Machine Learning

ITAM

Outline

- Unsupervised learning
- Clustering: Grouping and data segmentation
- Similarity measures
  - Transformation of ordinal, nominal and categorical variables
- Techniques
  - Partition methods
    - EM: k-medias
  - Density methods
  - Hierarchical methods

Objective

- Group data into categories or clusters such that instances that are more closely related belong to the same group
- Sometimes we also want a hierarchy that orders data accroding to their relatedness
  - E.g. A taxonomy

Unsupervised learning

- Clustering algorithms are unsupervised learning algorithms
  - There are no labeled examples from which the model is created
- Uses:
  - When labeling is expensive
  - When label change with time
  - When we want to find unsuspected relationships in the data that might be useful for classification
  - Whe we want to better understand the data
  - …..

Clustering techniques

- The first step is to define the appropriate criterion for similarity between data instances
  - Normaly this depends on the application
    - How similar are México and Uganda?
  - How similar is a 4 to a 5; a 4 to a 4.1?

Instance similarity
Similarity matrix

- On occasion we are provided witha similarity matrix. Silimarity between pairs

	México	Uganda	Holanda
México	1	0.4	0.3
Uganda	0.4	1	0.2
Holanda	0.3	0.2	1

- Usually they are simetric
- Some algorithms need a *difference* matrix.

Instance similarity
Attribute similarity

- If we don’t have such a matrix
  - Define a distance metric between the values in each attribute
  - Define a way to combine the similarities between the attributes
  - Eg. <México, 25> and <Uganda, 30>
    - The distance (similitud) between México and Uganda is 0.4. We can, for example, define the distance between 25 and 30 as |25-30|=5
    - We need a way to generete a single number from 0.4 and 5 (we will see ahead)

Instance similarity
Attribute similarity

- Quantitative variables
  - The most common is to define the similitud as:
    - $(X_{i,k} - X_{j,k})^2$
    - The squared difference of attribute k of instances i and j
    - There are, of course, others
      - Absolute value….

Instance similarity
Attribute similarity

- Ordinal variables
  - E.g.
    - Taste (horrible, so-so, delicious)
    - Grades (A, B, C, F)
  - They can be representes as an ordered list of succesive numbers
    - (i-1)/(M-1)
    - Where i is the i-th value and M is the total number of different values for the given attribute
    - horrible=0/2, so-so=1/2, delicious=2/2
  - Once tranformed they are treated as quantitative variables

Instance similarity
Attribute similarity

- Categorical variables
  - Distance matrix
  - Or substitute the variable for k indicator bits, one for each of its possible values. Each instance will then have k bits with one and only one set to 1 and the rest the value of 0. This is called one-hot encoding

Instance similarity
Distance between instances

- A common method is to use the Euclidean distance:
  - $\sqrt{(\sum(x_{i,k}-x_{j,k})^2)}$
  - The sum of the squared differences between attributes
  - We can omit the sqrt as it doen't alter the relative distances
- There are other option which might be more appropriate
  - Edit distance, Manhattan distance, Mahalanobis, etc.

Instance similarity
Distance between instances

- One danger is that attributes of different scales will affect the distance disproportionately
  - Eg. <distance_to_work, age>
    - Dato 1= <2500mts, 35>
    - Dato 2=<2400mts, 15>
    - Dato 3=<2300mts, 34>
    - The distance between d1y d2 is 10400 while between d1 and d3 is 40001.
    - It depends on the applications but at first glance its seems that d1 and d3 are more similar than d1 and d2
- Solution?

Instance similarity
Distance between instances

- Additionally we can add a weight w_k a to each attribute k the establish the importance of each attribute in determining similarity
  - $\sum w_k(x_{i,k}-x_{j,k})^2$
  - $\sum w_k=1$

Instance similarity
Distance between instances

- A good distance metric can be more important that the particular clustering algorithm used to segment the data

Clustering algorithms

- The objective of these algorithms is to divide data into groups such that the similitud between instance pairs within a group is greater that between intances in different groups (or clusters)

Clustering algorithms

- Two important categories :
  - Partition methods
    - Given n instances, classify into k disjoint categories.
    - Distance based: E.g. K-medias, A-priori
    - Density based: DBscan
  - Hierarchical methods
    - They create a hierarchy in the data. They create a tree of clusters, e.g., Cobweb

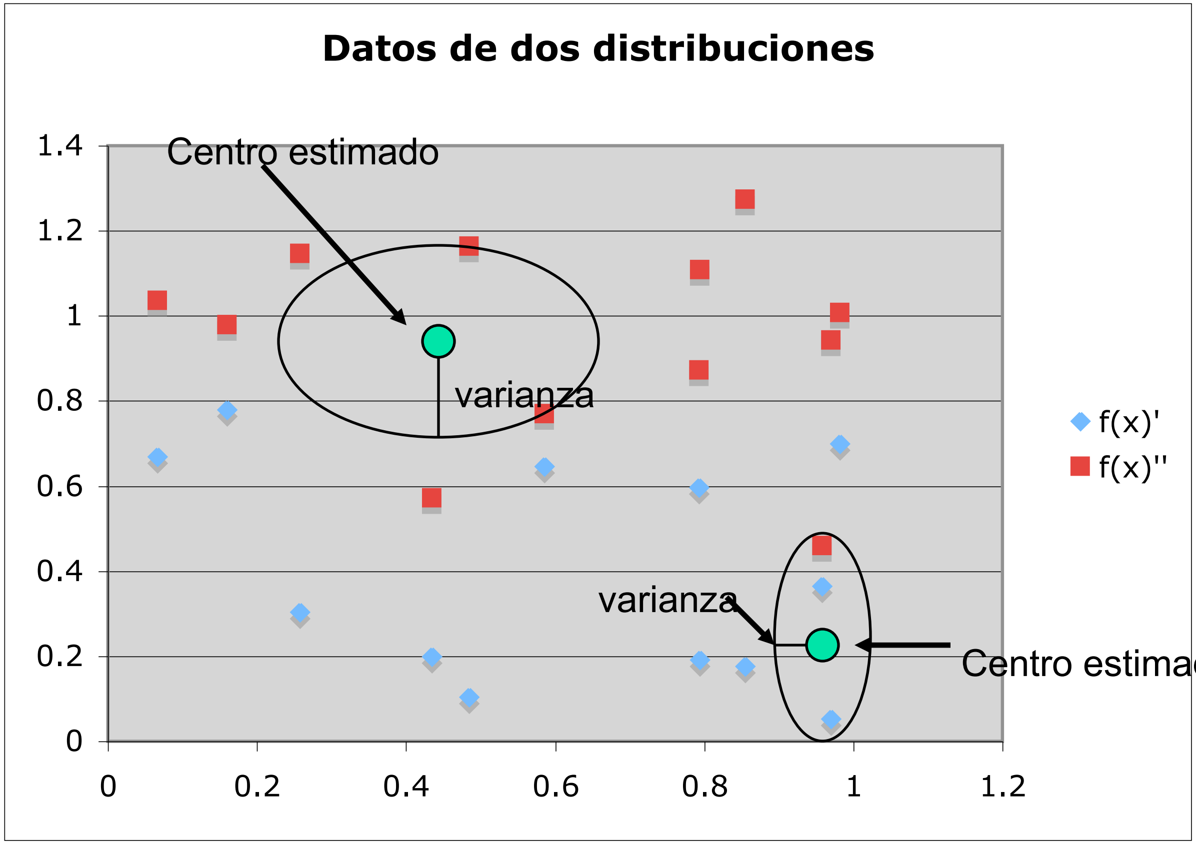
Clustering algorithms

- We will see:
  - Expectation Maximization
    - k-means
  - Hierarchical
  - DBSCAN
- A general question:
  - Can you learn anything from data without labels?
    - Depends on your assumptions
    - The goodness of your results depend on how your assumptions and reality match up

Clustering algorithms
Expectation Maximization (EM)

- We **asume** we know:que conocemos
  - The number of groups in which data is segmented
  - The distribution of the data
    - The most common assumption is that they are distributed normally
- We don’t know
  - The group each instance belongs to
    - Training data don’t have the value of the objetive function
  - The parameters of the distribution
    - If we assume the normal distribution we don’t know its mean and variance

Clustering algorithms
EM



The task is to estimate the unknown parameters of the underlying distributions (mean, variance,..)

Clustering algorithms k-means

- We are going to look at k-means which belongs to the EM family
- It is a very efective simplification of the more general algorithm

Clustering algorithms k-means

- The simplification consists in that we will only estimate the means of the distributions
  - The mean of the distribution i (D_i) :
    - $\mu_i=(1/w_i)\sum p_{i,j}\mathbf{x}_j$,   where
    - $p_{i,j}$: is the probability that the instance $\mathbf{x}_j$ was generated by distribution D_i
    - w_i: is the sum of all probabilities ($p_{i,j}$) for all instances $\mathbf{x}_j$ that belong to the distribution D_i

Clustering algorithms
k-means

- Additionally, since we dont want to estimate the variance, we can set the $p_{i,j}$ using the squared Euclidean distance from the data to the mean μ_i of the distrubution D_i
  - distance($\mathbf{x}_j$,μ_i)=$||\mathbf{x}_j - \mu_i||^2$

and approximate $p_{i,j}$ as

$$p_{i,j} \approx \begin{cases} 1 \text{ if } \mu_i \text{ is the closest mean to } \mathbf{x}_j, \\ 0 \text{ otherwise} \end{cases}$$

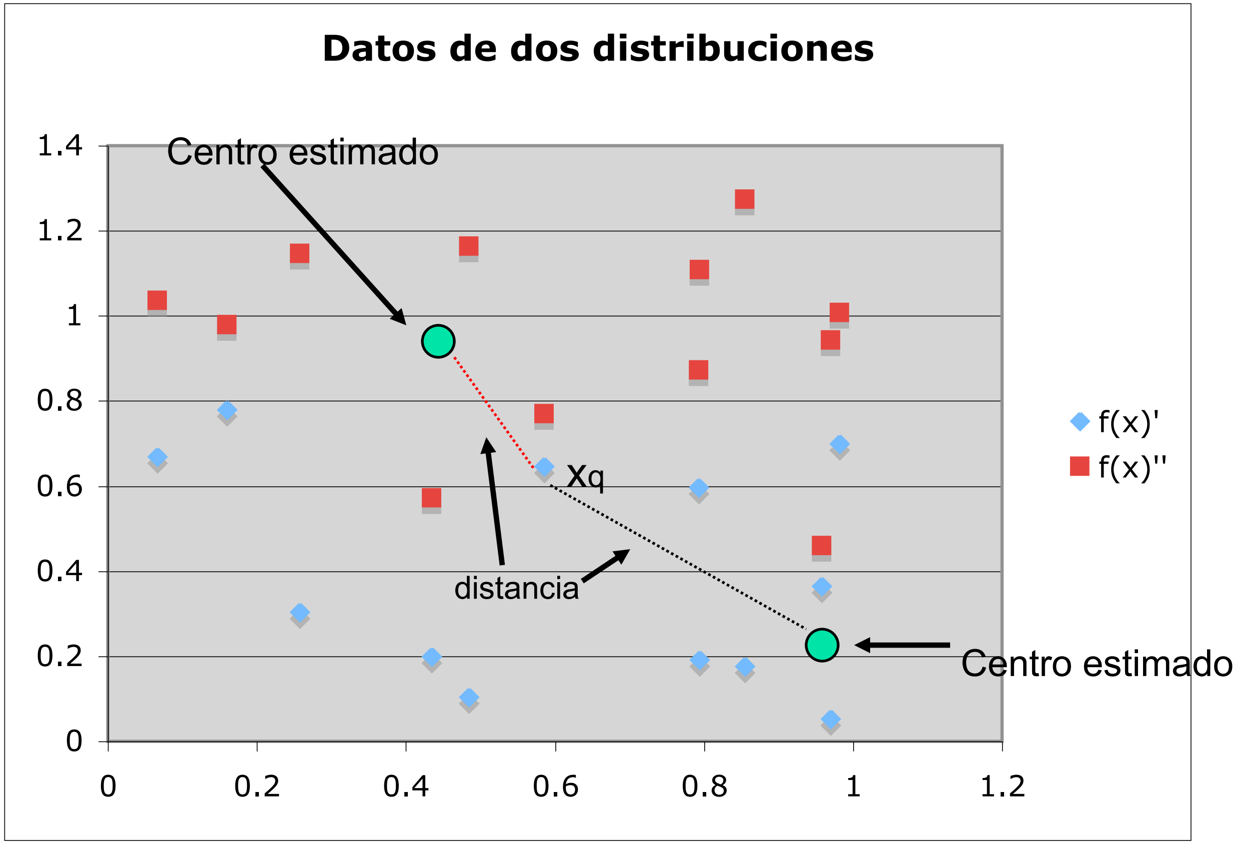
EM
General algorithm

- The algorithm consists of two phases
  - Expectation
    - Estimate the probability that each instance belongs to each of the distributions
  - Maximization
    - Recompute the means

Clustering algorithms k-means

- Given
  - n training examples
  - An integer k
  - Initial values for the means $\mu_1, \mu_2, \ldots, \mu_k$
- Do{
  - Classifly the n instances according to their closest mean μ_i
    - For each $\mathbf{x}_j$, compute the distance to each μ_i
      - Mínimo$_{i=1,k}$ (distancia($\mathbf{x}_j$, μ_i))
    - Classify $\mathbf{x}_j$ as member of its closest mean
  - Recalculate the new μ_i
    - μ_i<---$(1/w_i)\sum p_{i,j}\mathbf{x}_j$
- }while(ther is change in the values of $\mu_1, \mu_2, \ldots, \mu_k$)

Clustering algorithms
k-means



To classify x_q, you compute its distance to all the means.
Its classified as member of the closest mean

Clustering algorithms
Example k-means

DATOS	media 1	media 2	Pij (clase 1)	Pij (clase2)	Clase 1	Clase 2	media 1	media 2
	-0.5	1					0.144943	0.653655
	Distancia						Distancia	
0.67	1.17	0.33	0	1	0	0.67	0.525057	0.016345
0.19122452	0.691225	0.80878	1	0	0.19122	0	0.046282	0.46243
0.7	1.2	0.3	0	1	0	0.7	0.555057	0.046345
0.17606015	0.67606	0.82394	1	0	0.17606	0	0.031117	0.477595
0.103874	0.603874	0.89613	1	0	0.10387	0	0.041069	0.549781
0.646908	1.146908	0.35309	0	1	0	0.64691	0.501965	0.006747
0.19994854	0.699949	0.80005	1	0	0.19995	0	0.055006	0.453706
0.30341512	0.803415	0.69658	0	1	0	0.30342	0.158472	0.35024
0.0536079	0.553608	0.94639	1	0	0.05361	0	0.091335	0.600047
0.59716748	1.097167	0.40283	0	1	0	0.59717	0.452224	0.056488
0.87234622	1.372346	0.12765	0	1	0	0.87235	0.727403	0.218691
0.46032091	0.960321	0.53968	0	1	0	0.46032	0.315378	0.193334
0.97908235	1.479082	0.02092	0	1	0	0.97908	0.834139	0.325427
		Total =	5	8	0.72472	5.22924		Total =

Clustering algorithms
Example k-means

DATOS	media 1	media 2	Pij (clase 1)	Pij (clase2)	Clase 1	Clase 2	media 1	media 2
	0.144943	0.653655					0.171355	0.70369
	Distancia						**Distancia**	
0.67	0.525057	0.016345	0	1	0	0.67	0.498645	0.03369
0.19122	0.046282	0.46243	1	0	0.1912	0	0.019869	0.51246
0.7	0.555057	0.046345	0	1	0	0.7	0.528645	0.00369
0.17606	0.031117	0.477595	1	0	0.1761	0	0.004705	0.52763
0.10387	0.041069	0.549781	1	0	0.1039	0	0.067481	0.59982
0.64691	0.501965	0.006747	0	1	0	0.6469	0.475553	0.05678
0.19995	0.055006	0.453706	1	0	0.1999	0	0.028594	0.50374
0.30342	0.158472	0.35024	1	0	0.3034	0	0.13206	0.40027
0.05361	0.091335	0.600047	1	0	0.0536	0	0.117747	0.65008
0.59717	0.452224	0.056488	0	1	0	0.5972	0.425812	0.10652
0.87235	0.727403	0.218691	0	1	0	0.8723	0.700991	0.16866
0.46032	0.315378	0.193334	0	1	0	0.4603	0.288966	0.24337
0.97908	0.834139	0.325427	0	1	0	0.9791	0.807727	0.27539
		Total =	6	7				

Clustering algorithms Example k-means

DATOS	media 1	media 2	Pij (clase 1)	Pij (clase2)	Clase 1	Clase 2	media 1	media 2
	0.171355	0.70369					0.17136	0.70369
	Distancia						**Distancia**	
0.67	0.498645	0.03369	0	1	0	0.67	0.49864	0.03369
0.1912	0.019869	0.51246	1	0	0.1912	0	0.01987	0.51246
0.7	0.528645	0.00369	0	1	0	0.7	0.52864	0.00369
0.1761	0.004705	0.52763	1	0	0.1761	0	0.00471	0.52763
0.1039	0.067481	0.59982	1	0	0.1039	0	0.06748	0.59982
0.6469	0.475553	0.05678	0	1	0	0.6469	0.47555	0.05678
0.1999	0.028594	0.50374	1	0	0.1999	0	0.02859	0.50374
0.3034	0.13206	0.40027	1	0	0.3034	0	0.13206	0.40027
0.0536	0.117747	0.65008	1	0	0.0536	0	0.11775	0.65008
0.5972	0.425812	0.10652	0	1	0	0.5972	0.42581	0.10652
0.8723	0.700991	0.16866	0	1	0	0.8723	0.70099	0.16866
0.4603	0.288966	0.24337	0	1	0	0.4603	0.28897	0.24337
0.9791	0.807727	0.27539	0	1	0	0.9791	0.80773	0.27539
		Total =	6	7				

Clustering algorithms
k-means

- Some shortcomings
  - You have to pre-establish the number of clusters k
  - Its sensible to the initial values of the means

Other algorithms

- X-means: an extension of k-means that helps set k
- Fuzzy k-means
- K-medioids
- Hierarchical methods
  - Closest neighbors
  - Furthest neighbors
- Density methods
  - DBSCAN

Exercise

- Download data from UCI
  - I suggest Abalone
- Make a classification model using a the classification technique of your choice
- Execute k-means (do you need to separete in train and test before k-means?)
  - Train a model using the same technique as above for each cluster
  - Take care in combining the results for each cluster
- Compare results
  - Did any metric improve by preprocessing the data first with k-means?