Active Learning using Dirichlet Processes for Rare Class Discovery and Classification

Tom S. F. Haines & Tao Xiang
{thaines,txiang}@eecs.qmul.ac.uk

Queen Mary, University of London

30$^{\text{th}}$ August 2011

Roadmap

Note that code can be obtained from *thaines.com*

Active Learning

- Training a classifier consists of **collecting data**, then **labelling the data** and, finally, **fitting a model**.
- Data collection can often be automated, and model fitting is a problem of computation... labelling however typically requires human interaction, and is hence *expensive*.
- Active learning endeavours to minimise this expense. It orders the training exemplars to get as much performance as possible with the least effort.
- When to stop training is usually left to the user.

Discovery & Classification

- **Discovery** is when not all classes are known, and need to be found.
- **Classification** is where the classes are considered to be known but the boundaries between them need to be refined.
- Active learning is typically used to solve one of these problems at a time.
- Here we present an approach that tackles both problems *simultaneously*, with the express purpose of *maximising classification performance*.

Scenario

- We have a *pool* of items with which to train a *classifier*.
- The task of the active learner is to, given the current classifier, select the best item to be labelled by the *oracle*.
- After each item has had a label supplied the classifier is updated with the new information (It helps if an incremental learning method is used.).

Assumptions

- *Assumption 1*: That the item with the greatest probability of being misclassified should be selected.
- *Assumption 2*: That the classes have been drawn from a **Dirichlet process**. This is equivalent to assuming the items in the pool come from a **Dirichlet process mixture model**.

- An infinite number of classes to which entities may belong.
- Classifier is Bayesian, but this can be ignored with a *pseudo-prior*.

The Algorithm

Class assignment that the classifier, which cannot consider new classes, gives:

$$\text{cc} = \underset{c \in C}{\text{argmax}}\, P_c(c|\text{data})$$

Class assignment probability, including the possibility of a new class:

$$P_n(c \in C \cup \{\text{new}\}|\text{data}) \propto \begin{cases} \frac{m_c}{\sum_{k \in C} m_k + \alpha} P_c(\text{data}|c) & \text{if } c \in C \\ \frac{\alpha}{\sum_{k \in C} m_k + \alpha} P(\text{data}) & \text{if } c = \text{new} \end{cases}$$

Probability of misclassification:

$$P(\text{wrong}|\text{data}) = 1 - P_n(\text{cc}|\text{data})$$

Infinite Dirichlet Distribution

$$x \sim M(X), \quad X \sim D(\alpha, H), \quad x \in H$$

Finite Case	Infinite Case
D = Dirichlet distribution.	D = Dirichlet process.
X = Finite length vector, sum of all entries is 1.	X = Infinite length vector, sum of all entries is 1.
M = Multinomial distribution.	M = Infinite multinomial.
x = Individual atom.	x = Individual atom.
H = Set of arbitrary atoms, of size n.	H = Base measure, a from which atoms can be drawn. Often a standard distribution
$\alpha \in \mathbb{R}^n$ = Parameter for the Dirichlet distribution.	$\alpha \in \mathbb{R}$ = The concentration parameter.

Stick Breaking Construction

Remaining Stick→ 

$l_0 = 1$

Base Measure→ 

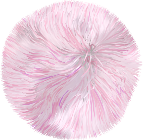
Stick Breaking Construction

Remaining Stick→ 

$l_1 = v_1$

$v_1 \sim \text{beta}(1, \alpha)$

$\beta_1 = 1 - v_1$



Base Measure→ 

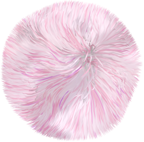
Stick Breaking Construction

Remaining Stick→ 

$l_2 = v_1 v_2$

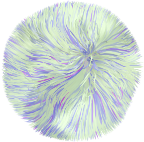
$v_1 \sim \text{beta}(1, \alpha)$

$\beta_1 = 1 - v_1$



$v_2 \sim \text{beta}(1, \alpha)$

$\beta_2 = v_1(1 - v_2)$



Base Measure→ 

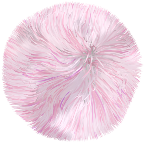
Stick Breaking Construction

Remaining Stick→ 

$l_3 = v_1 v_2 v_3$

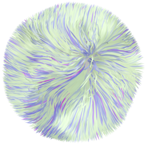
$v_1 \sim \text{beta}(1, \alpha)$

$\beta_1 = 1 - v_1$



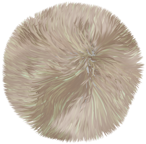
$v_2 \sim \text{beta}(1, \alpha)$

$\beta_2 = v_1(1 - v_2)$



$v_3 \sim \text{beta}(1, \alpha)$

$\beta_3 = v_1 v_2(1 - v_3)$



Base Measure→ 

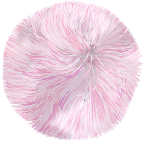
Stick Breaking Construction

Remaining Stick→ 

$l_n = \prod_{i=1}^{n} v_i$

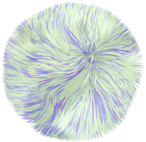
$v_1 \sim \text{beta}(1, \alpha)$

$\beta_1 = 1 - v_1$



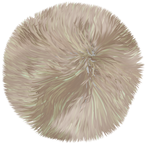
$v_2 \sim \text{beta}(1, \alpha)$

$\beta_2 = v_1(1 - v_2)$



$v_3 \sim \text{beta}(1, \alpha)$

$\beta_3 = v_1 v_2(1 - v_3)$



. . .

$v_n \sim \text{beta}(1, \alpha)$

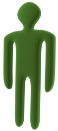
$\beta_n = \prod_{i=1}^{i-1} v_i(1 - v_n)$

. . .

Base Measure→ 

Chinese Restaurant Process

$\frac{\alpha}{\alpha}$



- Is $P(x|\alpha, H) = \int x \sim M(X), X \sim D(\alpha, H)dX$
- Customer enters the restaurant, has to choose where to sit.

Chinese Restaurant Process

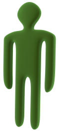
- An infinite number of tables are actually available, but as empty tables are equivalent the choice is meaningless.
- When sitting at an empty table a draw from the base measure (menu) is made - all customers at that table are then associated with that draw.

Chinese Restaurant Process

$\frac{\alpha}{\alpha+1}$

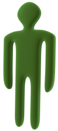
$\frac{1}{\alpha+1}$

- Tables are weighted by the number of customers sitting at them.



Chinese Restaurant Process

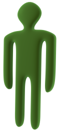
$\frac{\alpha}{\alpha+2}$ $\frac{1}{\alpha+2}$ $\frac{1}{\alpha+2}$



Chinese Restaurant Process

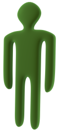
$\frac{\alpha}{\alpha+3}$ $\frac{2}{\alpha+3}$ $\frac{1}{\alpha+3}$

- Two people have sat at one of the tables - the same value has been drawn from the distribution twice.
- Consequentially, a continuous base distribution has been converted into a discrete distribution.



Chinese Restaurant Process

$\frac{\alpha}{\alpha+4}$ $\frac{3}{\alpha+4}$ $\frac{1}{\alpha+4}$



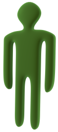
Chinese Restaurant Process

$\frac{\alpha}{\alpha+5}$

$\frac{3}{\alpha+5}$

$\frac{2}{\alpha+5}$

- The *rich get richer* - a table with lots of customers will attract more customers.

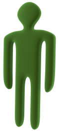


Chinese Restaurant Process

$$\frac{\alpha}{\alpha+6} \qquad \frac{1}{\alpha+6} \qquad \frac{3}{\alpha+6} \qquad \frac{2}{\alpha+6}$$

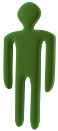
Chinese Restaurant Process

$$\frac{\alpha}{\alpha+7} \qquad \frac{1}{\alpha+7} \qquad \frac{4}{\alpha+7} \qquad \frac{2}{\alpha+7}$$



Chinese Restaurant Process

$$\frac{\alpha}{\alpha+8} \qquad \frac{2}{\alpha+8} \qquad \frac{4}{\alpha+8} \qquad \frac{2}{\alpha+8}$$



Chinese Restaurant Process

$$\frac{\alpha}{\alpha+\sum_{i=1}^{n} m_i} \qquad \frac{m_3}{\alpha+\sum_{i=1}^{n} m_i} \qquad \frac{m_2}{\alpha+\sum_{i=1}^{n} m_i} \qquad \frac{m_1}{\alpha+\sum_{i=1}^{n} m_i}$$

- m_i - The number of customers at table i.
- Whilst only four tables are shown the process goes on forever, leading to an infinite number of occupied tables.

The Algorithm, again

Class assignment probability, including the possibility of a new class:

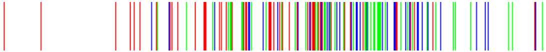
$$P_n(c \in C \cup \{\text{new}\}|\text{data}) \propto \begin{cases} \frac{m_c}{\sum_{k \in C} m_k + \alpha} P_c(\text{data}|c) & \text{if } c \in C \\ \frac{\alpha}{\sum_{k \in C} m_k + \alpha} P(\text{data}) & \text{if } c = \text{new} \end{cases}$$

Concentration parameter (α) needs to be estimated - use the Gibbs sampling method from Escobar & West '95.

Final entity selection is done probabilistically, using $P(\text{wrong})$ as a weighting.

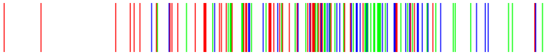
Demonstration

- Use Fisher *iris (orchid) classification problem* from 1936, reduced to 1D via PCA.

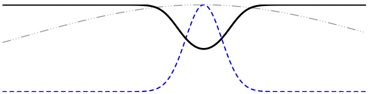


Demonstration

- Use Fisher *iris (orchid) classification problem* from 1936, reduced to 1D via PCA.

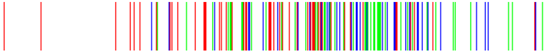


1 query:

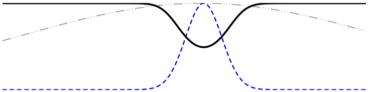


Demonstration

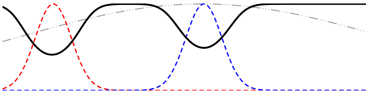
- Use Fisher *iris (orchid) classification problem* from 1936, reduced to 1D via PCA.



1 query:

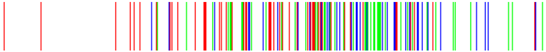


2 queries:

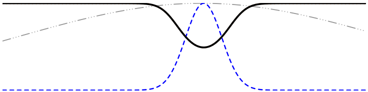


Demonstration

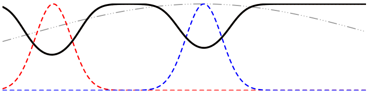
- Use Fisher *iris (orchid) classification problem* from 1936, reduced to 1D via PCA.



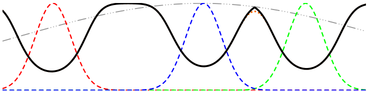
1 query:



2 queries:

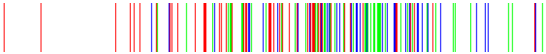


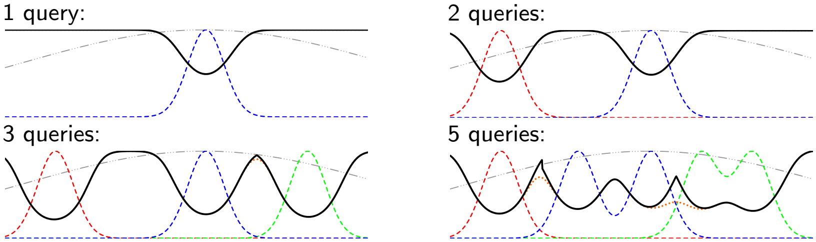
3 queries:



Demonstration

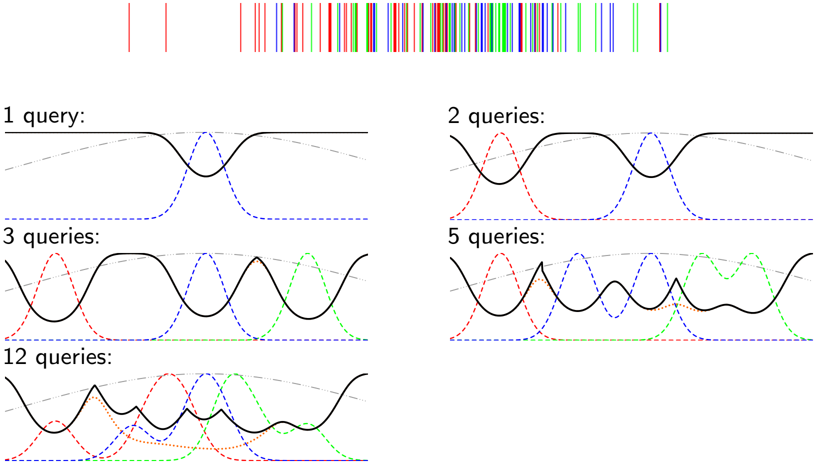
- Use Fisher *iris (orchid) classification problem* from 1936, reduced to 1D via PCA.



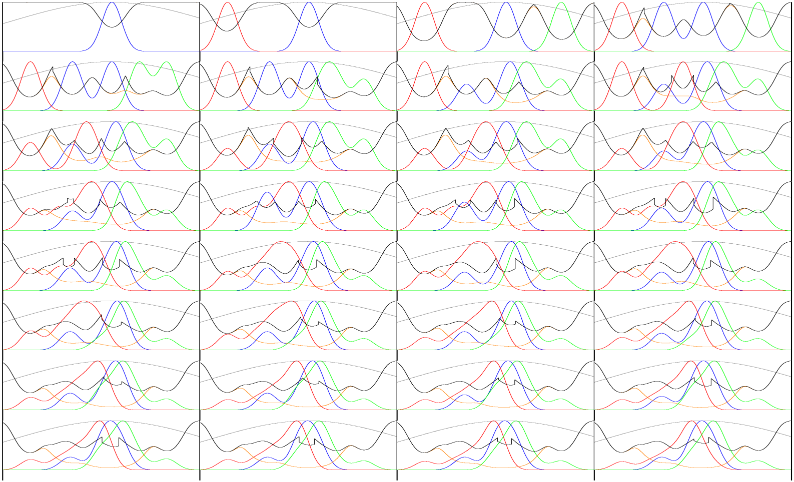


Demonstration

- Use Fisher *iris (orchid) classification problem* from 1936, reduced to 1D via PCA.



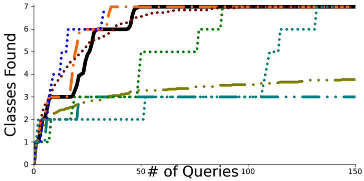
Demonstration (Bonus slide)

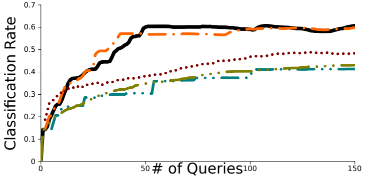


(First 32 queries, in reading order.)

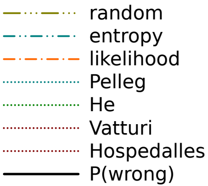
Shuttle

- Standard dataset from the UCI repository - included to compare with other algorithms.
- Seven classes; 78% of exemplars are in the largest class, 0.01% in the smallest.



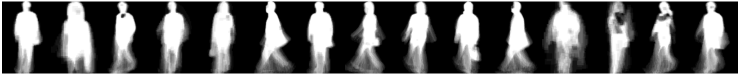


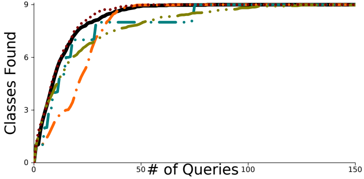
	shuttle	
	discovery	classification
random	486.2	53.5
entropy	423.5	51.8
likelihood	950.5	79.4
Pelleg	534.0	
He	768.5	
Vatturi	970.5	
Hospedales	933.2	61.8
P(wrong)	923.4	79.8

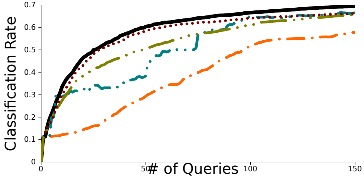


Gait

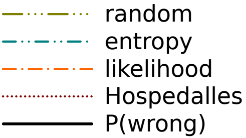
- Gait problem - recognising one of nine camera angles from a gait energy image. Geometric progression for sample sizes.





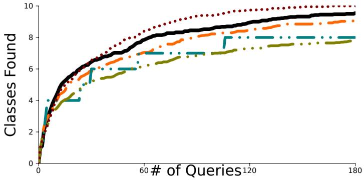


	gait	
	discovery	classification
random	1170.5	78.9
entropy	1183.8	75.3
likelihood	1171.7	56.5
Hospedalles	1253.1	84.8
P(wrong)	1241.9	88.4

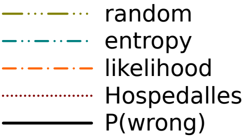


Digits

- Digits problem: Recognising the ten handwritten digits.

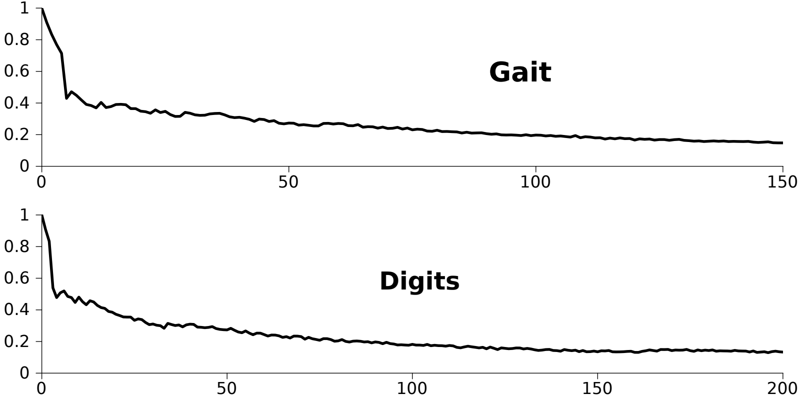


	digits	
	discovery	classification
random	915.2	54.6
entropy	974.0	57.1
likelihood	1060.2	61.9
Hospedales	1207.4	69.5
P(wrong)	1133.6	69.7



Interest in Finding New Classes

- Plots of the interest in finding a new class versus the number of queries.
- Glitch in graph due to concentration (α) estimation method requiring at least two classes.



Conclusions

- Simple to implement.
- Reasonable results.
- Minimal, if any, effort required for parameter tuning.
- Basic concept with many possible specialisations/improvements.

- It assumes a logarithmic relationship between # of classes and # of exemplars.
- Arguably better, if more complex, selection methods exist than the probability of misclassification.

The End

Questions?