Lecture 2:

Classification I

September 19, 2006

Class 0


```
Subject: EMPCA code download
Subject: Neighbourhood Component Analysis
Subject: Bell Canada Graduate Scholarships
Subject: Re: more info on ML/AI
Subject: Re: advisory committee
Subject: more experiments on structure learning
Subject: Thesis comments and corrections
Subject: IRIS In-kind contributions etc
Subject: neighbourhood components
Subject: short visit details
Subject: paper
Subject: [Fwd: Fwd: prize possibilities...]
Subject: Re: paper
Subject: RKHS revised
Subject: Matlab codes for KF+EM
Subject: Re: new idea
Subject: Revised Preliminary Grad Timetable
Subject: Re: Neurocomputing Review Request
Subject: Salary Benefit and Pensions - and Bulletin - Issues
Subject: nonnegative least squares
Subject: RE: visiting the UofT
Subject: Re: Amir
Subject: Re: PR266 (fwd)
Subject: Re: PR266 (fwd)
Subject: postdoc position
Subject: Re: multiplicative update algorithm
Subject: RE: dual trees in simpleKD?
Subject: Re: dual trees in simpleKD?
Subject: Re: future of 411 and 412 (fwd)
Subject: (Time-sensitive) Ontario Research Fund: Research Excellence Fund
Subject: [PDADC-L] Ontario Research Fund   [windows-1252] Research
Subject: new faculty arrivals
Subject: Hi, Mr. Roweis
Subject: your phd thesis
Subject: depth draft & volume paper & volume reviews
Subject: Re: letter and visas
Subject: savitch trick does work
Subject: Re: auton lab code
Subject: Locally Linear Embedding algorithm
Subject: EM algorithms
Subject: EM-algorithm
Subject: Microsoft_Research_Lecture=3A_Thierry_Arti=E8res=2C_=27A_g?=
Subject: (pas de sujet)
Subject: qualifying oral examination
Subject: Approval expiry reminder for protocol #12435
Subject: DCS Fall 2005 Courses
Subject: A possible review for psychometrika
Subject: ResearchNet has moved!
Subject: ResearchNet has moved! / Nouvelle adresse pour Recherchenet
Subject: [Fwd: Academic Handbook: Part II]
```

```
Subject: pool party
Subject: Re: Approval expiry reminder for protocol #12435
Subject: NCA
Subject: Optimal Component Analysis Paper
Subject: Re: talk
Subject: Re: salary question
Subject: PREA REPORT 2005 REMINDER
Subject: CSGSBS Island Picnic: Saturday, Sept 10
Subject:  Chair Instructions - inside SGS -Tang
Subject: "purely continuous" CPM-like model?
Subject: Re: Report of the Task Force on Faculty Governance (fwd)
Subject: convergence properties of projected gradient methods
Subject: Re: image restoration
Subject: blind deconvolution
Subject: L1-norm integration of estimated image gradients
Subject: Re: l1 integration of image gradients?
Subject: Re: image restoration
Subject: Re: DP k-means?
Subject: Re: DP k-means?
Subject: looking for a media contact for the AI program
Subject: Seminar on Wednesday August 31 in GB248
Subject: IEEE SP - Special Issue: Signal Processing Methods Genomics/Proteomics
Subject: Re: JMLR Manuscript 05-092
Subject: Re: Ali Rahimi
Subject: RE: gridx1
Subject: RE: gridx1
Subject: Google Alert - "machine learning"
Subject: NCA method help
Subject: preparation for next term’s teaching
Subject: Please Sir,,, I need your help in PhD
Subject: [Jmlr-ed-board] J.of Algorithms editorial board revolt
Subject: NYTimes article on CS
Subject: [Fwd: Amir Globerson]
Subject: Offer Letter - Globerson
Subject: CBIN NCE-NI participation invitation
Subject: Re: Max and DP K-means
Subject: Depth exam on Sept. 15, 2pm
Subject: Call for abstracts: Snowbird Learning Workshop 2006
Subject: preparing your Course Information Sheet
Subject: RE: ICML 2006 Senior Program Committee Invitation
Subject: kcorrect paper
Subject: updated version of yesterday’s memo
Subject: rooms for midterms
Subject: planning your TA contracts (allocation of hours)
Subject: submitting your TA Allocation of Hours form online
Subject: follow-up
Subject: Re: hello
Subject: Distungished Lecture Series, Fall 2005
Subject: Marking Scheme Form and Request for Examination
Subject: 05-06 Graduate Tinetable (with Room assignments)
```

Class 1

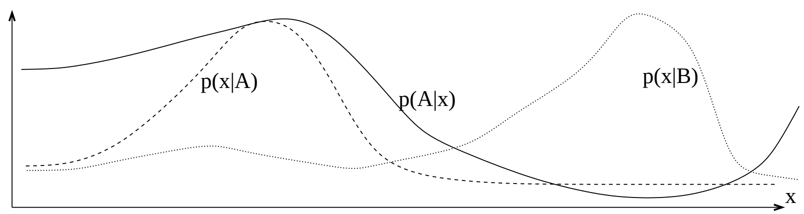

```
Subject: Daily St0ck Barometer against
Subject: real college girls Desmond
Subject: Delivery problems with your mail
Subject: FREE WEBCAM ACCESS - Better Than Phoney Sex - LIVE XXX CAMS 24 Hours!
Subject: Smart Solutions for your Pocket PC
Subject: OS-Adobe-Macromedia etc All under $15-$99 CDS
Subject: Rolex-factory Standard made wrist-watches
Subject: Force collectors to abide by Federal Law ,
Subject: Get an $80 Offer from Stamps.com today.
Subject: =?utf-8?B?T2ZmaWNlIFhQIC0gJDYwICBkZWxpbWl0cyBBbmdsb3Bob2JpYQ==?=
Subject: Buying the newest CDs over the net?
Subject: Get it up again
Subject: lasts for 36 hours
Subject: Buy cíalís wíthout embarrassment
Subject: Fw: Need soft_ ware? Cl- ick here.
Subject: Thank you for your loan request
Subject: Impress others, they will never know its not an Original-Rolex
Subject: we make it simple and quick
Subject: Now, it’s finally possible for you to enlarge your penis
Subject: fw:
Subject: All Windows software for cheap
Subject: it’s julie again -)
Subject: Top-level logo and business identity
Subject: pay less for Windows XP Professional
Subject: Doc"tors invent spe’rm. P|LLS                              ’
Subject: Friendly notification
Subject: Get a Bundle and we’ll give you the Book!
Subject: Once you go you’ll never stop.
Subject: This is what you’ve been wanting! johansen cyril
Subject: New Penny St0ck Idea For You befallen
Subject: FW:ÂËªº¦~«×¦n®Ñ¡I Learning-request
Subject: question
Subject: You left something the other night
Subject:  Hey.
Subject: shh come check out my secret. spi
Subject:  Hard as Rock McB
Subject: [Auto-Reply]   Hi there man-feel the power
Subject: Entrust your visual identity to us
Subject: Scream, get high, mix and match
Subject: Take action now, eliminate the threat.
Subject: Are you ready to get it?
Subject: AMATEUR TEEN WEBCAMS - FROM AROUND THE WORLD OR RIGHT NEXT DOOR
Subject: Premium online drvgs here
Subject: Re: in want is consistory
Subject: The Next Gangbuster Growth-stock? bread appalachia
Subject: There’s no reason why not have it.
Subject: For your information
Subject: Canada Day SALE
Subject: Jusst do it
Subject: O E M software
```

```
Subject: DÍSCOUNTED VÍAGRA
Subject: Sundays will never be the same
Subject:  Highly Recommended Ciialis aIg
Subject: I NEED YOUR ATTENTION ON THIS
Subject: New product! Cialis soft tabs.
Subject: Selling Tip of The Week June 30, 2005
Subject: Delivery Failure (roweis@cs.utoronto.ca)
Subject: let us save you money
Subject: best deals all day long
Subject:  ...
Subject: Exclusive-Offers for Quality-WristWatches
Subject: Today’s Cribsheet
Subject: Hot Stoxs in Play
Subject: i lost 100 pounds and completely changed my life.C
Subject:  Don’t Buy Viia-gra ix6r
Subject: Anvanced Penile Medication
Subject: Phamacy Online               incompetent
Subject: Circuits: Awkward Acronyms
Subject: Windows + Office only $80
Subject: =?iso-8859-1?B?UG9wdWxhciBzb2Z0IC0gd2hvbGVzYWxlIHByaWNl?=
Subject: Rolex-factory Standard made wrist-watches
Subject: We owe you: $90922
Subject: The Premier Investor Reports akin clayton
Subject: Small Cap Insight
Subject: Cheapest dru*gs on the net... guaranteed.
Subject: Get Víagra Onlíne Cheap! Ínternet Specíal!
Subject: perseus
Subject: The best you may make for is to be the #1 lover.
Subject: Play for real money
Subject: Cia-llis Softabs is the Best fmifV
Subject: Never leave your house backtrack
Subject: A marriage without love making isn’t a marriage at all.
Subject: It isn’t too good to be true. electrophorus
Subject: Re.Your bill
Subject: Let me help you out
Subject: Cure premature ejaculation
Subject: Exclusive notice
Subject: Confidence is back
Subject: Re: Details
Subject: real players
Subject: Make your rivals envy
Subject: All RX drugs sent out within one business day. irresistible
Subject: Phamacy Online               rollins
Subject: It’s so easy - find out yourself!
Subject: No worry by guile
Subject: New product! Cialis soft tabs.
Subject: beer makes your penis droop - brewers droop
Subject: soft at incredibly low prices qqf
Subject: Exclusive-Offers for Quality-WristWatches
Subject: she loves you rsvp
```

Reminder: Classification

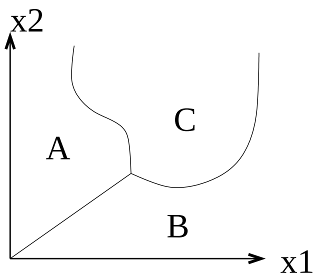
- Multiple inputs $\mathbf{x}$ (can be continuous, discrete or both).
- Single discrete output y.
- Goal: predict output on future unseen inputs.
- From a probabilistic point of view, we are using *Bayes rule*:

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y)p(y)}{\sum_{y'} p(\mathbf{x}|y')p(y')}$$



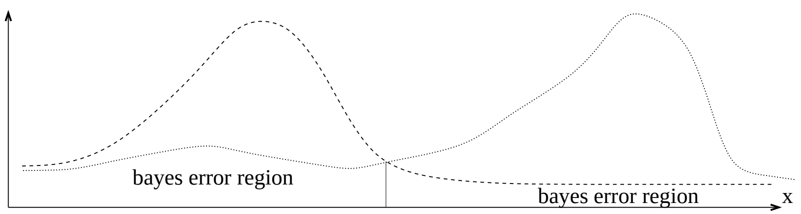
Voronoi Tessellation, Decision Surfaces

- For continuous inputs, we can view the problem as one of segmenting the input space into regions which belong to a single class, i.e. constant output.
- Such a segmentation is the "Voronoi tessellation" for our classifier.
- The boundaries between regions are the "decision surfaces".
- Training a classifier == defining decision surfaces.



Probabilistic Model, Bayes Error Rate

- Model original data as coming from joint pdf $p(\mathbf{x}, y)$. Classification == trying to learn conditional density $p(y|\mathbf{x})$.
- Even if we get the perfect model, our error rate may not be zero. Why? Classes may overlap.
- The best we could ever do if our cost function is number of errors is to guess $y^* = \mathrm{argmax}_y \; p(y|\mathbf{x})$.
  (The error rate of this procedure is known as the "Bayes error".)



K-Nearest-Neighbour

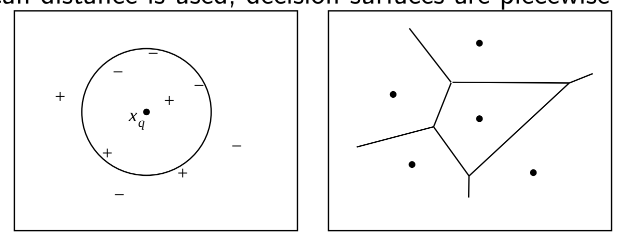
- Finally: a real algorithm!
- To classify a test point, chose the most common class amongst its K nearest neighbours in the training set.
- **Algorithm K-NN**

```
c-test ← KNN(K,x-train,c-train,x-test)   {
d(m,n) = distance between x-train(m) and x-test(n)
n(n,l) = index of l-th smallest entry of d(:,n) [*]
c(n,l) = c-train(n(n,l))
c-test(n) = most common value in c(n,1:K) [**]            }
```

- If ties at * when $l = K$, increase K for that n only.
- If ties at **, decrease K for that n only.
- confidence ≈ (#votes for class) / K
- Q: How should we select K? A: Cross-Validation (coming soon).

More on K-NN

- Typical distance = squared Euclidean $d(m, n) = \sum_d (x_d^m - x_d^n)^2$
- If Euclidean distance is used, decision surfaces are piecewise linear.



- Trick: remember the K^{th} smallest distance so far, and break out of the summation over dimensions if you exceed it.
- In low-d with lots of training points you can build "KD trees", "ball trees" or other data structures to speed up the query time.
- In high-d, save time by computing the distance of each training point from the min corner and using the "annulus bound".

Error Bounds for NN 8

- Amazing fact: asymptotically, err(1-NN) < 2 err(Bayes):

$$e_B \leq e_{1_{NN}} \leq 2e_B - \frac{M}{M-1}e_B^2$$

this is a tight upper bound, achieved in the "zero-information" case when the classes have identical densities.

- For K-NN there are also bounds. e.g. for two classes and odd K:

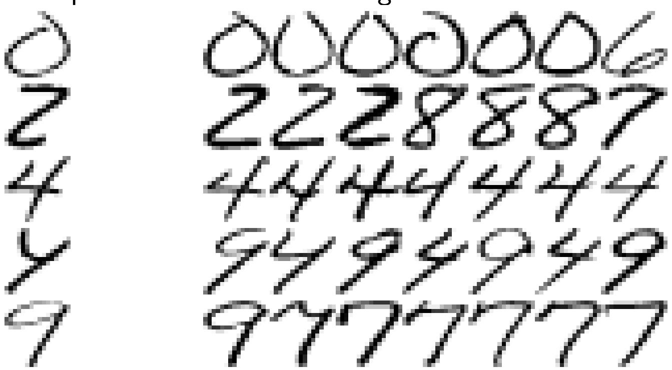
$$e_B \leq e_{K_{NN}} \leq \sum_{i=0}^{(K-1)/2} \binom{k}{i} \left[e_B^{i+1}(1-e_B)^{k-i} + e_B^{k-i}(1-e_B)^{i+1} \right]$$

- For more on these bounds, see the book *A Probabilistic Theory of Pattern Recognition*, by L. Devroye, L. Gyorfi & G. Lugosi (1996).

Example: USPS Digits 9

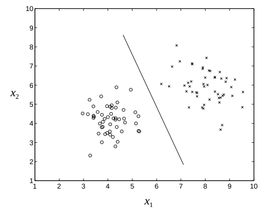
- Take 16x16 grayscale images (8bit) of handwritten digits.
- Use Euclidean distance in raw pixel space (dumb!) and 7-nn.
- Classification error (leave-one-out): 4.85%.

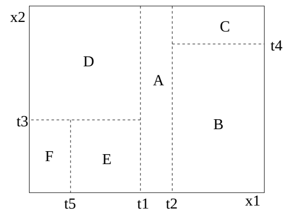
Example 7 Nearest Neighbours



Nonparametric (Instance-Based) Models 10

- Q: What are the parameters in K-NN? What is the complexity?
A: the scalar K *and the entire training set*.
Models which need the entire training set at test time but (hopefully) have very few other parameters are known as *nonparametric*, *instance-based* or *case based*.
- What if we want a classifier that uses only a small number of parameters at test time? (e.g. for speed or memory reasons)
Idea 1: single linear boundary, of arbitrary orientation
Idea 2: many boundaries, but axis-parallel & tree structured



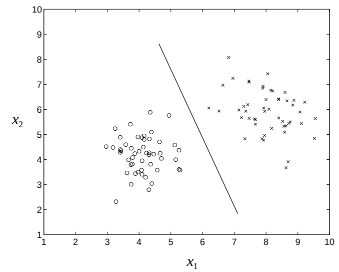


Linear Classification for Binary Output 11

- Goal: find the line (or hyperplane) which best separates two classes:

$$c(x) = \mathrm{sign}[\mathbf{x}^\top \underbrace{\mathbf{w}}_{weight} - \underbrace{w_0}_{threshold}]$$

- $\mathbf{w}$ is a vector perpendicular to decision boundary
- This is the opposite of non-parametric: only $d+1$ parameters!
- Typically we augment $\mathbf{x}$ with a constant term ± 1 ("bias unit") and then absorb w_0 into $\mathbf{w}$, so we don't have to treat it specially.

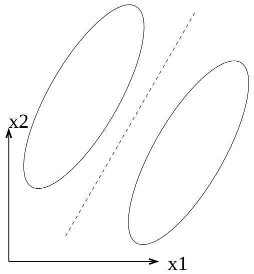


Fisher's Linear Discriminant

- Observation: If each class has a Gaussian distribution (with same covariances) then the Bayes decision boundary is linear:

$$\mathbf{w}^* = \Sigma^{-1}(\mu_0 - \mu_1)$$
$$w_0^* = \frac{1}{2}\mathbf{w}^\top(\mu_0 + \mu_1) - \mathbf{w}^\top(\mu_0 - \mu_1)\left[\frac{\log p_0 - \log p_1}{(\mu_0 - \mu_1)^\top \Sigma^{-1}(\mu_0 - \mu_1)}\right]$$

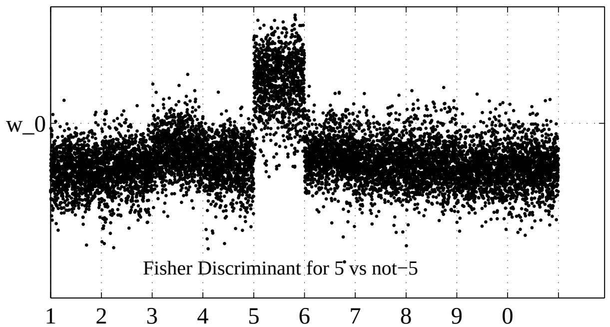
- Idea (Fisher'36):
  Assume each class is Gaussian even if they aren't!
  Fit μ_i and Σ as sample mean and sample covariance (shared).



- This also maximizes the ratio of *cross-class scatter* to *within class scatter*: $(\bar{z}_0 - \bar{z}_1)^2/(\mathrm{var}(z_0) - \mathrm{var}(z_1))$

Handwritten Digits again

Train to discriminant "5" from others.
Error = 3.59%

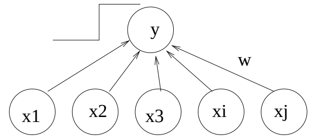


Perceptron Algorithm

- The architecture we are using

$$c(x) = \mathrm{sign}[\mathbf{x}^\top \mathbf{w} - w_0]$$

  can be thought of as a circuit/network.



- It was studied extensively in the 1960s and is known as a *perceptron*.
- There is another way to train the weights, other than Fisher.

Algorithm perceptronTrain (Rosenblatt'56)

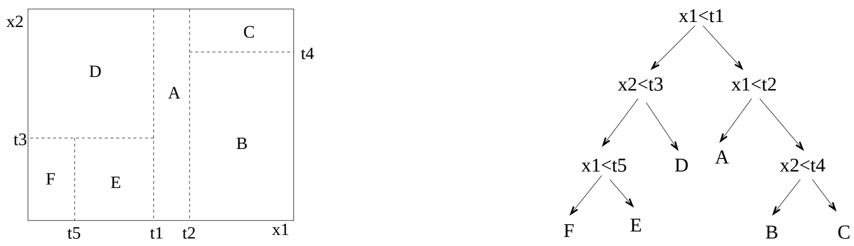
```
w ← perceptronTrain(x-train,c-train)   {
  w = ‘‘small’’ random values;
   do {   errors=0;
        for n=1:N {if(c-train(n) != sign[w’*xtrain(n)]) then {
        w = w + c - train(n)*xtrain(n);    errors++; } }
      } until(errors==0)
}
```


Perceptron Learning Rules

- Now: cycle through examples, when you make an error, add/subtract the example from the weight vector depending on its true class.
- Amazingly, for separable training sets, this always converges.
  (We absorb the threshold as a "bias" variable always equal to -1.)
- For non-separable datasets, you need to remember the sets of weights which you have seen so far, and combine them somehow.
- One way: keep the set that survived unchanged for the longest number of (random) pattern presentations. (Gallant's *pocket algorithm*.)
- Better way: Freund & Shapire's *voted perceptron* algorithm.
  Remember all sets and the length of time they survived.
- Perceptron, voted-perceptron, weighted-majority, kernel perceptron, Winnow, and other algorithms have a frumpy reputation but they are actually extremely powerful and useful, especially using the kernel trick. Try these before more complex classifiers such as SVMs!

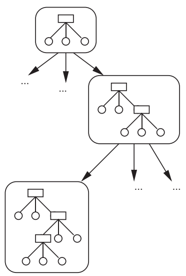
Decision Trees 16

- What if we want more than two regions?
- We could consider a fixed number of arbitrary linear segments but even cheaper is to use axis-aligned splits (one dimension each).
- If these form a hierarchical partition, then the classifier is called a *decision tree* or (axis-aligned) *classification tree*.
- Each internal node tests one attribute; leaves assign a class.
- Equivalent to a disjunction of conjunctions of constraints on attribute values (if-then rules).



Cost Function for Decision Trees 17

- Define a measure of "class impurity" in a set of examples. Push each example down the tree, how "pure" are leaves?
- Goal: minimize expected sum of impurity at leaves at test time.
- Two problems:
  1) We don't know true distribution $p(\mathbf{x}, y)$.
  2) Search: even if we knew $p(\mathbf{x}, y)$ finding optimal tree is NP.
- So we will take a suboptimal (greedy) approach.



Learning (Inducing) Decision Trees 18

- Need to pick the order of split axes and values of split points. Many algorithms: CART, ID3, C4.5, C5.0.
- Almost all have the following structure:
  1. Put all examples into the root node.
  2. At each node: search all dimensions, on each one chose split which most reduces impurity; chose the best split.
  3. Sort the data cases into the daughter nodes based on the split.
  4. Recurse until a leaf condition:
     – number of examples at node is too small
     – all examples at node have same class
     – all examples at node have same inputs
  5. Prune tree down to some maximum number of leaves. (Possibly using a different impurity measure than for growing.)

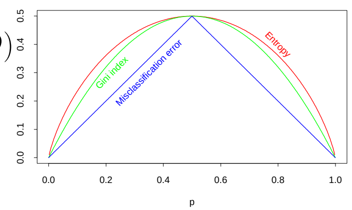
Impurity Measures 19

- When considering splitting data D at a node on x_i, we measure:

$$\mathrm{Gain}(D; x_i) = I(D) - \sum_{v \in split(x_i)} \frac{|D_{iv}|}{|D|} I(D_{iv})$$

- Common impurity measures:

Entropy: $I(D) = -\sum_c p_c(D) \log p_c(D)$ `(two classes)`
Misclass: $I(D) = 1 - p_{c^*}$
Gini: $I(D) = \sum_c \sum_{c' \neq c} p_c(D) p_{c'}(D)$
$= \sum_c p_c(D)(1 - p_c(D))$
(Gini is the avgerage error if we stochastically classify with node prior)

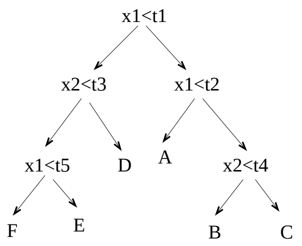


- These often favour multi-way splits.
- One solution: normalize by "split information":

$$S(D) = -\sum_v \frac{|D_{iv}|}{|D|} \log \frac{|D_{iv}|}{|D|}$$

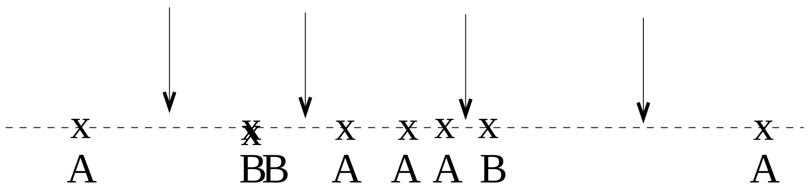
Restrict to Binary Splits

- A better solution is to always constrain ourselves to binary splits.
- For ordered discrete or real valued nodes, split is natural.
  Also easy to compute.
- For a discrete attribute with M settings, looks like we need to consider $2^M - 1$ splits. But for two classes, there is a trick:
  1. Order the settings according to $p(c|x_i = m)$.
  2. Search exhaustively over q, grouping first q and last $M - q$.
  3. Optimal split is one of those.



Real Valued Attributes

- For real valued attributes, what splits should we consider?
- Idea1: discretize the real value into M bins.
- Idea2: Search for a scalar value to split on.
  Sounds hard! Lots of real values. But there is a trick:
  Only need to consider splits at midpoints between observed values.
  In fact, only need to consider splits at midpoints between observed values with different classes.
- Complexity: $N \log N + 2N|C|$



Algorithm: DT


```
root of decision tree = SplitNode(train-data,nmin)

subtree ← SplitNode(D) {
c = most common class in D
if (all class(D) same) or (all x(D) same) or (size(D) < nmin)
then return a leaf of class c
else for each xi measure Gain(D;xi)
return a node which splits on best xi and has daughters:
- SplitNode(Div) for all split vals v with nonempty Div
- leaf of class c for values with empty Div                    }

G ← Gain(D,i) {
G = I(D)
for each value v in split(xi)
Div = cases in D with xi=v
G = G - I(Div)*size(Div)/size(D)                               }
```

Overfitting in Trees

- Just as with most other models, decision trees can overfit.
  In fact they are quite powerful.
- eg: Expressive power of binary trees
  Q: If all input and outputs are binary, what class of Boolean functions can DTs represent?
  A: All Boolean functions.
- Hence we must *regularize* to control capacity.
- Typically we do this by limiting the number of leaf nodes.
  Formally, we define: $\Phi(T) = \sum_{leaves} I(l) + \alpha|leaves|$.
- Minimizing this for any α is equivalent to finding the tree of a fixed size with smallest impurity. (cf. Lagrange multipliers).
- Practically, we achieve this via pruning.
  Often we use Gini/Entropy to grow tree and Misclass to prune it.

Pruning Decision Trees

- Finding the "optimal" pruned tree.
  It can be shown that if you start with a tree T_0 and insist on using a rooted subtree of it, the following sequence of trees contains the optimum tree for all numbers of leaves:
  1. Let U(node) = I(node)-I(subtree-rooted-at-node)
  2. Replace the non-leaf node with the smallest value of:
     U(node)/leaves-below-node
     with a leaf node having majority class.
- Even after pruning, decision trees still have problems:
  - cannot capture additive structure (OR), for this MARS is better
  - cannot deal with linear combinations of variables

DT Variants

- ID3 (Quinlan)
  - split values are all possible values of x_i
  - I(D) is entropy, no pruning
- C4.5, C5.0 (Quinlan)
  - binary splits
  - I(D) is entropy
  - error-pruning
  - "rule simplification"
- CART (Breiman et. al)
  - binary splits
  - I(D) is Gini
  - minimum-leaf subtree pruning

Open questions...

- How do we chose K in K-NN? (Cross-validation)
- How do we chose T_{max} for decision trees? (Cross-validation)
- Can Fisher's Discriminant overfit? (What do you think?)
- What about nearest-neighbour or tree-based models for regression as well as classification? (Good idea!)

Next class: Logistic regression, Neural Nets for Classification, Class-Conditional Models (Gaussian and Naive Bayes)