# Aprendizaje Estadístico

Modelado y Análisis de Redes de Telecomunicaciones

IIE - Facultad de Ingeniería

Curso 2014

# Introducción

- Algunos ejemplos de aprendizaje (son muchos!):
  - clasificar mail en spam o no basado en el contenido
  - predecir el precio de un producto dentro de seis meses basado en datos de la empresa y de la economia
  - identificar números de un código postal escrito a mano
  - identificar riesgo de cáncer basado en datos clínicos y demógraficos
  - clasificación de flujos (aplicaciones) en internet basado en el contenido de algunos pocos paquetes
  - predicción de la carga de un enlace basado en medidas pasadas
  - predicción de la QoS de un video basada en medidas activas (paq. de pruebas).

# Introducción

- Algunos ejemplos de aprendizaje (son muchos!):
  - clasificar mail en spam o no basado en el contenido
  - predecir el precio de un producto dentro de seis meses basado en datos de la empresa y de la economia
  - identificar números de un código postal escrito a mano
  - identificar riesgo de cáncer basado en datos clínicos y demógraficos
  - clasificación de flujos (aplicaciones) en internet basado en el contenido de algunos pocos paquetes
  - predicción de la carga de un enlace basado en medidas pasadas
  - predicción de la QoS de un video basada en medidas activas (paq. de pruebas).
  - y un muy largo etc.

# Aprendiendo de los datos

- Se considera un **output** que se quiere predecir: puede ser cuantitativo (continuo, e.g. carga del enlace) o cualitativo (discreto, e.g. QoS buena o mala)
- La predicción se basa en un conjunto de **features** (**inputs**) (e.g. serie temporal pasada de la carga del enlace o QoS experimentada por un paq. de prueba)

# Aprendiendo de los datos

- Se considera un **output** que se quiere predecir: puede ser cuantitativo (continuo, e.g. carga del enlace) o cualitativo (discreto, e.g. QoS buena o mala)
- La predicción se basa en un conjunto de **features** (**inputs**) (e.g. serie temporal pasada de la carga del enlace o QoS experimentada por un paq. de prueba)

- Se tiene una muestra de entrenamiento: conjunto de inputs y outputs
- El modelo de predicción (predictor, machine learning) se construye en base a la muestra de entrenamiento y permite la predicción del output para un nuevo input (desconocido)
- Un buen predictor es aquel que predice con exactitud (precisión) dicho output

# Aprendizaje Supervisado

- Lo anterior es un ejemplo de Aprendizaje Supervisado:
  - en la muestra de entrenamiento se tienen los outputs
- En el caso No Supervisado sólo se observan los features y el objetivo es principalmente organizar o agrupar esos datos: e.g. clustering

# Aprendizaje Supervisado

- Lo anterior es un ejemplo de Aprendizaje Supervisado:
  - en la muestra de entrenamiento se tienen los outputs
- En el caso No Supervisado sólo se observan los features y el objetivo es principalmente organizar o agrupar esos datos: e.g. clustering
- Formalmente se tiene un conjunto $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ tal que $X \in S_X$ son los inputs e $Y \in S_Y$ son los outputs
  - los espacios $S_X$ e $S_Y$ son muy generales: vectores, funciones, categorías...
- Una *learning machine* (predictor) es un algoritmo que dado un nuevo input $X$ predice el output $\widehat{Y} = f(X)$
- Si el output es continuo se dice que el problema es de *regresión* y si es discreto se dice que es de *clasificación*

# Aprendizaje Supervisado

- ¿Qué es un buen predictor? ¿Cómo se construye?
- La calidad del predictor depende de lo que se quiera obtener y no es una cuestión trivial
- Clasificación binaria ($Y \in \{-1, 1\}$): ¿qué error se mide?

# Aprendizaje Supervisado

- ¿Qué es un buen predictor? ¿Cómo se construye?
- La calidad del predictor depende de lo que se quiera obtener y no es una cuestión trivial
- Clasificación binaria ($Y \in \{-1, 1\}$): ¿qué error se mide?
  - número de datos bien/mal clasificados
  - sin embargo los errores pueden no ser simétricos (enfermo/no enfermo , spam/no spam)
  - puede interesar además el nivel de confianza en el resultado
  - incluso pueden depender del input: e.g. predicción metereológica en zonas urbanas/rurales

# Aprendizaje Supervisado

- ¿Qué es un buen predictor? ¿Cómo se construye?
- La calidad del predictor depende de lo que se quiera obtener y no es una cuestión trivial
- Clasificación binaria ($Y \in \{-1, 1\}$): ¿qué error se mide?
  - número de datos bien/mal clasificados
  - sin embargo los errores pueden no ser simétricos (enfermo/no enfermo , spam/no spam)
  - puede interesar además el nivel de confianza en el resultado
  - incluso pueden depender del input: e.g. predicción metereológica en zonas urbanas/rurales
- En el caso de varias clases o regresión puede ser más complicado aún

# Aprendizaje Supervisado

- ¿Qué es un buen predictor? ¿Cómo se construye?
- La calidad del predictor depende de lo que se quiera obtener y no es una cuestión trivial
- Clasificación binaria ($Y \in \{-1, 1\}$): ¿qué error se mide?
  - número de datos bien/mal clasificados
  - sin embargo los errores pueden no ser simétricos (enfermo/no enfermo , spam/no spam)
  - puede interesar además el nivel de confianza en el resultado
  - incluso pueden depender del input: e.g. predicción metereológica en zonas urbanas/rurales
- En el caso de varias clases o regresión puede ser más complicado aún
- En general lo que se hace es definir una función de pérdida y se busca minimizar el riesgo esperado

# Función de Pérdida

- Sea $L : S_X \times S_Y \times S_Y \to \mathbb{R}^+$ tal que:

  $L(x, u, y)$ *es el costo de decidir* $u$ *para el input* $x$ *siendo* $y$ *el verdadero output*

# Función de Pérdida

- Sea $L : S_X \times S_Y \times S_Y \to \mathbb{R}^+$ tal que:

  $L(x, u, y)$ *es el costo de decidir* $u$ *para el input* $x$ *siendo* $y$ *el verdadero output*

- Se definen diferentes funciones de pérdida dependiendo del problema.
- Problemas de clasificación:
  1. $L(x, u, y) = \mathbf{1}_{\{u \neq y\}}$ número de datos mal clasificados
  2. Si se consideran errores asimétricos o dependientes del input

$$L(x, u, y) = \begin{cases} 0 & y = u \\ c(x) & \text{otro caso} \end{cases}$$

  3. Soft Margin Loss: se puede definir

$$L(x, u, y) = \begin{cases} 0 & yu \geq 1 \\ 1 - yu & \text{otro caso} \end{cases}$$

     también se usa $(1 - yu)^2$
  4. Logistic Loss: $L(x, u, y) = \log(1 + \exp(-yu))$

# Función de Pérdida

- Para problemas de regresión :
  1. $L(x, u, y) = (u - y)^2$ corresponde a asumir ruido gaussiano aditivo en las observaciones
  2. $\epsilon$-Insensitive loss:

$$L(x, u, y) = \text{máx}(|y - u| - \epsilon, 0)$$

la idea es que desviaciones menores a $\epsilon$ no deben ser penalizadas

# Función de Pérdida

- Para problemas de regresión :
  1. $L(x, u, y) = (u - y)^2$ corresponde a asumir ruido gaussiano aditivo en las observaciones
  2. $\epsilon$-Insensitive loss:

$$L(x, u, y) = \text{máx}(|y - u| - \epsilon, 0)$$

     la idea es que desviaciones menores a $\epsilon$ no deben ser penalizadas

- Hay condiciones sobre la función $L$, por ejemplo que sea computacionalmente barata y que sea suficientemente regular para garantizar "solución única"

# Riesgo Esperado

- Asumimos que los pares $(X, Y)$ son v.a. *iid* con distribución conjunta $P(x, y) = P(y|x)P(x)$ (desconocida)
- Sea $\mathcal{F} = \{f : S_X \to S_Y\}$ el conjunto de todas las posibles funciones de $S_X$ en $S_Y$
  - un predictor (learning machine) es $f \in \mathcal{F}$
- Un buen predictor $f^*$ será aquel que minimice el riesgo esperado $R(f)$

$$R(f) = E(L(X, f(X), Y)) = \int_{S_X} \int_{S_Y} L(x, f(x), y) dP(x, y)$$

# Error Risk Minimization

- Dado que $P(x, y)$ es desconocida, se busca minimizar el *riesgo empírico*:

$$R_n(f) = \frac{1}{n} \sum_{i=1}^{n} L(X_i, f(X_i), Y_i)$$

# Error Risk Minimization

- Dado que $P(x, y)$ es desconocida, se busca minimizar el *riesgo empírico*:

$$R_n(f) = \frac{1}{n} \sum_{i=1}^{n} L(X_i, f(X_i), Y_i)$$

- Principio ERM (Error Risk Minimization): se define como predictor a $f_n^*$ tal que minimice $R_n(f)$.

# Error Risk Minimization

- Dado que $P(x, y)$ es desconocida, se busca minimizar el *riesgo empírico*:

$$R_n(f) = \frac{1}{n} \sum_{i=1}^{n} L(X_i, f(X_i), Y_i)$$

- Principio ERM (Error Risk Minimization): se define como predictor a $f_n^*$ tal que minimice $R_n(f)$.
- ¿Cúal es la validez de este principio? ¿ es $f_n^*$ una buena aproximación de $f^*$? o más importante aún ¿es $R_n(f_n^*)$ una buena aproximación de $R(f^*)$?
- Muchos de los resultados de Vapnik intentar probar que este principio es válido o más exactamente, bajo qué hipótesis lo es

# Error Risk Minimization

- El ERM es consistente si:

$$R(f_n) \underset{n\to\infty}{\xrightarrow{p}} \inf_{f\in\mathcal{F}} R(f) = R(f^*)$$

$$R_n(f_n) \underset{n\to\infty}{\xrightarrow{p}} \inf_{f\in\mathcal{F}} R(f) = R(f^*)$$

# Error Risk Minimization

- El ERM es consistente si:

$$R(f_n) \xrightarrow[n\to\infty]{p} \inf_{f\in\mathcal{F}} R(f) = R(f^*)$$

$$R_n(f_n) \xrightarrow[n\to\infty]{p} \inf_{f\in\mathcal{F}} R(f) = R(f^*)$$

- Teorema Fundamental del Learning (Vapnik): Si $a \leq L \leq b$ para toda $f \in \mathcal{F}$ son equivalentes:
  1. ERM es consistente
  2. El riesgo empírico $R_n(f)$ converge al riesgo esperado $R(f)$ en el siguiente sentido:

$$\lim_{n\to\infty} P\left(\sup_{f\in\mathcal{F}}(R_n(f) - R(f)) > \epsilon\right) = 0 \quad \forall\, \epsilon > 0$$

# Error Risk Minimization

- El ERM es consistente si:

$$R(f_n) \underset{n\to\infty}{\xrightarrow{p}} \inf_{f\in\mathcal{F}} R(f) = R(f^*)$$

$$R_n(f_n) \underset{n\to\infty}{\xrightarrow{p}} \inf_{f\in\mathcal{F}} R(f) = R(f^*)$$

- Teorema Fundamental del Learning (Vapnik): Si $a \leq L \leq b$ para toda $f \in \mathcal{F}$ son equivalentes:
  1. ERM es consistente
  2. El riesgo empírico $R_n(f)$ converge al riesgo esperado $R(f)$ en el siguiente sentido:

$$\lim_{n\to\infty} P\left(\sup_{f\in\mathcal{F}}(R_n(f) - R(f)) > \epsilon\right) = 0 \quad \forall\, \epsilon > 0$$

- Un caso sencillo donde se cumple lo anterior es cuando $S_X$ y $S_Y$ son finitos (hay convergencia uniforme)

# Teorema Fundamental del Learning

- Lo anterior es un resultado téorico ($R(f)$ no se puede calcular)
- La idea central es que la segunda condición se verifica si la familia de funciones $\mathcal{F}$ no es demasiado "grande"
- Hay que tener una medida de tamaño de $\mathcal{F}$: entropía, dimensión de Vapnik-Chervonenkis (VC)
- Hay muchos resultados en ese sentido, uno de ellos es el siguiente:

  Si $a \leq L \leq b$, para un $\eta$ dado, con probabilidad $1 - \eta$ se cumple que:

$$R(f) \leq R_n(f) + (b-a)\sqrt{\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n}}$$

  donde $h$ es la dimensión de VC.

# Teorema Fundamental del Learning

$$R(f) \leq R_n(f) + (b-a)\sqrt{\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n}}$$

- No depende de $P(x, y)$, sólo se asume que sean observaciones iid de acuerdo a alguna distribución
- El lado izquierdo no se puede calcular
- Si $h$ se puede calcular, eligiendo un $\eta$ chico, basta elegir la $f$ que minimice la cota para obtener el predictor que minimiza el error esperado
- Si $h$ es grande, no se puede decir nada sobre la elección de un predictor en particular
  - SVM tiene dimensión de VC infinita pero muy buena performance

# Error de Aproximación y de Estimación

- $f_n^*$ es el predictor que minimiza el riesgo empírico $R_n(f)$ para toda $f \in \mathcal{F}$
- Sea $\widehat{f}_n$ el predictor que minimiza $R_n(f)$ para toda $f$ en un subconjunto $\mathcal{F}_0 \subset \mathcal{F}$ (se está eligiendo un modelo)

# Error de Aproximación y de Estimación

- $f_n^*$ es el predictor que minimiza el riesgo empírico $R_n(f)$ para toda $f \in \mathcal{F}$
- Sea $\widehat{f}_n$ el predictor que minimiza $R_n(f)$ para toda $f$ en un subconjunto $\mathcal{F}_0 \subset \mathcal{F}$ (se está eligiendo un modelo)
- Hay dos tipos de errores:
  - Error de estimación: por minimizar $R_n(f)$ en lugar de $R(f)$
  - Error de aproximación: por buscar el mínimo en $\mathcal{F}_0$ y no en $\mathcal{F}$ (este error no se puede disminuir aunque se mejore la calidad de la muestra)

# Error de Aproximación y de Estimación

- $f_n^*$ es el predictor que minimiza el riesgo empírico $R_n(f)$ para toda $f \in \mathcal{F}$
- Sea $\widehat{f}_n$ el predictor que minimiza $R_n(f)$ para toda $f$ en un subconjunto $\mathcal{F}_0 \subset \mathcal{F}$ (se está eligiendo un modelo)
- Hay dos tipos de errores:
  - Error de estimación: por minimizar $R_n(f)$ en lugar de $R(f)$
  - Error de aproximación: por buscar el mínimo en $\mathcal{F}_0$ y no en $\mathcal{F}$ (este error no se puede disminuir aunque se mejore la calidad de la muestra)
- Compromiso entre ambos errores al momento de elegir $\mathcal{F}_0$
- Problema de sobreajuste (overfitting): si se define

$$f_n^*(x) = \begin{cases} Y_i & \text{si } x = X_i \\ 0 & \text{otro caso} \end{cases}$$

se tiene error nulo en la muestra de entrenamiento pero el error puede ser muy grande cuando se tiene una muestra distinta (capacidad de generalización)

# Capacidad de generalización

- C.J.C. Burges: "A machine with too much capacity is like a botanist with a photographic memory who, when presented with a new tree, concludes that it is not a tree because it has a different number of leaves from anything seen before; a machine with little capacity is like the botanist lazy brother, who declares that if it is green, it is a tree. Neither can generalized well"

# Capacidad de generalización

- C.J.C. Burges: "A machine with too much capacity is like a botanist with a photographic memory who, when presented with a new tree, concludes that it is not a tree because it has a different number of leaves from anything seen before; a machine with little capacity is like the botanist lazy brother, who declares that if it is green, it is a tree. Neither can generalized well"
- En general se construye el predictor en base a la muestra de entrenamiento y se mide su desempeño sobre una muestra distinta (muestra de validación)

# Capacidad de generalización

- C.J.C. Burges: "A machine with too much capacity is like a botanist with a photographic memory who, when presented with a new tree, concludes that it is not a tree because it has a different number of leaves from anything seen before; a machine with little capacity is like the botanist lazy brother, who declares that if it is green, it is a tree. Neither can generalized well"
- En general se construye el predictor en base a la muestra de entrenamiento y se mide su desempeño sobre una muestra distinta (muestra de validación)
- Dos ejemplos sencillos pero ilustrativos:
  - **Mínimos Cuadrados** (para un **Modelo Lineal**): importantes hipótesis sobre el modelo pero predicciones estables con posibles errores
  - **Vecinos Más Cercanos**: pocas hipótesis sobre el modelo pero las predicciones aunque acertadas pueden ser muy inestables

# Modelo Lineal y Mínimos Cuadrados

- Dado un input $X = (x_1, x_2, \ldots, x_p)$ se predice el output $Y$ según el modelo:

$$\widehat{Y} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

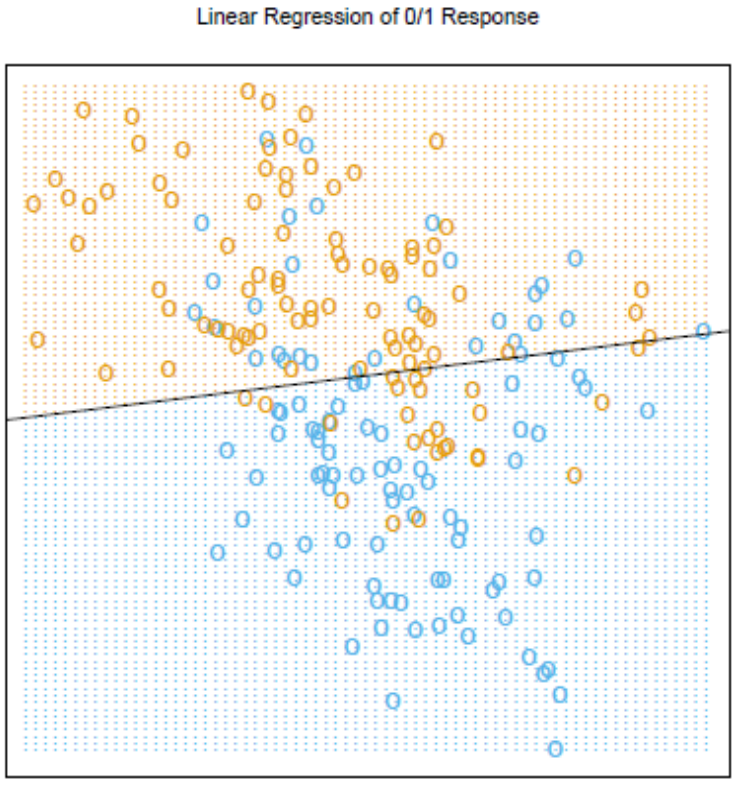
  o equivalente $\widehat{Y} = X^t \beta$ (agrego una coordenada a $X$)
- El vector $(X, \widehat{Y})$ representa un hiperplano
- Mínimos Cuadrados es el método más usado para ajustar el modelo: seleccionar $\beta$ tal que minimice la suma de los errores al cuadrado:

$$RSS(\beta) = \sum_{i=1}^{n} (Y_i - X_i^t \beta)^2$$

- Función cuadrática: el mínimo siempre existe y si la matriz $X^t X$ es no singular la solución (única) está dada por

$$\widehat{\beta} = (X^t X)^{-1} X^t Y$$

# Modelo Lineal y Mínimos Cuadrados



# Modelo Lineal y Mínimos Cuadrados

- Datos simulados: $Y = 0$ para el azul e $Y = 1$ para el naranja.
- Los valores ajustados se clasifican según la siguiente regla:

$$\widehat{G} = \begin{cases} \text{naranja} & \text{si } \widehat{Y} > 0{,}5 \\ \text{azul} & \text{si } \widehat{Y} \leq 0{,}5 \end{cases}$$

- Los datos se separan según la frontera de decisión $\left\{x : x^t \widehat{\beta} = 0{,}5\right\}$ donde $\beta$ es hallado por mínimos cuadrados (lineal en este caso)
- Hay errores en las dos clases
- Dependiendo de cómo se hayan generado los datos, este modelo es lo mejor que se pude hacer

# Vecinos Más Cercanos

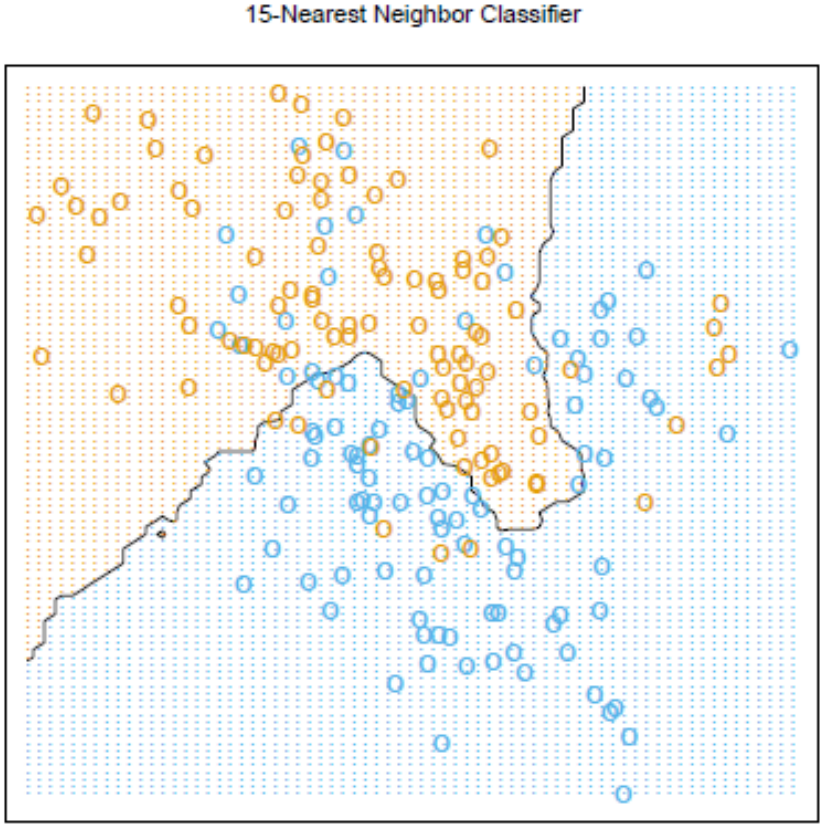
- Para definir $\widehat{Y}$ se consideran los inputs más "cercanos" de la muestra de entrenamiento
- En particular $k$ - vecinos más cercanos se define como:

$$\widehat{Y}(x) = \frac{1}{k} \sum_{X_i \in N_k(x)} Y_i$$

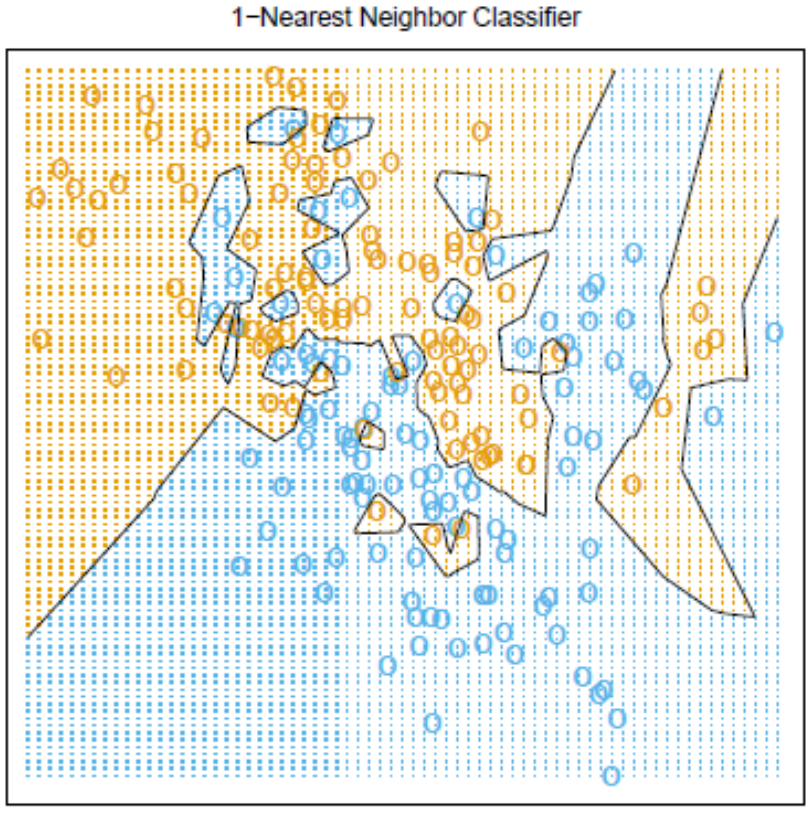
donde $N_k(x)$ es el entorno de $x$ definido por los $k$ puntos más cercanos.

- Cerca implica una métrica: norma euclidea
- Se promedian las $k$ observaciones más cercanas a $x$ en el espacio de los inputs
- Mismo ejemplo con 15 vecinos: $\widehat{Y}$ es la proporción de color en el entorno (se usa una grilla muy fina para colorear toda la región)

# Vecinos Más Cercanos



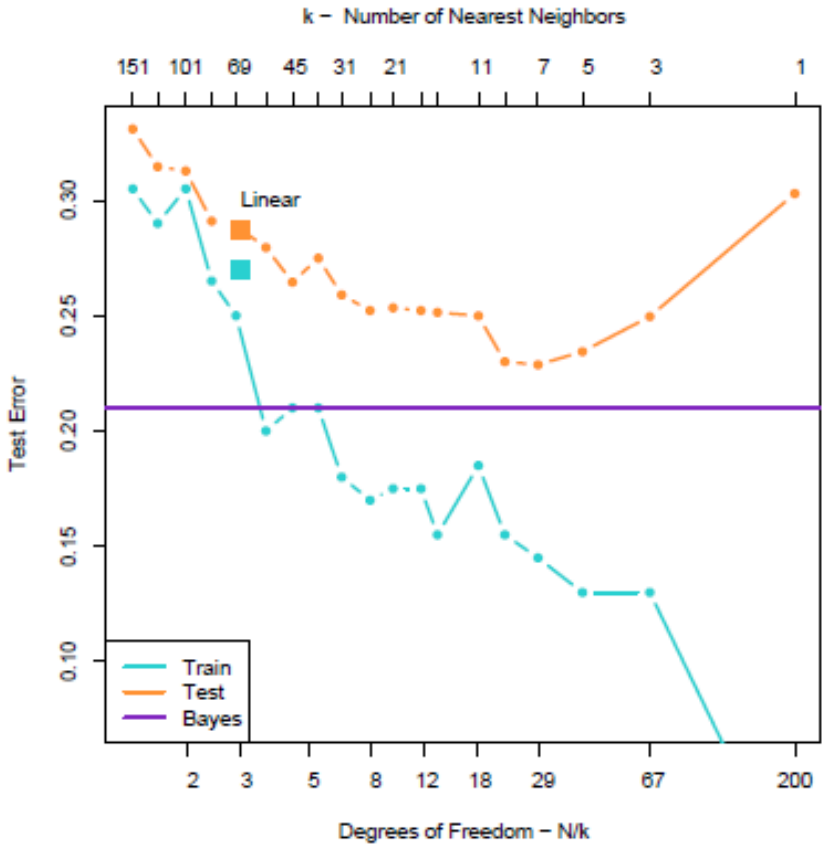
# Vecinos Más Cercanos



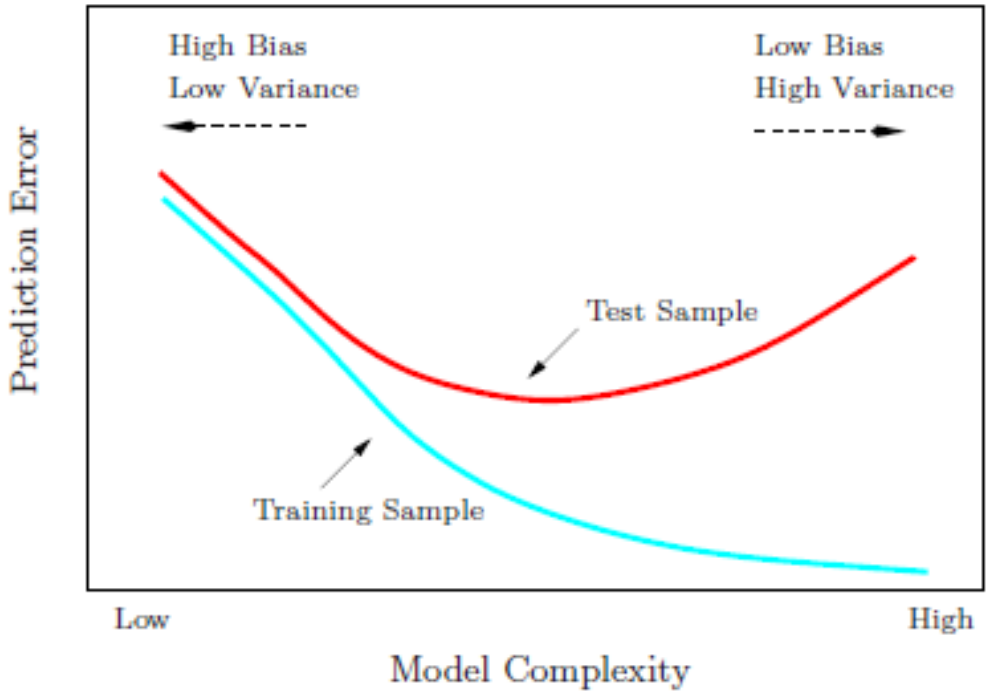
# Comparación

- Hay menos errores que en el modelo lineal (aunque más que en mosaico de voronoi!)
- En realidad habría que comparar sobre una muestra distinta a la de entrenamiento
- Número de parámetros:
  - Mínimos Cuadrados: $p$ parámetros
  - Vecinos Más Cercanos: 1 sólo parámetro ($k$) (en realidad el número efectivo es $n/k$ que en general es mayor a $p$)
- El modelo lineal da una decisión muy regular y estable (asume fuertemente que la relación es lineal)
- Vecinos más cercanos es más flexible pero poco estable (depende mucho de la posición de los puntos)

# Vecinos Más Cercanos



# Compromiso Sesgo - Varianza



# Un poco de formalidad ...

- Sea $L(Y, f(X)) = (Y - f(X))^2$ el error cuadrático como función de pérdida
- Sea $R(f) = E(Y - f(X))^2 = \int (y - f(x))^2 P(dx, dy)$ el error cuadrático de predicción
- Condicionando a $X$ se puede escribir

$$R(f) = E_X E_{Y|X}((Y - f(X))^2 | X)$$

y basta minimizar punto a punto, es decir

$$f(x) = \underset{c}{\operatorname{argmin}} \quad E_{Y|X}((Y - c)^2 | X = x)$$

de donde resulta que el mejor predictor es la esperanza condicional

$$f(x) = E(Y|X = x)$$

# Un poco de formalidad ...

- Vecinos más cercanos busca estimar la esperanza condicional promediando los puntos de la muestra en un entorno del punto dado:
  - la esperanza se aproxima tomando promedios
  - condicionando al punto se relaja condicionando a una región cerca del punto
- Hay problemas con la dimensión y la cantidad de puntos en el entorno (*course of dimensionality*, Bellman 1961)
- Los modelos lineales también entran en este contexto asumiendo que $f$ es lineal, es decir $f(x) = x^t\beta$, se calcula $R(f)$ y derivando se obtiene $\beta$ teóricamente

$$\beta = (E(X^tX))^{-1}E(X^tY)$$

que es lo mismo que antes pero tomando esperanza.

# Otros Métodos

- Muchas de las técnicas más usadas para clasificación y regresión son de hecho mejoras a estos métodos:
  - Métodos de Kernel: se usa una función que decrece a cero con la distancia (en lugar de 0/1)
  - Regresión Local: usa modelos lineales locales
  - Modelos lineales más transformación de los inputs
  - Redes Neuronales: suma de transformaciones no lineales de modelos lineales

# Referencias


- "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" Trevor Hastie, Robert Tibshirani, Jerome Friedman. http://www-stat.stanford.edu/ tibs/ElemStatLearn/
- "Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond" Bernhard Schlkopf, Alexander J.Smola.