Machine Learning

ITAM

Menu

- Instance based learning
  - K-nearest neighbors

Instance based learning

- Well, really all methods use instances to learn
- The difference is that methods in this category **store** a subset of the training data rather than deriving an explicit representation of the objetive function
  - Linear function, decision tree, neural net,…

Instance based learning

- This techniques memorize some examples and postpone generalization to the last moment. At predition time
  - They are known as lazy methods is CS since they leave the heavy computation until the last moment
- One advantage is that they can locally approximate an objective function
  - Useful when the global target can be properly approximated with a set of local approximations

Instance based learning

- Algunas características
  - Son técnicas de aprendizaje supervisado i.e., los ejemplos con los que se entrena tiene asociado un valor de la función de evaluación
  - Estas técnicas funcionan bien tanto para problemas de regresión como de clasificación
  - Desventajas
    - Costo computacional en línea
    - Desempeño degradado si las instancias tienen muchos atributos irrelevantes (más sobre esto después)

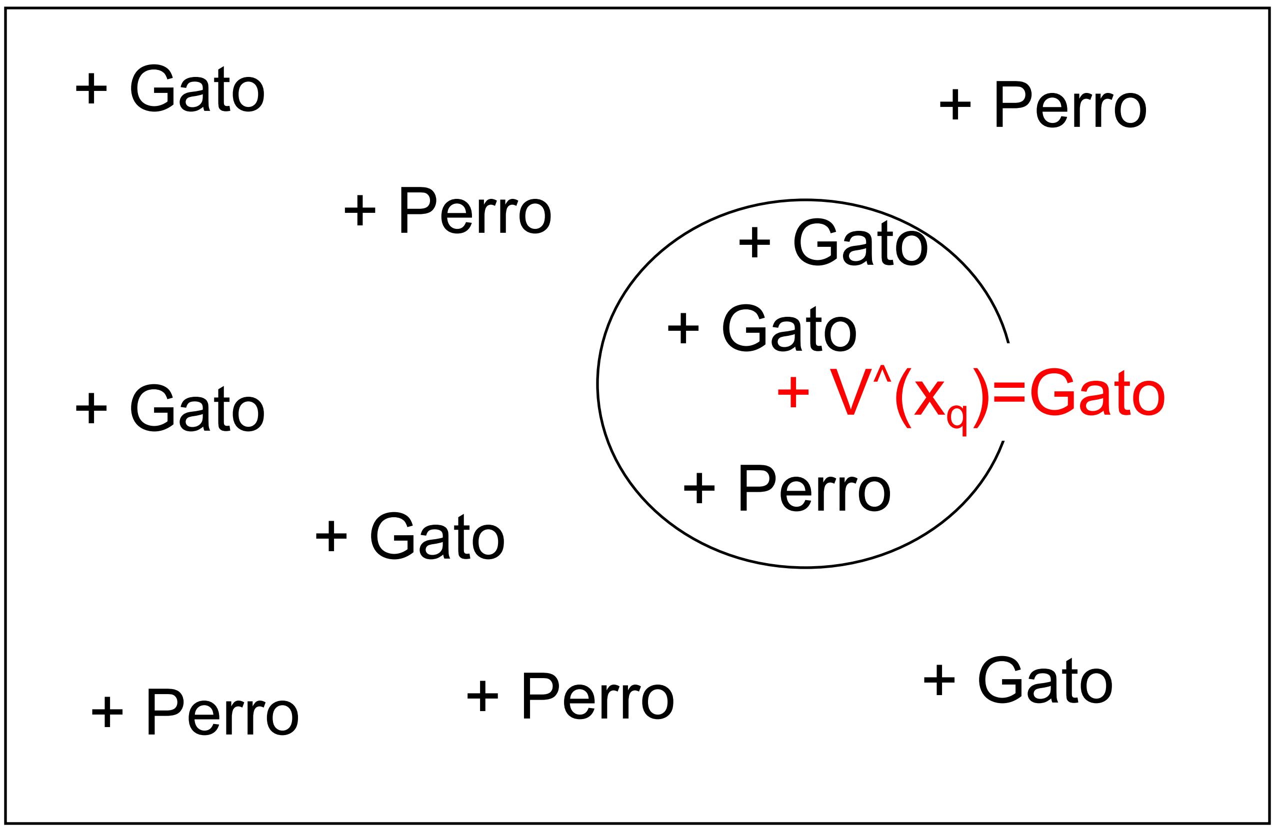
k-Nearest Neighbors

- We will see one algorithm from this family k-nearest neighbors
  - Basic version
  - Distance weighed version

k-Nearest Neighbors Basic version

- Each instance is considered as a point in n-dimensional space $\Re^n$, where n is the number of attributes in each instance
- The objective function can be discrete or continuous (classification or regression)
- In **discrete** case, to classify instance x_q the algorithm selects its k nearest instances (which it saved during training) and assigns the most common class amongst them
  - x_q=<1,0,0,1,1>
  - Suppose the k=3 nearest neighbors are:
    - (<1,1,0,1,1>,Gato)
    - (<1,0,0,0,1>,Perro)
    - (<0,0,0,1,1>,Gato)
  - The classification for x_q <1,0,0,1,1> will be Gato

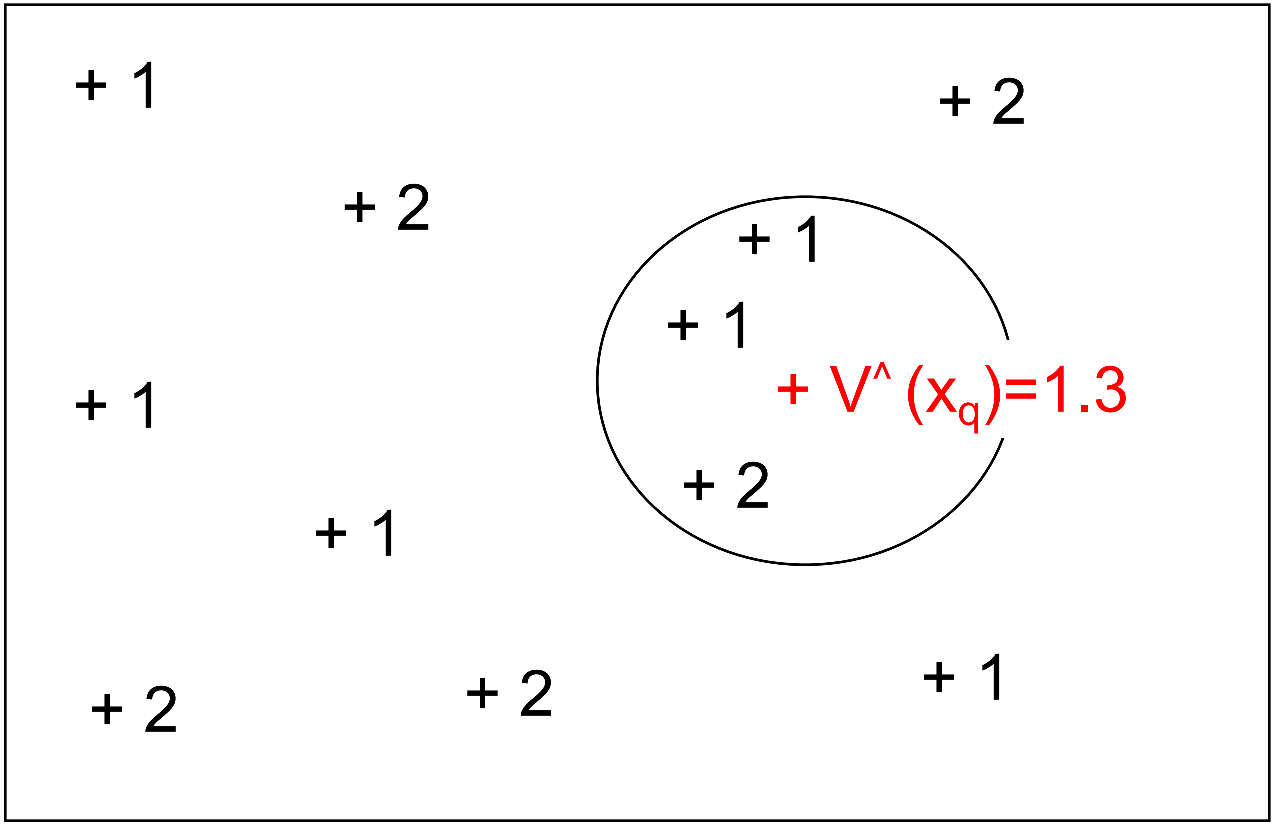
Example
Classify x_q, with k=3



k-Nearest Neighbors Basic version

- In the **continuos** case, the average of the objective function values of x_q´s k nearest instances are computed. For example, given
  - x_q=<1,0,0,1,1>
  - Assume the k=3 nearest neighbors are:
    - (<1,1,0,1,1>,1)
    - (<1,0,0,0,1>,2)
    - (<0,0,0,1,1>,1)
  - The output for <1,0,0,1,1> will be 4/3=1.3

Example
x_q with k=3



k-Nearest Neighbors Basic version

- How close instances are is computed using some distanve metric; for example Eulidean distance
  - The Euclidean distance between x_q and x_j is:

    distance$(x_q,x_j)=Sqrt(\Sigma_{r=1,n}(a_r(x_q)-a_r(x_j))^2)$

    -where $a_r(x_q)$ is the value of attribute r of instance x_q. The sum is over all n attributes

k-Nearest Neighbors Algorithm

- Training :
  - Store every instance (x, f(x))
  - (some implementations might choose representatives)
- Test:
  - Given x_q as input
  - Compute the distance between x_q and every stored instance
  - Classification:
    - Let $x_1 \ldots x_k$ be the k nearest neighbors to x_q
      - $V\hat{}(x_q)$<--- Most common from $\{f(x_1), f(x_2) \ldots f(x_k)\}$
  - Regression:
    - $V\hat{}(x_q)$<--1/k $\sum_{i=1,k} f(xi)$
    - The average value of the k nearest neighbors
    - Some implementations might use de median….

k-Nearest Neighbors
Classification example

- Data

Calif. Mate	Calif. Bio	Estudiante	dist. a Xq
8	8	Bueno	1
9	8	Bueno	1.41421356
7	9	Bueno	2.23606798
9	5	Malo	2.23606798
6	7	Malo	2
7	7	Malo	1

- Classification x_q=(Calif.Mate=8,Calif.Bio=7)

3-mas cercanos		
8	8	Bueno
7	7	Malo
9	8	Bueno

x_q is classified as Bueno

k-Nearest Neighbors Regression

- Datos

Calif. Mate	Calif. Bio	Estudiante	dist. a Xq
8	8	2	1
9	8	2	1.41421356
7	9	2	2.23606798
9	5	1	2.23606798
6	7	1	2
7	7	1	1

- Clasificación x_q=(Calif.Mate=8,Calif.Bio=7)

3-mas cercanos		Estudiante
8	8	2
7	7	1
9	8	2

x_q tests as 5/3=1.6

k-Nearest Neighbors
Weighed version

- One extension of the algorithm is to weigh the contributions of each neighbor by how close they are to x_q
  - The further away from x_q the less its influence
  - One possibility is to use the inverse squared distance to make the influence fall off quickly

k-Nearest Neighbors Weighed version

- We just need to modify the las line of the algorithm
- Classification
  - $f(x_q)$<--- Most common element of $\{w_1 f(x_1), w_2 f(x_2),\ldots, w_k f(x_k)\}$
    where $w_i=1/distancia(x_q,x_i)^2$. Note that to determine the most common element all the w_i associated with the same $f(x_i)$ must be added
- Regression
  - $f(x_q)$<-- $1/r\sum_{i=1,k} w_i f(xi)$
    where $r=\sum_{i=1,k} w_i$
- Note: If the distance between x_q y x_i is cero x_q is assigned the value of $f(x_i)$

k-Nearest Neighbors
Example

- Data

Calif. Mate	Calif. Bio	Estudiante	dist. a Xq
8	8	Bueno	2.01246118
9	8	Bueno	1.80277564
7	9	Bueno	3.38378486
9	5	Malo	1.20415946
6	7	Malo	3.00832179
7	7	Malo	2.06155281

- Clasification x_q=(Calif.Mate=8.9, Calif.Bio=6.2)

3-mas cercanos		dist. A Xq	wi	
8	8	2.012	0.24702679	bueno
9	8	1.802	0.30795725	bueno
9	5	1.204	0.68983786	malo

=	0.54(bueno)
	0.68(malo)

x_q es clasificado como malo

k-Nearest Neighbors

- Some characteristics
  - Robust to noisy data (with a large k)
  - Needs a lot of data
  - Slow to predict
  - Uses all attributes
    - This in contrast to decision trees
    - Problem: the curse of dimensionality

k-Nearest Neighbors
Example of the curse

- Data

Distancia	c3 de lluvia	Temp	Calif. Mate	Calif. Bio	Estudiante	dist. a Xq
500	50	25	8	8	Bueno	600.087494
1000	150	23	9	8	Bueno	148.667414
300	60	21	7	9	Bueno	800.255584
600	50	25	9	5	Malo	500.108988
300	100	23	6	7	Malo	802.249338
1500	40	21	7	7	Malo	400.00625

- Classification x_q=(1100,40,23,8,7)

3-mas cercanos					
1000	150	23	9	8	Bueno
1500	40	21	7	7	Malo
600	50	25	9	5	Malo

x_q is classified as Malo due to the new, irrelevant attributes (used to be classified as Bueno)

There is also a problem of scale. What to do?

k-Nearest Neighbors
Curse of dimensionality

- Some possible solutions
  - Choose relevant attributes
    - “Subset selection”
    - “Principal components”: Find a linear combination of the attributes
    - Information Gain: Similar to what we did for the tree
    - “Correlation Based Feature Selection” : Determine correlation among attributes and between attributes and dependent variable s
    - Factor analysis,….many more

Exercise

- Generate a data set with many circles distributed in a plane. The points inside the circles belong to category “in” and the point outside to category “out”
- Compare with an SVM or Neural net (choose one)
  - Please not all of you SVM