Machine Learning 2.13: Semi-supervised & Active Learning

Tom S. F. Haines
T.S.F.Haines@bath.ac.uk

UNIVERSITY OF BATH

Traditionally. . .

1. Collect data set
2. Train model
3. Test model
4. Use model

UNIVERSITY OF BATH

Traditionally...

1. **Collect data set**
2. Train model
3. Test model
4. Use model

- (for teaching) Often ignore data set...
  ...when it's arguably the most important part!

UNIVERSITY OF BATH

Data sets

- Often:
  - Collecting data = cheap
  - Labelling = expensive
- e.g. Doctors diagnosing x-rays
  (especially if segmenting)
- Can also be dangerous, e.g. invasive diagnostic procedures

UNIVERSITY OF BATH

Unlabelled data

- Semi-supervised learning:
  Not all data is labelled

UNIVERSITY OF BATH

Unlabelled data

- Semi-supervised learning:
  Not all data is labelled

- Active learning:
  Not all data is labelled...
  ...and the computer gets to decide what to label!

UNIVERSITY OF BATH

Unlabelled data

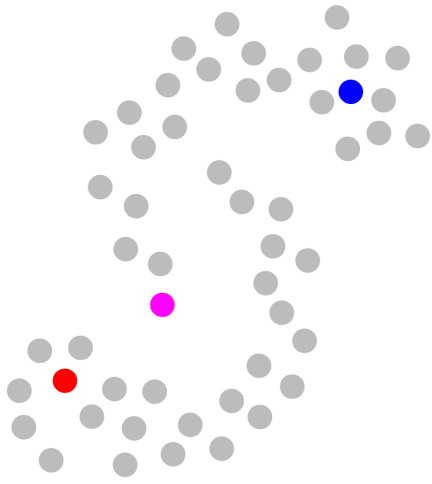
- Semi-supervised learning:
  Not all data is labelled
- Active learning:
  Not all data is labelled...
  ...and the computer gets to decide what to label!
- Related to:
  - Optimal experimental design (statistics)
  - Automated science
  - Hyperparameter optimisation
  - Machine teaching

Semi-supervised Learning

UNIVERSITY OF BATH

Distance

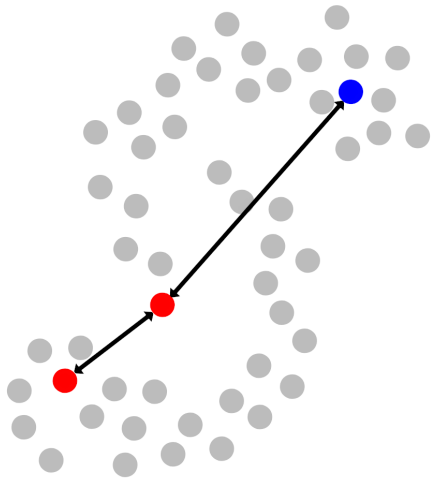
- Class of magenta point?



UNIVERSITY OF BATH

Distance

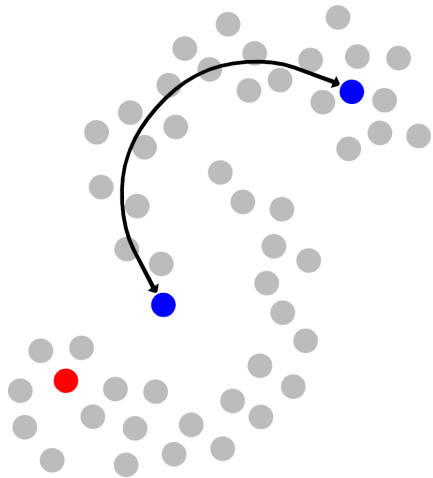
- Class of magenta point?
- Euclidean distance: Red



UNIVERSITY OF BATH

Distance

- Class of magenta point?
- Euclidean distance: Red
- **Manifold** of unlabelled points: Blue



UNIVERSITY OF BATH

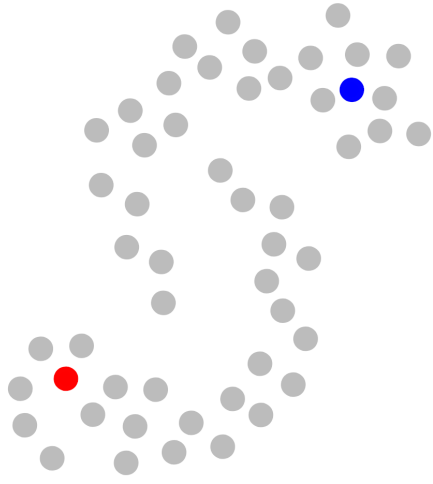
Semi-supervised learning

- Semi-supervised learning = classification on a manifold
- Most approaches:
  - Learn manifold (unsupervised, all exemplars)
  - Learn classifier on manifold (supervised, labelled exemplars)
- Improvements:
  - Simultaneous optimisation
  - Focusing on classification boundary

UNIVERSITY OF BATH

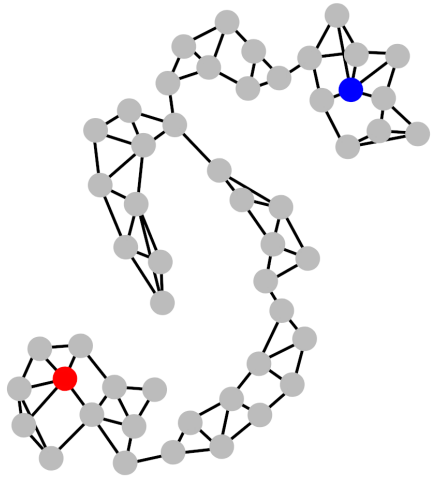
Simple algorithm

- Steps:



Simple algorithm

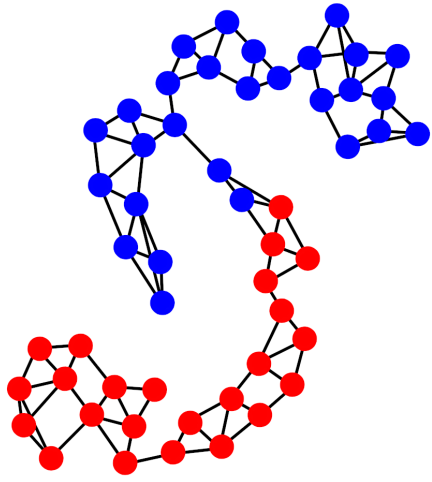
- Steps:
  1. Construct nearest neighbour graph (e.g. $N = 3$, PCA first)



UNIVERSITY OF BATH

Simple algorithm

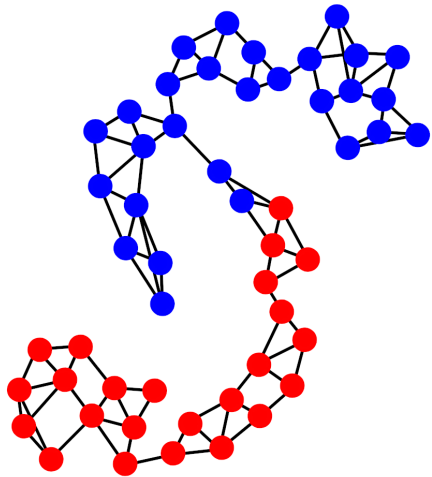
- Steps:
  1. Construct nearest neighbour graph (e.g. $N = 3$, PCA first)
  2. Label points with closest neighbour (on manifold with message passing)



UNIVERSITY OF BATH

Simple algorithm

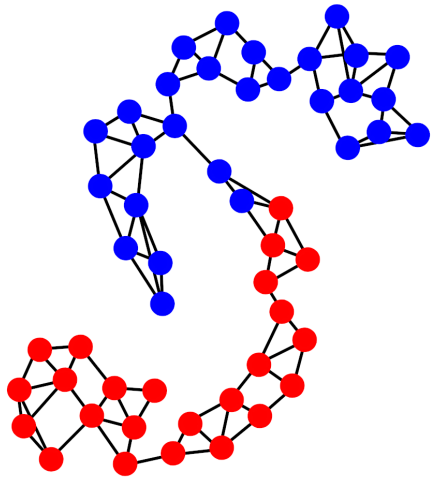
- Steps:
  1. Construct nearest neighbour graph
     (e.g. $N = 3$, PCA first)
  2. Label points with closest neighbour
     (on manifold with message passing)
  3. Classification with nearest neighbours
     (Euclidean space, all data)



UNIVERSITY OF BATH

Simple algorithm

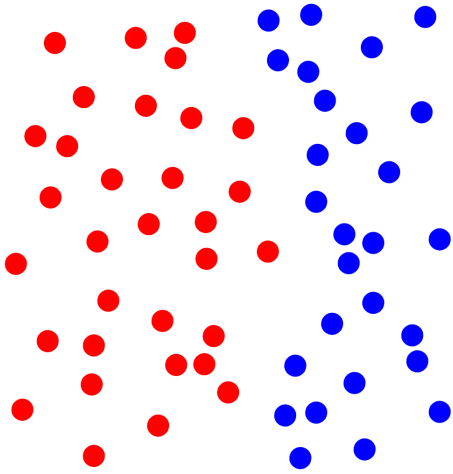
- Steps:
  1. Construct nearest neighbour graph
     (e.g. $N = 3$, PCA first)
  2. Label points with closest neighbour
     (on manifold with message passing)
  3. Classification with nearest neighbours
     (Euclidean space, all data)
- Poor robustness
- Assumes separated classes
- Better approaches exist
  (but are fundamentally the same idea)



Active Learning

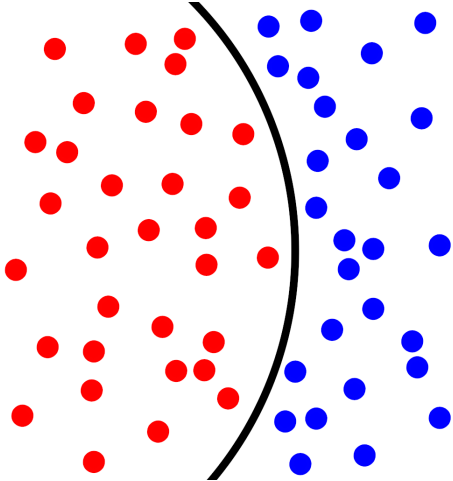
UNIVERSITY OF BATH

Unequal labels

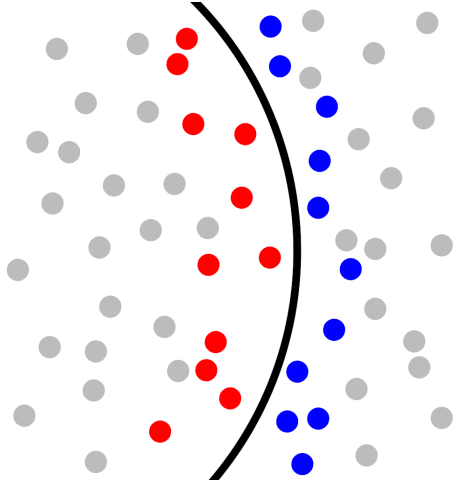


UNIVERSITY OF BATH

Unequal labels



Unequal labels



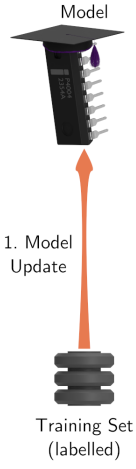
- Only need exemplars on edge!
  (same concept as support vector machine)

UNIVERSITY OF BATH

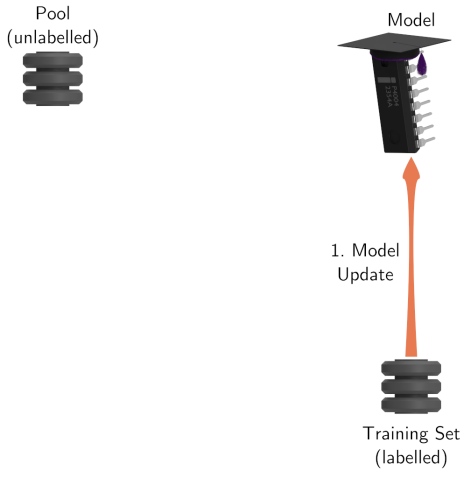
Active learning

Training Set
(labelled)

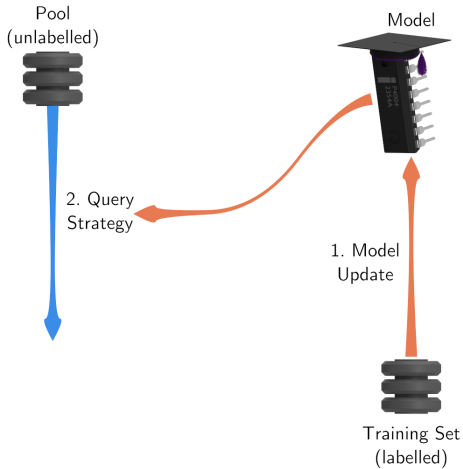
Active learning



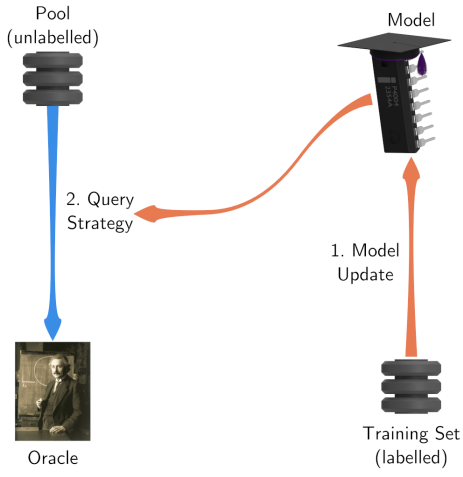
Active learning



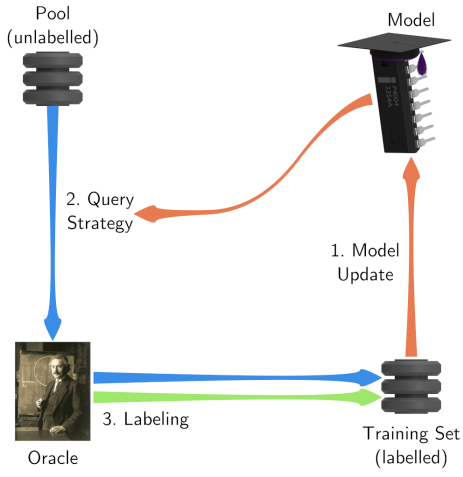
Active learning



Active learning



Active learning



UNIVERSITY OF BATH

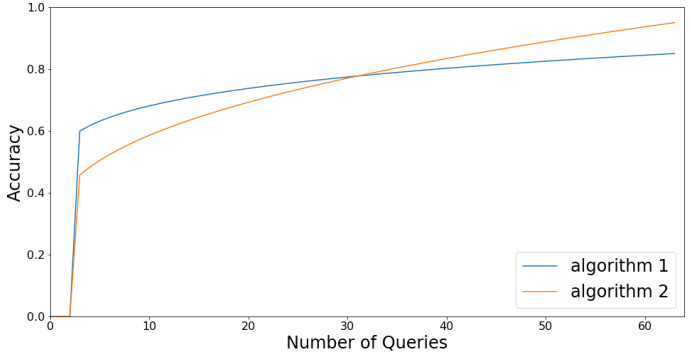
Measuring performance

- Goal: Maximum performance from least queries

UNIVERSITY OF BATH

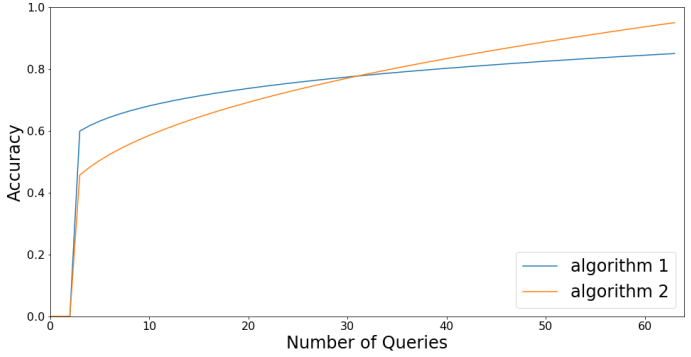
Measuring performance

- Goal: Maximum performance from least queries



Measuring performance

- Goal: Maximum performance from least queries



- Notes:
  - Use fake oracle (i.e use fully labelled test set)
  - Average many runs (stochastic approaches)
  - Single number: Area under curve, or accuracy at specified point
  - Algorithms do crossover – stopping point may matter!

Stopping

- Usually problem specific
- Running out of time / money
- Improvement no longer worth it
- Cost/benefit analysis

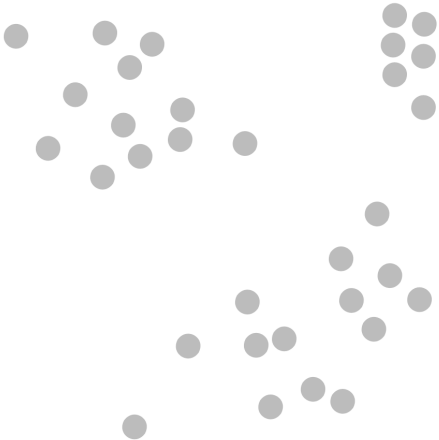
UNIVERSITY OF BATH

Baseline

- Randomly selecting an exemplar each time
- Does much better than you would imagine!
  (on balanced data sets)

UNIVERSITY OF BATH

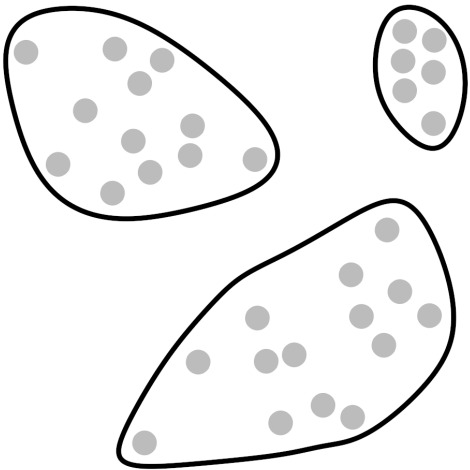
Clustering approach



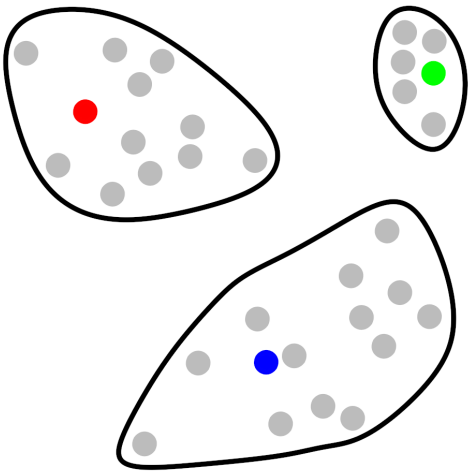
UNIVERSITY OF BATH

Clustering approach

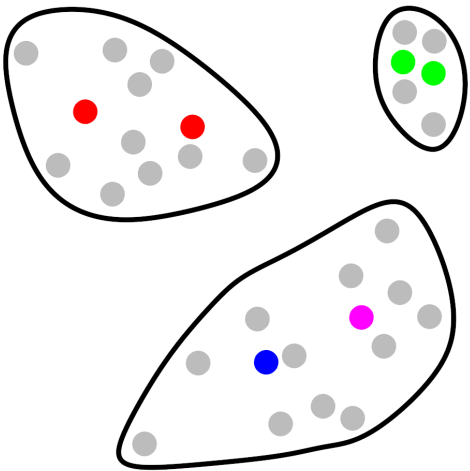
- Cluster data



Clustering approach

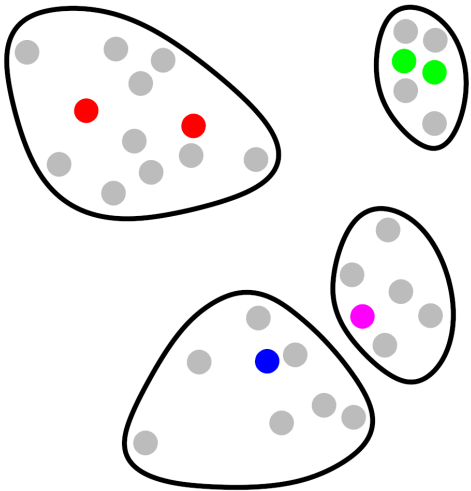


- Cluster data
- Exemplar in each cluster $\rightarrow$ Oracle



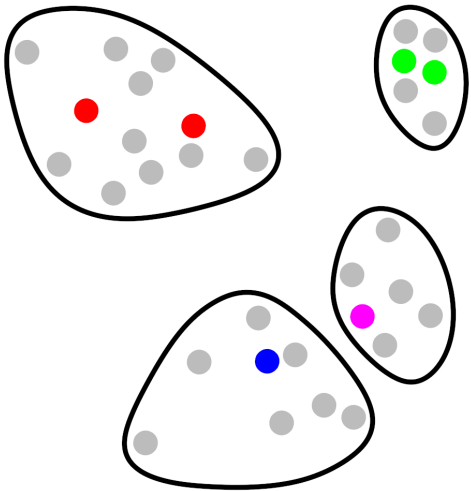
Clustering approach

- Cluster data
- Exemplar in each cluster $\rightarrow$ Oracle
- Repeat for confidence



Clustering approach

- Cluster data
- Exemplar in each cluster $\rightarrow$ Oracle
- Repeat for confidence
- Contradiction $\implies$ refine clustering



Clustering approach

- Cluster data
- Exemplar in each cluster $\rightarrow$ Oracle
- Repeat for confidence
- Contradiction $\implies$ refine clustering

- Keep going until confident
- Assumes classes are *separable*

UNIVERSITY OF BATH

Probabilistic querying

- Classifier probabilistic $\therefore$ have $P(c|x)$
  (c is a class $\in C$, x an exemplar)
- Approaches:

UNIVERSITY OF BATH

Probabilistic querying

- Classifier probabilistic $\therefore$ have $P(c|x)$
  (c is a class $\in C$, x an exemplar)
- Approaches:
  - Uncertainty (margin) sampling:
    Smallest difference between most and second most probable classes

$$\underset{x}{\operatorname{argmin}} \left(P(c_1|x) - P(c_2|x)\right) \qquad c_1 = \underset{c \in C}{\operatorname{argmax}} \left(P(c|x)\right) \qquad c_2 = \underset{c \in C - \{c_1\}}{\operatorname{argmax}} \left(P(c|x)\right)$$

UNIVERSITY OF BATH

Probabilistic querying

- Classifier probabilistic $\therefore$ have $P(c|x)$
  (c is a class $\in C$, x an exemplar)
- Approaches:
  - Uncertainty (margin) sampling:
    Smallest difference between most and second most probable classes

$$\underset{x}{\operatorname{argmin}} \left(P(c_1|x) - P(c_2|x)\right) \qquad c_1 = \underset{c \in C}{\operatorname{argmax}} \left(P(c|x)\right) \qquad c_2 = \underset{c \in C - \{c_1\}}{\operatorname{argmax}} \left(P(c|x)\right)$$

  - Least confident:
    Exemplar with whose most likely class is least probable

$$\underset{x}{\operatorname{argmin}} \left(\max_{c \in C} \left(P(c|x)\right) \right)$$

UNIVERSITY OF BATH

Probabilistic querying

- Classifier probabilistic $\therefore$ have $P(c|x)$
  (c is a class $\in C$, x an exemplar)
- Approaches:
  - Uncertainty (margin) sampling:
    Smallest difference between most and second most probable classes

$$\underset{x}{\operatorname{argmin}} \left(P(c_1|x) - P(c_2|x)\right) \qquad c_1 = \underset{c \in C}{\operatorname{argmax}} \left(P(c|x)\right) \qquad c_2 = \underset{c \in C - \{c_1\}}{\operatorname{argmax}} \left(P(c|x)\right)$$

  - Least confident:
    Exemplar with whose most likely class is least probable

$$\underset{x}{\operatorname{argmin}} \left(\max_{c \in C} \left(P(c|x)\right) \right)$$

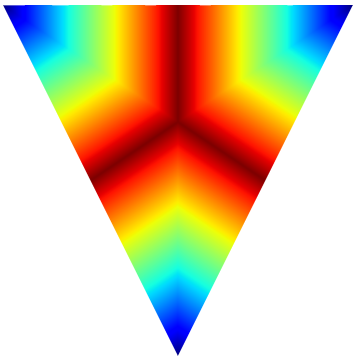
  - Entropy:
    Exemplar with maximum entropy class probabilities

$$\underset{x}{\operatorname{argmax}} \left(-\sum_{c \in C} P(c|x) \log \left(P(c|x)\right) \right)$$

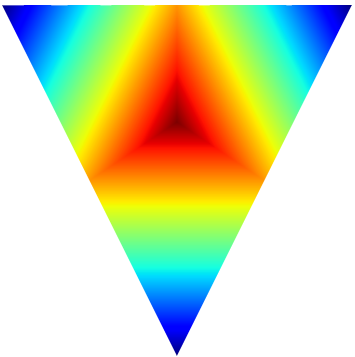
UNIVERSITY OF BATH

Visualisation

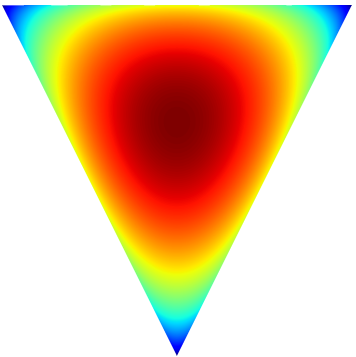
- Class in each corner
- Probabilities blended linearly



Uncertainty (margin) sampling



Least confident



Entropy

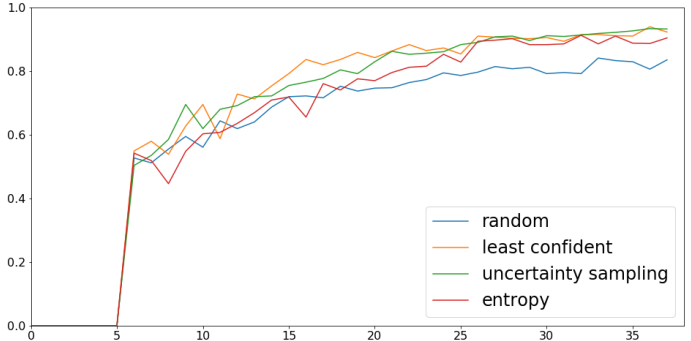
UNIVERSITY OF BATH

Example

- Problem: Identifying glass at crime scene
  Expensive $\rightarrow$ need broken glass!

Example

- Problem: Identifying glass at crime scene
  Expensive → need broken glass!



UNIVERSITY OF BATH

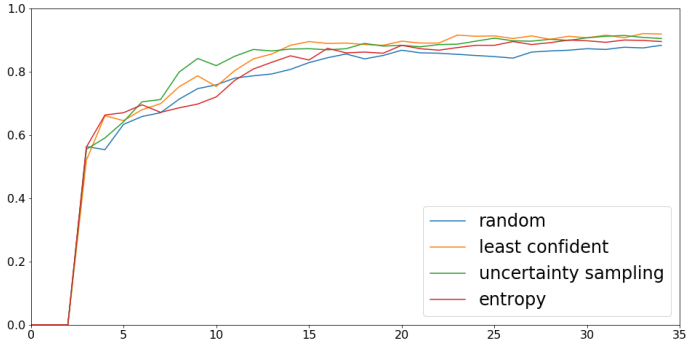
Random is competitive!

- Problem: Identifying seeds (visually near-identical)
  Expensive $\rightarrow$ wait for plant to grow!

UNIVERSITY OF BATH

Random is competitive!

- Problem: Identifying seeds (visually near-identical)
  Expensive → wait for plant to grow!



- Balanced data only – real data usually imbalanced

Bias

- Active learning selects biased sample (by design)

Bias

- Active learning selects biased sample (by design)
- Assume pool unbiased $\therefore$ classify pool to
  - Estimate class probabilities
  - Calibrate probability response (Kolmogorov–Smirnov)
- Combine with semi-supervised learner

UNIVERSITY OF BATH

Query by committee

- Ensemble model: Select least consistent exemplar
- *Probabilistic querying* specialisation
  (included for terminology)

- Usually done with Bayesian parameter averaging
  (models drawn from posterior)
- Slow

UNIVERSITY OF BATH

Expected model change

- *Select exemplar that could cause biggest change to model*
- Calculate *expected model change* for each exemplar:
  1. Hallucinate labelling it with each class
  2. Update model for each hallucination; measure parameter change
  3. Calculate expected change from class assignment probabilities (of current model)
- *Parameter change* problematic
- Really slow

UNIVERSITY OF BATH

Expected error reduction

- Fixes *expected model change*

- Define error:

$$= \int L\left(P(c|x), P_D(c|x)\right) dx$$

where

- $L(\cdot, \cdot)$ = loss between distributions
- $P(c|x)$ = true distribution (unknown)
- $P_D(c|x)$ = learned distribution

Expected error reduction

- Fixes *expected model change*

- Define error:

$$= \int L\left(P(c|x), P_D(c|x)\right) dx$$

where

- $L(\cdot, \cdot)$ = loss between distributions
- $P(c|x)$ = true distribution (unknown)
- $P_D(c|x)$ = learned distribution

- Approximate:
  - Monte Carlo integration with data
  - $P_D(c|x)$ is best estimate of $P(c|x)$

$$= \frac{1}{|P|} \sum_{x \in P} L\left(P_D(c|x), P_D(c|x)\right)$$

(P = all data, labelled and unlabelled)

- Probabilistic querying costs $\approx$ error (depending on choice of $L(\cdot)$)

Expected error reduction

- Fixes *expected model change*

- Define error:

$$= \int L\left(P(c|x), P_D(c|x)\right) dx$$

where

- $L(\cdot, \cdot)$ = loss between distributions
- $P(c|x)$ = true distribution (unknown)
- $P_D(c|x)$ = learned distribution

- Approximate:
  - Monte Carlo integration with data
  - $P_D(c|x)$ is best estimate of $P(c|x)$

$$= \frac{1}{|P|} \sum_{x \in P} L\left(P_D(c|x), P_D(c|x)\right)$$

(P = all data, labelled and unlabelled)

- Probabilistic querying costs $\approx$ error (depending on choice of $L(\cdot)$)

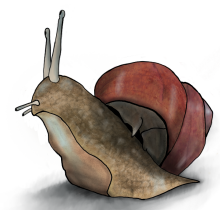
- Hallucinate future as before, and select best expectation

- Horrifically slow
- 2, or more, moves into future: Insanely slow

Snails

- Previous approaches are slow
- Human may be waiting

- Theoretically nice, pragmatically useless
- Can optimise; incremental learning a given



Active Discovery

Active discovery

- Thus far: Classes known in advance
- Often not true, e.g. only 0.001% of stars may be evidence of new physics

UNIVERSITY OF BATH

Active discovery

- Thus far: Classes known in advance
- Often not true, e.g. only 0.001% of stars may be evidence of new physics
- **Active discovery**: Finds unknown classes
- Often outlier detection:
  Fit density estimate, give low probability exemplars to oracle

UNIVERSITY OF BATH

Active discovery and learning

- Simultaneously:
  - Discover new classes
  - Refine classification of existing
- Exploration/exploitation trade off

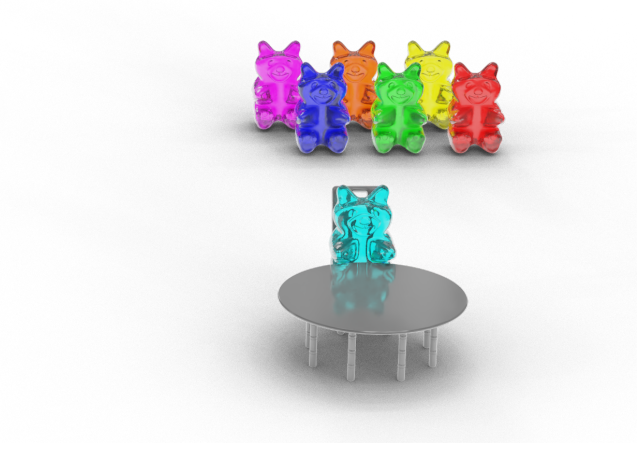
UNIVERSITY OF BATH

Chinese restaurant process

Customers queuing outside restaurant

UNIVERSITY OF BATH

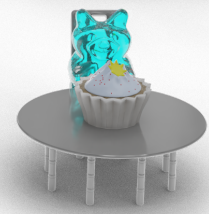
Chinese restaurant process



First takes a seat at a new table

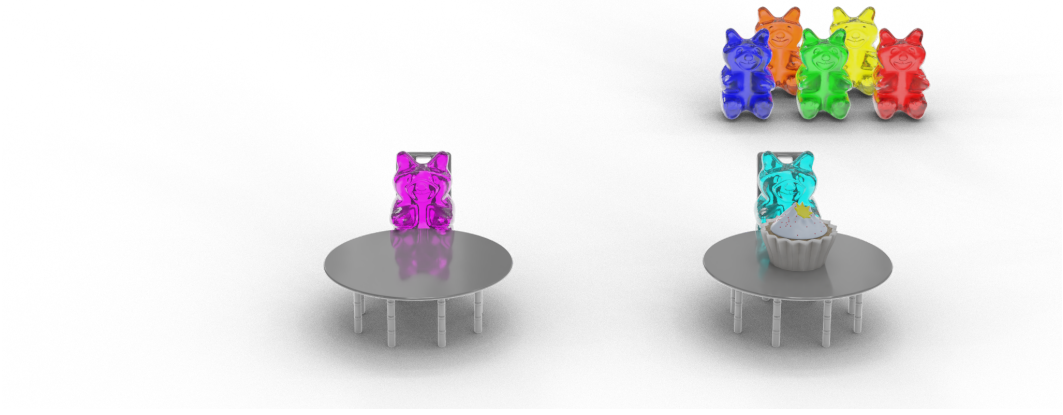
UNIVERSITY OF BATH

Chinese restaurant process



As first at table they choose what is eaten (draw from base measure)

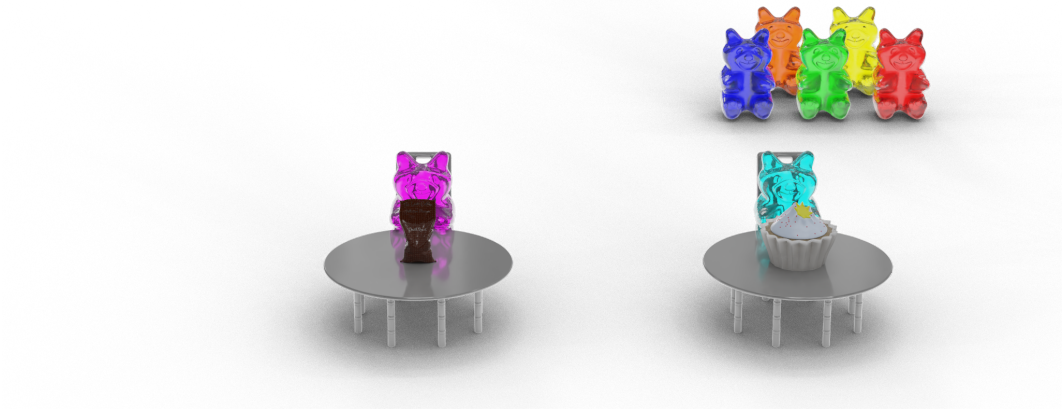
Chinese restaurant process



Second happens to choose a new table, $P(\texttt{new}) = \frac{\alpha}{\alpha+1}$

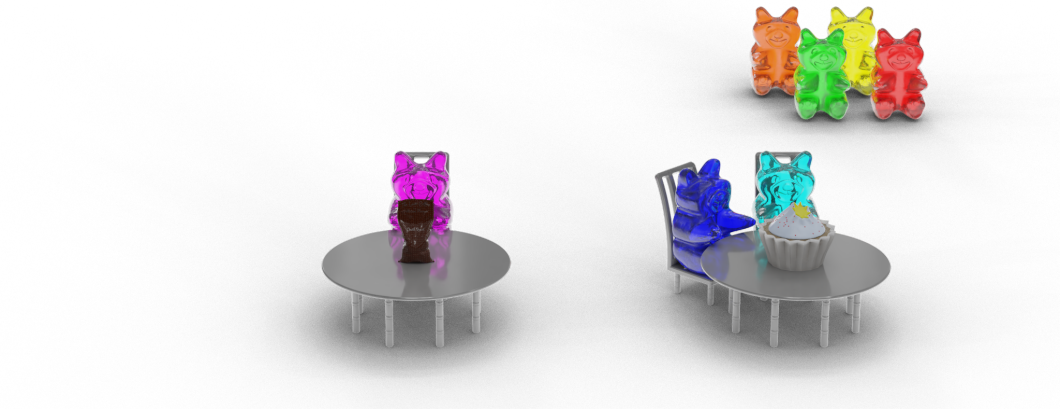
UNIVERSITY OF BATH

Chinese restaurant process



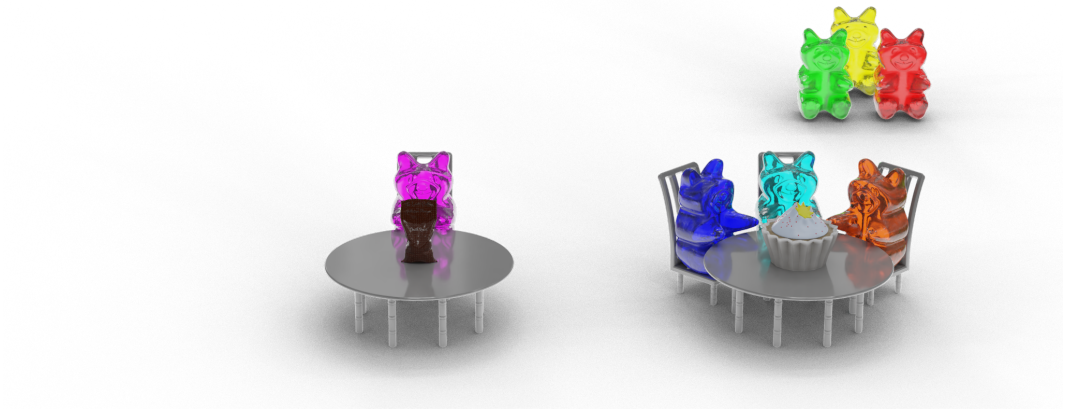
New table means new menu choice (draw from base measure)

Chinese restaurant process



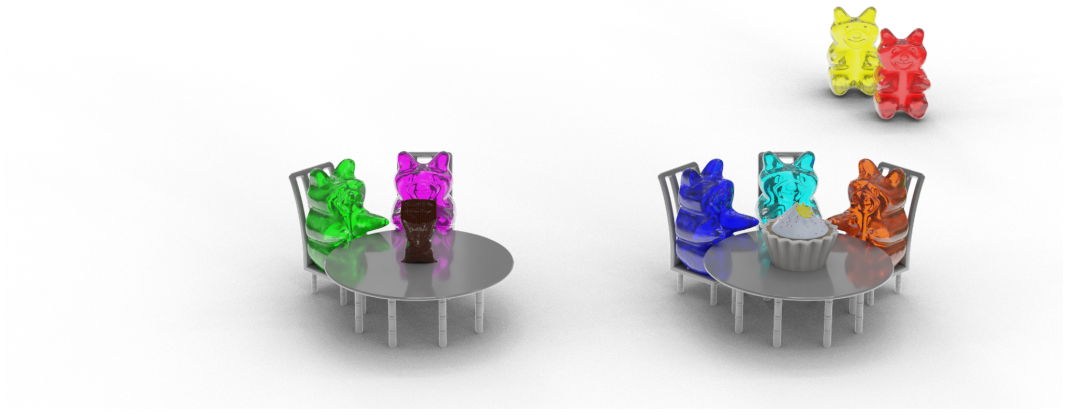
Third happens to sit at first table, $P(\texttt{table}_1) = \frac{1}{\alpha+2}$

Chinese restaurant process



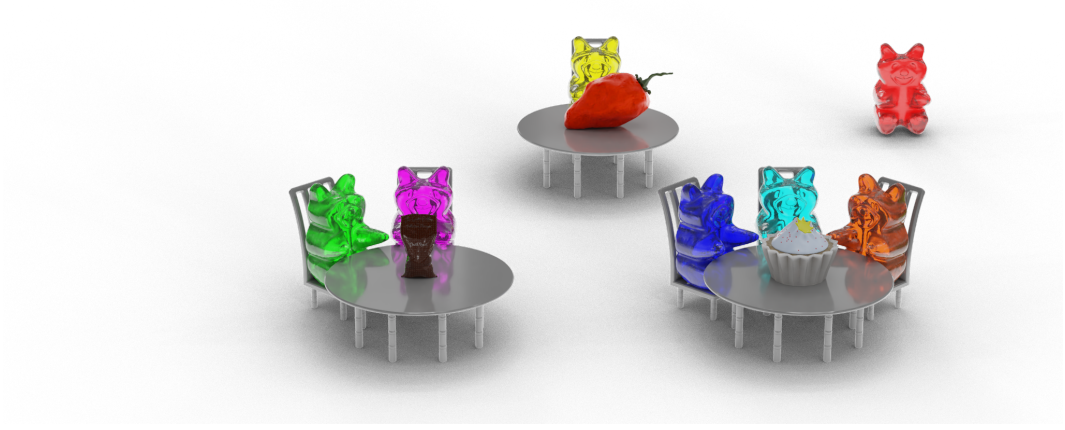
So does the fourth, $P(\texttt{table}_1) = \frac{2}{\alpha+3}$
Note "rich get richer" property

Chinese restaurant process



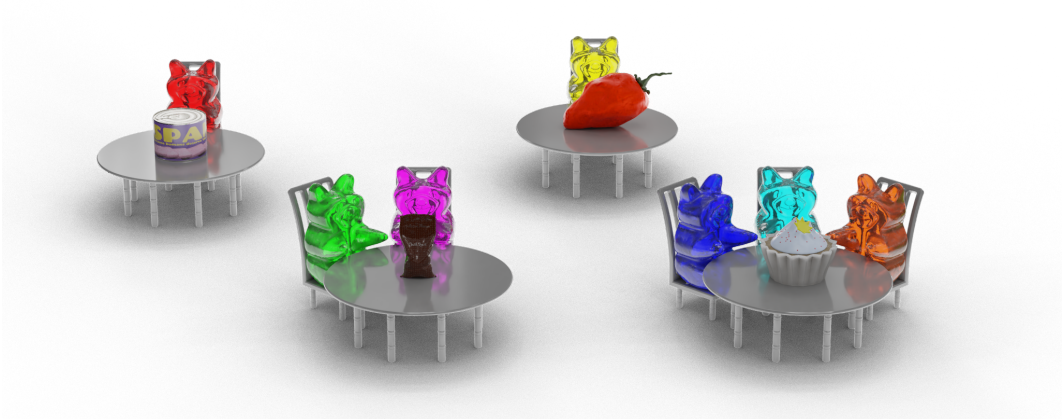
Fifth goes for second table, $P(\texttt{table}_2) = \frac{1}{\alpha+4}$

Chinese restaurant process



Sixth goes for a new table, $P(\texttt{new}) = \frac{\alpha}{\alpha+5}$

Chinese restaurant process

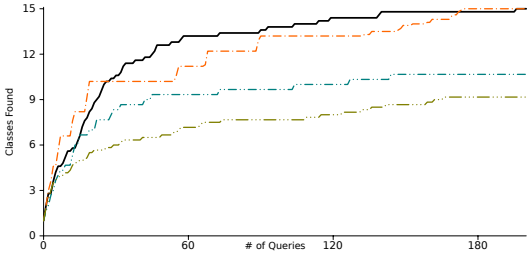


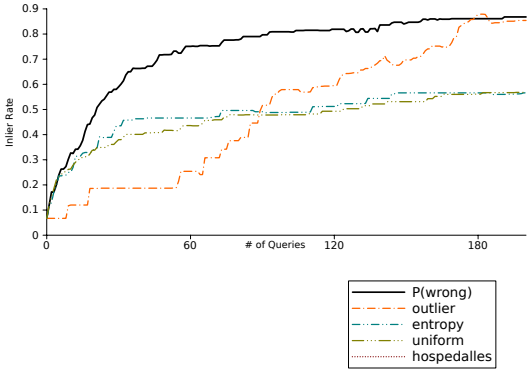
Seventh goes for a new table, $P(\texttt{new}) = \frac{\alpha}{\alpha+6}$
Note that the probability of a new table goes down with more customers

UNIVERSITY OF BATH

$P(\text{wrong})$

- Dirichlet process prior over classes = Chinese restaurant process
- Probability of *something new*! (discovery part)
- Intrusion detection problem – KDD99





Variants

- Regression
- Batch: Select many exemplars at once
- Cost sensitive: Labelling cost varies; multiple labelling options
  (e.g. does the image contain a cat vs click on the cat vs segment the cat)
- Online: Exemplars are seen once, and a decision to label has to be made immediately
- Unreliable oracles: Oracles make mistakes
  (Amazon mechanical Turk)

UNIVERSITY OF BATH

Which approach?

- You can't run a proper experiment!
  - Identify and test similar problems
  - Intuition for the rest!
  - Verify after the fact
- Factor in needs of humans

UNIVERSITY OF BATH

Summary

- Semi-supervised learning: Uses unlabelled data
- Active learning: Computer asks for help
- Active discovery: Computer finds new things
- Many approaches
- Be cautious!

UNIVERSITY OF BATH

Further reading

- Elegant semi-supervised paper:
  "*Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions*",
  by Zhu, Ghahramani & Lafferty (2003)

- Summary of active learning:
  "*Active Learning Literature Survey*",
  by B. Settles (2008)

- $P(\text{wrong})$ paper:
  "*Active rare class discovery and classification using dirichlet processes*",
  by Haines & Xiang (2014)

- Combined semi-supervised learning and active learning:
  "*Hierarchical Subquery Evaluation for Active Learning on a Graph*",
  by Mac Aodha, Campbell, Kautz & Brostow (2014)

UNIVERSITY OF BATH

Sources

- Glass identification:
  `https://archive.ics.uci.edu/ml/datasets/Glass+Identification`
- Seed identification:
  `http://archive.ics.uci.edu/ml/datasets/seeds`
- Intrusion detection:
  `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`