Machine Learning
Tutorial KNN and Decision Trees

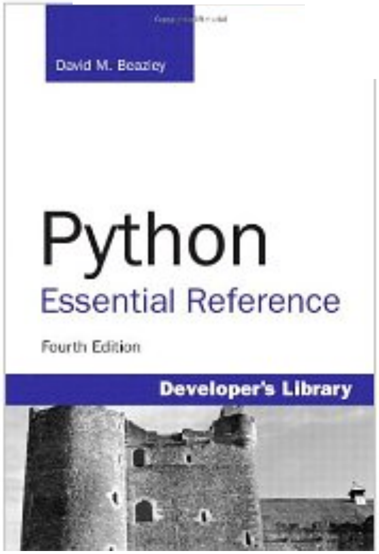
Suggestions for reading:

- KNN, probabilistic KNN : Barber, chapter 14
- NCA: original paper by Goldberg at al.
- Decision Trees: Murphy, chapter 16.2

Python and IPython Books

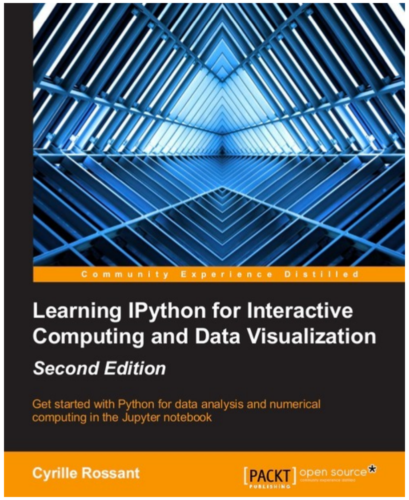
- **Learning Python:**
  Python Essential Reference (2012)
  by David M. Beazley, Safari Books
  (especially **chapter 1: A Tutorial Introduction (25 pages))**

  free eAccess: https://eaccess.ub.tum.de/login



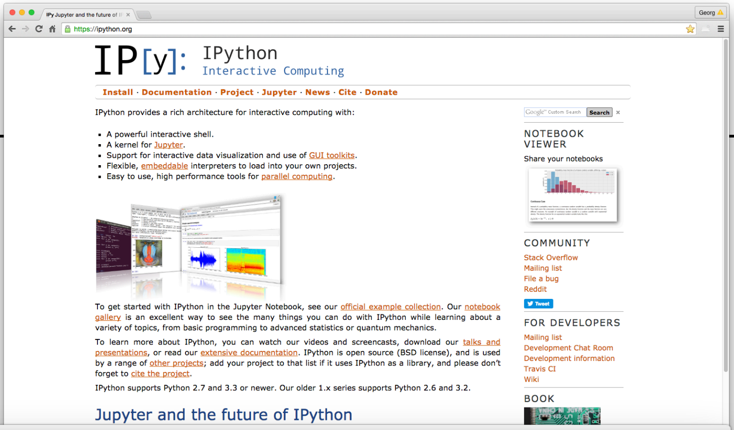
- **Learning IPython / Reference for IPython:**
  Learning IPython for Interactive Computing and Data Visualization (SECOND EDITION) by Cyrille Rossant, 175 pages, Packt Publishing, October 25 2015
  (Especially (free) chapter 1.4. A crash course on Python)

  free access: http://nbviewer.ipython.org/github/ipython-books/minibook-2nd-code/blob/master/chapter1/14-python.ipynb
  (do not try to open this ipynb with Jupyter directly. Instead, download all the ipynb's from the book from Github: https://github.com/ipython-books/minibook-2nd-code → 14-python.ipynb)


Community Experience Distilled
Learning IPython for Interactive Computing and Data Visualization
Second Edition
Get started with Python for data analysis and numerical computing in the Jupyter notebook
Cyrille Rossant
[PACKT] open source*

IPython Website

- **https://ipython.org/** **or** https://github.com/ipython-books/minibook-2nd-code
- → Installation of IPython / Jupyter:
  Anaconda – package (has numpy / scipy preinstalled)
- run :
  jupyter notebook
  → notebook appears in browser:


IPy Jupyter and the future of IF
Georg
https://ipython.org
IP[y]: IPython
Interactive Computing
Install · Documentation · Project · Jupyter · News · Cite · Donate
Google Custom Search
Search

IPython provides a rich architecture for interactive computing with:

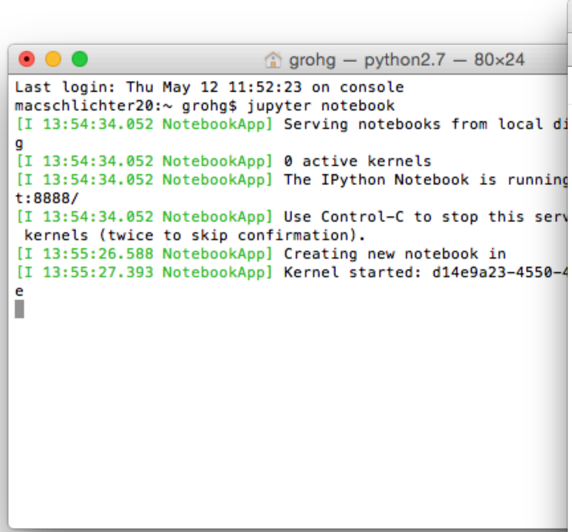
- A powerful interactive shell.
- A kernel for Jupyter.
- Support for interactive data visualization and use of GUI toolkits.
- Flexible, embeddable interpreters to load into your own projects.
- Easy to use, high performance tools for parallel computing.

To get started with IPython in the Jupyter Notebook, see our official example collection. Our notebook gallery is an excellent way to see the many things you can do with IPython while learning about a variety of topics, from basic programming to advanced statistics or quantum mechanics.

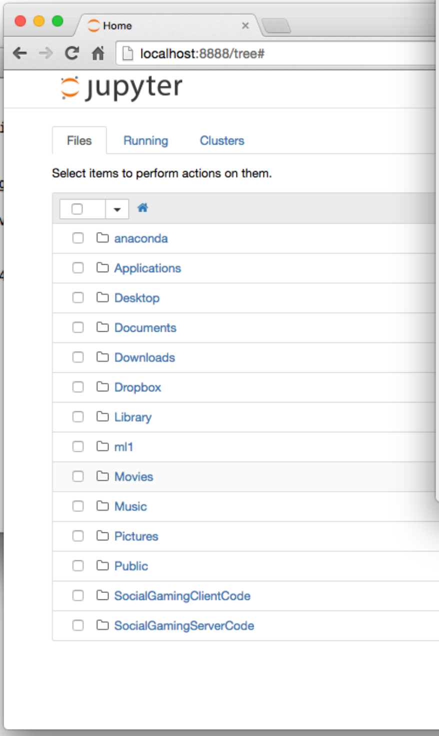
To learn more about IPython, you can watch our videos and screencasts, download our talks and presentations, or read our extensive documentation. IPython is open source (BSD license), and is used by a range of other projects; add your project to that list if it uses IPython as a library, and please don't forget to cite the project.

IPython supports Python 2.7 and 3.3 or newer. Our older 1.x series supports Python 2.6 and 3.2.

Jupyter and the future of IPython

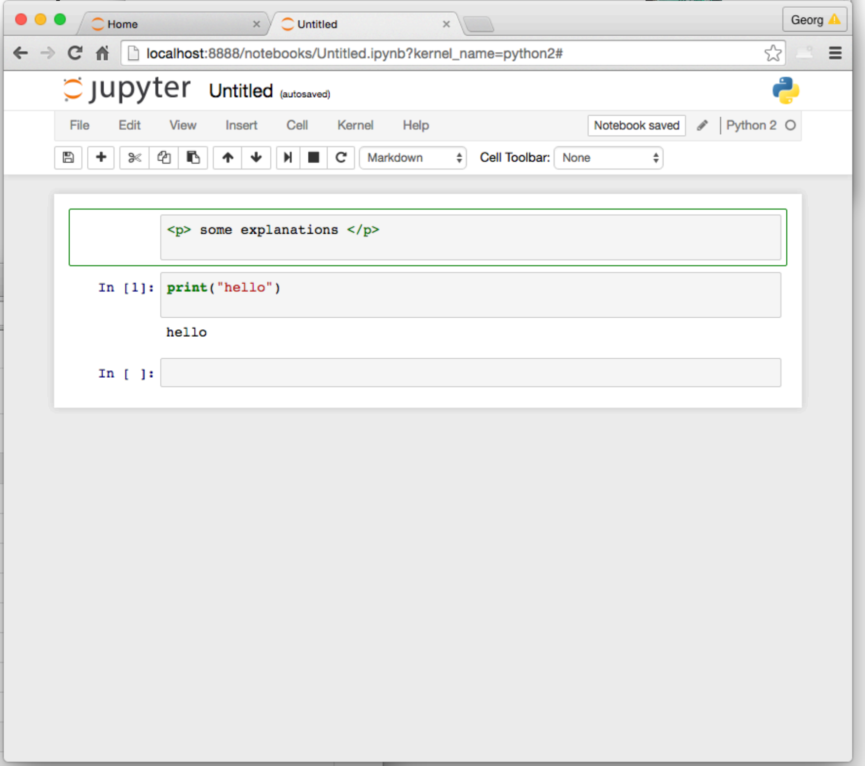


```
Last login: Thu May 12 11:52:23 on console
macschlichter20:~ grohg$ jupyter notebook
[I 13:54:34.052 NotebookApp] Serving notebooks from local di
g
[I 13:54:34.052 NotebookApp] 0 active kernels
[I 13:54:34.052 NotebookApp] The IPython Notebook is running
t:8888/
[I 13:54:34.052 NotebookApp] Use Control-C to stop this serv
 kernels (twice to skip confirmation).
[I 13:55:26.588 NotebookApp] Creating new notebook in
[I 13:55:27.393 NotebookApp] Kernel started: d14e9a23-4550-4
e
```



Select items to perform actions on them.

- anaconda
- Applications
- Desktop
- Documents
- Downloads
- Dropbox
- Library
- ml1
- Movies
- Music
- Pictures
- Public
- SocialGamingClientCode
- SocialGamingServerCode



```
<p> some explanations </p>

In [1]: print("hello")

        hello

In [ ]:
```

Python: Crash Course

- Recommendation if unfamiliar with python: Work through [Beazley 2013] Part 1, chapter 1: A Tutorial Introduction (25 pages) or [Rossant 2015] 1.4. A crash course on Python (an ipynb interactive IPython-Notebook)

- http://wiki.python.org/moin/BeginnersGuide/Programmers
  is a list of other Python tutorials and books.
- Python go-to-Webpage: http://www.python.org
- NumPy go-to-Webpage: http://www.scipy.org
- NumPy basics → NumPy User Guide.
  NumPy library reference → NumPy Reference Guide
  Both documents available at http://docs.scipy.org/doc/

Problems w.r.t. Decision Trees

Problem 1: Build a decision tree for the dataset $\mathcal{D}^1$ below.

Day	Outlook	Temperature	Humidity	Wind	Play Tennis?
D1	sunny	hot	high	weak	No
D2	sunny	hot	high	strong	No
D3	overcast	hot	high	weak	Yes
D4	rain	mild	high	weak	Yes
D5	rain	cool	normal	weak	Yes
D6	rain	cool	normal	strong	No
D7	overcast	cool	normal	strong	Yes
D8	sunny	mild	high	weak	No
D9	sunny	cool	normal	weak	Yes
D10	rain	mild	normal	weak	Yes
D11	sunny	mild	normal	strong	Yes
D12	overcast	mild	high	strong	Yes
D13	overcast	hot	normal	weak	Yes
D14	rain	mild	high	strong	No

Use the *ID3* algorithm. In contrast to CART, ID3 allows for multiway splits and exhausts all possible values for a feature when that feature is chosen for a split. The criterion that determines the best split is information gain, which in turn is based on entropy.

Problem 1: Build a decision tree for the dataset $\mathcal{D}^1$ below.

Day	Outlook	Temperature	Humidity	Wind	Play Tennis?
D1	sunny	hot	high	weak	No
D2	sunny	hot	high	strong	No
D3	overcast	hot	high	weak	Yes
D4	rain	mild	high	weak	Yes
D5	rain	cool	normal	weak	Yes
D6	rain	cool	normal	strong	No
D7	overcast	cool	normal	strong	Yes
D8	sunny	mild	high	weak	No
D9	sunny	cool	normal	weak	Yes
D10	rain	mild	normal	weak	Yes
D11	sunny	mild	normal	strong	Yes
D12	overcast	mild	high	strong	Yes
D13	overcast	hot	normal	weak	Yes
D14	rain	mild	high	strong	No

Use the *ID3* algorithm. In contrast to CART, ID3 allows for multiway splits and exhausts all possible values for a feature when that feature is chosen for a split. The criterion that determines the best split is information gain, which in turn is based on entropy.

The information gain of an attribute A at node t is given by $\Delta i(A, t)$. In contrast to be before, there might be more than two outcomes for a feature test, i.e., there are as many branches as there are number of values for attribute A.

$$\Delta i(A, t) = i_H(t) - \sum_{v \in \text{values}(A)} p(A = v) i_H(t_v)$$

where t_v denotes the subset of samples in t where A takes value v and

$$i_H(t) = -\left[p(\text{Tennis} = \text{Yes}) \log p(\text{Tennis} = \text{Yes}) + p(\text{Tennis} = \text{No}) \log p(\text{Tennis} = \text{No})\right].$$

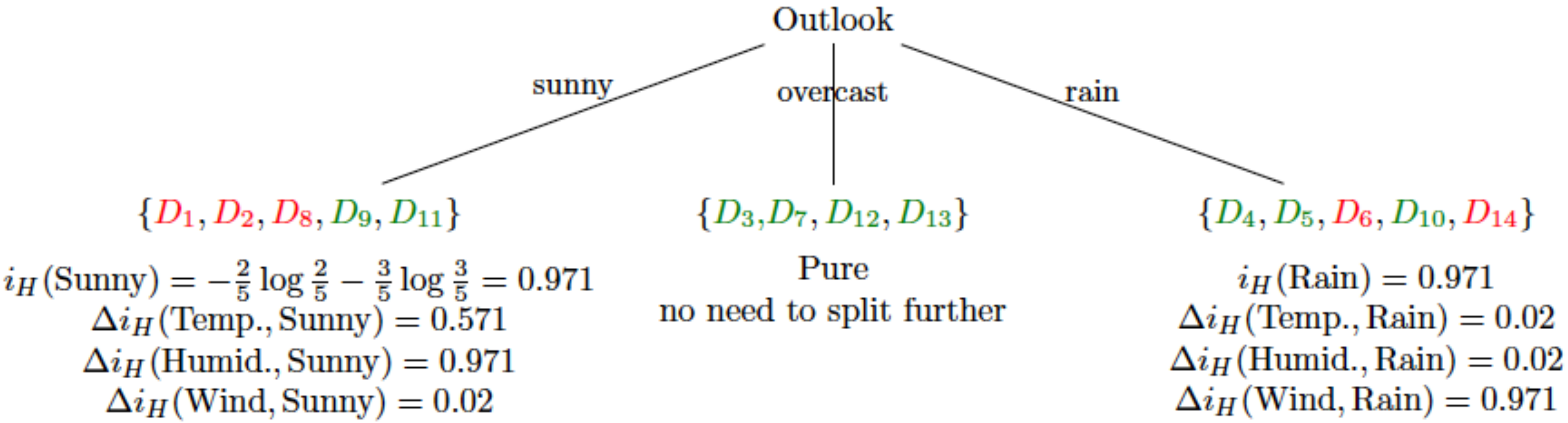
We start out with

$$i_H(\mathcal{D}) = -\frac{9}{14}\log\frac{9}{14} - \frac{5}{14}\log\frac{5}{14} = 0.940286$$

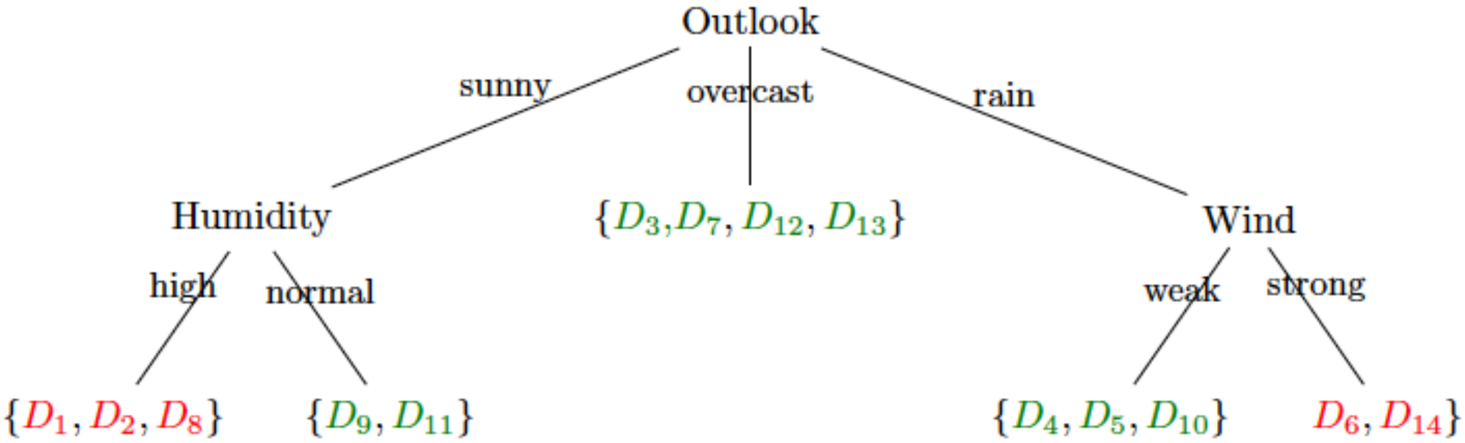
The information gain for the attributes Outlook, Temperature, Humidity and Wind respectively calculates as:

$$\begin{aligned}
\Delta i_H(\text{Temp.}, 0) &= i_H(\mathcal{D}) - \Big(\underbrace{\frac{4}{14}(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2})}_{\text{Temp. = hot}} + \underbrace{\frac{6}{14}(-\frac{4}{6}\log\frac{4}{6} - \frac{2}{6}\log\frac{2}{6})}_{\text{Temp. = mild}} + \underbrace{\frac{4}{14}(-\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4})}_{\text{Temp. = cool}}\Big) \\
&= 0.0292 \\
\Delta i_H(\text{Humid.}, 0) &= i_H(\mathcal{D}) - \Big(\underbrace{\frac{7}{14}(-\frac{4}{7}\log\frac{4}{7} - \frac{3}{7}\log\frac{3}{7})}_{\text{Humid. = high}} + \underbrace{\frac{7}{14}(-\frac{6}{7}\log\frac{6}{7} - \frac{1}{7}\log\frac{1}{7})}_{\text{Humid. = normal}}\Big) \\
&= 0.1518 \\
\Delta i_H(\text{Wind}, 0) &= i_H(\mathcal{D}) - \Big(\underbrace{\frac{8}{14}(-\frac{6}{8}\log\frac{6}{8} - \frac{2}{8}\log\frac{2}{8})}_{\text{Wind = weak}} + \underbrace{\frac{6}{14}(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2})}_{\text{Wind = strong}}\Big) \\
&= 0.0481 \\
\Delta i_H(\text{Outlook}, 0) &= i_H(\mathcal{D}) - \Big(\underbrace{\frac{5}{14}(-\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5})}_{\text{Outlook = sunny}} + \underbrace{\frac{4}{14}(-1\log 1 - 0\log 0)}_{\text{Outlook = overcast}} \underbrace{\frac{5}{14}(-\frac{3}{5}\log\frac{3}{5} - \frac{2}{4}\log\frac{2}{4})}_{\text{Outlook = rain}}\Big) \\
&= 0.2468
\end{aligned}$$

The first attribute to split on is therefore the attribute Outlook. This split leads to the following intermediate result:


Outlook
sunny
overcast
rain
$\{D_1, D_2, D_8, D_9, D_{11}\}$
$\{D_3, D_7, D_{12}, D_{13}\}$
$\{D_4, D_5, D_6, D_{10}, D_{14}\}$
$i_H(\text{Sunny}) = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} = 0.971$
$\Delta i_H(\text{Temp.}, \text{Sunny}) = 0.571$
$\Delta i_H(\text{Humid.}, \text{Sunny}) = 0.971$
$\Delta i_H(\text{Wind}, \text{Sunny}) = 0.02$
Pure
no need to split further
$i_H(\text{Rain}) = 0.971$
$\Delta i_H(\text{Temp.}, \text{Rain}) = 0.02$
$\Delta i_H(\text{Humid.}, \text{Rain}) = 0.02$
$\Delta i_H(\text{Wind}, \text{Rain}) = 0.971$

With the information gains in the children nodes Sunny and Rain, the final tree for $\mathcal{D}$ is



Original slides from Tom Mitchell (CMU):

Top-Down Induction of Decision Trees

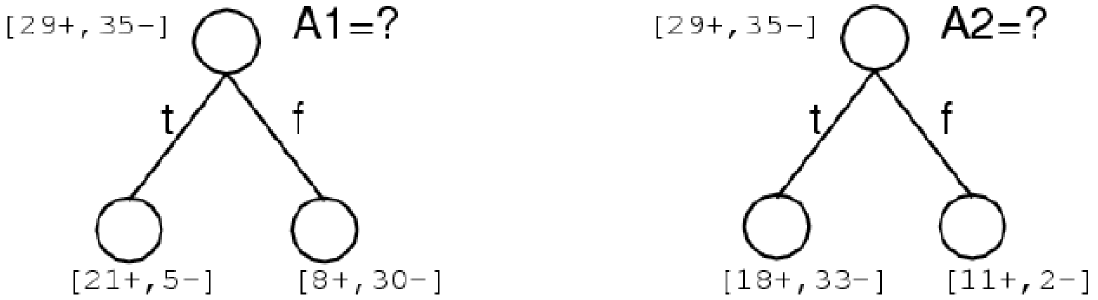
[ID3, C4.5, Quinlan]

node = Root

Main loop:

1. $A \leftarrow$ the “best” decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?



source: slides on ID3: Tom Mitchell (Carnegie Mellon): ML, Spring 2011

Top-Down Induction of Decision Trees

[ID3, C4.5, Quinlan]

node = Root

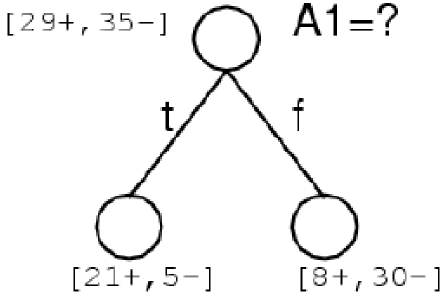
Main loop:

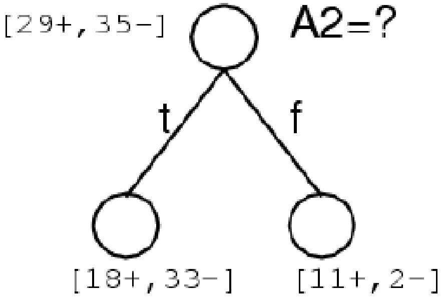
1. $A \leftarrow$ the “
2. Assign A a
3. For each va
   node
4. Sort trainin
5. If training
   STOP, Els

Remark: This algorithm is just a special version of the algorithm 16.1 in Murphy

(algorithm 16.1. in Murphy encompasses CART, ID3, C4.5…)

Which attribute is best?





source: slides on ID3: Tom Mitchell (Carnegie Mellon): ML, Spring 2011

Entropy

Entropy *H(X)* of a random variable *X*

# of possible values for X

$$\boxed{H(X) = -\sum_{i=1}^{n} P(X=i)\log_2 P(X=i)}$$

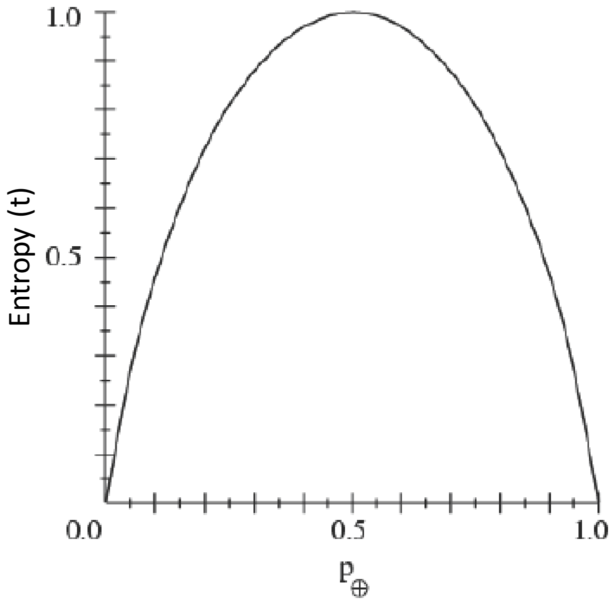
H(X) is the expected number of bits needed to encode a randomly drawn value of *X* (under most efficient code)

Why? Information theory:

- Most efficient code assigns $-\log_2 P(X=i)$ bits to encode the message $X=i$
- So, expected number of bits to code one random *X* is:

$$\sum_{i=1}^{n} P(X=i)(-\log_2 P(X=i))$$

Sample Entropy



- t is a sample of training examples
- $p_{\oplus}$ is the proportion of positive examples in t
- $p_{\ominus}$ is the proportion of negative examples in t
- Entropy measures the impurity of t

$$H(\mathrm{t}) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Entropy

Entropy $H(X)$ of a random variable X

$$\boxed{H(X) = -\sum_{i=1}^{n} P(X=i) \log_2 P(X=i)}$$

Specific conditional entropy $H(X|Y=v)$ of X given $Y=v$:

$$H(X|Y=v) = -\sum_{i=1}^{n} P(X=i|Y=v) \log_2 P(X=i|Y=v)$$

Conditional entropy $H(X|Y)$ of X given Y :

$$H(X|Y) = \sum_{v \in values(Y)} P(Y=v) H(X|Y=v)$$

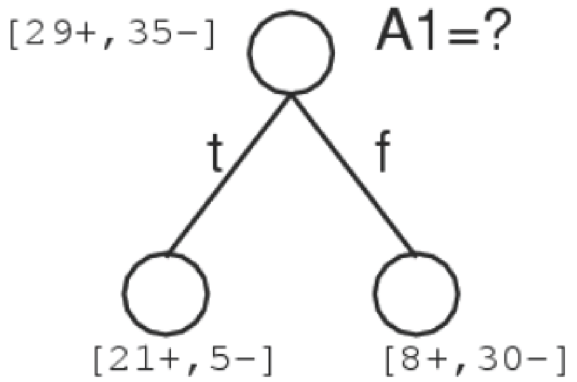
Mututal information (aka Information Gain) of X and Y :

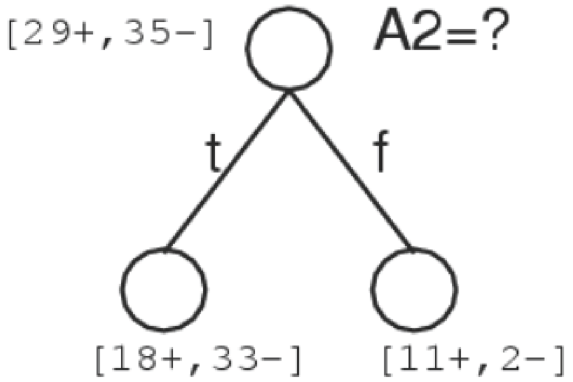
$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Information Gain is the mutual information between input attribute A and target variable Y

Information Gain is the expected reduction in entropy of target variable Y for data sample t due to sorting on variable A

$$Gain(t, A) = I_t(A, Y) = H_t(Y) - H_t(Y|A)$$



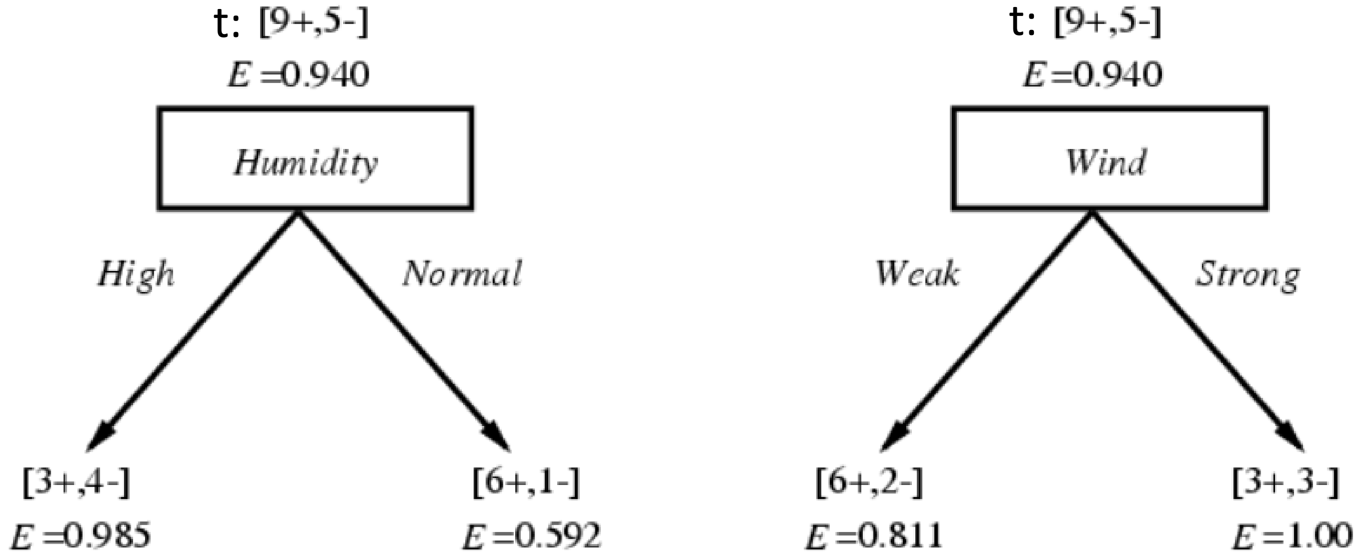


Training Examples

Day	Outlook	Temperature	Humidity	Wind	PlayTen
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

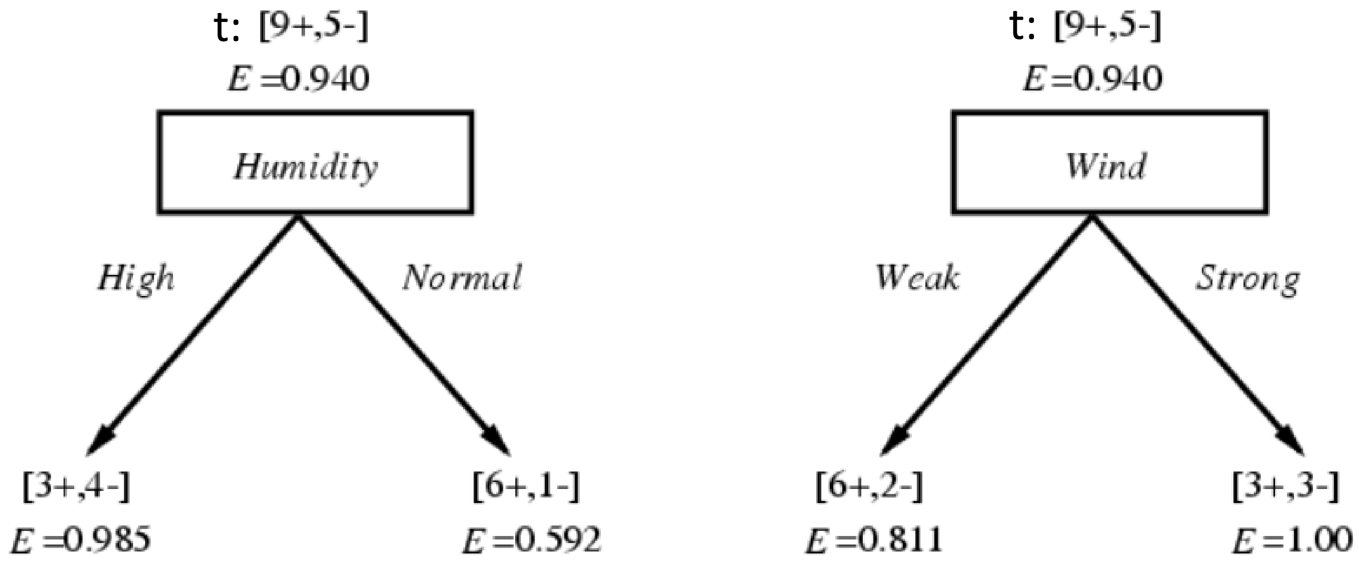
Selecting the Next Attribute

Which attribute is the best classifier?



Selecting the Next Attribute

Which attribute is the best classifier?



t: [9+,5-]

E =0.940

Humidity

High

Normal

[3+,4-]

E =0.985

[6+,1-]

E =0.592

t: [9+,5-]

E =0.940

Wind

Weak

Strong

[6+,2-]

E =0.811

[3+,3-]

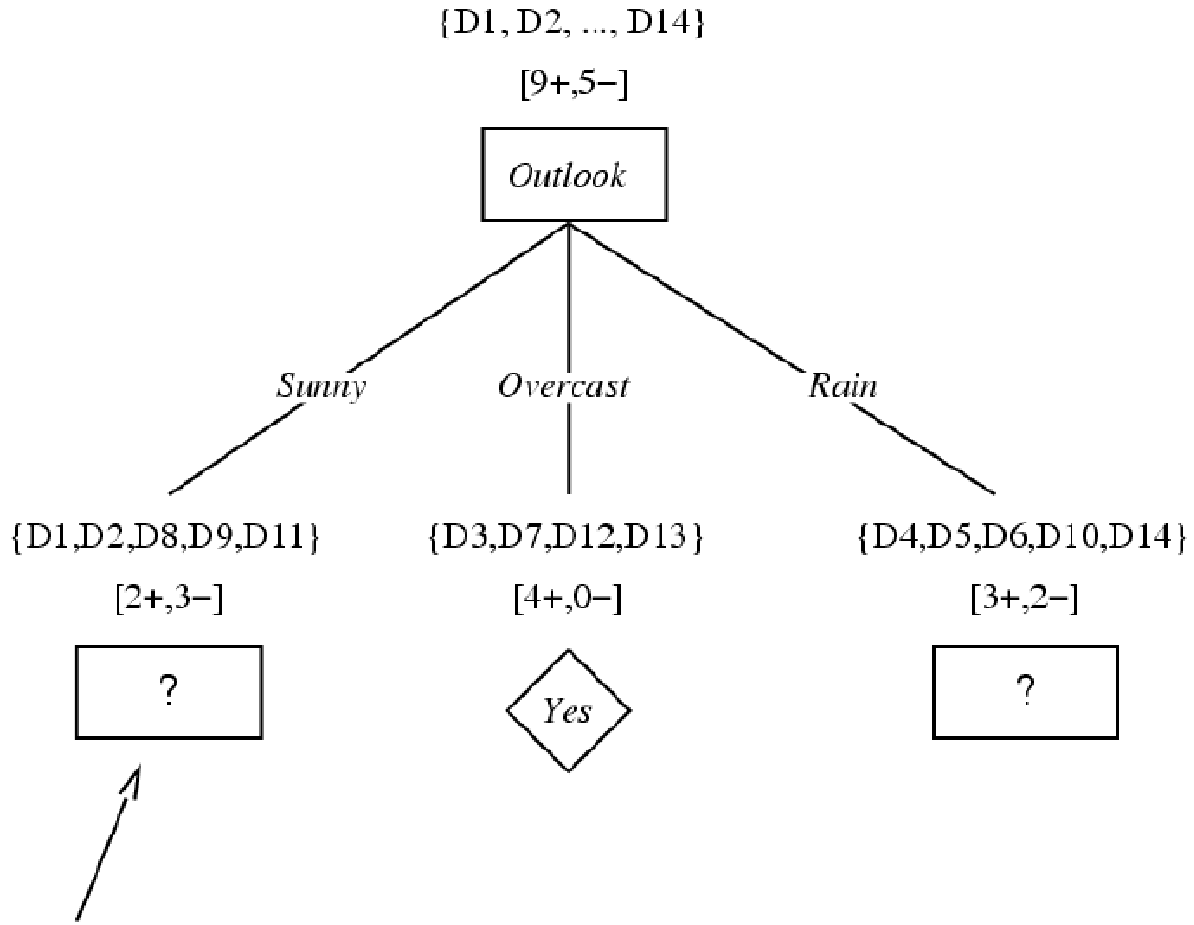
E =1.00

Gain (t , *Humidity*)

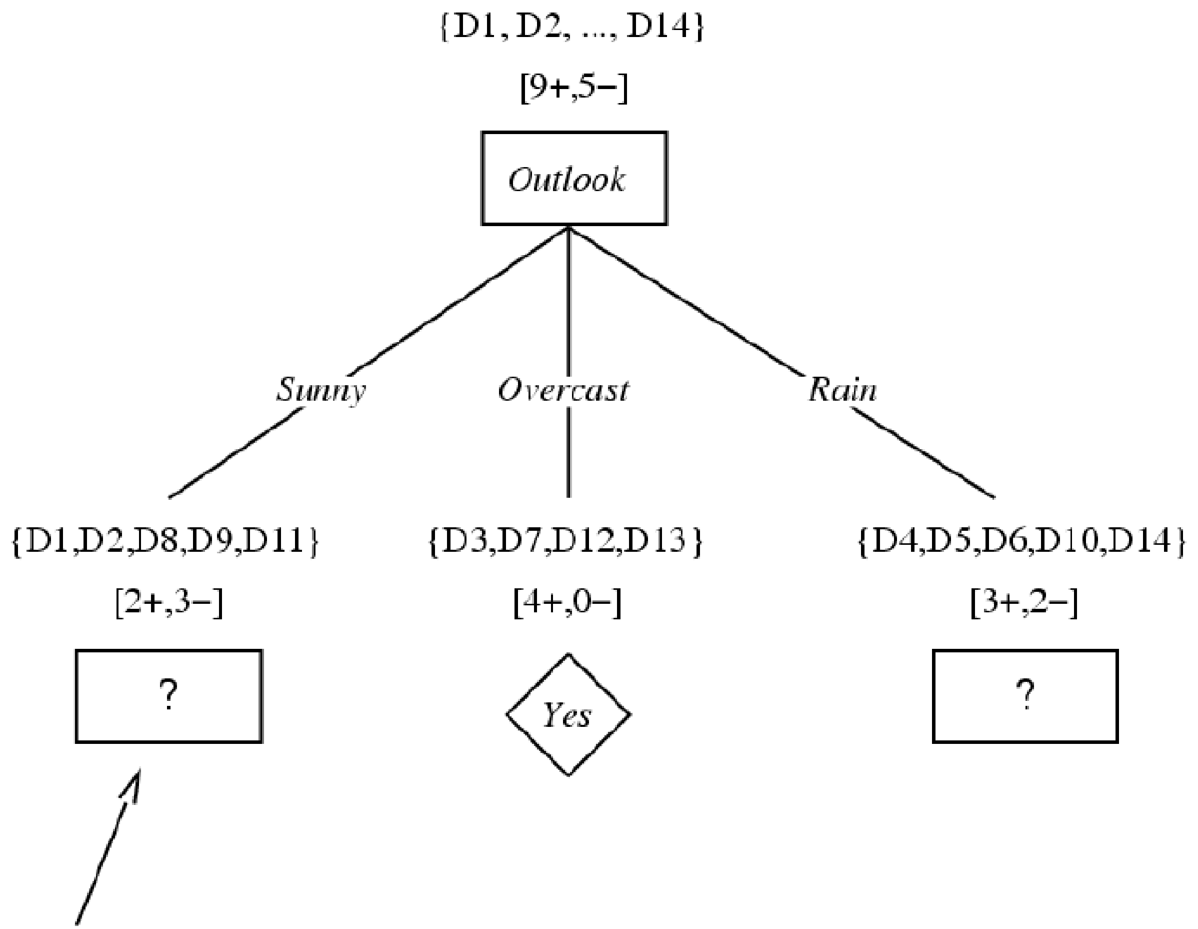
= .940 - (7/14).985 - (7/14).592
= .151

Gain (t ,*Wind*)

= .940 - (8/14).811 - (6/14)1.0
= .048



Which attribute should be tested here?



{D1, D2, ..., D14}

[9+,5−]

Outlook

Sunny *Overcast* *Rain*

{D1,D2,D8,D9,D11}

[2+,3−]

?

{D3,D7,D12,D13}

[4+,0−]

Yes

{D4,D5,D6,D10,D14}

[3+,2−]

?

Which attribute should be tested here?

$t_{sunny} = \{D1, D2, D8, D9, D11\}$

$Gain\,(t_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 = .970$

$Gain\,(t_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

$Gain\,(t_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$

end of Tom Mitchell’s slides

Problem 2: We consider decision trees for a two-class classification problem with classes 0 and 1. Let $\Phi(p,q)$ be a strictly concave function defined on $0 \leq p, q \leq 1$ such that

- $\Phi(1,0) = \Phi(0,1)$ is minimal;
- $\Phi(\frac{1}{2}, \frac{1}{2})$ is maximal.

Then, for $i(t) = \Phi(p(c=0 \mid t), p(c=1 \mid t))$, $\Delta i(s,t) = i(t) - p_R\, i(t_R) - p_L\, i(t_L)$ and any split s, show that

$$\Delta i(s,t) \geq 0,$$

with equality if and only if $p(c=i \mid t) = p(c=i \mid t_L) = p(c=i \mid t_R)$ for both $i = 0, 1$[2].

Hint: Strict concavity for $\Phi(p,q)$ means that for p_1, q_1, p_2, q_2 and $\alpha \in [0,1]$

$$\Phi((1-\alpha)p_1 + \alpha p_2, (1-\alpha)q_1 + \alpha q_2) > (1-\alpha)\Phi(p_1,q_1) + \alpha\Phi(p_2,q_2)$$

[2] adapted from G. Louppe. 2014. Understanding Random Forests. PhD Thesis

Problem 2: We consider decision trees for a two-class classification problem with classes 0 and 1. Let $\Phi(p, q)$ be a strictly concave function defined on $0 \leq p, q \leq 1$ such that

- $\Phi(1, 0) = \Phi(0, 1)$ is minimal;
- $\Phi(\frac{1}{2}, \frac{1}{2})$ is maximal.

Then, for $i(t) = \Phi(p(c = 0 \mid t), p(c = 1 \mid t))$, $\Delta i(s, t) = i(t) - p_R\, i(t_R) - p_L\, i(t_L)$ and any split s, show that

$$\Delta i(s, t) \geq 0,$$

with equality if and only if $p(c = i \mid t) = p(c = i \mid t_L) = p(c = i \mid t_R)$ for both $i = 0, 1$[2].

Hint: Strict concavity for $\Phi(p, q)$ means that for p_1, q_1, p_2, q_2 and $\alpha \in [0, 1]$

$$\Phi((1-\alpha)p_1 + \alpha p_2, (1-\alpha)q_1 + \alpha q_2) > (1-\alpha)\Phi(p_1, q_1) + \alpha\Phi(p_2, q_2)$$

Let us first remark that

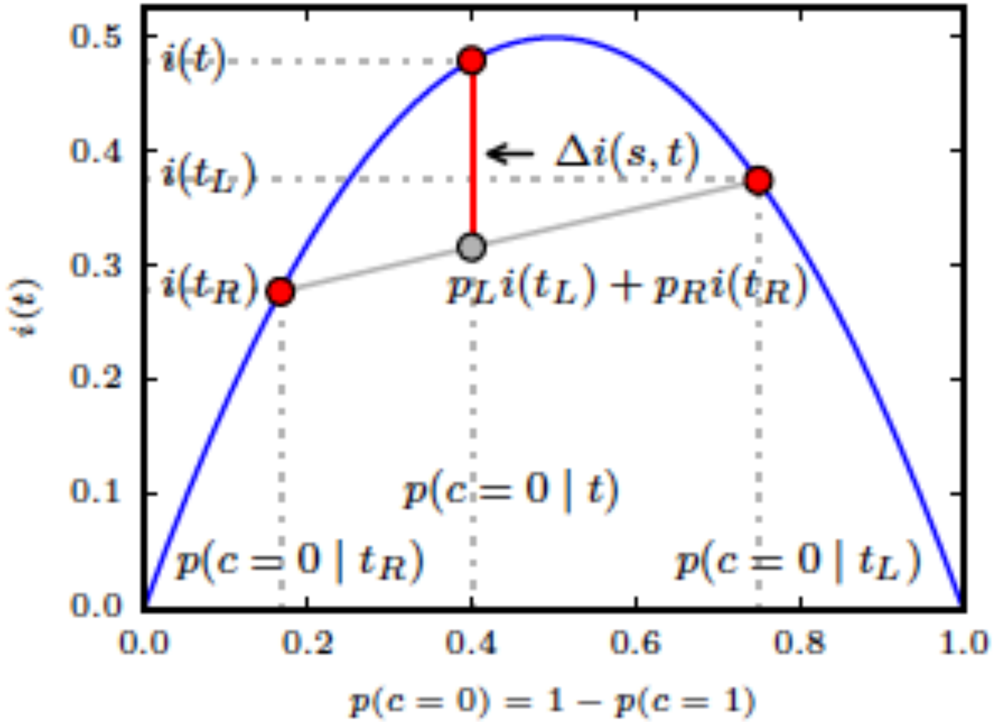
$$p(c = i \mid t) = \frac{N_{c_i t}}{N_t} = \frac{N_{c_i t_L} + N_{c_i t_R}}{N_t} = \frac{N_{t_L}}{N_t}\frac{N_{c_k t_L}}{N_{t_L}} + \frac{N_{t_R}}{N_t}\frac{N_{c_k t_R}}{N_{t_R}}$$
$$= p_L\, p(c = i \mid t_L) + p_R\, p(c = i \mid t_R)$$

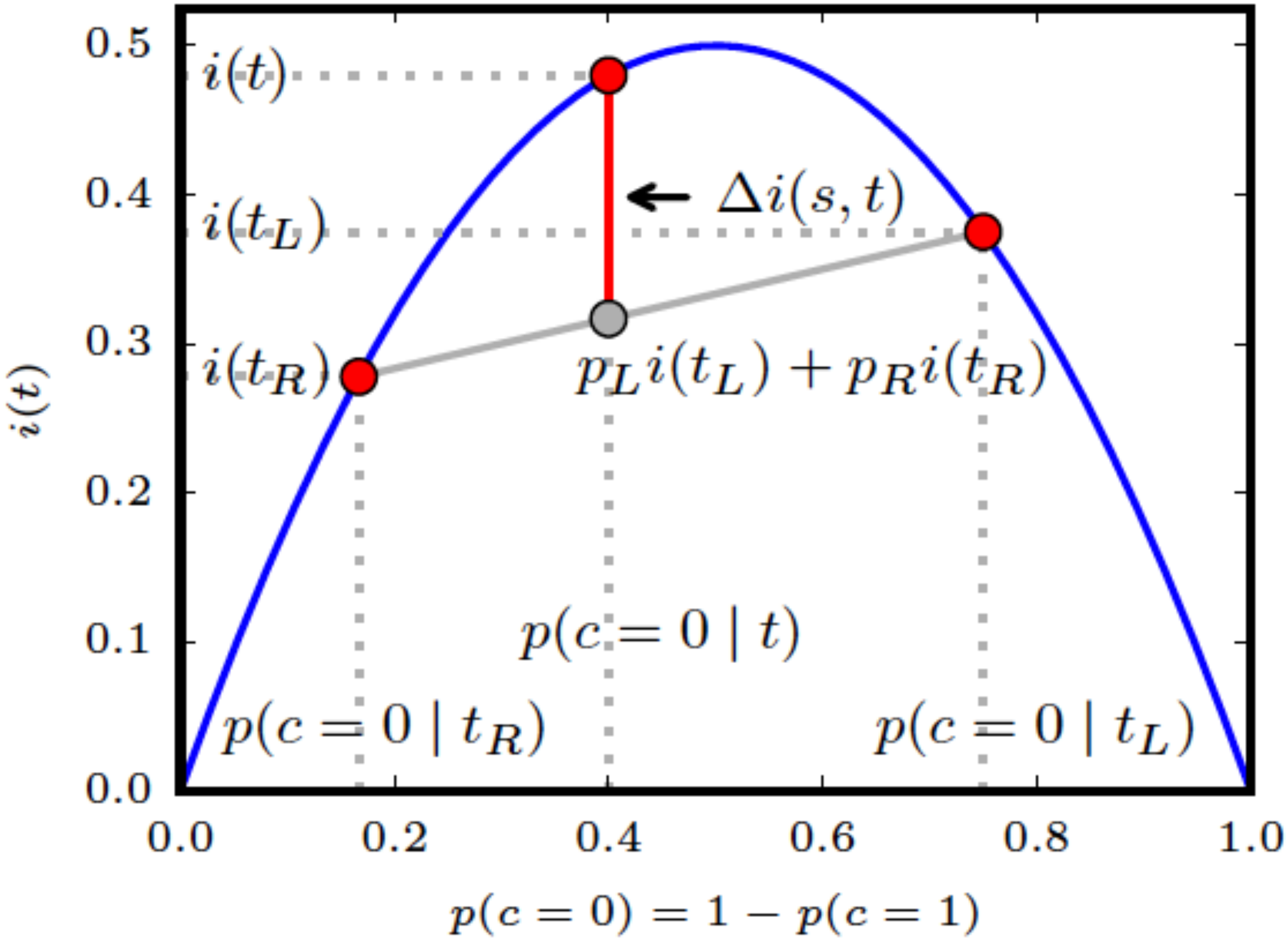
By strict concavity,

$$\begin{aligned} i(t) &= \Phi(p(c=0 \mid t),\ p(c=1 \mid t)) \\ &= \Phi(p_L\, p(c=0 \mid t_L) + p_R\, p(c=0 \mid t_R),\ p_L\, p(c=1 \mid t_L) + p_R\, p(c=1 \mid t_R)) \\ &\geq\ p_L\, \Phi(p(c=0 \mid t_L),\ p(c=1 \mid t_L)) + p_R\, \Phi(p(c=0 \mid t_R),\ p(c=1 \mid t_R)) \\ &= p_L\, i(t_L) + p_R\, i(t_R) \end{aligned}$$

with equality if and only if $p(c=i \mid t) = p(c=i \mid t_L) = p(c=i \mid t_R)$ for both $i = 0, 1$

We can get a better understanding of what this means by looking at a plot of $i(t)$ with respect to $p(c=0)$ (given some node t).




$i(t)$
$i(t_L)$
$i(t_R)$
$\Delta i(s,t)$
$p_L i(t_L) + p_R i(t_R)$
$p(c = 0 \mid t)$
$p(c = 0 \mid t_R)$
$p(c = 0 \mid t_L)$
$i(t)$
0.0
0.1
0.2
0.3
0.4
0.5
0.0
0.2
0.4
0.6
0.8
1.0
$p(c = 0) = 1 - p(c = 1)$

It's important to grasp why the gray point has y-value $p_L i(t_L) + p_R i(t_R)$. This follows from the fact that points on the line defined by $(p(c=0 \mid t_R), i(t_R))$ and $(p(c=0 \mid t_L), i(t_L))$ are expressed by

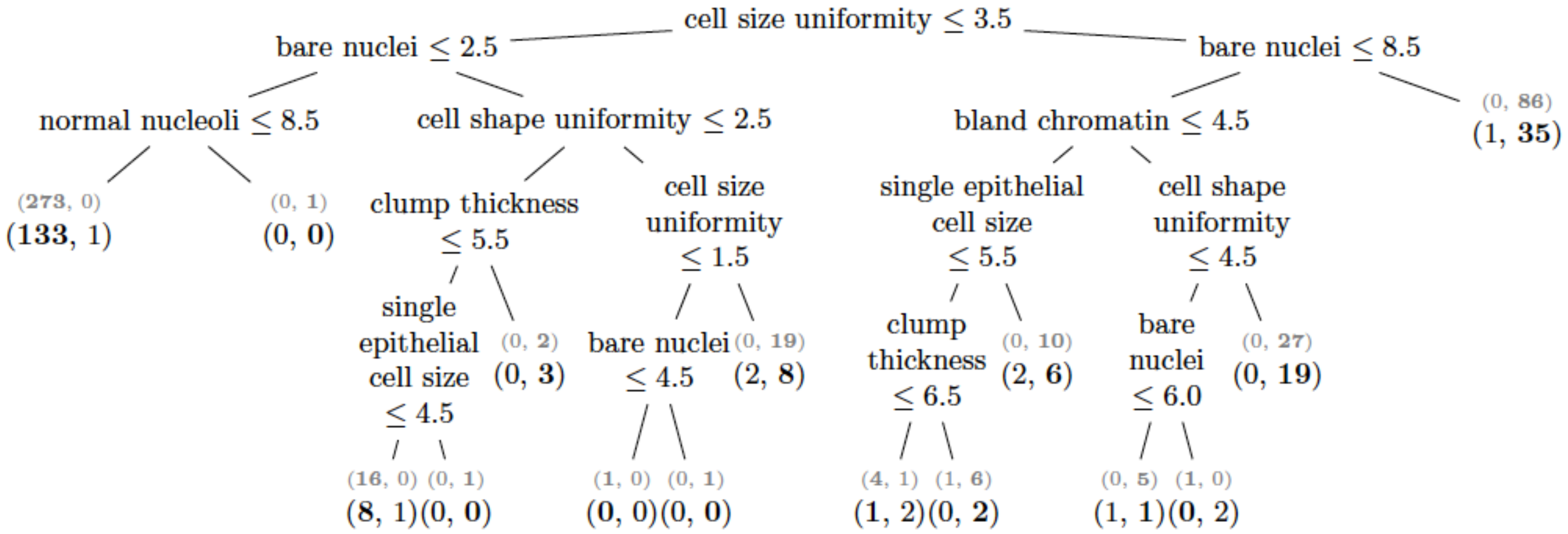
$$f(x) = \frac{i(t_L) - i(t_R)}{p(c=0 \mid t_L) - p(c=0 \mid t_R)} x + i(t_R) - \frac{i(t_L) - i(t_R)}{p(c=0 \mid t_L) - p(c=0 \mid t_R)} p(c=0 \mid t_R)$$

with $f(p(c=0 \mid t_R)) = i(t_R)$ and $f(p(c=0 \mid t_R)) = i(t_L)$. Since f is a linear function, evaluating it at $p(c=i \mid t) = p_L \, p(c=i \mid t_L) + p_R \, p(c=i \mid t_R)$ leads to $p_L i(t_L) + p_R i(t_R)$.

In the lecture, we discussed some issues with the missclassification rate $i_E(t) = 1 - \max_j p(c=j \mid t)$. Now you know that these arise because the missclassification rate is not a strictly concave function.

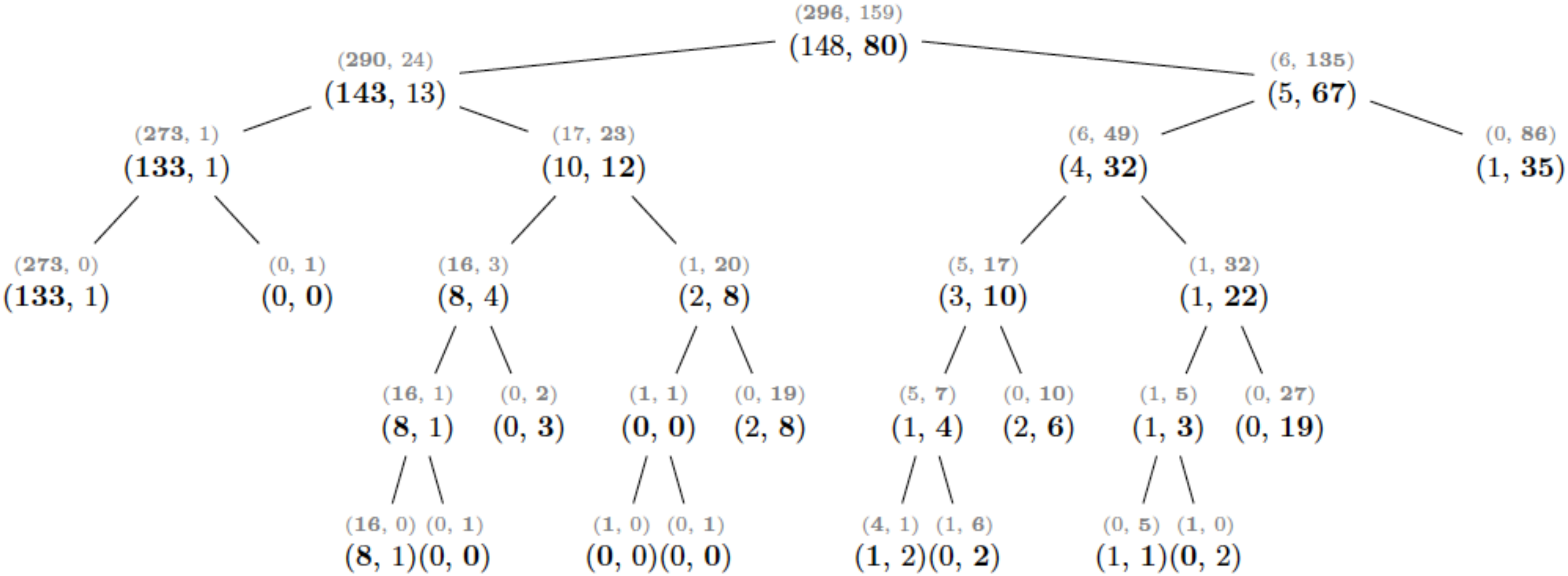
2 Pruning

You trained a tree on 455 samples of the popular Wisconsin breast cancer dataset to a maximum depth of 5 and noted the number of samples of each class in the leaves as tuples $(n_{\text{benign}}, n_{\text{malignant}})$ in gray font, indicating the class label for each leaf in bold. You realize that many leaves contain only a small number of samples and decide that you want to prune the tree. Luckily, you held out 228 samples during training so that you can now perform reduced error pruning on the final tree. You note down the number of validation samples in every leaf in black font and start pruning.



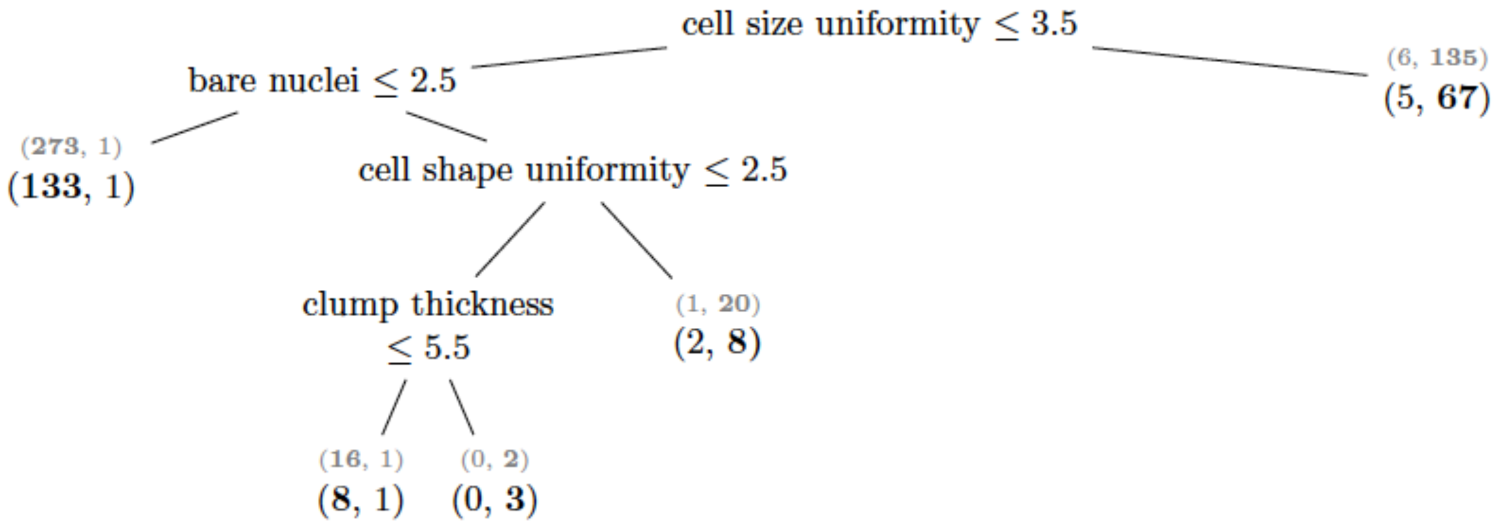
Problem 3: Which nodes do you prune? What are the error rates on both your training and validation set before and after pruning?

error rates	before	after
training set	2/455≈0.004	9/455≈0.020
validation set	12/228≈0.053	9/228≈0.039



	# validation samples misclassified	decision
before pruning	**12**	
prune node ‘normal nucleoli ≤ 8.5’	12	prune*
prune node ‘single epithelial cell size ≤ 4.5’	12	prune*
prune node ‘clump thickness ≤ 5.5’	15	keep
prune node ‘bare nuclei ≤ 4.5’	12	prune*
prune node ‘cell size uniformity ≤ 1.5’	12	prune*
prune node ‘cell shape uniformity ≤ 2.5’	19	keep
prune node ‘bare nuclei ≤ 2.5’	21	keep
prune node ‘clump thickness ≤ 6.5 ’	**11**	prune
prune node ‘single epithelial cell size ≤ 5.5’	11	prune*
prune node ‘bare nuclei ≤ 6.0’	**9**	prune
prune node ‘cell shape uniformity ≤ 4.5’	9	prune*
prune node ‘bland chromatin ≤ 4.5’	9	prune*
prune node ‘bare nuclei ≤ 8.5’	9	prune*
prune node ‘cell size uniformity ≤ 3.5’	80	keep

*) We could also decide to keep nodes that do not affect the validation accuracy in either way, although it may be natural to prefer smaller trees.


cell size uniformity ≤ 3.5
bare nuclei ≤ 2.5
(6, 135)
(5, 67)
(273, 1)
(133, 1)
cell shape uniformity ≤ 2.5
clump thickness ≤ 5.5
(1, 20)
(2, 8)
(16, 1)
(8, 1)
(0, 2)
(0, 3)

Problem 4: Now assume that the consequences (cost!) of misclassifying malignant as benign (type II error) are 10 times as high as classifying benign as malignant (type I error). Do you still prefer the pruned version of your tree?

Problem 4: Now assume that the consequences (cost!) of misclassifying malignant as benign (type II error) are 10 times as high as classifying benign as malignant (type I error). Do you still prefer the pruned version of your tree?

$$\text{cost}_{\text{original}} = w_{\text{II}} n_{\text{II}} + w_{\text{I}} n_{\text{I}} = 10 \cdot 6 + 6 = 66$$
$$\text{cost}_{\text{pruned}} = 10 \cdot 2 + 7 = 27$$

You would still prefer the pruned version.

3 Random Forests

In this exercise we will investigate the effect of two main parameters in Random Forests, namely the number of trees and the number of features randomly chosen at each node.

Problem 5: Suppose a random forest classifier discriminates between only two different classes. Assuming that the outputs of all trees are independent and have the same individual error rate ϵ, write down the probability of a random forest making the wrong prediction given the (odd!) number of trees n and the error rate of a single tree ϵ.

3 Random Forests

In this exercise we will investigate the effect of two main parameters in Random Forests, namely the number of trees and the number of features randomly chosen at each node.

Problem 5: Suppose a random forest classifier discriminates between only two different classes. Assuming that the outputs of all trees are independent and have the same individual error rate ϵ, write down the probability of a random forest making the wrong prediction given the (odd!) number of trees n and the error rate of a single tree ϵ.

The probability that exactly k trees make the wrong prediction:

$$p(n_{\text{wrong}} = k) = \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k}$$

The ensemble only makes the wrong prediction if at least half of the trees make the wrong prediction.

$$p(\text{wrong}) = \sum_{k=(n+1)/2}^{n} p(n_{\text{wrong}} = k) = \sum_{k=(n+1)/2}^{n} \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k}$$

Let's say we have $n = 99$ trees where each tree has individual error rate $\epsilon = 25\%$:

$$p(\text{wrong}) = \sum_{k=50}^{99} \binom{99}{k} 0.25^k 0.75^{99-k} \approx 4.38485 \times 10^{-8}$$

Problem 6: In practice, the limited number of training samples does not allow us to build a large number of independent trees. Consider a Random Forest with $n = 35$, where sets of five trees are mutually independent but where each of the trees within a set has the same output. How does this affect the accuracy of the random forest in contrast to a Random Forest with all independent trees?

Problem 6: In practice, the limited number of training samples does not allow us to build a large number of independent trees. Consider a Random Forest with $n = 35$, where sets of five trees are mutually independent but where each of the trees within a set has the same output. How does this affect the accuracy of the random forest in contrast to a Random Forest with all independent trees?

This corresponds to a Random Forest with $n = 7$ independent trees and thus decreases the accuracy.

Problem 7: Every tree of a Random Forest is built using a bootstrap sample of n samples chosen drawn from n available samples with replacement. Show that the probability that a certain sample s is used to build a certain tree T is $p(s \in T) \approx .632$ for large n.

Problem 7: Every tree of a Random Forest is built using a bootstrap sample of n samples chosen drawn from n available samples with replacement. Show that the probability that a certain sample s is used to build a certain tree T is $p(s \in T) \approx .632$ for large n.

Not choosing a particular sample s means, we have to draw one of the $n-1$ samples for each of the n draws.

$$p(s \in T) = 1 - p(s \notin T) = 1 - \left(\frac{n-1}{n}\right)^n$$

For $n = 1000$, $p(s \in T) \approx 0.63230458$.

Alternatively, we can also consider the limit when n approaches ∞. We use the substitution $m = n-1$

$$\begin{aligned}
\lim_{n\to\infty}\left(\frac{n-1}{n}\right)^n &= \lim_{m\to\infty}\left(\frac{m}{m+1}\right)^{m+1} = \lim_{m\to\infty}\left(\frac{1}{\frac{m+1}{m}}\right)^{m+1} \\
&= \lim_{m\to\infty}\left(\frac{1}{1+\frac{1}{m}}\right)^{m+1} = \lim_{m\to\infty}\left(\frac{1}{1+\frac{1}{m}}\right)^{m} \times \frac{1}{1+\frac{1}{m}} = \frac{1}{e} \\
&\Rightarrow \lim_{n\to\infty} p(s \in T) = 1 - \frac{1}{e} \approx 0.63212056
\end{aligned}$$

Problem 8: If you have $n = 10$ samples, how many different bootstrap samples are possible?

Problem 8: If you have $n = 10$ samples, how many different bootstrap samples are possible?

If you consider only different possible unique examples in a bootstrap sample the number is equal to the number of subsets of the set of n patterns (excluding the empty subset):

$$\sum_{k=1}^{n} \binom{n}{k} = (\sum_{k=0}^{n} \binom{n}{k}) - 1 = 2^n - 1 = 1023.$$

If you want to consider Bootstrap samples that contain the same examples but with different "multiples" (such as AAABC and ABBCC) (examples that are used more than once have more impact than others) the solution is

$$\binom{2n-1}{n} = 92378.$$

because the number of k element multisets over an n element set is

$$\binom{n+k-1}{k}$$

(number of ways to choose with replacement) and we choose $k = n$ times. (see page 18f in Steger "Diskrete Strukturen 1", Springer 2002 or the "arbitrary mapping" column and the "distinguishable urns and non-distinguishable balls" row in the twelve-fold way.)

Problem 9: Apart from taking bootstrap samples, variance between trees is introduced by randomly sampling only $d < D$ of the total number of D features at any node to determine the best split. Which effect does the parameter d have on the error rates?

Problem 9: Apart from taking bootstrap samples, variance between trees is introduced by randomly sampling only $d < D$ of the total number of D features at any node to determine the best split. Which effect does the parameter d have on the error rates?

If d is small, the individual trees are "weaker" (larger error rate ϵ) since they use splits at every node that are less optimal. In turn the trees are more independent of each other. If d is large, the individual predictive power of the trees is higher but they are less independent of each other, which corresponds to having a smaller number of different trees n.

In practice, the optimal value of d is determined by heuristics (e.g. $d = \sqrt{D}$ for classification).

Problems w.r.t. KNN

4 Probabilistic k-NN

Assume that you have two classes. Let N_0 be the number of samples in class 0, N_1 the number of samples in class 1 (this implies that $p(c = 0) = \frac{N_0}{N_0+N_1}$ and $p(c = 1) = \frac{N_1}{N_0+N_1}$). Let $\boldsymbol{x}^*$ be a point that you want to classify.

Problem 10: Consider the ratio $\frac{p(c=0|\boldsymbol{x}^*)}{p(c=1|\boldsymbol{x}^*)}$. Show that for small σ^2 the following approximation holds:

$$\frac{p(c = 0 \mid \boldsymbol{x}^*)}{p(c = 1 \mid \boldsymbol{x}^*)} \approx \frac{\exp\left((-\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2)/(2\sigma^2)\right)}{\exp\left((-\|\boldsymbol{x}^* - \boldsymbol{x}_1\|^2)/(2\sigma^2)\right)}$$

Hint: You may assume that if σ^2 is very small, then the closest data points for each class (denoted by $\boldsymbol{x}_0$ for class 0 and $\boldsymbol{x}_1$ for class 1) will dominate the sum over the exponentials.

We have

$$p(\boldsymbol{x}^* \mid c = i) = \frac{1}{N_i} \frac{1}{(2\pi\sigma^2)^{D/2}} \sum_{\boldsymbol{x}_n \in \text{class } i} e^{-\frac{\|\boldsymbol{x}^* - \boldsymbol{x}_n\|^2}{(2\sigma^2)}}$$

and

$$p(c = i \mid \boldsymbol{x}^*) = \frac{p(\boldsymbol{x}^* \mid c = i)\; p(c = i)}{p(\boldsymbol{x}^* \mid c = 0)\; p(c = 0) + p(\boldsymbol{x}^* \mid c = 1)\; p(c = 1)}$$

Both $p(c = 0 \mid \boldsymbol{x}^*)$ and $p(c = 1 \mid \boldsymbol{x}^*)$ have the same denominator – this denominator cancels. Furthermore, $1/(2\pi\sigma^2)^{D/2}$ cancels, too. Thus

$$\frac{p(c = 0 \mid \boldsymbol{x}^*)}{p(c = 1 \mid \boldsymbol{x}^*)} \approx \frac{\exp\left((-\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2)/(2\sigma^2)\right)}{\exp\left((-\|\boldsymbol{x}^* - \boldsymbol{x}_1\|^2)/(2\sigma^2)\right)} \; \frac{p(c = 0)N_1}{p(c = 1)N_0}$$

where we assume (without proof) that the closest points dominate the sum over the exponentials. With the class probabilities $p(c)$ we get

$$\frac{p(c = 0)N_1}{p(c = 1)N_0} = \frac{N_1 N_0/(N_0 + N_1)}{N_0 N_1/(N_0 + N_1)} = 1.$$

Problem 11: Show that in the limit case ($\sigma \to 0$) this approximation classifies $\boldsymbol{x}^*$ as class 0 if it is closer to $\boldsymbol{x}_0$ than to $\boldsymbol{x}_1$.

Problem 11: Show that in the limit case ($\sigma \to 0$) this approximation classifies $\boldsymbol{x}^*$ as class 0 if it is closer to $\boldsymbol{x}_0$ than to $\boldsymbol{x}_1$.

If $\boldsymbol{x}^*$ is closer to $\boldsymbol{x}_0$ than to $\boldsymbol{x}_1$, then

$$\frac{p(c=0 \mid \boldsymbol{x}^*)}{p(c=1 \mid \boldsymbol{x}^*)} \approx \frac{\exp\left((-\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2)/(2\sigma^2)\right)}{\exp\left((-\|\boldsymbol{x}^* - \boldsymbol{x}_1\|^2)/(2\sigma^2)\right)} = \frac{1}{\exp\left((\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 - \|\boldsymbol{x}^* - \boldsymbol{x}_1\|^2)/(2\sigma^2)\right)} > 1$$

because $(\|\boldsymbol{x}^* - \boldsymbol{x}_0\|^2 - \|\boldsymbol{x}^* - \boldsymbol{x}_1\|^2) < 0$.

Problem 12: How does σ relate to k?

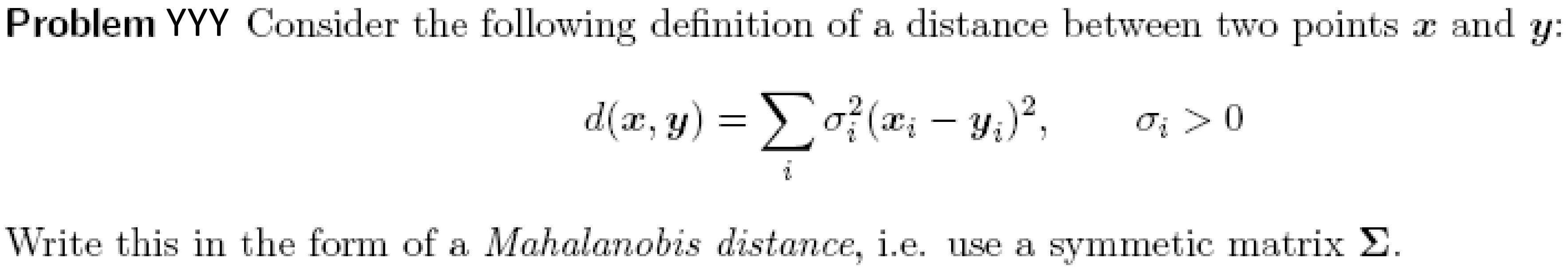
Problem 12: How does σ relate to k?

In the probabilistic k-NN scheme, if σ is large, the probability mass centered on each point extends far into the neighborhood. This corresponds to looking at a large neighborhood (large k).

Problem YYY Consider the following definition of a distance between two points $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_i \sigma_i^2 (\boldsymbol{x}_i - \boldsymbol{y}_i)^2, \qquad \sigma_i > 0$$

Write this in the form of a *Mahalanobis distance*, i.e. use a symmetic matrix $\boldsymbol{\Sigma}$.



Problem YYY Consider the following definition of a distance between two points $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$d(\boldsymbol{x},\boldsymbol{y}) = \sum_i \sigma_i^2 (\boldsymbol{x}_i - \boldsymbol{y}_i)^2, \qquad \sigma_i > 0$$

Write this in the form of a *Mahalanobis distance*, i.e. use a symmetic matrix $\boldsymbol{\Sigma}$.

Mahalanobis distance ($\boldsymbol{\Sigma}$ is *positive (semi) definite* and *symmetric*):

$$d(x,y) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^T \boldsymbol{\Sigma} (\boldsymbol{x}-\boldsymbol{y})}$$

or $d(x,y) = (\boldsymbol{x}-\boldsymbol{y})^T \boldsymbol{\Sigma}\, (\boldsymbol{x}-\boldsymbol{y})$

Problem YYY Consider the following definition of a distance between two points $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_i \sigma_i^2 (\boldsymbol{x}_i - \boldsymbol{y}_i)^2, \qquad \sigma_i > 0$$

Write this in the form of a *Mahalanobis distance*, i.e. use a symmetic matrix $\boldsymbol{\Sigma}$.

$$\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2).$$

Let $\sigma_{\ i}$ be the scale of feature i.

Then we then want to have $\Sigma = diag(\frac{1}{{\sigma_1}^2}, \frac{1}{{\sigma_2}^2}, \ldots, \frac{1}{{\sigma_n}^2})$

5 Neighbourhod Component Analysis

Problem 13: Calculate the gradient of the NCA objective as given in the slides. The following matrix identity[3] is helpful:

$$\frac{\partial tr(A^T AB)}{\partial A} = A(B + B^T)$$

tr is the *trace* of a matrix. You may also want to use the shorthand $x_{ij} = (x_i - x_j)$.

[1] for more of these things: K. B. Petersen and M. S. Pedersen. 2008. The Matrix Cookbook. Technical Report.

5 Neighbourhod Component Analysis

Problem 13: Calculate the gradient of the NCA objective as given in the slides. The following matrix identity[3] is helpful:

$$\frac{\partial tr(A^T A B)}{\partial A} = A(B + B^T)$$

tr is the *trace* of a matrix. You may also want to use the shorthand $x_{ij} = (x_i - x_j)$.

To compute $\frac{\partial p_{ij}}{\partial A}$, first write $||Ax_i - Ax_j||^2$ as $||A(x_i - x_j)||^2 = ||Ax_{ij}||^2 = (Ax_{ij})^T Ax_{ij} = x_{ij}^T A^T A x_{ij} = tr(x_{ij}^T A^T A x_{ij}) = tr(A^T A x_{ij} x_{ij}^T)$, where we used the cyclic property of the trace ($tr(A_1 A_2 \dots A_n) = tr(A_2 \dots A_n A_1) = tr(A_n A_2 \dots A_{n-1})$), so that the matrix identity is applicable. *Note: This is an important trick!* Now

$$\begin{aligned}
\frac{\partial p_{ij}}{\partial A} &= \frac{\exp(-||Ax_{ij}||^2)(-1)A(x_{ij}x_{ij}^T + x_{ij}x_{ij}^T)\sum_{k\neq i}\exp(-||Ax_{ik}||^2)}{\left(\sum_{k\neq i}\exp(-||Ax_{ik}||^2)\right)^2} \\
&- \frac{\exp(-||Ax_{ij}||^2)\left[\sum_{k\neq i}\exp(-||Ax_{ik}||^2)(-1)A(x_{ik}x_{ik}^T + x_{ik}x_{ik}^T)\right]}{\left(\sum_{k\neq i}\exp(-||Ax_{ik}||^2)\right)^2} \\
&= -2Ap_{ij}\left(x_{ij}x_{ij}^T - \sum_{k\neq i} p_{ik}x_{ik}x_{ik}^T\right)
\end{aligned}$$

We are actually interested in $\frac{\partial f}{\partial A}$ and by using $p_i = \sum_{j\in C_i} p_{ij}$ and reordering the above equation we obtain the result given in the slides.

Assume You are not yet fit in matrix calculus.
Can You do it („on foot“) in components ?

$$\frac{\partial f(A)}{\partial A_{mn}} = \tag{1}$$

$$\frac{\partial}{\partial A_{mn}} \sum_i \sum_{j \in C_i} \frac{exp(-||A(x^{(i)} - x^{(j)})||^2)}{\sum_{k \neq i} exp(-||A(x^{(i)} - x^{(k)})||^2)} = \tag{2}$$

$$\frac{\partial}{\partial A_{mn}} \sum_i \sum_{j \in C_i} \frac{exp(-||Ax^{(ij)}||^2)}{\sum_{k \neq i} exp(-||Ax^{(ik)}||^2)} = \tag{3}$$

$$\sum_i \sum_{j \in C_i} \left[\frac{\frac{\partial}{\partial A_{mn}} exp(-||Ax^{(ij)}||^2)}{\sum_{k \neq i} exp(-||Ax^{(ik)}||^2)} \right. \tag{4}$$

$$\left. - \frac{exp(-||Ax^{(ij)}||^2)}{\left(\sum_{k \neq i} exp(-||Ax^{(ik)}||^2)\right)^2} \frac{\partial}{\partial A_{mn}} \sum_{k \neq i} exp(-||Ax^{(ik)}||^2) \right] =$$

$$\sum_i \sum_{j \in C_i} \left[\frac{\frac{\partial}{\partial A_{mn}} exp(-||Ax^{(ij)}||^2)}{\sum_{k \neq i} exp(-||Ax^{(ik)}||^2)} \right. \tag{4}$$

$$\left. - \frac{exp(-||Ax^{(ij)}||^2)}{\left(\sum_{k \neq i} exp(-||Ax^{(ik)}||^2)\right)^2} \frac{\partial}{\partial A_{mn}} \sum_{k \neq i} exp(-||Ax^{(ik)}||^2) \right] =$$

$$\sum_i \sum_{j \in C_i} \left[\frac{exp(-||Ax^{(ij)}||^2)}{\sum_{k \neq i} exp(-||Ax^{(ik)}||^2)} \frac{\partial}{\partial A_{mn}} (-||Ax^{(ij)}||^2) \right. \tag{5}$$

$$\left. - \frac{exp(-||Ax^{(ij)}||^2)}{\left(\sum_{k \neq i} exp(-||Ax^{(ik)}||^2)\right)^2} \sum_{k \neq i} \frac{\partial}{\partial A_{mn}} exp(-||Ax^{(ik)}||^2) \right] =$$

$$\sum_i \sum_{j \in C_i} \left[p_{ij} \frac{\partial}{\partial A_{mn}} (-||Ax^{(ij)}||^2) \right. \tag{6}$$

$$\left. - p_{ij} \frac{1}{\sum_{k \neq i} exp(-||Ax^{(ik)}||^2)} \sum_{k \neq i} exp(-||Ax^{(ik)}||^2) \frac{\partial}{\partial A_{mn}} (-||Ax^{(ik)}||^2) \right]$$

$$\sum_i \sum_{j \in C_i} \left[p_{ij} \frac{\partial}{\partial A_{mn}} (-||Ax^{(ij)}||^2) - p_{ij} \sum_{k \neq i} p_{ik} \frac{\partial}{\partial A_{mn}} (-||Ax^{(ik)}||^2) \right]; \tag{7}$$

$$\frac{\partial}{\partial A_{mn}}(-||Ax^{(ij)}||^2) = \tag{8}$$

$$-\frac{\partial}{\partial A_{mn}}\sum_a \left(\sum_b A_{ab}x_b^{(ij)}\right)^2 = \tag{9}$$

$$-\sum_a \frac{\partial}{\partial A_{mn}} \left(\sum_b A_{ab}x_b^{(ij)}\right)^2 = \tag{10}$$

$$-\sum_a 2 \left(\sum_b A_{ab}x_b^{(ij)}\right) \frac{\partial}{\partial A_{mn}} \sum_c A_{ac}x_c^{(ij)} = \tag{11}$$

$$-2\sum_a \left(\sum_b A_{ab}x_b^{(ij)}\right) \sum_c \frac{\partial}{\partial A_{mn}} A_{ac}x_c^{(ij)} = \tag{12}$$

$$-2\sum_a \left(\sum_b A_{ab}x_b^{(ij)}\right) \sum_c \delta_{ma}\delta_{cn}x_c^{(ij)} = \tag{13}$$

$$-2\sum_a \left(\sum_b A_{ab}x_b^{(ij)}\right) \delta_{ma}x_n^{(ij)} = \tag{14}$$

$$-2\sum_b A_{mb}x_b^{(ij)}x_n^{(ij)} = \tag{15}$$

$$-2\left(Ax^{(ij)}\right)_m x_n^{(ij)} = \tag{16}$$

$$-2\left(Ax^{(ij)}(x^{(ij)})^T\right)_{mn} \tag{17}$$