiers $\binom{n}{2}$ to build ***but…***
    - The pairwise classification are way much faster
    - The classifications are balanced (easier to find the best regularization)

… so that in many cases it is clearly faster than one-vs-all

Confusion Matrix

- Visualize which classes are more difficult to learn
- Can also be used to compare two different classifiers
- Cluster classes and go hierachical [Godbole, ‘02]



BLAST classification of proteins in 850 superfamilies

Classification of 20 news groups

Predicted classes

Actual classes

Classname		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
alt.atheism	1	251	6	1	3	32	1	1	2	1	2	0	0	0	0	0	0	0	0	0	0
soc.religion.christian	2	9	277	0	1	6	0	0	1	0	0	0	0	0	1	2	2	0	0	0	1
sci.space	3	3	1	273	1	0	1	2	0	1	1	9	0	0	1	2	3	0	0	1	1
talk.politics.misc	4	2	0	3	213	24	3	0	17	3	0	0	0	0	0	0	1	0	1	33	0
talk.religion.misc	5	88	36	2	23	132	0	1	0	0	0	0	0	0	0	0	2	0	1	15	0
rec.autos	6	0	0	0	3	1	272	0	0	0	7	1	2	1	6	4	1	0	0	2	0
comp.windows.x	7	1	1	2	1	0	1	246	0	2	2	30	5	3	1	1	2	1	1	0	0
talk.politics.mideast	8	0	3	1	18	0	0	0	275	0	1	0	0	0	0	0	0	0	1	1	0
sci.crypt	9	1	0	1	2	1	0	3	0	284	0	3	0	1	0	0	1	0	0	3	0
rec.motorcycles	10	0	0	0	1	0	4	1	0	0	286	1	2	0	1	2	1	0	0	1	0
comp.graphics	11	0	1	2	1	1	0	10	1	2	0	243	23	7	3	3	3	0	0	0	0
comp.sys.ibm.pc.hardware	12	0	0	0	0	0	2	7	0	1	0	5	243	23	12	3	1	3	0	0	0
comp.sys.mac.hardware	13	0	0	1	1	0	2	1	0	0	0	7	10	260	8	9	1	0	0	0	0
sci.electronics	14	1	0	1	0	1	5	2	0	2	0	7	13	13	245	6	3	0	1	0	0
misc.forsale	15	0	1	4	2	0	12	1	0	0	4	1	19	10	8	233	1	0	1	1	2
sci.med	16	0	1	5	0	1	1	0	0	0	1	2	0	2	7	2	275	0	1	1	1
comp.os.mswindows.misc	17	1	0	2	0	1	1	58	1	3	0	38	71	17	3	6	0	97	1	0	0
rec.sport.baseball	18	2	1	1	0	0	0	0	0	0	0	4	0	0	0	1	1	0	282	1	7
talk.politics.guns	19	0	0	0	9	5	1	0	0	1	0	0	0	0	1	0	0	1	1	281	0
rec.sport.hockey	20	0	1	0	0	0	1	0	0	0	2	0	0	1	1	0	0	0	3	0	291

[Godbole, ‘02]

Calibration

How to measure the confidence in a class prediction?

Crucial for:

1. Comparison between different classifiers
2. Ranking the prediction for ROC/Precision-Recall curve
3. In several application domains having a measure of confidence for each individual answer is very important (e.g. tumor detection)

Some methods have an implicit notion of confidence e.g. for SVM the distance to the class boundary relative to the size of the margin other like logistic regression have an explicit one.

Calibration

Definition: the decision function f of a classifier is said to be *calibrated* or *well-calibrated* if

$$\mathbf{P}(x \text{ is correctly classified} \,|\, f(x) = s) \simeq s$$

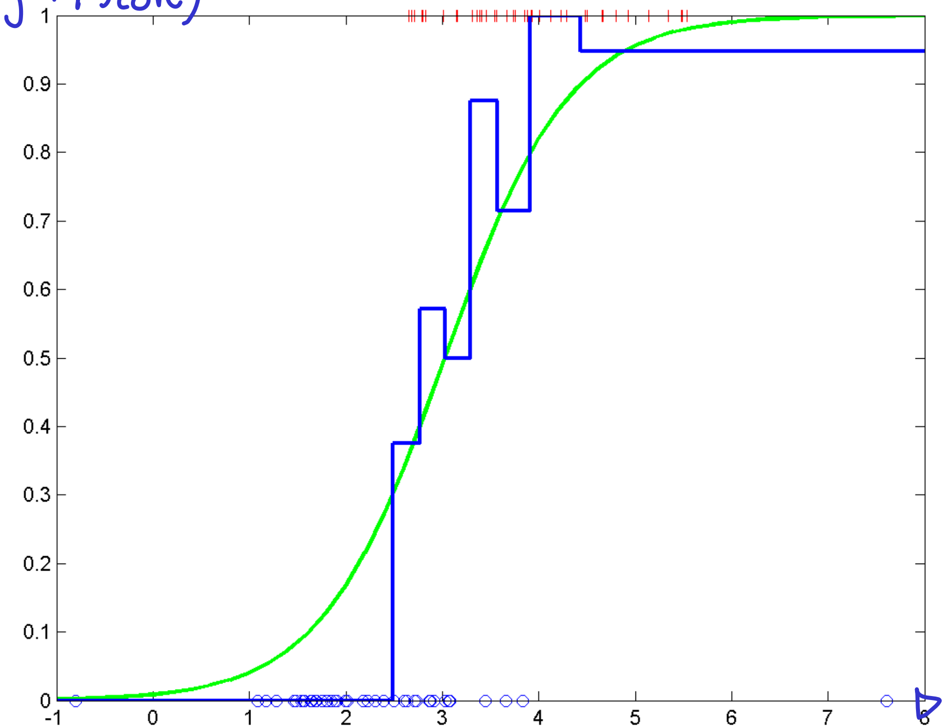
Informally f is a good estimate of the probability of classifying correctly a new datapoint x which would have output value x.

Intuitively if the "raw" output of a classifier is g you can calibrate it by estimating the probability of x being well classified given that g(x)=y for all y values possible.

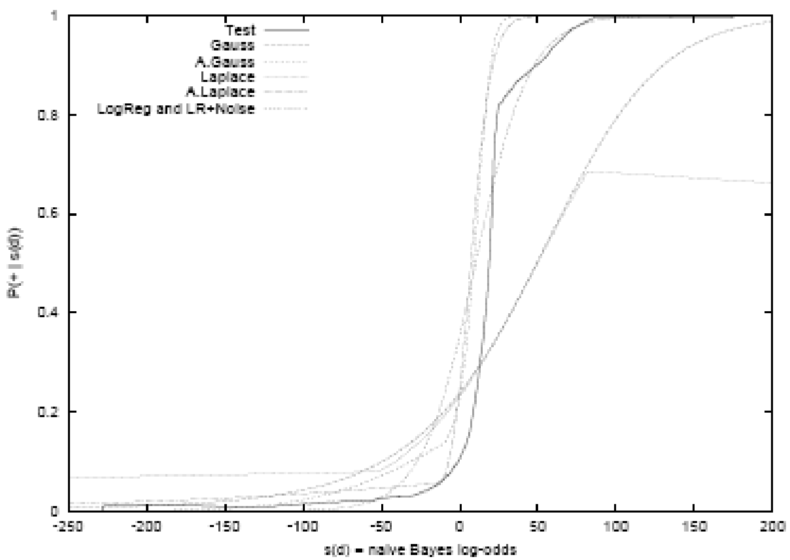
Calibration

Example: a logistic regression, or more generally calculating a Bayes posterior should yield a reasonably *well-calibrated* decision function.

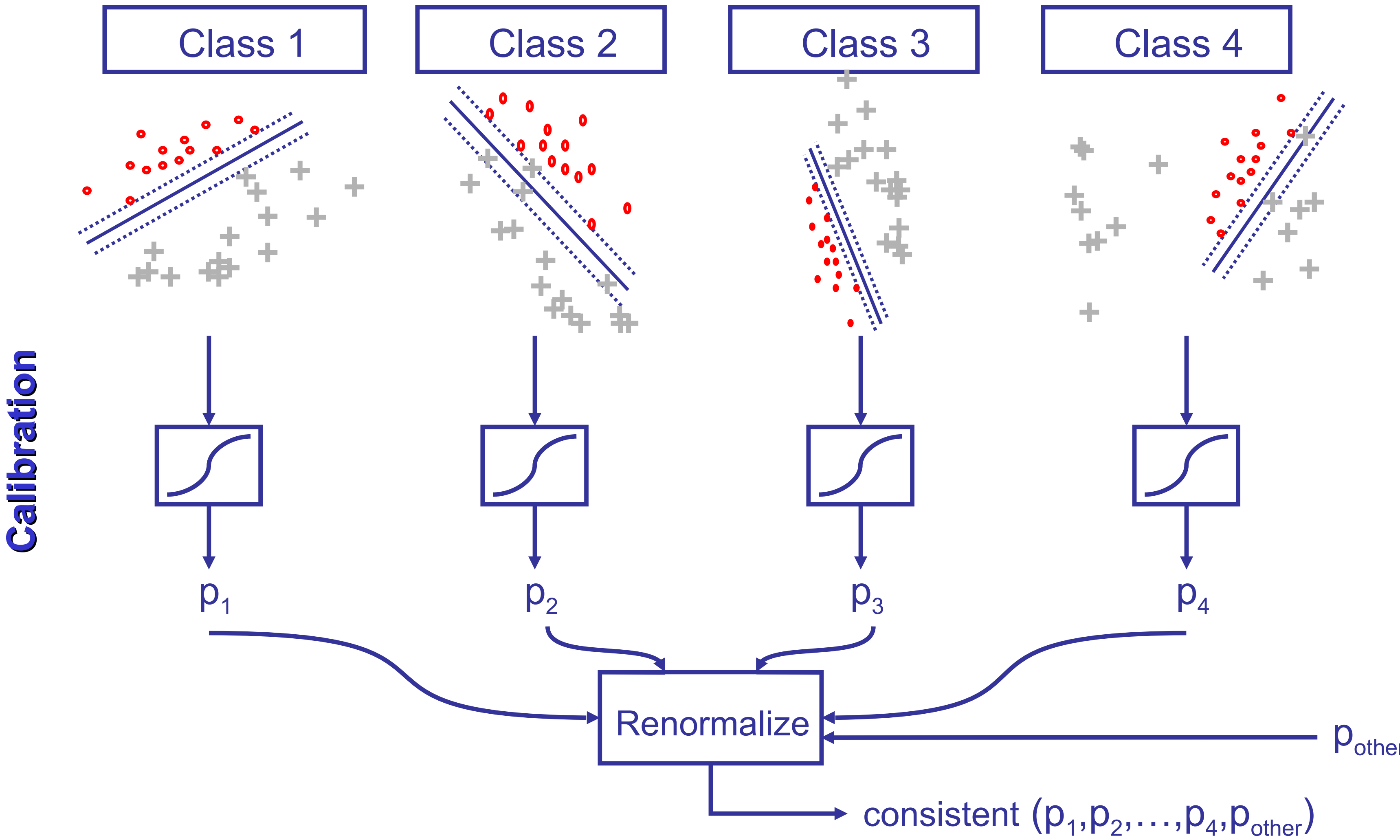
prob($y=1$ | score)



score e.g. $w \cdot x_i + b$



Combining OVA calibrated classifiers



Other methods for calibration

- Simple calibration
  - Logistic regression
  - Intraclass density estimation + Naïve Bayes
  - Isotonic regression
- More sophisticated calibrations
  - Calibration for A-vs-A by Hastie and Tibshirani