

Empirical Approaches to Multilingual Lexical Acquisition

Lecturer: Timothy Baldwin



THE UNIVERSITY OF
MELBOURNE

Lecture 2

Introduction to Machine Learning

Machine Learning (ML)

- Hypothesis: pre-existing data repositories contain a lot of potentially important information
- Mission of ML: find it
- Definition of ML:
automatic extraction of **valid**, **novel**, **useful** and **comprehensible** knowledge (rules, regularities, patterns, constraints, models) from arbitrary sets of data

Underlying Motivation

- *We are drowning in data, but starving for knowledge!*
- Data = raw information
- Knowledge = set of patterns or models behind the data

ML Example: the *Cool/Cute* Classifier

- According to my 2 y.o. son:

<i>Entity</i>	<i>Class</i>	<i>Entity</i>	<i>Class</i>
self	cute	sports car	cool
self as baby	???	tiger	cool
big brother (4 y.o.)	cool	Hello Kitty	cute
big sister (6 y.o.)	cute	spoon	???
Mummy	cute	water	???

- What would we predict the class for the following to be:

train, koala, book on ML

Yeah yeah, but what's in it for me?

- Scenario 1:

You are a supermarket manager, wishing to boost sales without increasing expenditure

- Solution:

Strategically position products to entice consumers to spend more:

beer next to chips?

beer next to bathroom cleaner?

Yeah yeah, but what's in it for me?

- Scenario 1:

You are a supermarket manager, wishing to boost sales without increasing expenditure

- Solution:

Strategically position products to entice consumers to spend more

beer next to chips?

beer next to bathroom cleaner?

ASSOCIATION RULES

- Scenario 2:

You are an archaeologist in charge of classifying a mountain of fossilised bones, and want to quickly identify any “finds of the century” before sending the bones off to a museum

- Solution:

Identify bones which are of different size/dimensions/characteristics to others in the sample and/or pre-identified bones

- Scenario 2:

You are an archaeologist in charge of classifying a mountain of fossilised bones, and want to quickly identify any “finds of the century” before sending the bones off to a museum

- Solution:

Identify bones which are of different size/dimensions/characteristics to others in the sample and/or pre-identified bones

CLUSTERING, OUTLIER DETECTION

- Scenario 3:

You are an archaeologist in charge of classifying a mountain of fossilised bones, and want to come up with a consistent way of determining the species and type of each bone which doesn't require specialist skills

- Solution:

Identify some easily measurable properties of bones (size, shape, number of "lumps", ...) and compare any new bones to a pre-classified DB of bones

- Scenario 3:

You are an archaeologist in charge of classifying a mountain of fossilised bones, and want to come up with a consistent way of determining the species and type of each bone which doesn't require specialist skills

- Solution:

Identify some easily measurable properties of bones (size, shape, number of "lumps", ...) and compare any new bones to a pre-classified DB of bones

SUPERVISED CLASSIFICATION

- Scenario 4:

You are in charge of developing the next “release” of Coca Cola, and want to be able to estimate how well received a given recipe will be

- Solution:

Carry out tast tests over various “recipes” with varying proportions of sugar, caramel, caffeine, phosphoric acid, coca leaf extract, ... (and any number of “secret” new ingredients), and estimate the function which predicts customer satisfaction from these numbers

- Scenario 4:

You are in charge of developing the next “release” of Coca Cola, and want to be able to estimate how well received a given recipe will be

- Solution:

Carry out tast tests over various “recipes” with varying proportions of sugar, caramel, caffeine, phosphoric acid, coca leaf extract, ... (and any number of “secret” new ingredients), and estimate the function which predicts customer satisfaction from these numbers

REGRESSION

Machine Learning vs. Data Mining

- Machine learning tends to:
 - ★ be more concerned with theory than applications
 - ★ largely ignore questions of run time/scalability
- Data mining tends to:
 - ★ be more concerned with (business) applications than theory
 - ★ talk a lot about databases and run time/scalability
- Fuzzy dividing line between the two
- Google Munich is hiring “machine learning” and not “data mining” experts (according to Google sponsored links ca. 15/1/2008)

- We will refer to everything as “machine learning” for the purposes of this course
- In doing so, we will tend to shy away from many of the high-end scalability issues

Lexical Acquisition: Machine Learning or Data Mining?

- Is lexical acquisition more related to machine learning or data mining?
- As good empiricists, we answer question via Google counts (carried out on 15/1/2008):

	lexical_acquisition	*
machine_learning	6,460	1,710,000
data_mining	3,920	3,540,000
*	10,900	9,630,000,000

- **Pointwise mutual information** is an information-theoretic measure to determine the relative “association” between two events:

$$I(A; B) = \log \frac{P(A, B)}{P(A)P(B)}$$

NB: the pointwise mutual information between two independent events is 0

- $I(\text{lexical_acquisition}; \text{machine_learning}) \approx 11.7$

$$I(\text{lexical_acquisition}; \text{data_mining}) \approx 9.9$$

- Conclusion: lexical acquisition has more of a machine learning than data mining orientation

DATA REPRESENTATION

Terminology

- The input to a machine learning system consists of:
 - ★ **Instances**: the individual, independent examples of a concept also known as **exemplars**
 - ★ **Attributes**: measuring aspects of an instance also known as **features**
 - ★ **Concepts**: things that we aim to learn

Example: weather.nominal dataset

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
⋮	⋮	⋮	⋮	⋮

Example: weather.nominal dataset

Outlook	Temperature	Humidity	Windy	Play
INSTANCE ₁ sunny	hot	high	FALSE	no
INSTANCE ₂ sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
⋮	⋮	⋮	⋮	⋮

Example: weather.nominal dataset

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
		⋮	⋮	⋮

What's a Concept?

- Styles of learning concepts:
 - ★ Association learning:
 - detecting associations between features
 - ★ Clustering:
 - grouping similar instances into clusters
 - ★ Classification learning:
 - predicting a discrete class
 - ★ Regression:
 - predicting a numeric quantity

Association learning

- Detect frequent patterns, associations, correlations, or causal structures among sets of items or objects in dataset
- Frequent pattern: pattern (set of items, sequence, etc.) that occurs frequently in a database
- Any kind of structure is considered interesting, and no *a priori* sense of what we hope to predict
- Potential for massive number of association rules

Top-10 Association Rules for weather.nominal

```
# java weka.associations.Apriori -t data/weather.nominal.arff
```

1. humidity=normal windy=FALSE ==> play=yes
2. temperature=cool ==> humidity=normal
3. outlook=overcast ==> play=yes
4. temperature=cool play=yes ==> humidity=normal
5. outlook=rainy windy=FALSE ==> play=yes
6. outlook=rainy play=yes ==> windy=FALSE
7. outlook=sunny humidity=high ==> play=no
8. outlook=sunny play=no ==> humidity=high
9. temperature=cool windy=FALSE ==> humidity=normal play=yes
10. temperature=cool humidity=normal windy=FALSE ==> play=yes

Full weather.nominal Dataset

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

Clustering

- Finding groups of items that are similar
- Clustering is **unsupervised**
- The class of an example is not known
- Success often measured subjectively

Clustering over weather.nominal

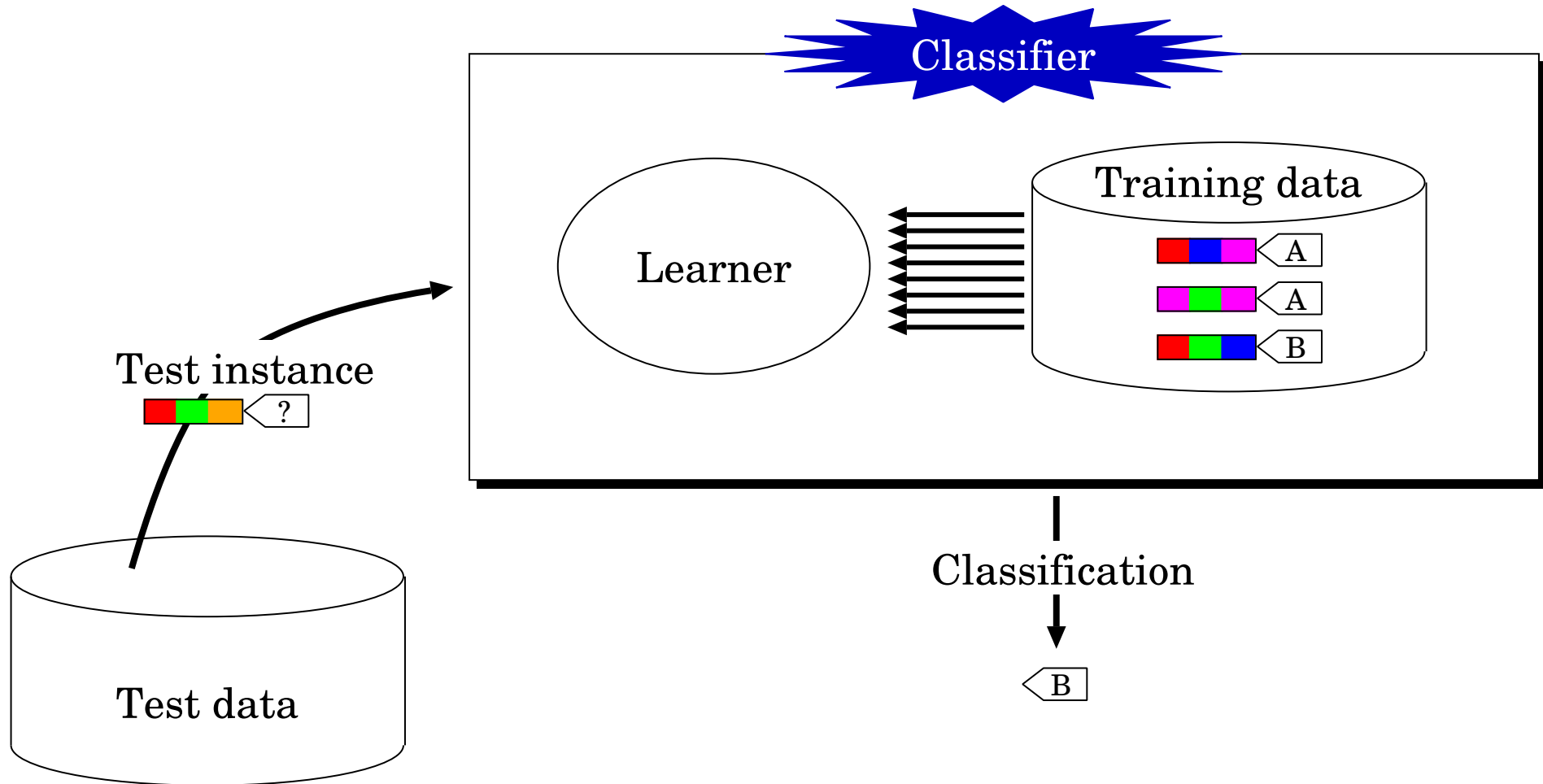
Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
⋮	⋮	⋮	⋮	⋮

Example Clusters for `weather.nominal`

Outlook	Temperature	Humidity	Windy	Cluster	Play
sunny	hot	high	FALSE	0	no
sunny	hot	high	TRUE	0	no
overcast	hot	high	FALSE	0	yes
rainy	mild	high	FALSE	1	yes
rainy	cool	normal	FALSE	1	yes
rainy	cool	normal	TRUE	1	no
overcast	cool	normal	TRUE	1	yes
sunny	mild	high	FALSE	0	no
sunny	cool	normal	FALSE	1	yes
rainy	mild	normal	FALSE	1	yes
sunny	mild	normal	TRUE	1	yes
overcast	mild	high	TRUE	1	yes
overcast	hot	normal	FALSE	0	yes
rainy	mild	high	TRUE	1	no

Classification learning

- Scheme is provided with actual outcome or **class**
- The learning algorithm is provided with a set of classified **training data**
- Measure success on fresh data for which class labels are known (**test data**)
- Classification learning is **supervised**



Example Predictions for `weather.nominal`

Outlook	Temperature	Humidity	Windy	Actual	Classified
sunny	hot	high	FALSE	no	
sunny	hot	high	TRUE	no	
overcast	hot	high	FALSE	yes	
rainy	mild	high	FALSE	yes	
rainy	cool	normal	FALSE	yes	
rainy	cool	normal	TRUE	no	
overcast	cool	normal	TRUE	yes	
sunny	mild	high	FALSE	no	
sunny	cool	normal	FALSE	yes	
rainy	mild	normal	FALSE	yes	
sunny	mild	normal	TRUE	(yes)	no
overcast	mild	high	TRUE	(yes)	yes
overcast	hot	normal	FALSE	(yes)	yes
rainy	mild	high	TRUE	(no)	yes

A Word on Supervision

- **Supervised** methods have prior knowledge of a closed set of classes and set out to discover and categorise new instances according to those classes
- **Unsupervised** methods dynamically discover the classes in the process of categorising the instances [STRONG DEFINITION]

OR

- **Unsupervised** methods categorise instances without the aid of pre-classified data [WEAK DEFINITION]

Regression

- Classification learning, but class is (continuous) numeric
- Learning is supervised
- Also known as **numeric prediction**

Example Predictions for `weather.nominal`

Outlook	Humidity	Windy	Play	Actual Temp	Classified Temp
sunny	85	FALSE	no	85	
sunny	90	TRUE	no	80	
overcast	86	FALSE	yes	83	
rainy	96	FALSE	yes	70	
rainy	80	FALSE	yes	68	
rainy	70	TRUE	no	65	
overcast	65	TRUE	yes	64	
sunny	95	FALSE	no	72	
sunny	70	FALSE	yes	69	
rainy	80	FALSE	yes	75	
sunny	70	TRUE	yes	(75)	68.8
overcast	90	TRUE	yes	(72)	76.2
overcast	75	FALSE	yes	(81)	70.6
rainy	91	TRUE	no	(71)	76.5

What's in an Attribute?

- Each instance is described by a fixed feature vector of attributes
- Attributes generally come in two types:
 1. nominal
 2. continuous

Nominal Attributes

- Values are distinct symbols (e.g. {sunny,overcast,rainy})
 - ★ values themselves serve only as labels or names
- Also called **categorical**, **enumerated**, or **discrete** (NB. “enumerated” and “discrete” imply an order which tends not to exist)
- Special case: dichotomy (“boolean” attribute)
- No relation is implied among nominal values (no ordering or distance measure), and only equality tests can be performed

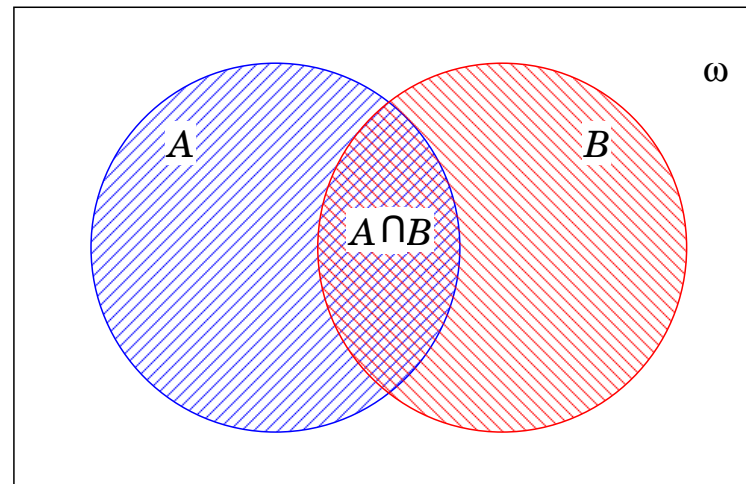
Continuous Attributes

- Ratio quantities are real-valued attributes with a well-defined zero point and (usu.) no explicit upper bound
- Also called **numeric**
- Example: attribute distance
 - Distance between an object and itself is zero
- All mathematical operations are allowed

(VERY) BASICS OF PROBABILITY AND INFORMATION THEORY

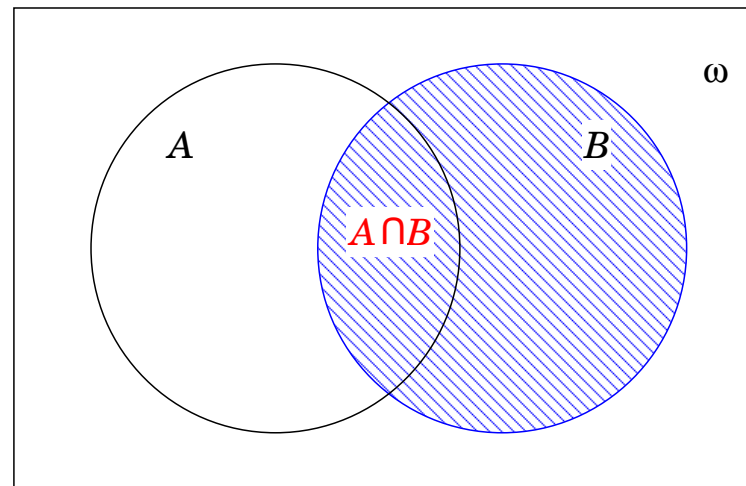
Basics of Probability Theory

- **Joint probability** ($P(A, B)$): the probability of both A and B occurring = $P(A \cap B)$



$$P(\text{ace, heart}) = \frac{1}{52}, \quad P(\text{heart, red}) = \frac{1}{4}$$

- **Conditional probability** ($P(A|B)$): the probability of A occurring given the occurrence of $B = \frac{P(A \cap B)}{P(B)}$



$$P(\text{ace}|\text{heart}) = \frac{1}{13}, \quad P(\text{heart}|\text{red}) = \frac{1}{2}$$

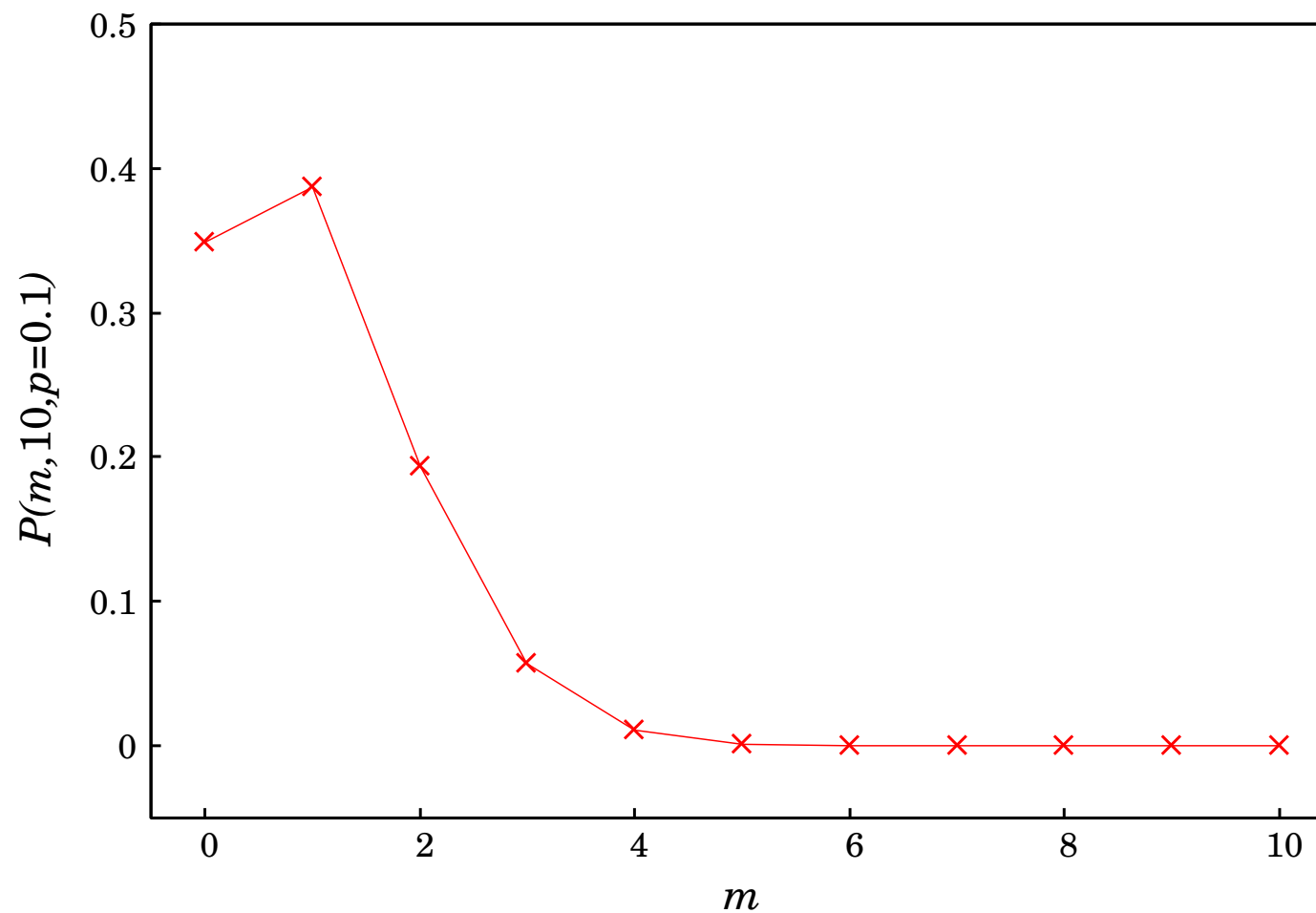
- **Multiplication rule:** $P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$
- **Prior probability** ($P(A)$): the probability of A occurring, given no additional knowledge about A
- **Posterior probability** ($P(A|B)$): the probability of A occurring, given background knowledge about event(s) B leading up to A
- **Independence:** A and B are independent iff $P(A \cap B) = P(A)P(B)$

Binomial Distributions

- A **binomial distribution** results from a series of independent trials with only two outcomes (i.e. **Bernoulli trials**)
e.g. multiple coin tosses ($\langle H, T, H, H, \dots, T \rangle$)
- The probability of an event with probability p occurring exactly m out of n times is given by

$$P(m, n, p) = \frac{n!}{m!(n-m)!} p^m (1-p)^{n-m}$$

Binomial Example: $P(m, 10, p = 0.1)$



Entropy

- Given a probability distribution, the information (in bits) required to predict an event is the distribution's **entropy** or **information value**
- The entropy of a discrete random event x with possible states $1, ..n$

is:

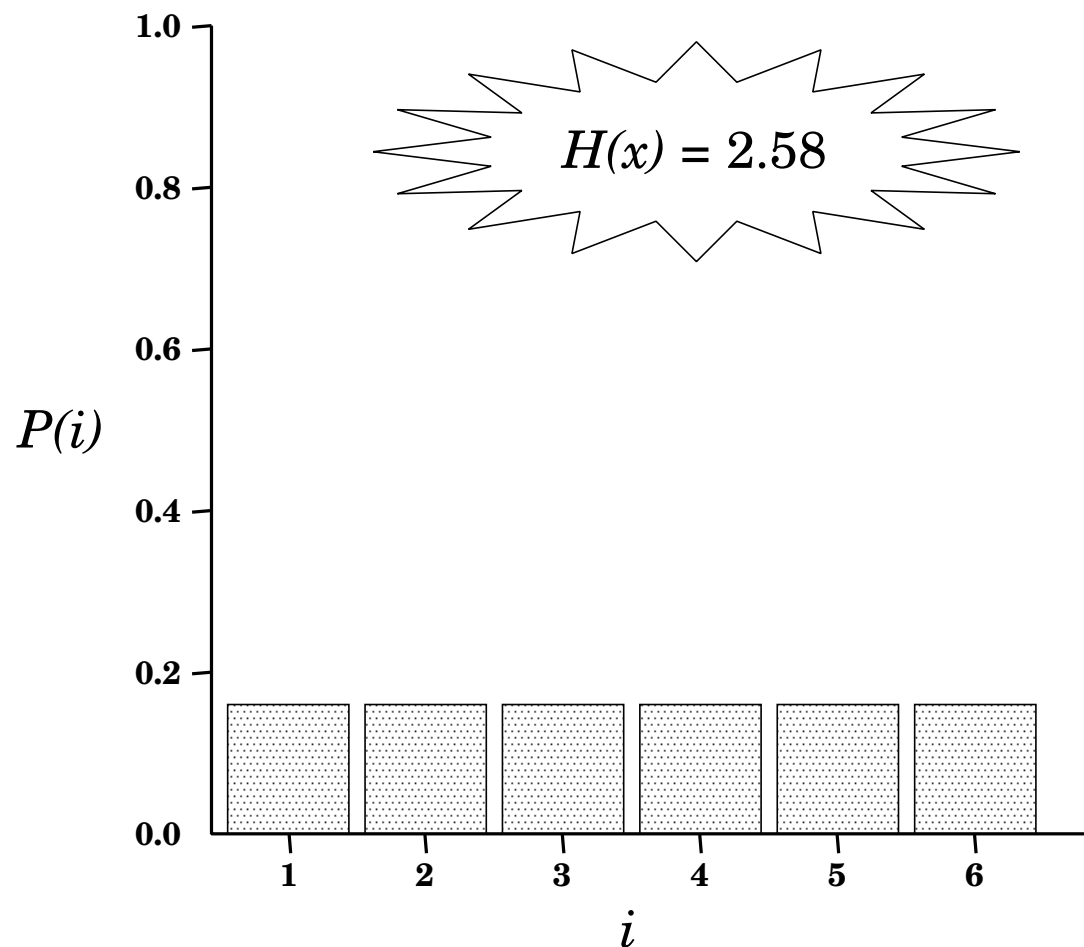
$$\begin{aligned} H(x) &= - \sum_{i=1}^n P(i) \log_2 P(i) \\ &= \frac{\text{freq}(\ast) \log_2(\text{freq}(\ast)) - \sum_{i=1}^n \text{freq}(i) \log_2(\text{freq}(i))}{\text{freq}(\ast)} \end{aligned}$$

where $0 \log_2 0 =^{def} 0$

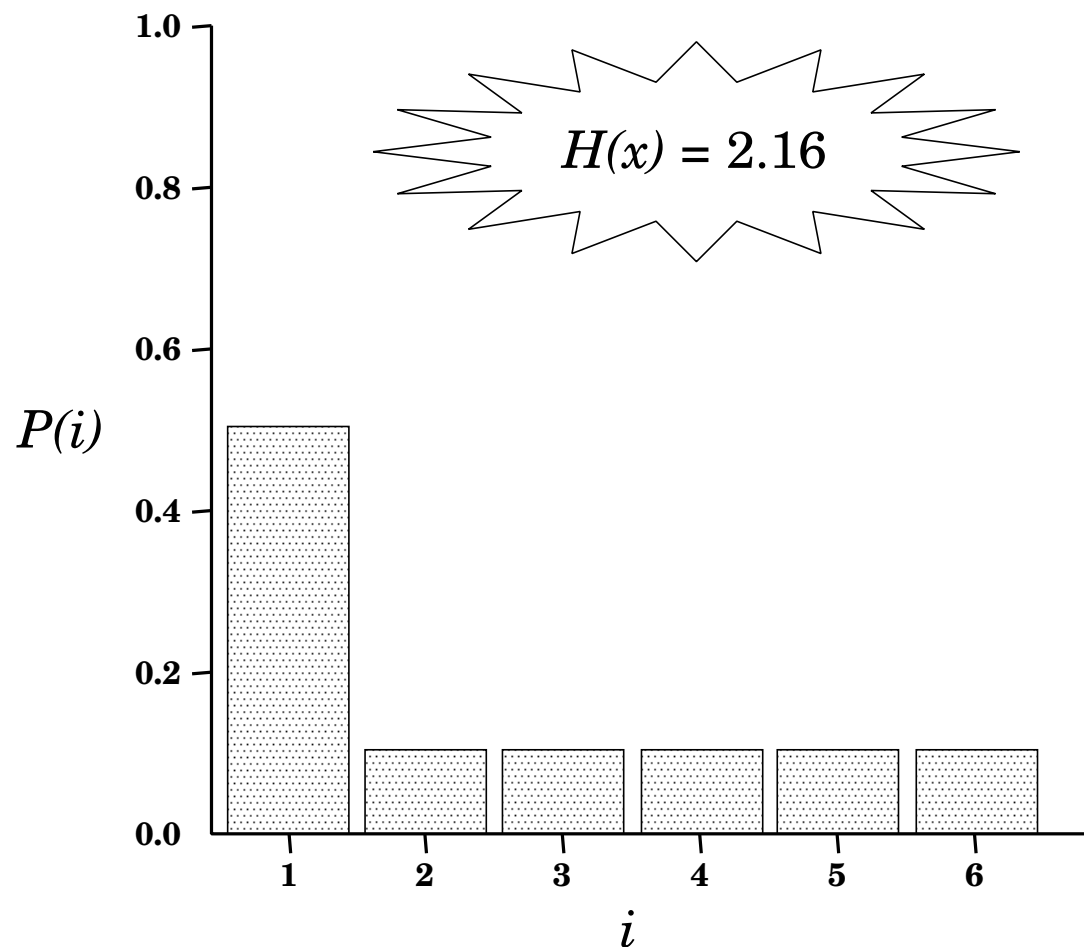
Interpreting Entropy Values

- A high entropy value means x is boring (uniform/flat)
- A low entropy value means x is varied (“peaky”)

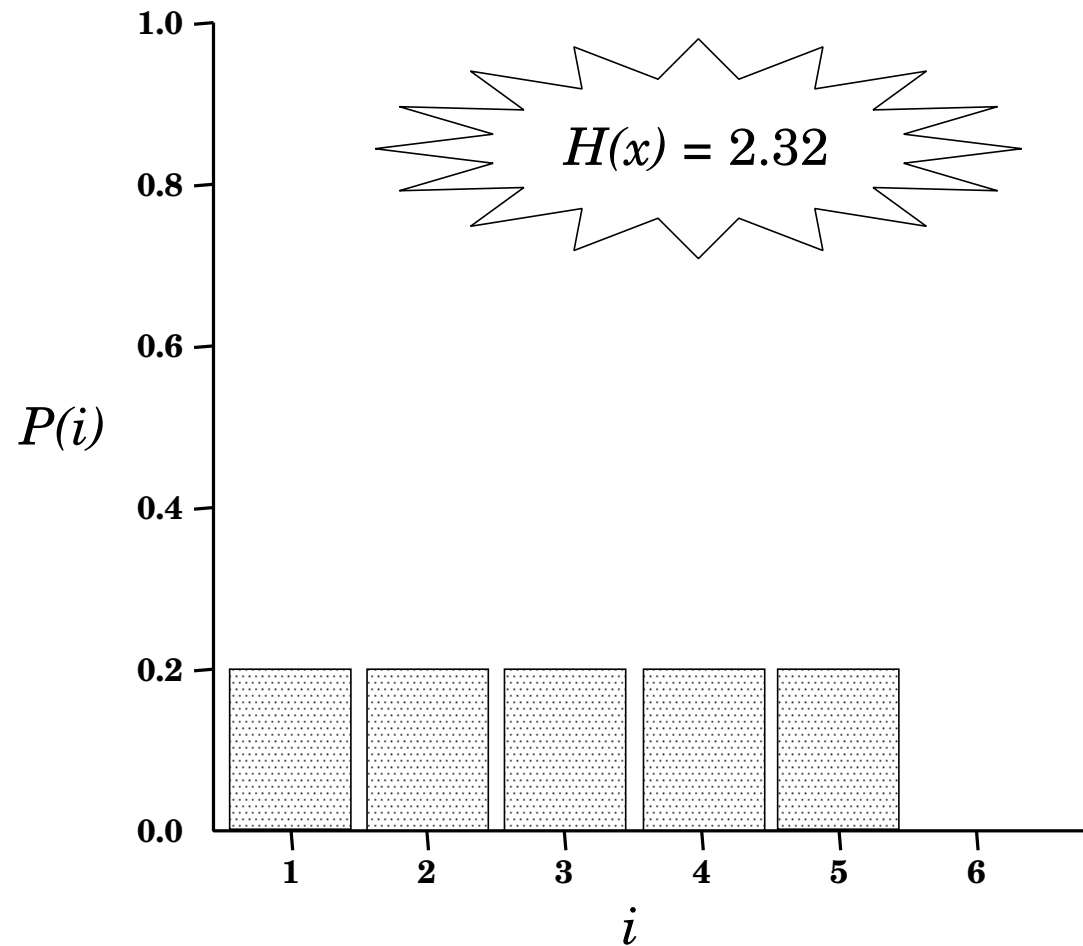
Entropy of Loaded Dice (1)



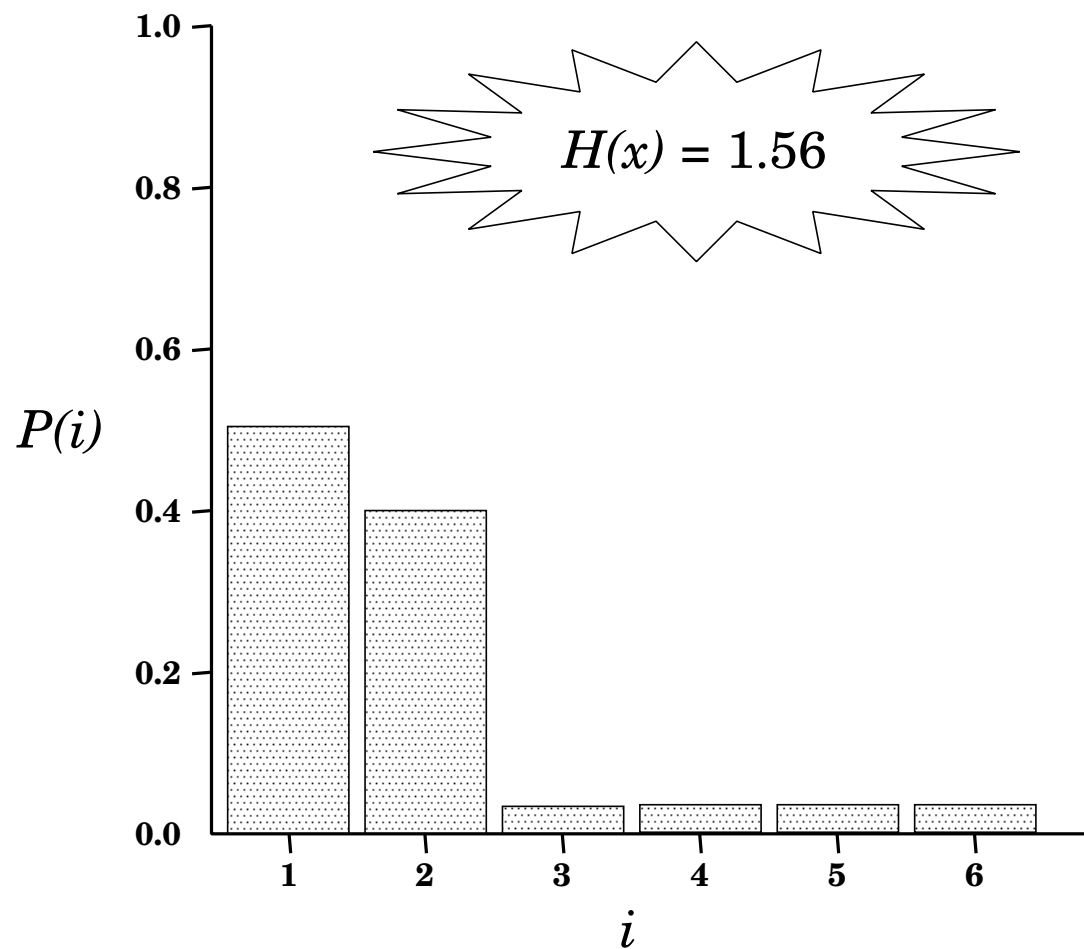
Entropy of Loaded Dice (2)



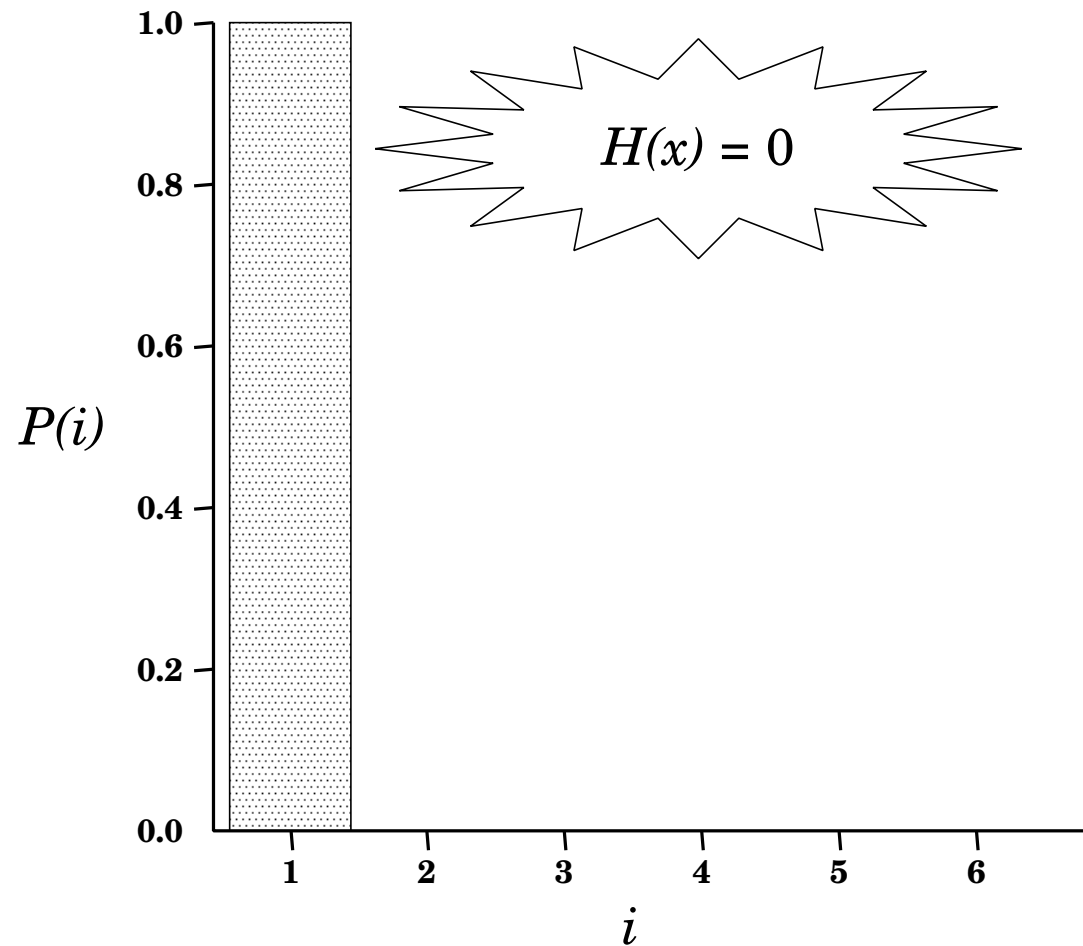
Entropy of Loaded Dice (3)



Entropy of Loaded Dice (4)



Entropy of Loaded Dice (5)



Estimating the Probabilities

- The most obvious way of generating the probabilities is via **maximum likelihood estimation** (MLE), using the frequency counts in the training data:

$$\hat{P}(c_j) = \frac{\text{freq}(c_j)}{\sum_k \text{freq}(c_k)}$$
$$\hat{P}(x_i|c_j) = \frac{\text{freq}(x_i, c_j)}{\text{freq}(c_j)}$$

- Based on this, a term frequency representation such as:

$$\vec{x} = \langle 233, 0, 2, 0, 0, 1, 0, \dots \rangle$$

would turn into something like:

$$\vec{x} = \langle 0.1, 0, 0.0002, 0, 0, 0.0001, 0, \dots \rangle$$

EVALUATION

Baselines vs. Benchmarks

- **Baseline** = naive method which we would expect any reasonably well-developed method to better
 - e.g. for a novice marathon runner, the time to walk 42km
- **Benchmark** = established rival technique which we are pitching our method against
 - e.g. for a marathon runner, the time of our last marathon run/the world record time/3 hours/...
- “Baseline” often used as umbrella term for both meanings

The Importance of Baselines

- Baselines are important in establishing whether any proposed method is doing better than “dumb and simple”
 - “dumb” methods often work surprisingly well
- Baselines are valuable in getting a sense for the intrinsic difficulty of a given task (cf. accuracy = 5% vs. 99%)
- In formulating a baseline, we need to be sensitive to the importance of positives and negatives in the classification task
 - limited utility of a baseline of `unsuitable` for a classification task aimed at detecting potential sites for new diamond mines

True/false Positives/negatives

- Basis of evaluation metrics is generally a **confusion matrix**:

		<i>Predicted</i>	
		<i>Y</i>	<i>N</i>
<i>Actual</i>	<i>Y</i>	true positive (TP)	false negative (FN)
	<i>N</i>	false positive (FP)	true negative (TN)

Accuracy

- The simplest form of evaluation is in terms of **classification accuracy** (aka **accuracy** or **success rate**)
- Accuracy is the proportion of instances for which we have correctly predicted the label
- For binary classifier:

$$\text{ACC} = \frac{TP + TN}{TP + FP + FN + TN}$$

- Alternatively, we sometimes talk about the **error rate**:

$$\text{ER} = \frac{FP + FN}{TP + FP + FN + TN}$$

N.B. $\text{ER} = 1 - \text{ACC}$

Precision and Recall

- If we wish to focus on only how well we have identified the positives and **not** what we have correctly ignored (or equivalently, performance relative to a single class of interest), we use **precision** and **recall**

$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{Recall} = \frac{TP}{TP + FN}$$

F-score

- In applications where we make individual decisions for each data point rather than generating a monolithic ranking (e.g. spam filtering), **F-score** gives us an overall picture of system performance:

$$\text{F-score} = (1 + \beta^2) \frac{PR}{R + \beta^2 P}$$

where P = precision and R = recall [**weighted harmonic mean**]

- Set β depending on how much we care about false negatives vs. false positives (cf. intelligence filtering vs. spam filtering) ... conventionally $\beta = 1$

Bias and Variance in Evaluation

- The (training) **bias** of a classifier is the average distance between the expected value and the estimated value
- The (test) **variance** of a classifier is the standard deviation between the expected and estimated value
- The **noise** in a dataset is the inherent variability of the training data
- In evaluation, we aim to minimise classifier bias and variance (but there's not a lot we can do about noise!)

Holdout

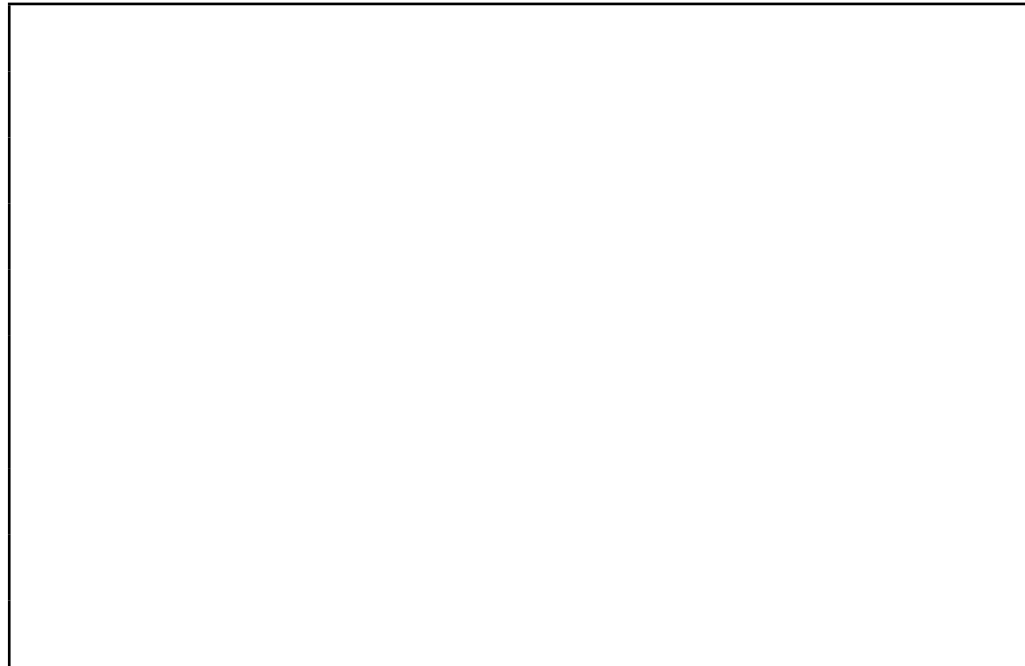
- Train a classifier over a fixed training dataset, and evaluate it over a held-out test dataset
- Advantages:
 - ★ simple to work with
 - ★ high reproducibility
- Disadvantages:
 - ★ trade-off between more training and more test data (variance vs. bias)
 - ★ representativeness of the training and test data

Random Subsampling

- Perform holdout over multiple iterations, randomly selecting the training and test data (maintaining a fixed size for each dataset) on each iteration
- Evaluate by taking the average across the iterations
- Advantages:
 - ★ reduction in variance and bias over “holdout” method
- Disadvantages:
 - ★ trade-off between more training and more test data

Cross Validation: Input

- Take our entire dataset:



Cross Validation: Partitioning

- Split up into N equal-sized partitions P_i :

P_1
P_2
P_3
P_4
P_5
P_6
P_7
P_8
P_9
P_{10}

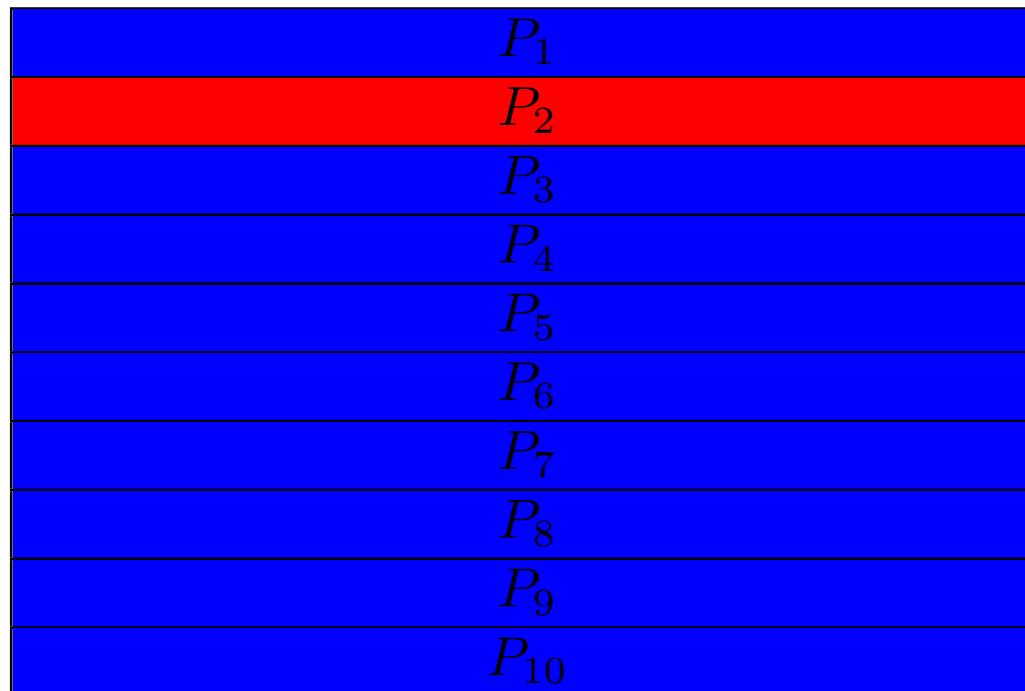
Cross Validation: Fold 1

- For each $i = 1 \dots N$, take P_i as the test data and $\{P_j : j \neq i\}$ as the training data

P_1
P_2
P_3
P_4
P_5
P_6
P_7
P_8
P_9
P_{10}

Cross Validation: Fold 2

- For each $i = 1 \dots N$, take P_i as the test data and $\{P_j : j \neq i\}$ as the training data



Cross Validation: Fold 3

- For each $i = 1 \dots N$, take P_i as the test data and $\{P_j : j \neq i\}$ as the training data



Cross Validation: Fold i

- And so on ...

Cross Validation: Evaluate

- Evaluate according to the average across the N iterations
- N is generally set to 10 (but possibly run over multiple iterations, with different partitions)
- Advantages:
 - ★ minimises bias and variance
 - ★ makes effective use of training data
- Disadvantages:
 - ★ efficiency

Stratified Cross Validation

- To further reduce variance and bias, we can additionally **stratify** the data = partition the data so as to maintain the overall class distribution within individual partitions
- Subtle questions about whether we are accurately modelling novel data instances or giving our classifier a push in the right direction?

Summary

- What are the 4 basic “flavours” of machine learning?
- What is the difference between supervised and unsupervised ML?
- How is data standardly represented in ML?
- What is entropy and what is its relevance to ML?
- What are accuracy, precision, recall and F-score?
- What are the different ways of partitioning up the data in evaluation, and what are the relative strengths and weaknesses of each?

References

- KOHAVI, RON. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1137–43.
- TAN, PANG-NING, MICHAEL STEINBACH, and VIPIN KUMAR. 2006. *Introduction to Data Mining*. Addison Wesley.
- WITTEN, IAN H., and EIBE FRANK. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco, USA: Morgan Kaufmann.
- , and —— . 2005. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco, USA: Morgan Kaufmann, second edition.