COMP30027 Machine Learning
Semi-supervised Learning

Semester 1, 2019

Jeremy Nicholson & Tim Baldwin & Karin Verspoor

Lecture Outline

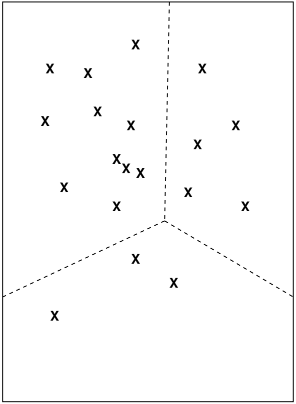
1. Clustering
2. Cluster Evaluation
3. Semi-Supervised Learning
4. Active Learning
5. Summary

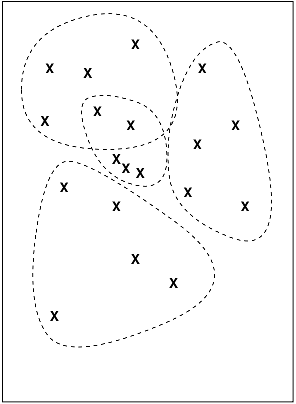
Clustering: Basics

- Clustering = (truly) unsupervised learning; no explicit or implicit definition of class
- Basic contrasts:
  - Exclusive vs. overlapping clustering
  - Deterministic vs. probabilistic clustering
  - Hierarchical vs. partitioning clustering
  - Incremental vs. batch clustering

Source(s): Tan et al. [2006, pp487–495], Jain et al. [1999]

Exclusive vs. Overlapping Clustering



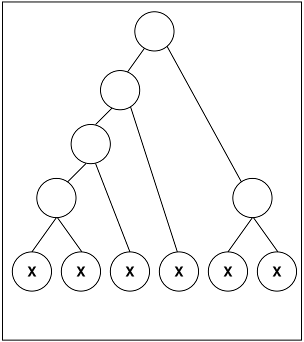


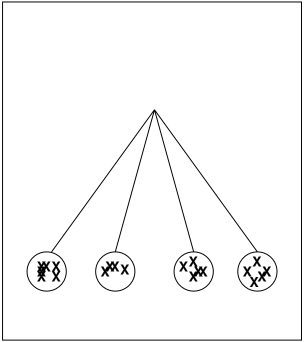
Deterministic vs. Probabilistic Clustering

Instance	**Cluster**
1	2
2	3
3	2
4	1
⋮	⋮

Instance	***Cluster*** 1	2	3	4
1	0.01	0.87	0.12	0.00
2	0.05	0.25	0.67	0.03
3	0.00	0.98	0.02	0.00
4	0.45	0.39	0.08	0.08
⋮		⋮		

Hierarchical vs. Partitioning Clustering





k-means Clustering Refresher

- Given *k*, the *k*-means algorithm is implemented in four steps ("Lloyd's algorithm"):
  (1) Select *k* points at random to act as seed clusters
  (2) Compute seed points as the centroids of the clusters of the current partition (the **centroid** is the centre, i.e., mean point, of the cluster)
  (3) Assign each instance to the cluster with the nearest centroid
  (4) Go back to 2, stop when no reassignments
- Exclusive, deterministic, partitioning, batch clustering method

Source(s): Tan et al. [2006, pp496–515], Jain et al. [1999]

"Soft" k-means Clustering I

- Is it possible to have a probabilistic ("soft") version of k-means, where each instance is probabilistically assigned to each of the k clusters? ... why, yes, using a softmax function:

  (1) Set $t = 0$; randomly initialise the centroids $\mu_1^0, \mu_2^0, \ldots, \mu_k^0$

  (2) Soft-assign each instance x_j to a cluster based on:

$$z_{ij} = \frac{\exp\left[-\beta \|x_j - \mu_i^t\|\right]}{\sum_l \exp\left[-\beta \|x_j - \mu_l^t\|\right]}$$

  $\beta > 0$, and is the "stiffness parameter"

"Soft" k-means Clustering II

(3) Update each of the centroids:

$$\mu_i^{t+1} = \frac{\sum_j z_{ij} x_j}{\sum_j z_{ij}}$$

(4) Set $t = t + 1$; go back to 2 until centroids stabilise

- Overlapping, probabilistic, partitioning, batch clustering method

Clustering via Finite Mixtures

- **A finite mixture** is a mixed distribution with k component distributions
- We use finite mixtures to model the distribution of attribute–value pairs in each cluster

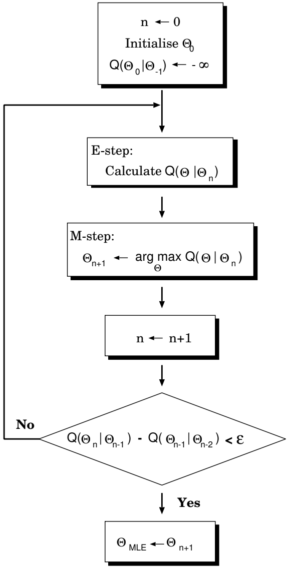
Attribute		***Cluster*** 1	2
outlook	sunny	0.4	0.3
	overcast	0.3	0.3
	rainy	0.3	0.4
temperature	hot	0.4	0.2
	mild	0.5	0.3
	cool	0.1	0.5
⋮	⋮	⋮	

The EM "Algorithm" I

- **EM (Expectation Maximisation) algorithm** = quasi-Newton parameter estimation method with guaranteed "positive" hill-climbing characteristics relative to the gradient of log-likelihood
- Used to estimate (hidden) parameter values or cluster membership
- Basic idea: generalisation of (soft) k-means:
  - based on the current estimate of the parameters Θ_n, calculate the expected log-likelihood (= **E(xpectation) step**)
  - compute the new parameter distribution Θ_{n+1} from Θ_n, that maximises the log-likelihood (= **M(aximisation) step**)

The EM “Algorithm” II

- Not so much an algorithm as a family of algorithms
- Example estimation tasks:
    - estimate the values of missing values based on features with known values
    - estimate the component distributions of two loaded dice from a sample set of their sum over N rolls


n ← 0
Initialise Θ_0
$Q(\Theta_0|\Theta_{-1}) \leftarrow -\infty$
E-step:
Calculate $Q(\Theta|\Theta_n)$
M-step:
$\Theta_{n+1} \leftarrow \arg\max_{\Theta} Q(\Theta|\Theta_n)$
n ← n+1
$Q(\Theta_n|\Theta_{n-1}) - Q(\Theta_{n-1}|\Theta_{n-2}) < \varepsilon$
No
Yes
$\Theta_{MLE} \leftarrow \Theta_{n+1}$

Measuring Convergence: Log Likelihood

- The log likelihood for a given finite mixture is:

  $$L = \sum_i \log \sum_j P(C_j)P(X_i|C_j)$$

  where each X_i is an instance, and each C_j is a "class"
- This gives us an estimate of the "goodness" of the cluster model, and is guaranteed to increase on each iteration of the algorithm
- Convergence can be measured by the relative difference in log likelihood from one iteration to the next; once this falls below a certain predefined level ϵ, we can consider the estimate to have converged

EM Algorithm: Reflections

- Advantages:
  - guaranteed "positive" hill climbing behaviour
  - fast to converge
  - results in probabilistic cluster assignment
  - (relatively) simple but powerful method
- Disadvantages:
  - possibility of getting stuck in a local maximum
  - still rely on arbitrary k (but ...)
  - tends to overfit data if "over-trained"

Lecture Outline

1. Clustering
2. Cluster Evaluation
3. Semi-Supervised Learning
4. Active Learning
5. Summary

Evaluating the Cluster Outputs

- We have recognised that the output of hard/soft *k*-means, and the EM algorithm are sensitive to the seed centroids and initial class assignment, resp.
- Given different methods for coming up with seeds, orderings, etc., how can we compare them to work out if one cluster analysis is "better" than another?
    - evaluation relative to held-out test data?
    - subjective evaluation?
    - similarity between clusters over multiple iterations?

Source(s): Tan et al. [2006, pp532–555], Jain et al. [1999]

Applications of Clustering Evaluation

- While recognising the inherent difficulties in evaluating a given cluster analysis, clustering evaluation has applications in:
  - comparing competing analyses from a given algorithm
  - determining the optimal number of clusters for a given dataset
  - evaluating how well the analysis fits the data
  - comparing clustering algorithms
  - hyperparameter tuning of a given clustering algorithm (no. clusters, size of clusters, ...)

Measures of Cluster Validity

- Clustering evaluation measures come in two basic types:
  - **Unsupervised**: how cohesive are individual clusters/how separate is one cluster from other clusters?
  - **Supervised**: how well do cluster labels match externally supplied class labels?
- Ideally, we would like to have evaluation measures which are independent of the objective functions built into clustering algorithms

Unsupervised Evaluation I

- A "good" cluster analysis should have one or both of:
  - high **cluster cohesion**, i.e. instances in a given cluster should be closely related to each other

$$\mathit{cohesion}(C_i) = \frac{1}{\sum_{\mathbf{x},\mathbf{y} \in C_i} \mathit{proximity}(\mathbf{x}, \mathbf{y})}$$

  - high **cluster separation**, i.e. instances in different clusters should be distinct from each other

$$\mathit{separation}(C_i, C_j) = \sum_{\mathbf{x} \in C_i, \mathbf{y} \in C_{j \neq i}} \mathit{proximity}(\mathbf{x}, \mathbf{y})$$

Unsupervised Evaluation II

- The implementation details will often depending on whether our clustering method is prototype or graph-based, deterministic or probabilistic, etc., etc.

Cluster Compactness: Squared Errors

- One way of evaluating the quality of clusters (esp. for *k*-means), is via the **sum of squared errors** (SSE):

$$\mathrm{SSE} = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \mathit{proximity}(\mathbf{x}, \mathbf{c_i})^2$$

  where $\mathbf{c_i}$ is the centroid of cluster C_i

- Often Euclidean distance is the proximity measure
- For nominal attributes, Hamming distance:

Squared Errors: Example

SSE = 38

(sunny,mild,high,FALSE)

(sunny,hot,high,FALSE)
(sunny,hot,high,TRUE)
(overcast,hot,high,FALSE)
(rainy,mild,high,FALSE)
(sunny,mild,high,FALSE)
(overcast,mild,high,TRUE)
(rainy,mild,high,TRUE)

$SSE_1 = 18$

(overcast,cool,normal,TRUE)

(rainy,cool,normal,TRUE)
(overcast,cool,normal,TRUE)
(sunny,cool,normal,FALSE)
(rainy,mild,normal,FALSE)
(sunny,mild,normal,TRUE)
(overcast,hot,normal,FALSE)
(rainy,cool,normal,FALSE)

$SSE_2 = 20$

Other Measures of Cohesion and Separation

- Graph-based cohesion ($\mathcal{I}_1$):

$$\sum_{i=1}^{k} \frac{|C_i|\,(|C_i|-1)}{\sum_{\mathbf{x},\mathbf{y}\in C_i} proximity(\mathbf{x},\mathbf{y})}$$

- Prototype-based separation ($\mathcal{E}_1$):

$$\sum_{i=1}^{k} |C_i| proximity(\mathbf{c},\mathbf{c_i})$$

- Graph-based separation and cohesion ($\mathcal{G}_1$):

$$\sum_{i=1}^{k} \left[\frac{1}{\sum_{\mathbf{x},\mathbf{y}\in C_i} proximity(\mathbf{x},\mathbf{y})} \sum_{j=1, j\neq i}^{k} \sum_{\mathbf{x}\in C_i, \mathbf{y}\in C_j} proximity(\mathbf{x},\mathbf{y}) \right]$$

Supervised Evaluation

- Supervised evaluation of cluster "validity" measures the degree to which predicted class labels match the actual class labels, e.g. based on the distribution of actual class labels within each cluster:

$$purity = \sum_{i=1}^{K} \frac{|C_i|}{N} \max_j P_i(j)$$

$$entropy = \sum_{i=1}^{K} \frac{|C_i|}{N} H(x_i)$$

where x_i is the distribution of actual class labels in cluster i

Example Calculation I

- Calculate the entropy and purity of the following clustering output:

Cluster	`Play = yes`	`Play = no`	Entropy	Purity
1	4	0	0	1
2	4	4	1	0.5
Total	8	4	0.67	0.67

Example Calculation II

- Calculate the entropy and purity of the following clustering output:

Cluster	`Play = yes`	`Play = no`	Entropy	Purity
1	2	0	0	1
2	6	4	0.97	0.6
Total	8	4	0.81	0.67

Example Calculation III

- Calculate the entropy and purity of the following clustering output:

Cluster	`Play = yes`	`Play = no`	Entropy	Purity
1	0	0	—	—
2	8	4	0.92	0.67
Total	8	4	0.92	0.67

Lecture Outline

Taking Stock I

- Given a set of labelled training data, is it ever preferable to use an unsupervised ML method, rather than a supervised method?
- "...knowing something is better than knowing nothing." (citation needed)
- Generally, any supervised method will get better Accuracy than any unsupervised method (see Project 1, for example) ... given enough data

Taking Stock II

- To date, we have talked a lot about supervised learning — where we have assumed (fully) labelled training data — and a little about unsupervised learning — where we have (fully) unlabelled training data
- What if we had a small amount of labelled training data, and lots of unlabelled training data?
- What if we had a large amount of data, but only a limited budget to label training data?

Semi-Supervised Learning

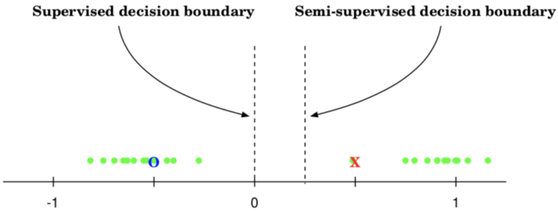
- Semi-supervised learning = learning from both labelled and unlabelled data
- **Semi-supervised classification**:
    - training data = l labelled instances $\{(\boldsymbol{x_i}, y_i)\}$
    - and u unlabelled instances $\{\boldsymbol{x_j}\}$; often $u \gg l$
    - Goal: learn a better classifier f from $l \cup u$ than is possible from l alone

Source(s): Zhu [2005], Zhu [2009]

Why Semi-Supervised Learning?

- Data is (often) cheap and abundant; labelling tends to be expensive
  - example: Switchboard corpus – 400 hours of annotation time per hour of speech data
- In the clustering case, even if we don't know what the class set is, we may have some domain knowledge indicating inter-instance compatibility

Example of Unlabelled Data Impacting on Learning



- Decision boundary shifted by unlabelled data (based on assumption that each class is a coherent group)

Source(s): Zhu [2005], Zhu [2009]

Self Training

- Perhaps the simplest example of semi-supervised learning is **self training**:

  1: Initialise: $L = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{l}$, $U = \{\boldsymbol{x_j}\}_{j=l+1}^{l+u}$
  2: Repeat:
  3: Train f_i from L using supervised learning
  4: Apply f_i to U using supervised learning (predict)
  5: Identify a subset U' of U where $f_i(\boldsymbol{x_j})$ is "confident"
  6: $U \leftarrow U \setminus U'$
  7: $L \leftarrow L \cup U'$ s.t. $U' = \{(\boldsymbol{x_j}, f_i(\boldsymbol{x_j}))\}$
  8: Until L is unchanged from one iteration to the next

- Also known as "bootstrapping"

Source(s): Zhu [2005], Zhu [2009]

Self Training Example

- Propagating 1-nearest neighbour:

1: Initialise: $L = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{l}$, $U = \{\boldsymbol{x_j}\}_{j=l+1}^{l+u}$

2: Repeat:

3: Select $\boldsymbol{x}, \boldsymbol{x'} = \arg\min_{\boldsymbol{x} \in U, \boldsymbol{x'} \in L} \min d(\boldsymbol{x}, \boldsymbol{x'})$

4: $U \leftarrow U \setminus \{\boldsymbol{x}\}$

5: $L \leftarrow L \cup \{(\boldsymbol{x}, y\prime)\}$

6: Until $U = \phi$

Source(s): Zhu [2005], Zhu [2009]

Lecture Outline

1. Clustering
2. Cluster Evaluation
3. Semi-Supervised Learning
4. Active Learning
5. Summary

Active Learning

- **Active learning** builds off the hypothesis that a classifier can achieve higher accuracy with fewer training instances if it is allowed to have some say in the selection of the training instances
- The underlying assumption is that labelling is a finite resource, which should be expended in a way which optimises machine learning effectiveness
- Active learners pose *queries* (unlabelled instances) for labelling by an *oracle* (e.g. a human annotator)

Source(s): Settles [2010]

Active Learning: Query Strategies I

- One simple query strategy is to query those instances the classifier is least confident of the classification for:

$$x^* = \arg\max_{\boldsymbol{x}}(1 - P_\theta(\hat{y}|\boldsymbol{x}))$$

where $\hat{y} = \arg\max_y P_\theta(y|\boldsymbol{x})$

Active Learning: Query Strategies II

- Alternatively, it may be appropriate to perform "margin sampling":

$$x_M^* = \arg\min_{\boldsymbol{x}} P_\theta(\hat{y}_1|\boldsymbol{x}) - P_\theta(\hat{y}_2|\boldsymbol{x})$$

  where $\hat{y}_1$ and $\hat{y}_2$ are the first and second most-probable label predictions for $\boldsymbol{x}$

- Or better still, to use entropy as an uncertainty measure:

$$x_H^* = \arg\max_{\boldsymbol{x}} - \sum_{y_i} P_\theta(y_i|\boldsymbol{x}) \log_2 P_\theta(y_i|\boldsymbol{x})$$

Active Learning: Query Strategies III

- A more complex strategy involving multiple classifiers is **query-by-committee** (QBC), where a suite of classifiers is trained over a fixed training set L, and the instance where there is the highest disagreement is selected for querying
- QBC assumes that it is possible to generate a suite of heterogeneous base classifiers, much like ...
- Determination of relative disagreement can again occur via entropy, or alternatively via one-vs-rest relative entropy

Active Learning: Practicalities

- Active learning is used increasingly widely, but must be handled with some care:
  - empirically shown to be a robust strategy, but a theoretical justification has proven elusive
  - querying is inherently biased towards a particular class set and learning approach(es), which may limit the general utility of the resulting dataset
  - results to suggest that active learning is more highly reliant on "clean" labelling

Source(s): Settles [2010]

Lecture Outline

1. Clustering
2. Cluster Evaluation
3. Semi-Supervised Learning
4. Active Learning
5. Summary

Clustering Summary

- What basic contrasts are there in different clustering methods?
- What is soft k-means and how does it work?
- What is EM clustering and how does it work?
- How are hard/soft k-means and EM clustering similar and different?
- What basic approaches are there to cluster evaluation?
- What elements are focused on in unsupervised cluster evaluation, and how are these implemented in different evaluation measures?
- What is the basis of supervised cluster evaluation, and how is this proceduralised in purity and entropy?

Semi-Supervised Summary

- What is semi-supervised learning?
- What is self-training, and how does it operate?
- What is active learning?
- What are the main sampling strategies in active learning?
- Outline a selection of query strategies in active learning.

References I

A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

Burr Settles. Active learning literature survey. Technical report, Department of Computer Sciences, University of Wisconsin, Madison, 2010. `http://pages.cs.wisc.edu/~bsettles/pub/settles.activelearning.pdf`.

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison, 2005. `http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf`.

Xiaojin Zhu. Tutorial on semi-supervised learning. `http://pages.cs.wisc.edu/~jerryzhu/pub/sslchicago09.pdf`, 2009.