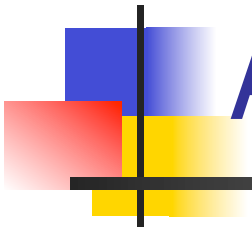
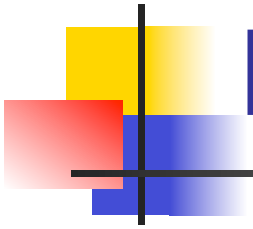


Aprendizaje de Máquina

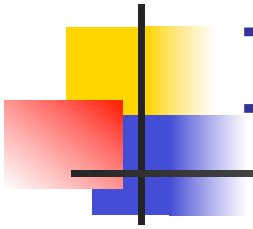
ITAM



Menú

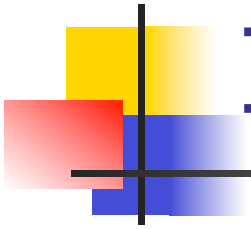
- Aprendizaje en base a instancias
  - K-vecinos cercanos

Aprendizaje en Base a Instancias



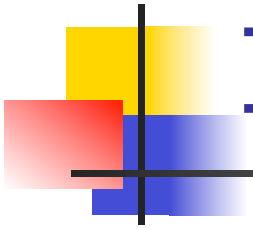
- Todos los métodos, en realidad, han aprendido usando instancias
- La diferencia es que los métodos de ésta categoría **conservan** un subconjunto de los ejemplos de entrenamiento, en lugar de derivar en representación explicita de la función objetivo
  - Función lineal, árbol de decisión, una red neuronal, etc.

Aprendizaje en Base a Instancias



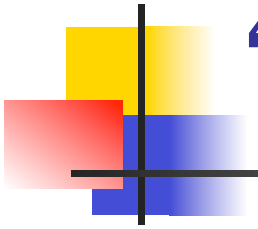
- Estas técnicas memorizan algunos ejemplos y posponen generalizar hasta el momento en que una nueva instancia debe ser clasificada
  - Se conocen como métodos desidiosos ("lazy") pues dejan la realización del cómputo hasta el último momento (cuando hay que clasificar alguna instancia)
- Una ventaja es que pueden estimar una función objetivo (F.O.) especializada localmente para cada instancia a clasificar
  - Esto es bueno cuando la F.O. global es muy compleja pero puede ser expresada como un conjunto de aproximaciones locales

Aprendizaje en Base a Instancias



- 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)

Algoritmo k-Vecinos Cercanos "k-Nearest Neighbors"

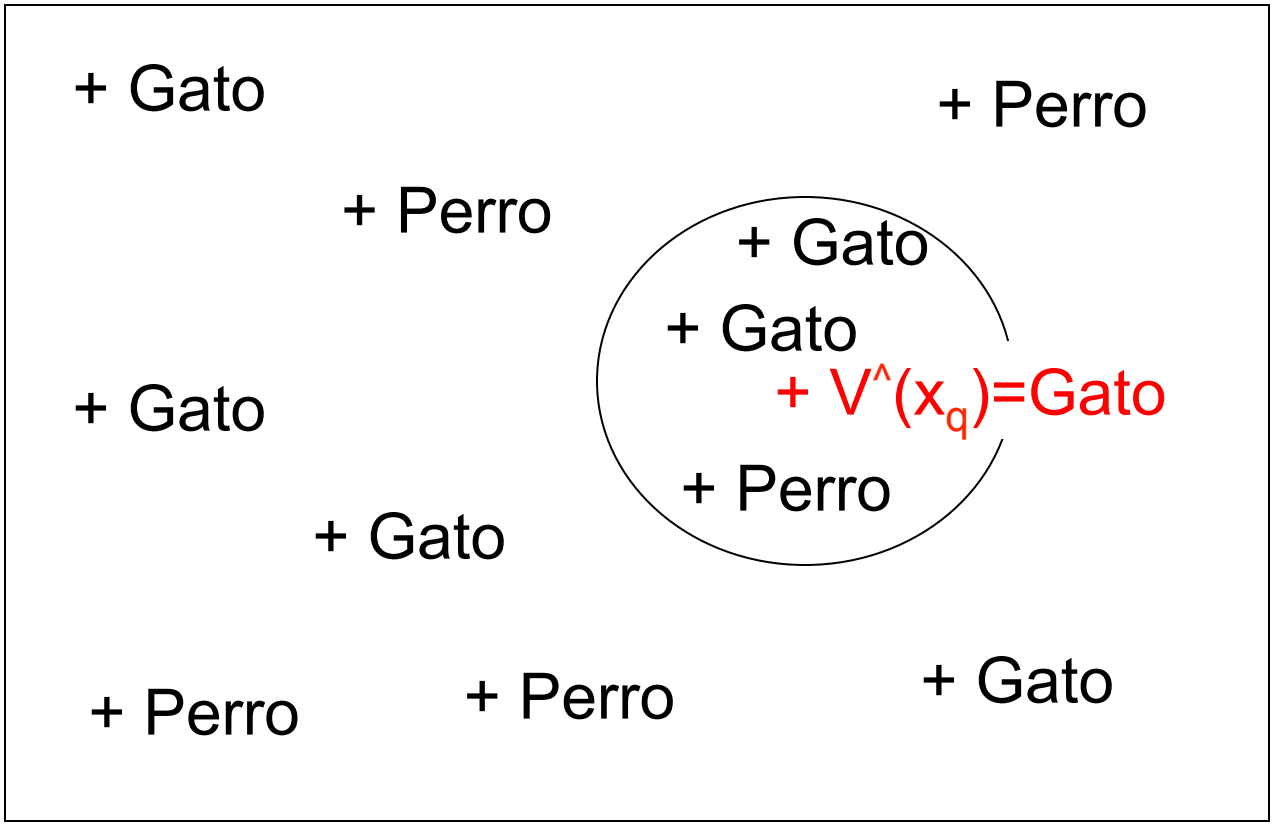


- Vamos a ver un algoritmo de esta familia llamado k-Vecinos Cercanos
  - Versión básica
  - Versión ponderada por distancia

Algoritmo k-Vecinos Cercanos Versión Básica

- Cada instancia es considerada como un punto en el espacio n-dimensional $\Re^n$, donde n es el número de atributos de cada instancia
- La función objetivo puede ser tanto real como discreta (categórica)
- En el caso de ser **discreta**, el algoritmo selecciona los k vecinos más cercanos a la instancia a clasificar x_q y le asigna la categoría más común entre estos. Por ejemplo, dada la instancia
  - x_q=<1,0,0,1,1>
  - Supongamos que los k=3 vecinos más cercanos son:
    - (<1,1,0,1,1>,Gato)
    - (<1,0,0,0,1>,Perro)
    - (<0,0,0,1,1>,Gato)
  - La clasificación de <1,0,0,1,1> será Gato

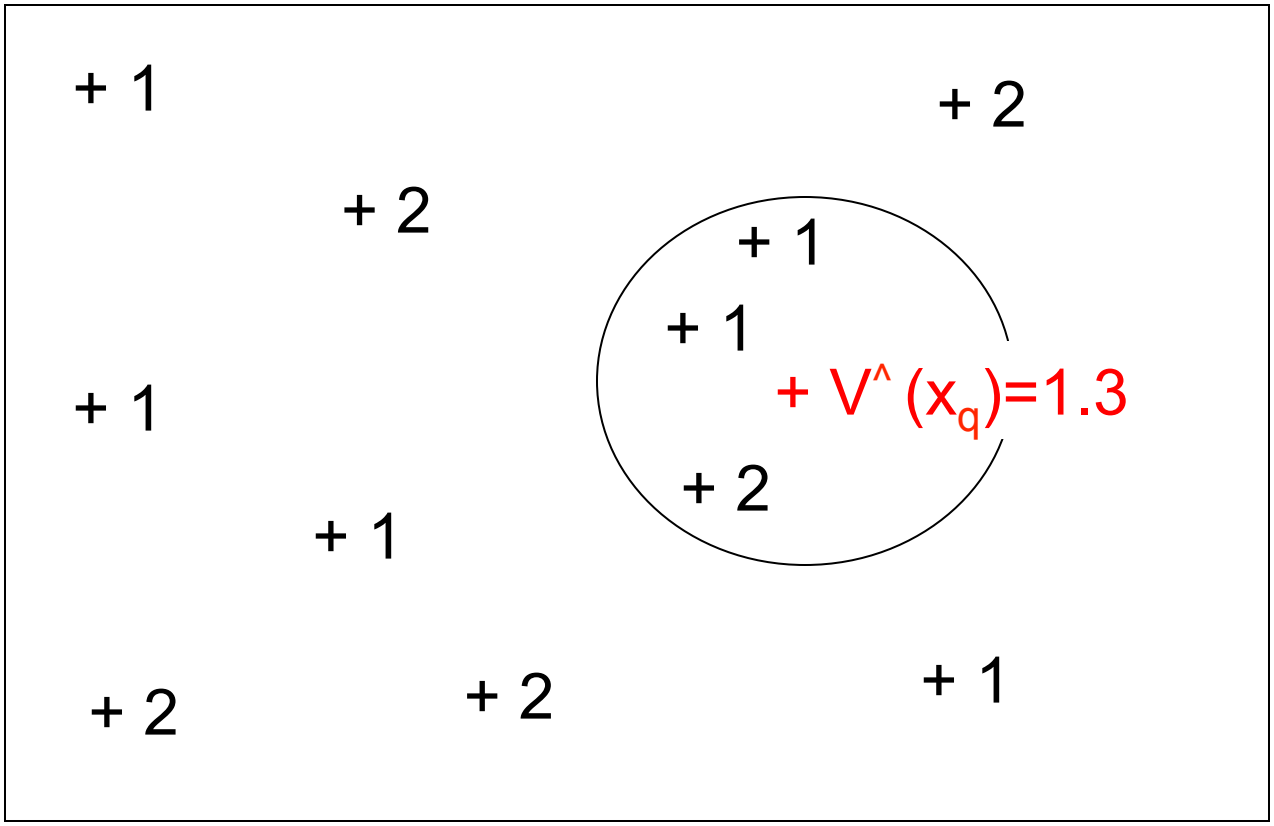
Ejemplo
Clasificar x_q, con k=3



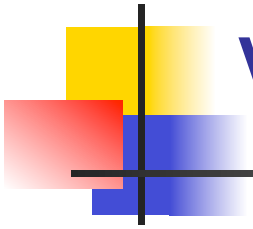
k-Vecinos Cercanos Versión Básica

- En el caso de ser **continua**, se calcula el promedio de los valores de las F.O. de los k vecinos más cercanos a la instancia a clasificar (x_q). Por ejemplo, dada
  - x_q=<1,0,0,1,1>
  - Supongamos que los k=3 vecinos más cercanos son:
    - (<1,1,0,1,1>,1)
    - (<1,0,0,0,1>,2)
    - (<0,0,0,1,1>,1)
  - La clasificación de <1,0,0,1,1> será 4/3=1.3

Ejemplo
Clasificar x_q, con k=3



k-Vecinos Cercanos
Versión Básica



- La cercanía entre instancias se calcula usando alguna medida de distancia; por ejemplo, la distancia Euclidiana
  - La distancia Euclidiana entre x_q y x_j es:

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

    -donde $a_r(x_q)$ es el valor del atributo r de la instancia x_q. La suma se realiza para cada uno de los n atributos

k-Vecinos Cercanos
Algoritmo Versión Básica Discreta

- El algoritmo de entrenamiento:
  - Guardar cada ejemplo (x, f(x))
- Algoritmo de clasificación para valores discretos
  - Dada la instancia x_q a clasificar
  - Calcular la distancia de cada ejemplo de entrenamiento a x_q. Sean x_1 a x_k los k ejemplos más cercanos a x_q
    - $V\hat{}(x_q)$<--- Elemento más común de $\{f(x_1), f(x_2) \ldots f(x_k)\}$

k-Vecinos Cercanos
Algoritmo Versión Básica Continua

- Cuando la función objetivo es continua, podemos adaptar el algoritmo de la siguiente manera:
- El algoritmo de entrenamiento:
  - Guardar cada ejemplo (x f(x))
- Algoritmo de clasificación para valores continuos
  - Dada la instancia x_q a clasificar
  - Calcular la distancia de cada ejemplo de entrenamiento a x_q. Sean x_1 a x_k los k ejemplos más cercanos a x_q
    - $V\hat{}(x_q) \leftarrow -1/k \sum_{i=1,k} f(xi)$
    - El promedio de la f.o. de los k valores más cercanos

k-Vecinos Cercanos
Ejemplo Caso Discreto

- Datos

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

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

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

x_q es clasificado como Bueno

k-Vecinos Cercanos
Ejemplo Caso Continuo

- Datos

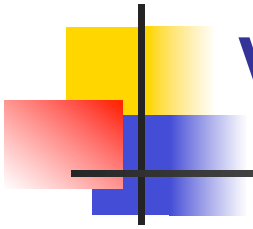
Calif. Mate	Calif. Bio	Estudiante	dist. a Xq
8	8	2	1
9	8	2	1.41421356
7	9	2	2.23606798
9	5	1	2.23606798
6	7	1	2
7	7	1	1

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

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

x_q es clasificado como 5/3=1.6

k-Vecinos Cercanos
Versión Ponderada



- Una extensión obvia del algoritmos es ponderar la contribución de cada uno de los k vecinos con respecto a su distancia a x_q (el punto a clasificar)
  - Mientras mas lejano a x_q menor influencia en su clasificación
  - Se puede utilizar el inverso del cuadrado de la distancia, para que disminuya rápido la influencia de los vecinos lejanos

k-Vecinos Cercanos Versión Ponderada

- Para ponderar contribuciones de cada dato con el inverso de la distancia cuadrada, modificación la última línea de los algoritmos
- Caso discreto
  - $f(x_q)$<--- Elemento más común de $\{w_1 f(x_1), w_2 f(x_2), \ldots, w_k f(x_k)\}$
    donde $w_i = 1/\text{distancia}(x_q, x_i)^2$. Note que para determinar cuál es el más común, se deben sumar las w_i de todas las $f(x_i)$ que sean iguales
- Caso continuo
  - $f(x_q) \leftarrow 1/r \sum_{i=1,k} w_i f(xi)$
    donde $r = \sum_{i=1,k} w_i$
- Nota: Si las distancia entre x_q y x_i cero se asigna a x_q el valor de $f(x_i)$

k-Vecinos Cercanos

Ejemplo Versión Ponderada (Caso Discreto)

- Datos

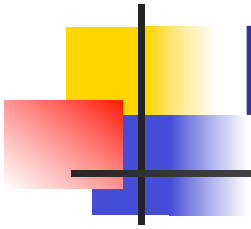
Calif. Mate	Calif. Bio	Estudiante	dist. a Xq
8	8	Bueno	2.01246118
9	8	Bueno	1.80277564
7	9	Bueno	3.38378486
9	5	Malo	1.20415946
6	7	Malo	3.00832179
7	7	Malo	2.06155281

- Clasificación 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-Vecinos Cercanos

- Algunas características
  - Robusto a ruido (usa muchos ejemplos para clasificar)
  - Necesita muchos ejemplos
  - Utiliza todos los atributos
    - En contraste con los árboles de decisión
    - Esto puede ser un problema: la maldición de la dimensionalidad

k-Vecinos Cercanos
Ejemplo de la Maldición

- Datos

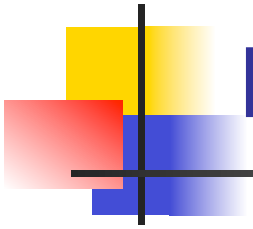
Distancia	c3 de lluvia	Temp	Calif. Mate	Calif. Bio	Estudiante	dist. a Xq
500	50	25	8	8	Bueno	600.087494
1000	150	23	9	8	Bueno	148.667414
300	60	21	7	9	Bueno	800.255584
600	50	25	9	5	Malo	500.108988
300	100	23	6	7	Malo	802.249338
1500	40	21	7	7	Malo	400.00625

- Clasificación x_q=(1100,40,23,8,7)

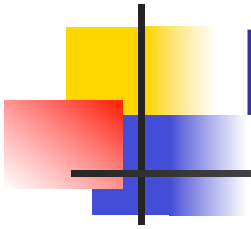
3-mas cercanos					
1000	150	23	9	8	Bueno
1500	40	21	7	7	Malo
600	50	25	9	5	Malo

x_q es clasificado como malo gracias a los nuevos atributos Irrelevantes (antes lo habíamos clasificado como bueno)

k-Vecinos Cercanos
La maldición de la dimensionalidad



- Algunas posibles soluciones
  - Escoger atributos relevantes
    - “Subset selection”
    - “Principal components”: Técnica basada en algebra lineal que determina los atributos o la combinación lineal de atributos que mejor separan los datos
    - Ganancia Informática: Parecido a lo que haremos para determinar nodos en el árbol de decisión
    - “Correlation Based Feature Selection” : Calcula la correlación entre diferentes subconjuntos de atributos para determinar cuáles son redundantes y cuáles esenciales
    - Y muchas mas.



Ejercicio

- Genere un conjunto de datos compuesto por muchos círculos dispersos en el espacio de dos dimensiones. Los puntos dentro de los círculos pertenecen a la categoría "dentro" y los que caen fuera a "fuera"
- Compare con SVM o con RN (escoga una) (No todos SVM por favor)