INTELLIGENT SYSTEMS LABORATORY

Lesson 9
Fundamentals of Machine Learning

Mathematics and Statistics for Data Science

Tapanan Yeophantong
Vincent Mary School of Science and Technology
Assumption University

Content

- Machine learning & its applications
- Decision tree learning
- Bayesian learning
- Nearest-neighbours algorithms
- Performance Evaluation

Creating the Knowledge

- How do we code “knowledge” into a software?
  - Knowledge encoded in the program
  - Knowledge in the database (e.g. rules)
  - Mathematical model or function
- Or, we can make the program obtain the knowledge by *learning*.

- When speaking of intelligent systems in modern times, we often refer to those with a learning capability.

Machine Learning

The ability of an intelligent program to create *knowledge* from *data* to improve its performance over time without being explicitly programmed.

Data = Observations

Machine learning often relies on *statistical techniques* to give the machines the ability to learn.

Observations → Knowledge

Example	Sky	Air	Humidity	Wind	Water	Forecast	EnjoySport?
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

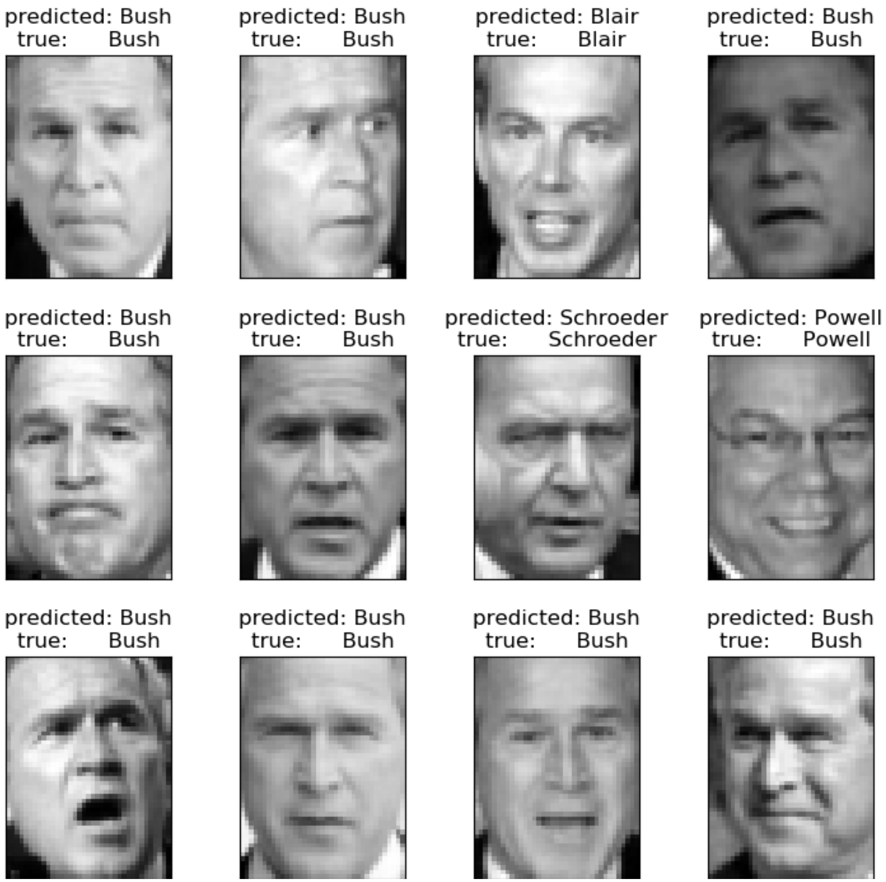
What *kind of days* does Simbe enjoy playing sports?

Data for Machine Learning

- Generally, we need structured data (like the ones in class).
- Unfortunately, they do not come by so nicely.

- Typically, we have to deal with data like:
  - Signal data, e.g. images, videos
  - Unstructured text
  - Temporal or spatiotemporal data
- And dealing with lots of them (i.e. Big data).

Computer Vision



- Computer vision requires collection of images.
- We teach machines by showing it a lot of images,which are seen as an array of pixels.
- Pixels are transformed into feature vectors (structured data) for training.

Natural Language Processing

Bag of Words Example

Document 1

The quick brown fox jumped over the lazy dog's back.

Document 2

Now is the time for all good men to come to the aid of their party.

Term	Document 1	Document 2
aid	0	1
all	0	1
back	1	0
brown	1	0
come	0	1
dog	1	0
fox	1	0
good	0	1
jump	1	0
lazy	1	0
men	0	1
now	0	1
over	1	0
party	0	1
quick	1	0
their	0	1
time	0	1

Stopword List
for
is
of
the
to

- NLP systems require collection of documents.
- Documents can be seen as a collection of terms (e.g. characters or words).
- Terms are transformed into feature vectors (structured data) for training.

Learning $\neq$ Memorizing

… well, most of the time!

Concept Learning

- To generalise a concept from examples, we:
  - Define a space of potential candidates for the concept
  - Select a candidate that best represents the concept

- Remember that we do not know the concept.

Concept "EnjoySport"

Example	Sky	Air	Humidity	Wind	Water	Forecast	EnjoySport?
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

How many candidates for the concept?

Do we have enough examples?

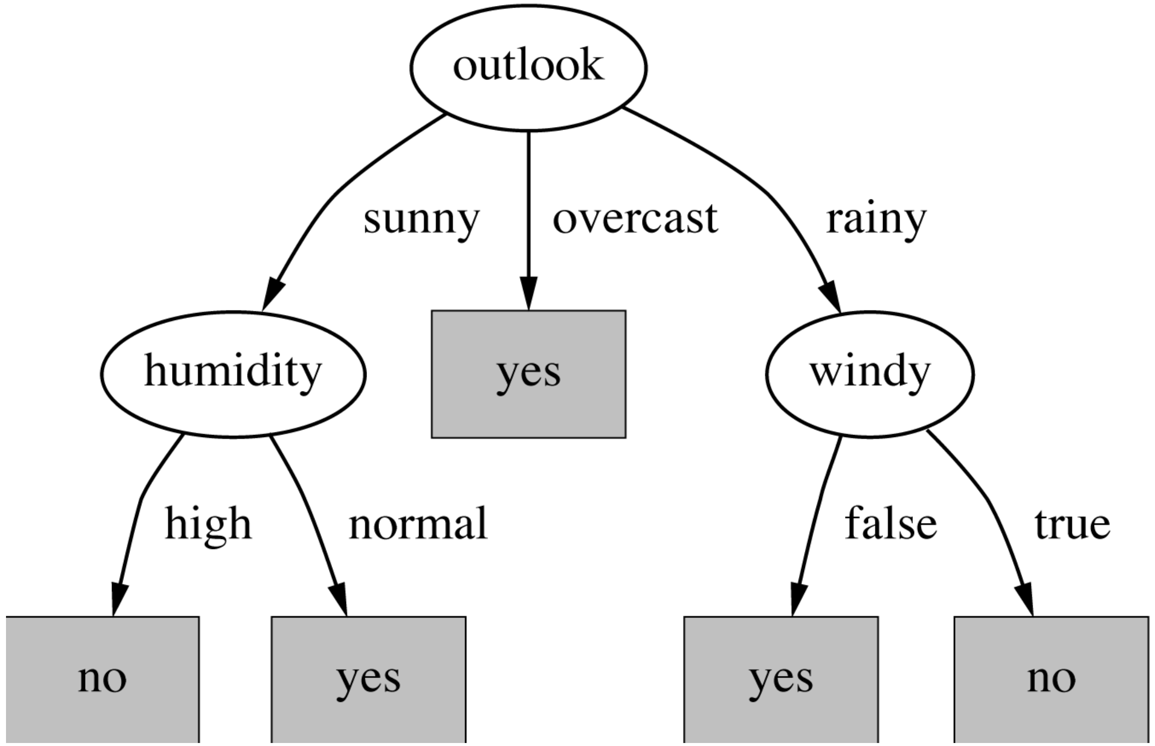
Inductive Learning Hypothesis

Any "candidate" found to approximate the target concept well over a sufficiently large set of examples will also approximate the target concept well over other unobserved examples.

Example: “PlayTennis” Concept

Day	Outlook	Temp	Humidity	Wind	*PlayTennis*
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

Decision Tree



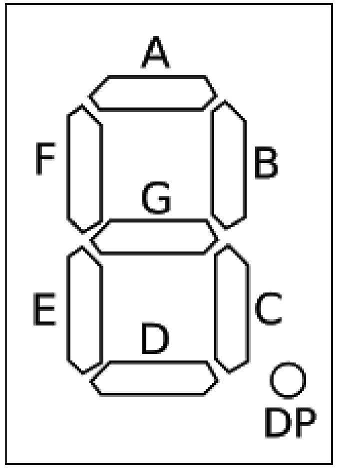
Will the customer play tennis if the day is sunny and cool, with high humidity and strong wind?

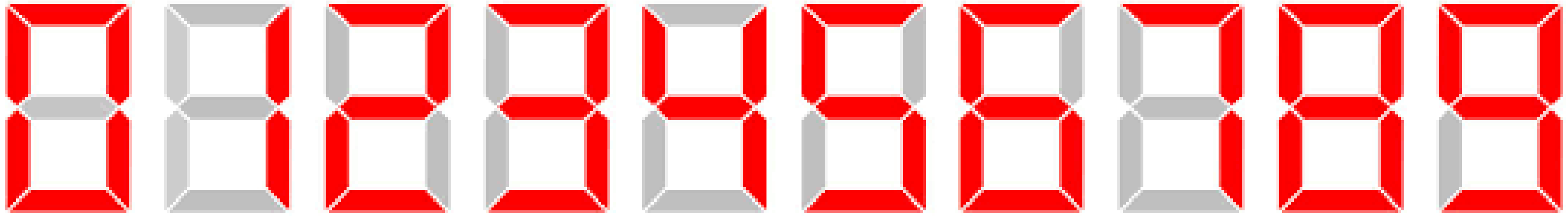
Naïve Bayes

	weather			temperature			humidity		windspeed	
	sunny	overcast	rain	cool	hot	mild	normal	high	weak	strong
N	3/4	0/4	1/4	1/4	2/4	1/4	1/4	3/4	2/4	2/4
Y	2/7	3/7	2/7	2/7	2/7	3/7	5/7	2/7	5/7	2/7
	5/11	3/11	3/11	3/11	4/11	4/11	6/11	5/11	7/11	4/11

Will the customer play tennis if the day is sunny and cool, with high humidity and strong wind?

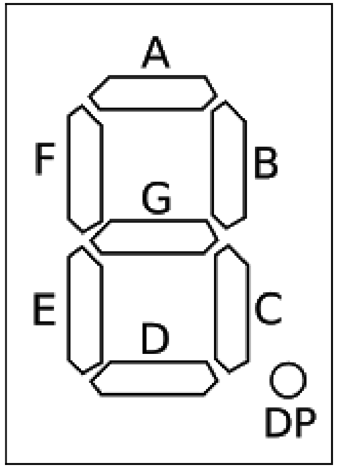
7-Segment LED Display

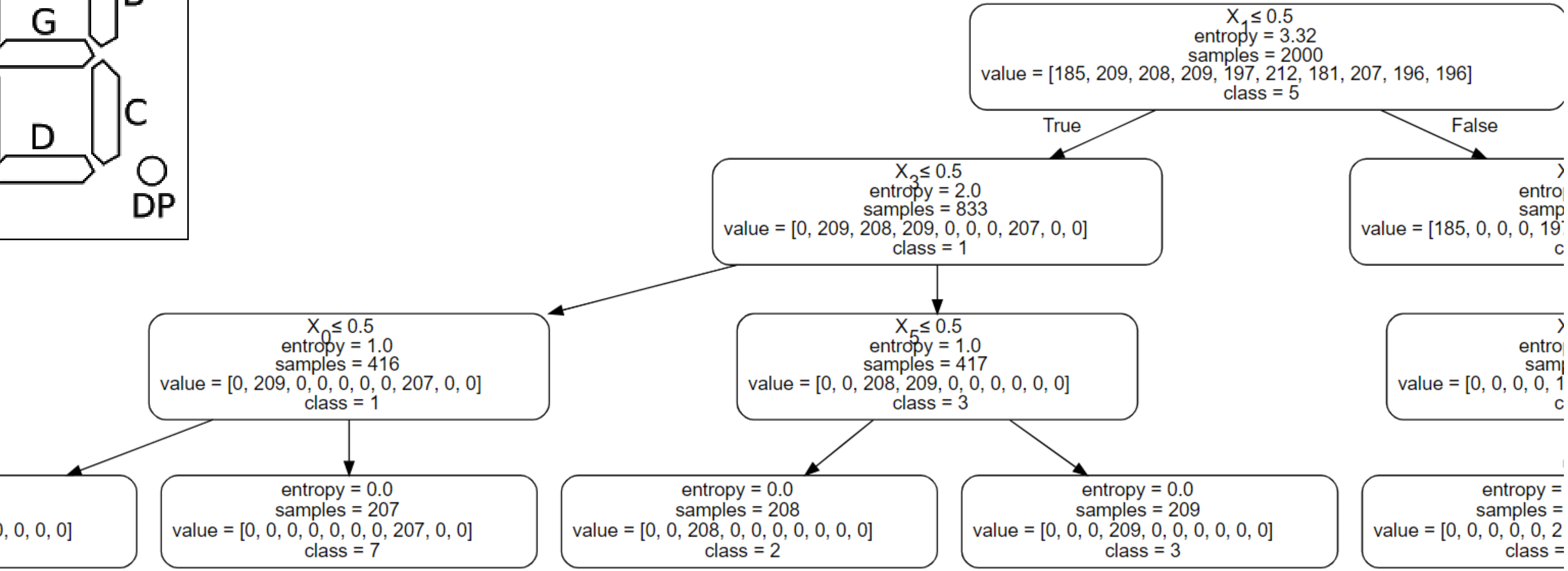




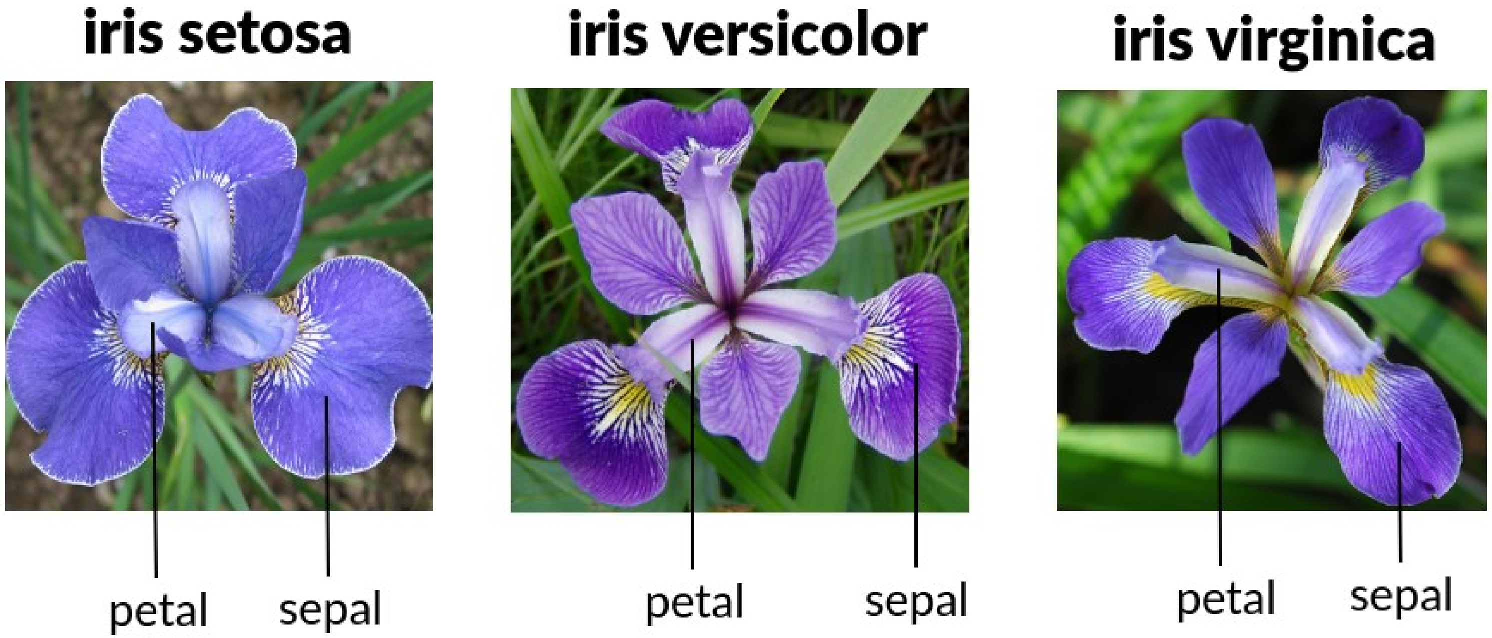
How would a machine learn to recognize these digits (e.g. using decision tree)?

Decision Tree for LED Display





Iris Classification



Content

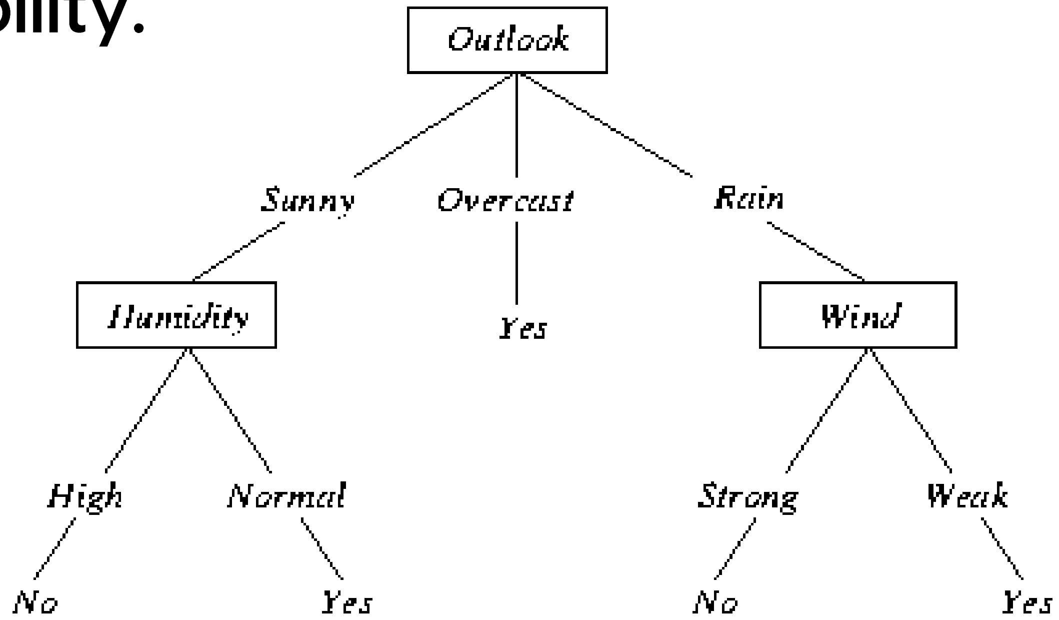
- Machine learning & its applications
- Decision tree learning
- Bayesian learning
- Nearest-neighbours algorithms
- Performance Evaluation

Overview of Decision Trees

- Widely used, practical method for inductive inference.
- Approximate target functions as trees.
- Robust to noisy data and able to learn disjunctive rules.
- Use a completely expressive hypothesis space.

Tree Representation

- Learned function is represented as a decision tree.
- Learned trees can also be represented as if-then rules to improve human readability.



Day	Outlook	Temp	Humidity	Wind	PlayTennis
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

Basic Algorithm

- Top-down, greedy search through a hypothesis space of possible decision trees.
- Evaluate each attribute to see how well it can classify training examples.
- Best attribute is selected.
- Repeat for each node of the tree (starting from root).

Attribute Selection

- An attribute is good when:
  - For one value we get all instances as positive.
  - For other value we get all instances as negative.
- An attribute is poor when:
  - It provides no discrimination; that is, for each value, there is the same number of positive and negative instances.

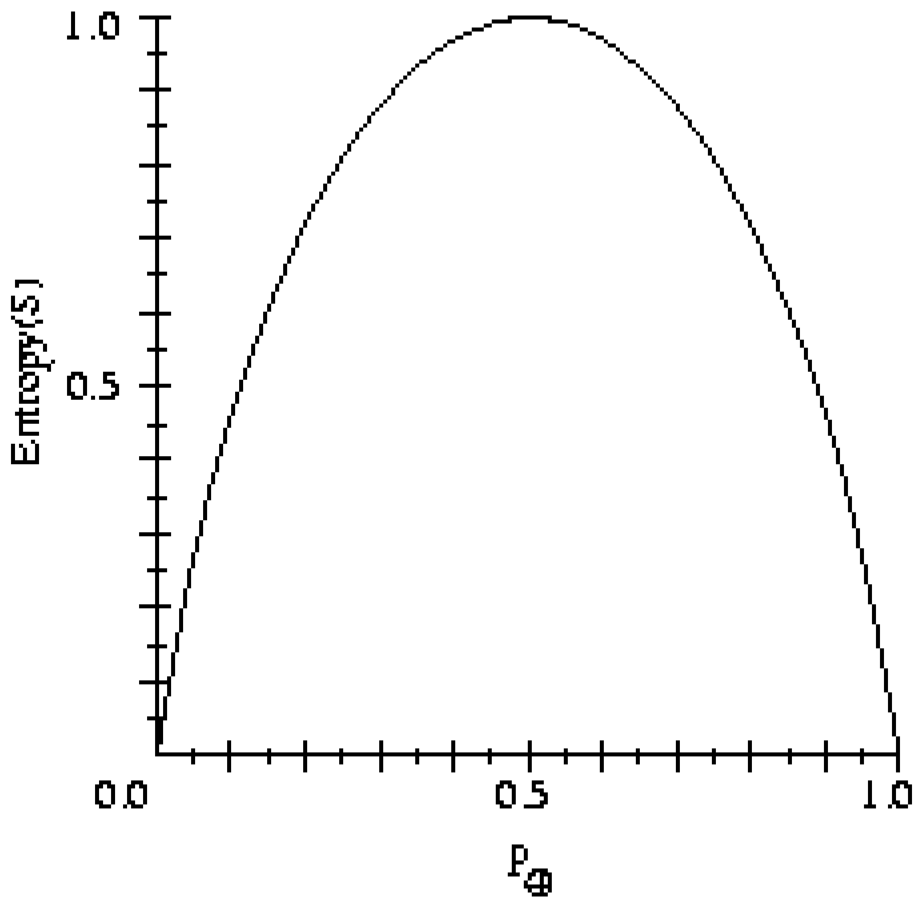
Entropy & Information Gain

- Entropy characterises the (im)purity of an arbitrary collection of examples.

$$Entropy(S) \equiv -p_{+} \log_2 p_{+} - p_{-} \log_2 p_{-}$$

- Information Gain is defined in terms of Entropy, an an expected *reduction in entropy* caused by partitioning the examples according to the attribute.

Entropy Function

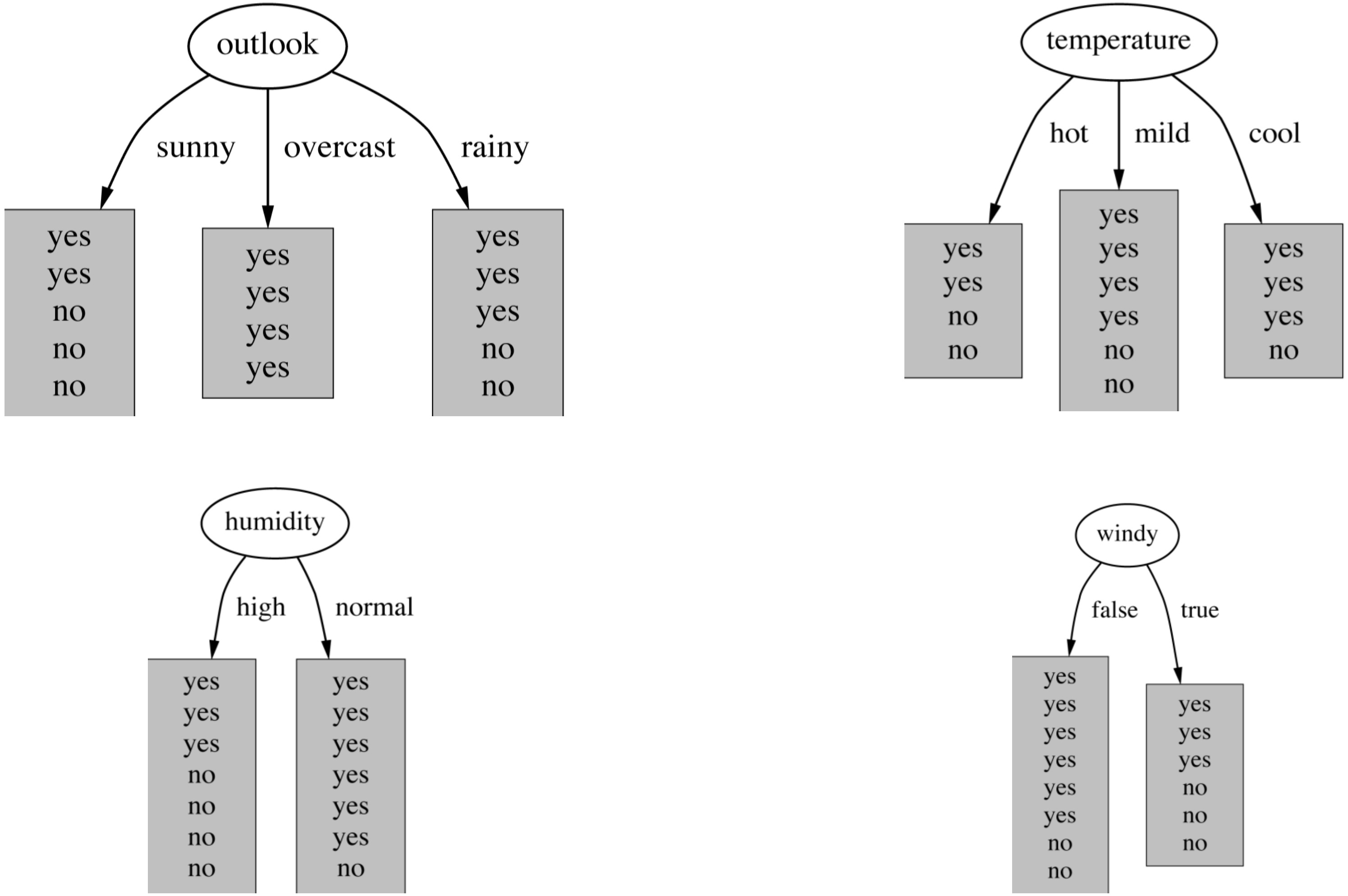


Information Gain

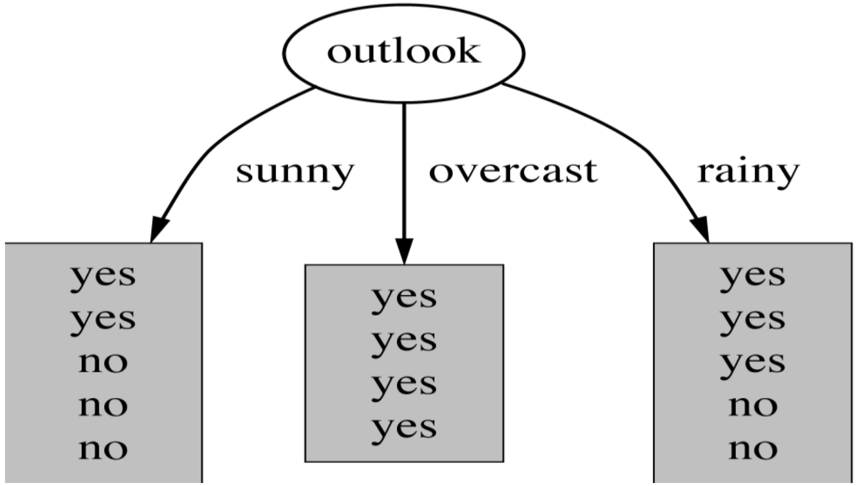
- Information Gain(S,A) of an attribute A is the reduction in entropy caused by partitioning the examples according to the attribute A:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Tree Stumps for Weather Data



Computation Example



Info[2,3] = 0.97 bits

Info[4,0] = 0.00 bits

Info[3,2] = 0.97 bits

Info[2,3] = entropy(2/5, 3/5)

= -2/5 log 2/5 - 3/5 log 3/5

= 0.97 bits

Average information of subtree (weighted):

= (0.97 x 5/14) + (0.00 x 4/14) + (0.97 x 5/14)

= 0.693 bits

Information of all training samples:

info[9,5] = 0.94

gain(outlook) = 0.94 - 0.693 = 0.247 bits

Information Gain: Summary

- Gain(outlook) = 0.94 – 0.693 = 0.247
- Gain(temperature)= 0.94 – 0.911 = 0.029
- Gain(humidity)= 0.94 – 0.788 = 0.152
- Gain(windy)= 0.94 – 0.892 = 0.048

- argMax {0.247, 0.029, 0.152, 0.048} = outlook
- Select outlook as the splitting attribute of tree.

Overfitting

- Overfitting occurs when:
  - There is noise in the data, or
  - The number of training examples is too small to produce a representative sample of the true target function.
- Overfitting is a significant practical difficulty for decision tree learning and many other learning methods.

Content

- Machine learning & its applications
- Decision tree learning
- Bayesian learning
- Nearest-neighbours algorithms
- Performance Evaluation

Overview of Bayesian Models

- Probabilistic approach to inference.
- Assume quantities are governed by probability distributions.
- Make decisions reasoning about probabilities together with observed data.
- Provides quantitative approach to weighing evidence.

Probability Theory

- A fully specified probability model associates a numerical probability P(ω) with each possible world.
- Every possible world has a probability between 0 and 1 and that the total probability of the set of possible worlds is 1; that is,

$$0 \leq P(\omega) \leq 1 \text{ for every } \omega \text{ and } \sum_{\omega \in \Omega} P(\omega) = 1$$

- In statistics, probabilities are assigned to events.
- In AI, they are assigned to propositions.

Conditional Probability

- Unconditional or prior probability is the degree of belief in a proposition in the *absence* of other information.
  - For examples, `P(X=1)` or `P(head)`.
  - Also called "priors" or "a priori".
- Conditional or posterior probability is the degree of belief in proposition *given* some other information is true.
  - `P(A|B)` is the probability that A is true given that B is true.
  - Also called "posteriors" or "posteriori".

Product Rule

- The definition of conditional probability can be written in a form called the product rule:

$$P(a \wedge b) = P(a \mid b)P(b)$$

- For *a* and *b* to be true,
  - *b* needs to be true, and
  - *a* needs to be true given *b*.

Bayes' Theorem

- Bayes' Theorem is derived from the product rule where:

$$P(b \mid a) = \frac{P(a \mid b)P(b)}{P(a)}$$

- This simple equation happens to be in many modern AI systems for probabilistic inferences.

Naïve Bayes

$$v_{MAP} = \arg\max_{c_j \in C} P(c_j \mid e_1, e_2 \ldots e_n)$$

$$v_{MAP} = \arg\max_{c_j \in C} \frac{P(e_1, e_2 \ldots e_n \mid c_j) P(c_j)}{P(e_1, e_2 \ldots e_n)}$$

$$= \arg\max_{c_j \in C} P(e_1, e_2 \ldots e_n \mid c_j) P(c_j)$$

$$\boxed{c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_i P(e_i \mid c_i)}$$

Day	Outlook	Temp	Humidity	Wind	PlayTennis
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

Content

- Machine learning & its applications
- Decision tree learning
- Bayesian learning
- Nearest-neighbours algorithms
- Performance Evaluation

Instance-based Learning

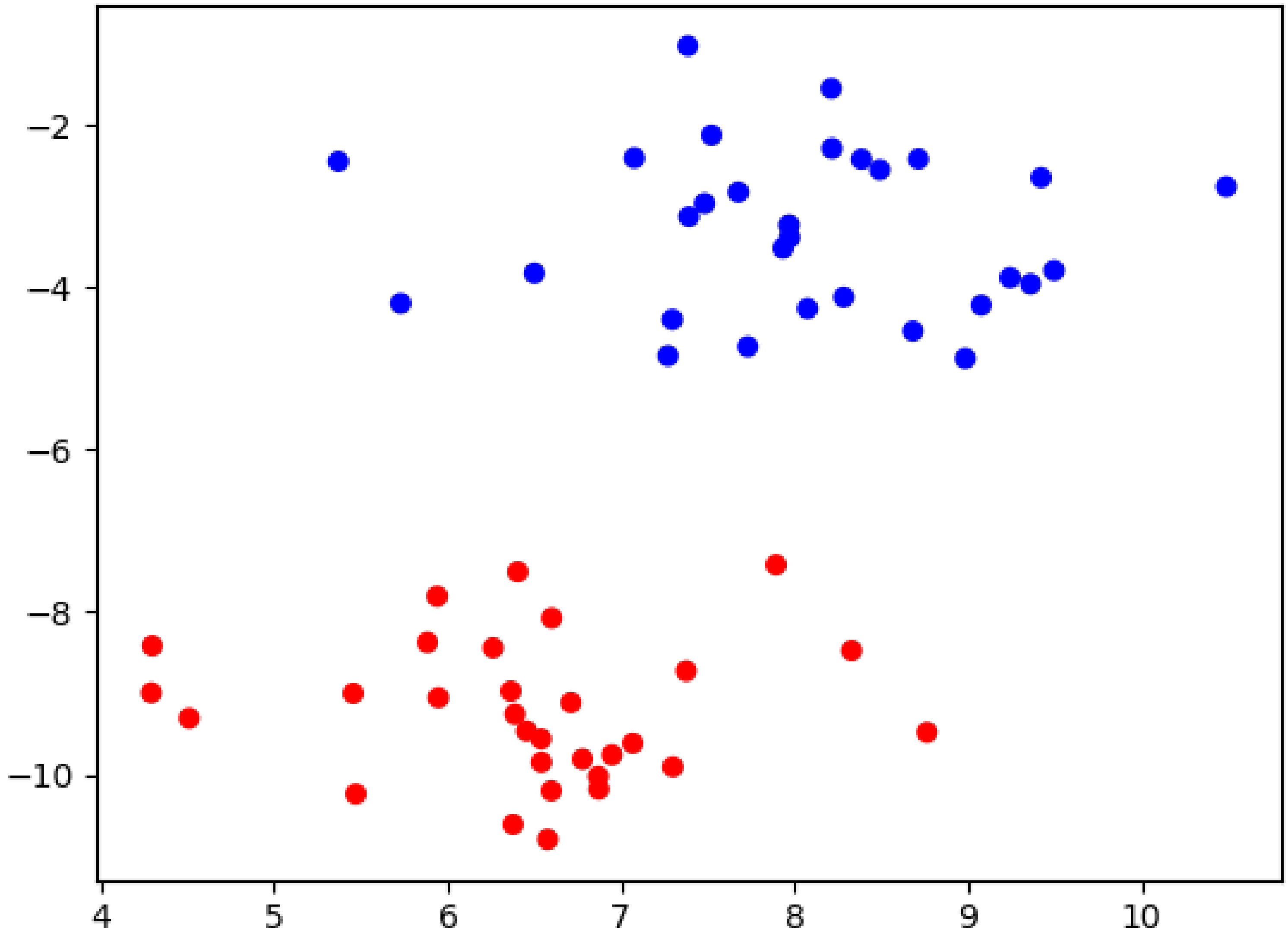
- Also known as the lazy learners.
- Delay learning by storing training examples and using them only when a query instance is observed.
- Approaches to approximating both real-valued and discrete-valued target functions.

Nearest-neighbour Learning

- Most basic instance-based method.
- Assumes all instances correspond to points in the n-dimensional space $\Re^n$.
- Nearest neighbours are normally defined in terms of the standard Euclidean distance.

k-Nearest Neighbour

- An instance-based algorithm for approximating real-valued or discrete-valued target functions.
- Target function value for a new query is estimated from the known values of the *k nearest training examples*.
- The value returned by the algorithm is the most common value among $\mathbf{k}$ training examples nearest to $\mathbf{x}_q$.



−2
−4
−6
−8
−10
4
5
6
7
8
9
10

Uniform-Weighted KNN

- For *discrete-valued* target function (classification):

$$\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i))$$

- For *real-valued* target function (regression):

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

Distance-Weighted KNN

- For *discrete-valued* target function (classification):

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\arg\max} \sum_{i=1}^{k} w_i \delta(v, f(x_i))$$

- For *real-valued* target function (regression):

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$$

- Where:

$$w_i \equiv \frac{1}{d(x_q, x_i)^2}$$

Content

- Machine learning & its applications
- Decision tree learning
- Bayesian learning
- Nearest-neighbours algorithms
- Performance Evaluation

Performance Evaluation

- Accuracy Score
- Confusion Matrix
- Precision, Recall, and F1
- Precision-Recall Curve

Confusion Matrix

Class Predicted by your model

Actual Classes

$$
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 54 & 0 & 0 & 0 & 17 \\
1 & 0 & 36 & 0 & 1 & 6 \\
2 & 0 & 0 & 66 & 5 & 18 \\
3 & 0 & 0 & 0 & 273 & 15 \\
4 & 0 & 0 & 0 & 0 & 367
\end{array}
$$

Precision, Recall and F1

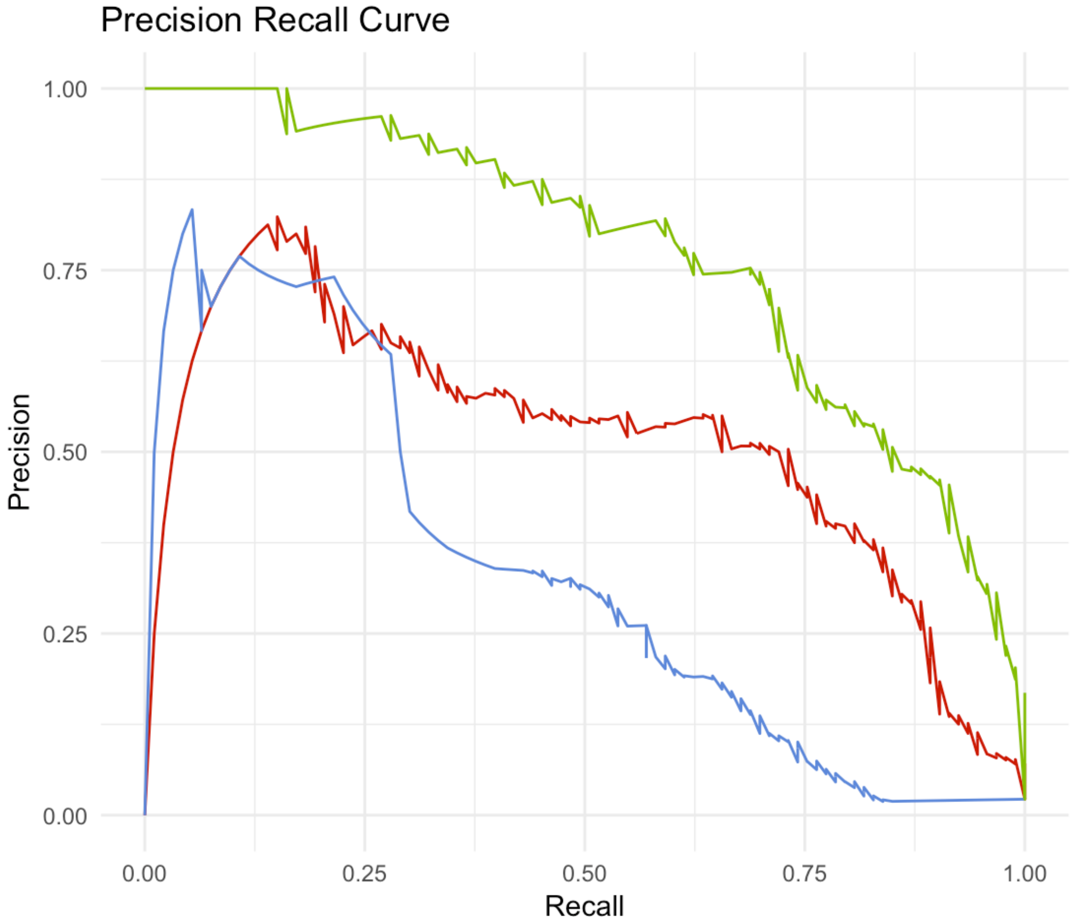
	precision	recall	f1-score	support
0.0	0.86	0.90	0.88	543
1.0	0.83	0.76	0.79	342
avg / total	0.85	0.85	0.85	885

$$\text{precision} = \frac{tp}{tp + fp},$$

$$\text{recall} = \frac{tp}{tp + fn},$$

$$F_\beta = (1 + \beta^2) \frac{\text{precision} \times \text{recall}}{\beta^2 \text{precision} + \text{recall}}.$$

Precision-Recall Curve



$$\mathrm{AveP} = \int_0^1 p(r)dr$$

$$\mathrm{AveP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1.0\}} p_{\mathrm{interp}}(r)$$

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP(q)}}{Q}$$