Selección de Atributos

Aprendizaje Automático INF-393 II-2018

Ricardo Ñanculef

UTFSM Campus San Joaquín

Table of contents

Introducción

Propósitos de la Selección de Atributos

Dar una representación adecuada a los datos es fundamental para obtener buenos resultados en aprendizaje automático.

En el paradigma dominante, los datos de entrada se representan como vectores de atributos de la forma $x \in \mathbb{R}^d$. La i-ésima dimensión de esta representación corresponde a la medición de una característica o variable que denotaremos X_i.

Dos problemas asociados a esta representación son los siguientes:

1. Determinar el grado de asociación o dependencia que tiene la variable de salida Y con respecto a una determinada característica X_i.
2. Encontrar un pequeño subconjunto de atributos $X_{i_1}, X_{i_2}, \ldots, X_{i_B}$ que permita obtener una representación "más compacta" de los datos.

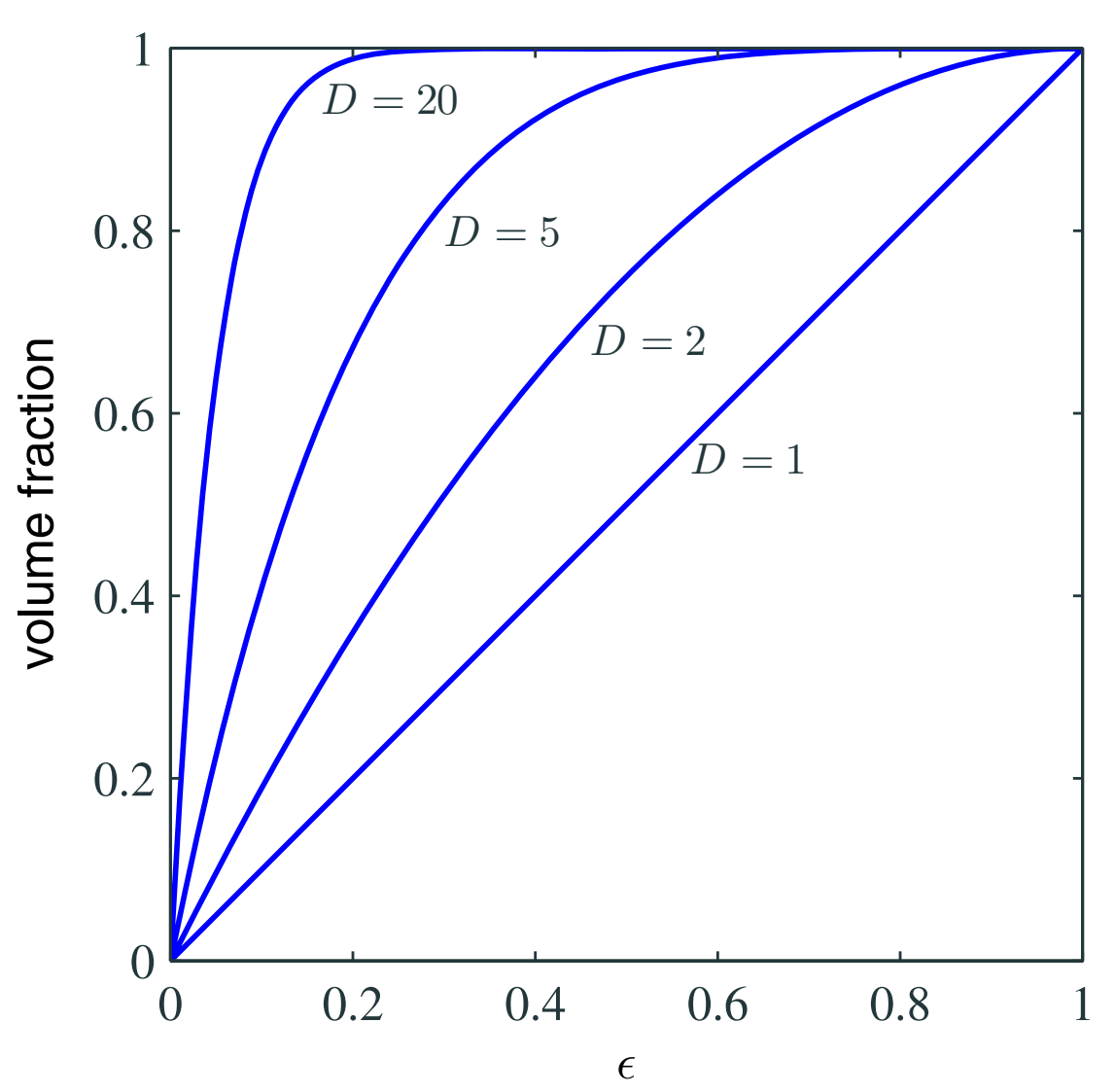
Propósitos de la Selección de Atributos

Encontrar esta representación "más compacta" de los datos puede ser importante por muchos motivos:

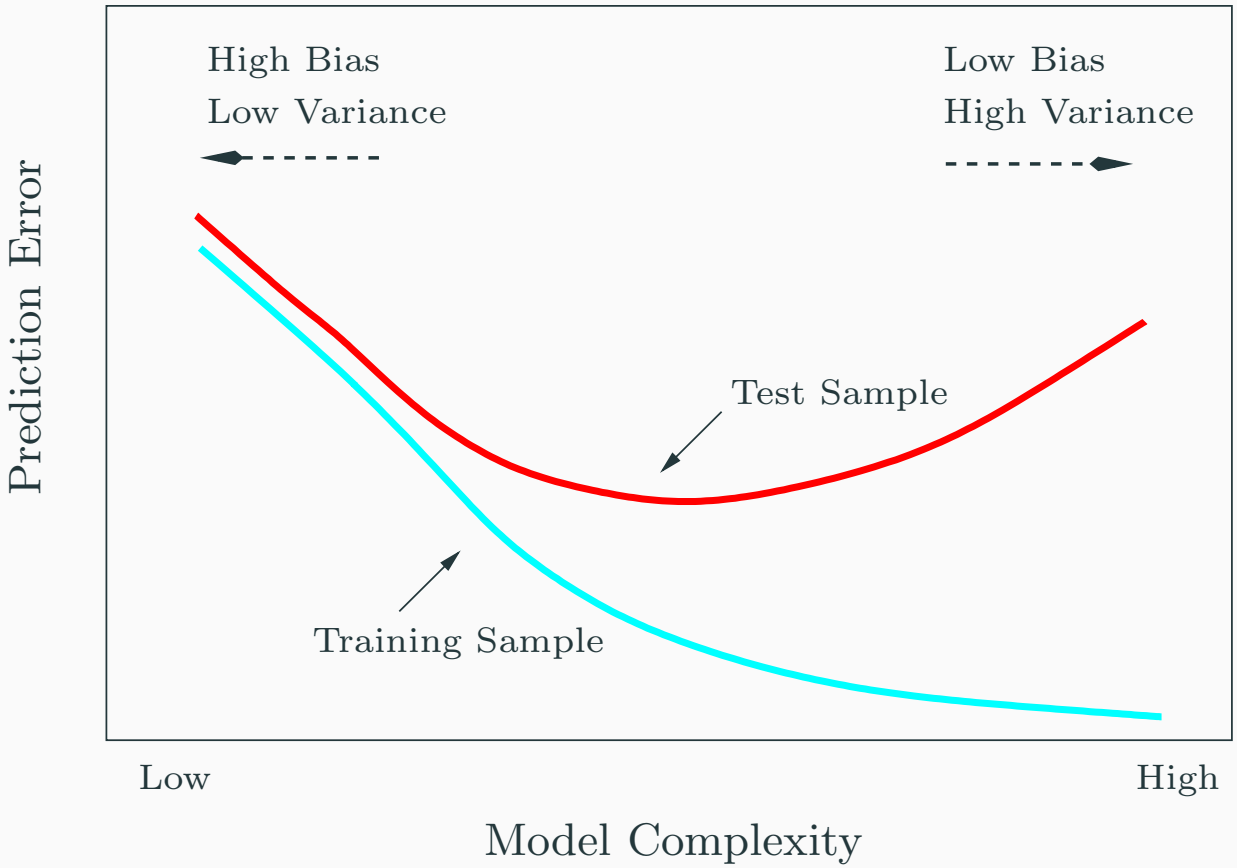
1. Hacer el modelo más comprensible.
2. Eficiencia computacional (tiempo & espacio).
3. Lidiar con la maldición de la dimensionalidad.
4. Evitar overfitting y por lo tanto obtener una mayor capacidad predictiva.

Estos problemas se estudian en la literatura bajo el nombre de selección de atributos o selección de características (feature selection).

Selección de Atributos & Maldición de la Dimensionalidad



Selección de Atributos & Overfitting



Una medida intuitiva de la complejidad de un modelo es el número de parámetros libres (entrenables). En general esta intuición es incorrecta.

Selección de Atributos & Overfitting

Sin embargo, la intuición anterior es correcta en algunos casos. Por ejemplo, la familia de modelos lineales de la forma $f(x) = w^T x + b$ tiene efectivamente complejidad (dimensión VC) dada por $c = d + 1$, donde d es el número de parámetros[1]. Recordar que

Con probabilidad $1 - \eta$,

$$R(f) \leq \hat{R}(f) + \sqrt{\frac{c \log\left(\frac{2n}{c} + 1\right) - \log\left(\frac{\eta}{4}\right)}{n}} \tag{1}$$

[1] Ver por ejemplo Hastie et al. *Elements of Statistical Learning*, Sección 7.9.

Tres Grandes Familias

Los métodos utilizados organizarse en 3 grandes grupos[2]:

1. Métodos de Filtrado (filter methods): Otorgan un puntaje (score) a cada atributo individual que es independiente de los demás.
2. Métodos tipo Wrapper (wrapper methods): Buscan el mejor subconjunto de atributos re-entrenando el modelo (clasificador o regresor) con distintos subconjuntos candidatos.
3. Métodos Embedidos (embedded methods): Buscan el mejor subconjunto de atributos, optimizando una función objetivo propia, que busca ser independiente del modelo a entrenar y/o evitar el costo de múltiples re-entrenamientos.

[2]Ver por ejemplo el paper de Isabelle Guyon & André Elisseeff: *An Introduction to Variable and Feature Selection.* JMLR 2003.

Métodos de Filtrado

Generalidades

Un método de filtrado asigna un puntaje individual a cada atributo que es (o debiese ser) proporcional al grado de asociación o dependencia de este con variable de salida y.

1. Permite hacer un ranking de las variables del problema.
2. Permite seleccionar el subconjunto de K atributos más relevantes.

Limitaciones:

1. Un atributo individualmente muy relevante puede volverse irrelevante si otros atributos están presentes.
2. Los K atributos más relevantes individualmente no necesariamente corresponden al conjunto de K atributos que maximizan la capacidad predictiva del modelo.

Correlación Lineal

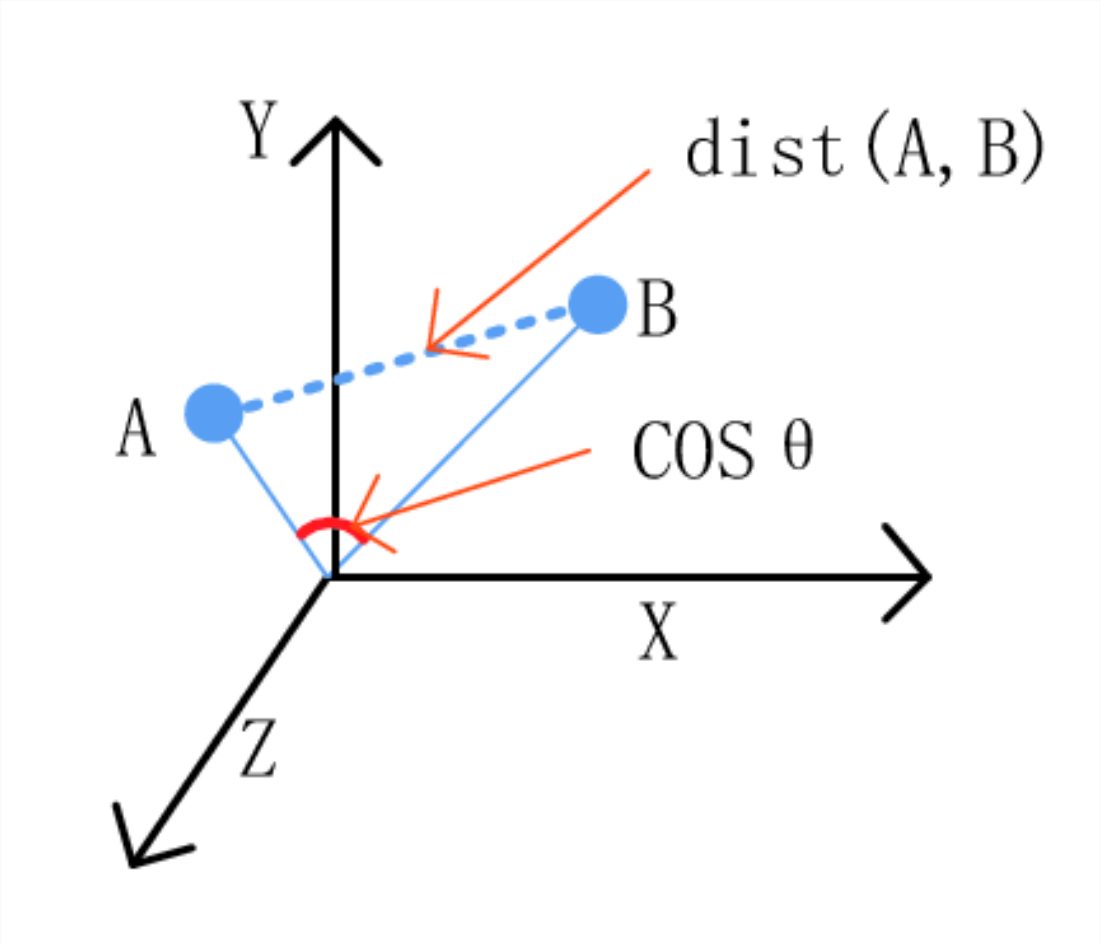
Asumiendo que Y es continua, un criterio simple y popular para medir la relevancia de una característica X_i es el coeficiente de correlación de Pearson

$$\rho(i) = \frac{cov(X_i, Y)}{\sqrt{\text{var}(X_i)\text{var}(Y)}} \,. \tag{2}$$

Dados n ejemplos, $\{(x^{(\ell)}, y^{(\ell)})\}_\ell$, es posible estimar $\rho(i)$ como

$$\hat{\rho}(i) = \frac{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)(y^{(\ell)} - \bar{y})}{\sqrt{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2}\sqrt{\sum_\ell (y^{(\ell)} - \bar{y})^2}} \,. \tag{3}$$

Interpretación Geométrica



Correlación & Z-score

Si hacemos una regresión lineal de Y sobre X_i, es decir, ajustamos el modelo $Y = a_i X_i + b_i + \epsilon_i$ usando mínimos cuadrados, obtenemos que el estimador MV de a_i viene dado por

$$\hat{a}_i = \frac{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)(y^{(\ell)} - \bar{y})}{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2} \,. \tag{4}$$

Asumiendo $\epsilon_i \sim \mathcal{N}(0, 1)$ obtenemos que

$$\mathrm{STD}(\hat{a}_i) = \frac{\sigma}{\sqrt{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2}} \tag{5}$$

Correlación & Z-score

De este modo, un ranking de los atributos en base a $\rho(i)$ es equivalente a un ranking en base al denominado Z-score de la variable i,

$$Z(i) = \frac{\hat{a}_i}{\mathrm{STD}(\hat{a}_i)} = \frac{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)(y^{(\ell)} - \bar{y})}{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2} \frac{\sqrt{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2}}{\hat{\sigma}^2} \tag{6}$$

$$= \hat{\sigma}^2 \frac{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)(y^{(\ell)} - \bar{y})}{\sqrt{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2}} \propto \rho(i)\,.$$

Bajo el supuesto $\epsilon_i \sim \mathcal{N}(0,1)$, y bajo la hipótesis $H_0 : a_i = 0$, $Z(i) \sim \mathcal{T}_{n-2}$ permite efectuar un contraste contra $H_0 : a_i \neq 0$ y obtener un subconjunto de coeficientes relevantes en base a un p-valor.

Recordar la descomposición

$$\sum_\ell (y^{(\ell)} - \bar{y})^2 = \sum_\ell (\hat{y}_i^{(\ell)} - \bar{y})^2 + \sum_\ell (\hat{y}_i^{(\ell)} - y^{(\ell)})^2 \qquad (7)$$
$$SST = SSR(i) + SSE(i)\,,$$

con $\hat{y}_i^{(\ell)} = \hat{a}_i x^{(\ell)} + \hat{b}$. Por lo tanto, un ranking de los atributos en base a $\rho(i)$ o a $Z(i)$ es equivalente a un ranking en base a

$$R_i^2 = \frac{SSR(i)}{SST} = \rho(i)^2\,,$$

Correlación & F-score

Notar también que el denominado F-score de la variable i satisface

$$F(i) = \frac{\frac{SSR(i)}{1}}{\frac{SSE(i)}{n-2}} = \frac{\frac{SSR(i)}{SST}}{\frac{SSE(i)}{SST(n-2)}} = (n-2)\frac{\frac{SSR(i)}{SST}}{1-\frac{SSR(i)}{SST}} = \frac{(n-2)\rho(i)^2}{(1-\rho(i)^2)}\,,$$

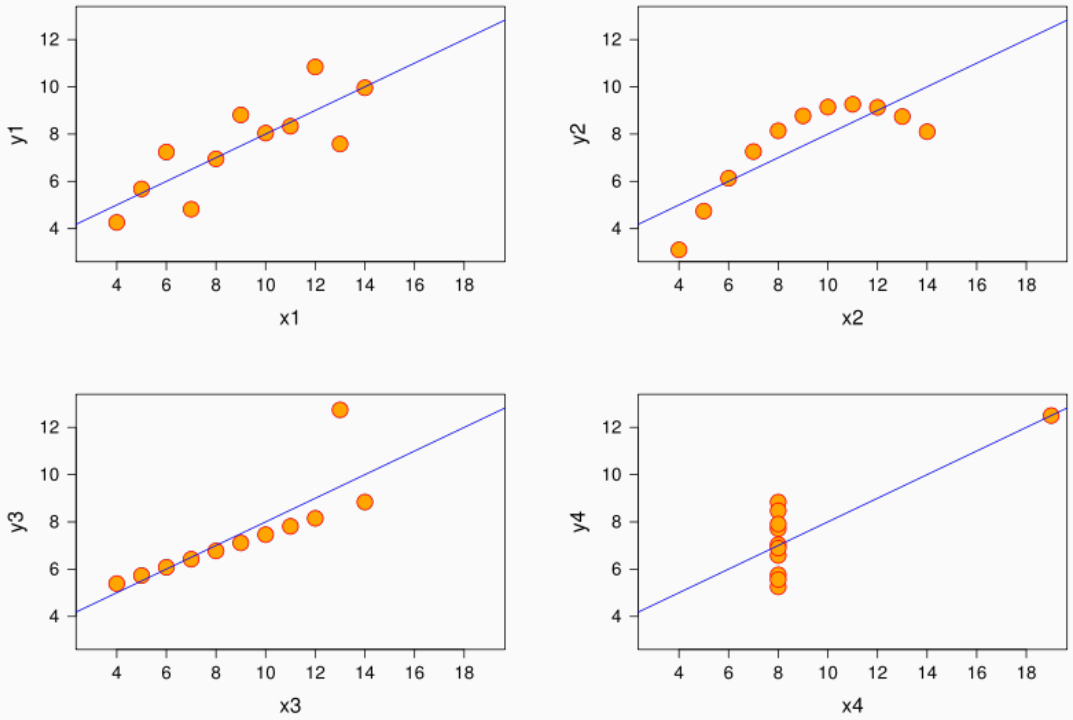
Por lo tanto,

$$F(i) > F(j) \Rightarrow \frac{\rho(i)^2}{1-\rho(i)^2} > \frac{\rho(j)^2}{1-\rho(j)^2} \Rightarrow \rho(i)^2 > \rho(j)^2\,, \qquad (8)$$

Es, decir, un ranking de los atributos en base a $F(i)$ es equivalente a un ranking en base a $Z(i)$. Bajo el supuesto $\epsilon_i \sim \mathcal{N}(0,1)$, y bajo la hipótesis $H_0 : a_i = 0$, $F(i) \sim \mathcal{F}_{1,n-2}$ permite efectuar un contraste equivalente a aquel obtenido en base a $Z(i)$.

Límites del Coeficiente de Correlación

Naturalmente, si Y es categórica, el uso del coeficiente de correlación es muy discutible. Además, éste permite detectar solo relaciones de dependencia lineal.



Información Mutua

En teoría de la información, la **información mutua (MI)** de dos variables aleatorias X e Y, con función de probabilidad conjunta $f(x, y)$ y marginales $f(x)$ e $f(y)$ respectivamente, se define como

$$I(X, Y) = \mathbb{E}_{x,y} \log \left(\frac{f(x, y)}{f(x)f(y)} \right) = \int \int f(x, y) \log \left(\frac{f(x, y)}{f(x)f(y)} \right) dxdy \quad (9)$$
$$= \mathbb{E}_y \mathbb{E}_{x|y} \log \left(\frac{f(x|y)}{f(x)} \right) = \mathbb{E}_x \mathbb{E}_{y|x} \log \left(\frac{f(y|x)}{f(y)} \right),$$

y es una medida de la dependencia entre X e Y o de la información que una variable contiene con respecto a la otra.

Es fácil mostrar que $I(X, Y) \geq 0$ y que $I(X, Y) = 0$ si y sólo si X e Y son estadísticamente independientes.

Información Mutua en Teoría de la Información

Recordando la definición de la **divergencia de Kulback-Leibler**,

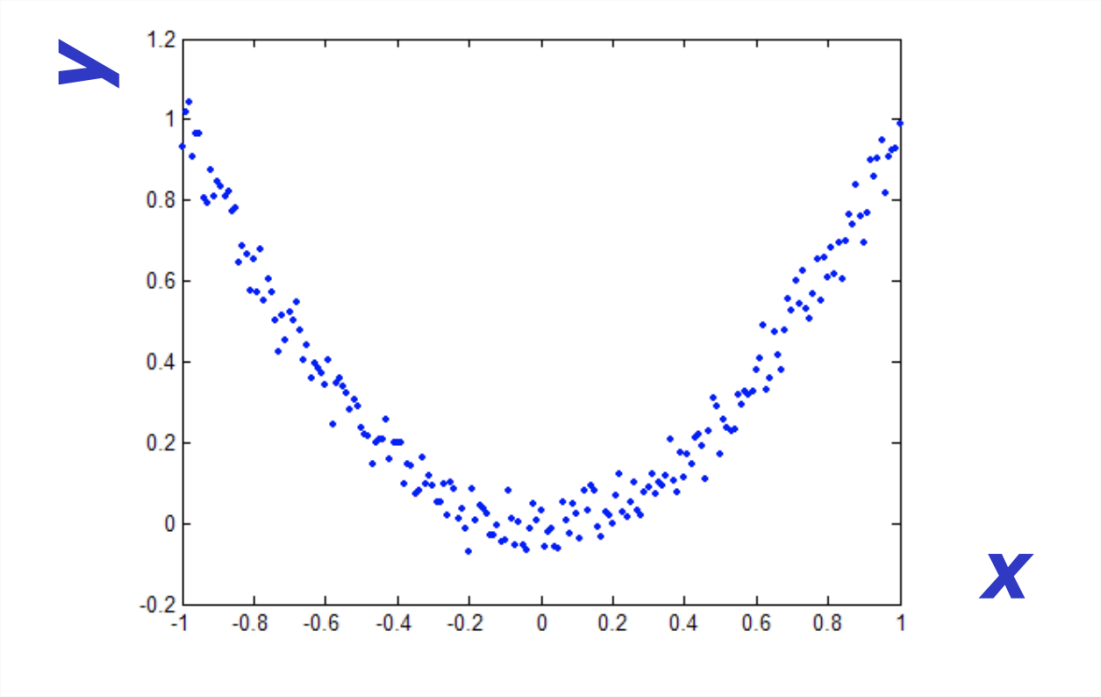
$$D_{KL}(P||Q) = \mathbb{E}_z \log\left(\frac{P(z)}{Q(z)}\right) = \int_z P(z) \log\left(\frac{P(z)}{Q(z)}\right) dz \tag{10}$$

obtenemos que la información mutua (MI) de dos variables aleatorias es simplemente la "distancia" entre la distribución que obtendríamos asumiendo variables independientes, con respecto a la función de probabilidad conjunta real.

⋆ **La MI representa entonces la pérdida de eficiencia de codificación que se produce cuando codificamos X usando $f(x)$ en vez de $f(x|y)$, ó (equivalentemente) cuando codificamos Y usando $f(y)$ en vez de $f(y|x)$.**

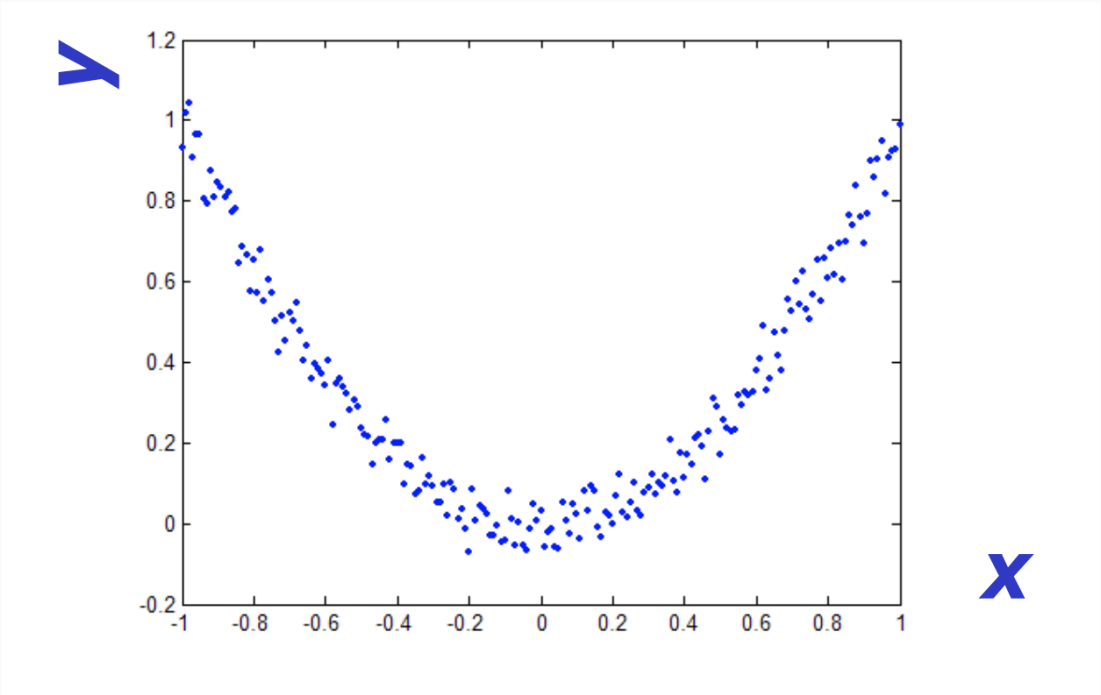
Información Mutua & No-Linealidad

A diferencia de la correlación, la información mutua permite detectar dependencias no-lineales entre dos variables. Por ejemplo, consideremos la siguiente situación en que, $X \sim U(-1, 1)$, $Y = X^2 + \epsilon$ y $Z \sim U(-1, 1)$ (Z independiente de X e Y).



Información Mutua & No-Linealidad

Usando una muestra de 100 puntos, obtenemos los siguientes resultados



	Y, Y	Y, X	Y, Z
Correlación	1	0.046	0.052
MI	2.258	1.199	0.003

Información Mutua & Selección de Características

La información mutua $I(X_i, Y)$ entre un atributo X_i y la respuesta Y es una de las medidas más estudiadas para selección de características.

Naturalmente, en la práctica no conocemos la f.d.p de las variables en cuestión. Tenemos sólo un conjunto de n ejemplos, $\{(x^{(\ell)}, y^{(\ell)})\}_\ell$, desde los cuales debemos construir un estimador de $I(X_i, Y)$.

Una posibilidad es construir estimadores de $f(x_i, y)$, $f(x_i)$ y $f(y)$, para luego utilizar el estimador

$$\hat{I}(X_i, Y) = \sum_{x,y} \hat{f}(x_i, y) \log \left(\frac{\hat{f}(x_i, y)}{\hat{f}(x_i)\hat{f}(y)} \right) . \tag{11}$$

Caso Discreto

Cuando tanto X_i como Y son variables categóricas con valores posibles $\{v_1, v_2, \ldots, v_M\}$ y $\{c_1, c_2, \ldots, c_K\}$ respectivamente, esto se traduce en

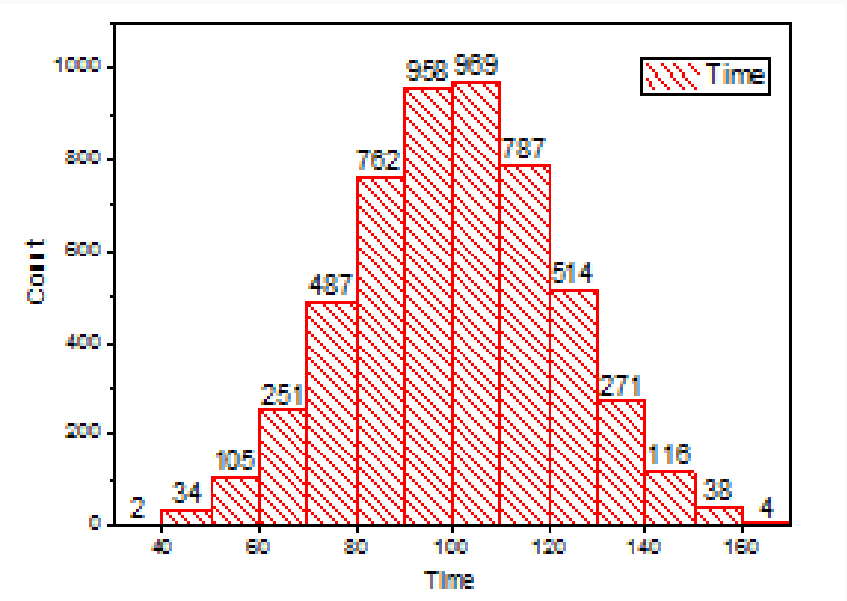
$$\hat{I}(X_i, Y) = \sum_{j,k} p(v_j, c_k) \log \left(\frac{p(v_j, c_k)}{p(v_j)p(c_k)} \right) , \tag{12}$$

donde $p(v_j, c_k)$ es la fracción de datos donde $X_i = v_j$ e $Y = c_k$, $p(v_j)$ es la fracción de datos donde $X_i = v_j$ y $p(c_k)$ es la fracción de datos tales que $Y = c_k$.

Caso Continuo

Cuando una de las variables es continua (típicamente X_i), la estimación de $f(x_i)$ requiere de un método de agregación.

El método más simple consiste en usar un histograma. Este método resulta bastante sensible al número de bins utilizados y/o al criterio para definir el largo de los bins [3].

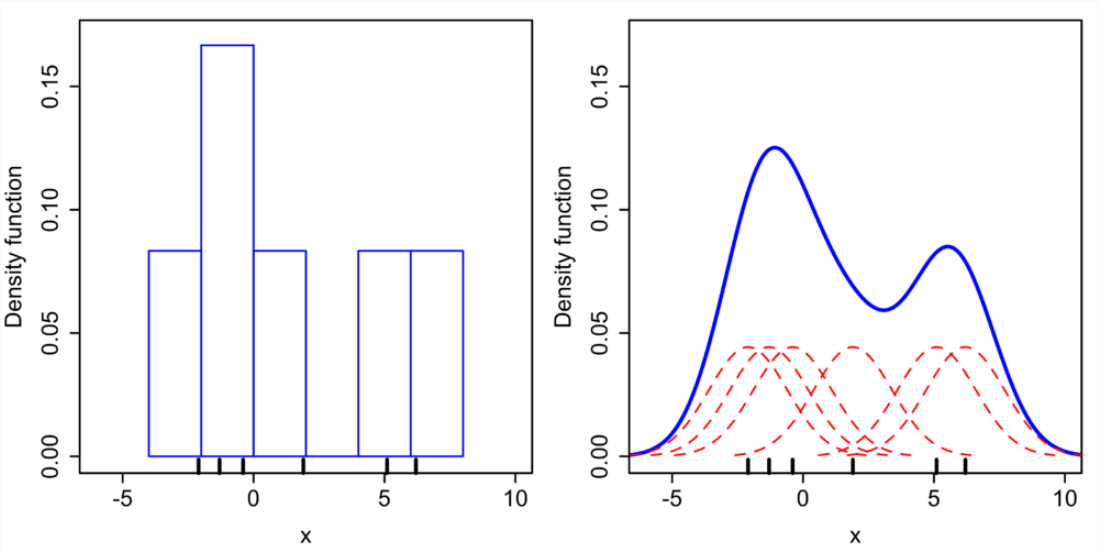


[3] Ver Kraskov et al. *Estimating Mutual Information*, Physical Review E, 2004.

Estimadores Basados Superposición de Kernels

Una alternativa más sofisticada consiste en usar estimadores basados en kernel (kernel density estimators)[4], una generalización del clásico método denominado *ventanas de Parzen*.

Este método puede generar una estimación más suave que aquella basada en un histograma, sobre todo en muestras pequeñas.



[4]Ver Moon et al. *Estimation of mutual information using kernel density estimators*, Physical Review E, 1995.

Estimadores Basados Superposición de Kernels

Idea (Parzen): La probabilidad concentrada en un punto x se puede estimar considerando un pequeño volumen V en torno a punto x y asumiendo que la f.d.p es aproximadamente constante en ese volumen

$$\int_V p(x)dx \approx |V|p(x)\,, \tag{13}$$

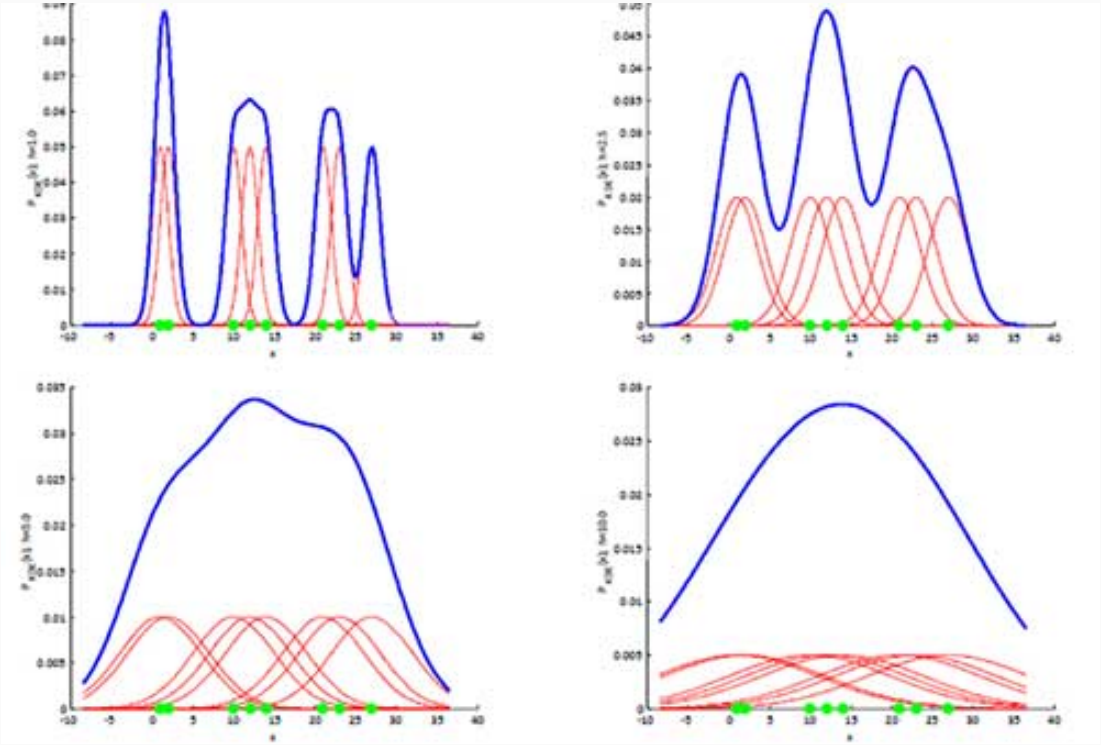
Además, si tenemos una muestra, podemos contar la fracción de datos k_n/n que caen dentro del volumen V, en modo de obtener

$$|V|p(x) \approx \frac{k_n}{n} \;\Rightarrow p(x) \approx \frac{k_n}{n|V|} = \frac{1}{n}\sum_{\ell} \frac{I(x_i - x)}{|V|}\,, \tag{14}$$

donde $I(x_i - x)$ verifica si el dato x_i está en el volúmen en torno a x.

Efecto del Bandwidth

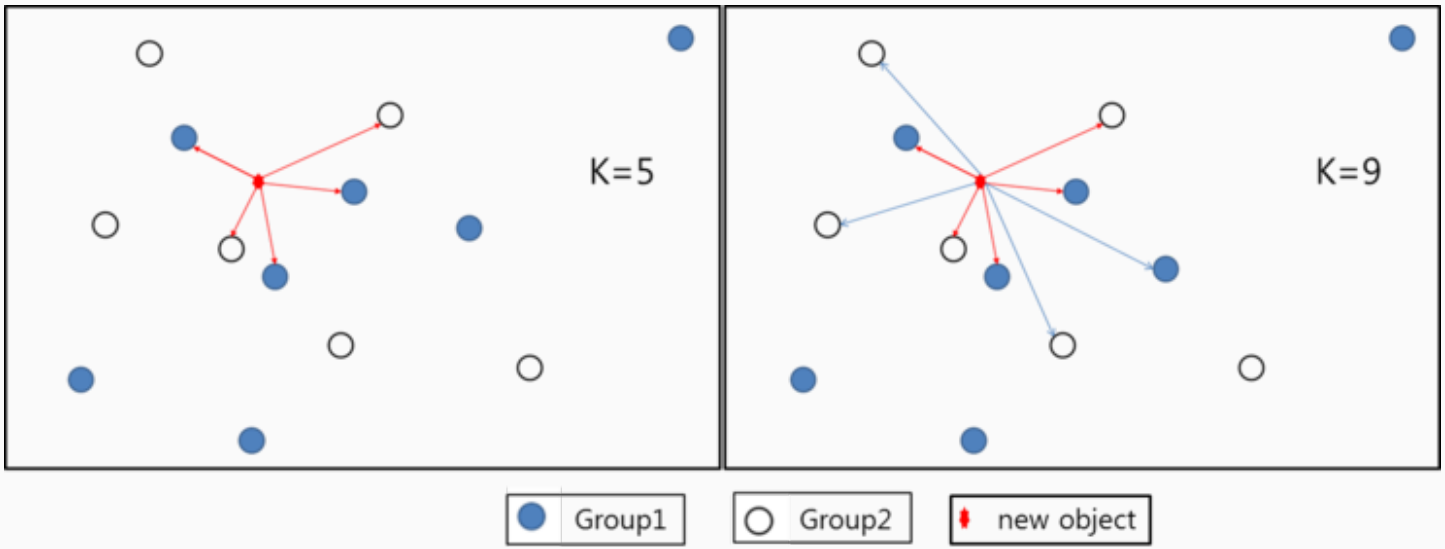
Sin embargo, los resultados de este método pueden resultar muy sensible a la elección de *bandwidth* (tamaño del volumen), que define el grado de "influencia" de cada kernel.



Uno de los métodos más efectivos para elegir este parámetro consiste en maximizar la verosimilitud sobre un conjunto de validación (distinto al conjunto de entrenamiento).

Estimación Basada en KNN

Los métodos modernos para estimar la información mutua se basan en el cálculo de los vecinos más cercanos de los datos de entrenamiento. Naturalmente, esto requiere la definición de una métrica (e.g. Euclidiana).



Estimación Basada en KNN

Preliminares: Notemos primero que la MI entre dos variables X, Y se puede calcular como

$$I(X,Y) = H(X) + H(Y) - H(X,Y)\,, \tag{15}$$

donde

$$H(Z) = -\mathbb{E}_z \log(f(z)) = \int f(z)\log(f(z))dx\,, \tag{16}$$

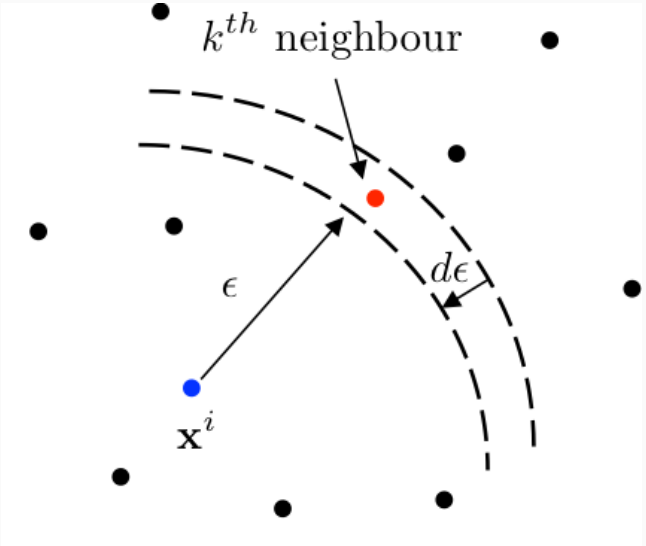
es la entropía de Z.

Por lo tanto, si podemos estimar la entropía de una variable (y de un par $Z = (X,Y)$), podemos estimar la MI. Notemos también que si tenemos un conjunto de n datos $\{z^{(\ell)}\}_\ell$, podemos estimar $H(Z)$ como

$$\hat{H}(Z) = -\frac{1}{n}\sum_\ell \log(f(z^{(\ell)})) = \frac{1}{n}\sum_\ell \log\left(\frac{1}{f(z^{(\ell)})}\right)\,. \tag{17}$$

Estimación Basada en KNN

Idea (Kraskov, 2003)[5]: Estimar $f(z)$ usando la distancia ϵ al k-esimo vecino más cercano de z en la muestra.



Sea $p(\epsilon)$ la distribución de probabilidad de ϵ y consideremos un diferencial de $d\epsilon$ en torno al k-ésimo vecino más cercano de un punto $z \sim f(z)$ (aleatorio porque z es aleatorio).

[5] Detalles en Kraskov et al. *Estimating Mutual Information*, Physical Review E, 2004.

Estimación Basada en KNN

Por un lado, tenemos que la probabilidad de que el k-ésimo vecino más cercano de z esté a una distancia entre ϵ y $\epsilon + d\epsilon$ se puede aproximar como $p(\epsilon)d\epsilon$. Pero para que ello ocurra, exactamente $k-1$ puntos deben estar a una distancia menor y $n-k$ a una distancia mayor (sino, ϵ no sería la distancia al k-ésimo vecino más cercano). Es decir,

$$p(\epsilon)d\epsilon = \binom{n-1}{1}\binom{n-2}{k-1}\frac{dP_z(\epsilon)}{d\epsilon}d\epsilon\,(P_z(\epsilon))^{k-1}\,(1-P_\ell(\epsilon))^{n-k-1}\,, \tag{18}$$

donde $P_z(\epsilon)$ es la probabilidad de observar un punto en una bola de radio ϵ en torno a z,

$$P_z(\epsilon) = \int_{B((z,\epsilon)} f(\tilde{z})d\tilde{z} \tag{19}$$

Estimación Basada en KNN

Si consideramos diferentes z, distribuidos de acuerdo a $f(z)$, observaremos diferentes distancias al k-ésimo vecino más cercano, distribuidas de acuerdo a $p(\epsilon)$. Por lo tanto,

$$\mathbb{E}_z \log P_z(\epsilon) = \int \log P_z(\epsilon) p(\epsilon) d\epsilon \tag{20}$$

Usando la ecuación (18) tenemos que

$$\begin{aligned}\mathbb{E}_z \log P_z(\epsilon) &= \int \log P_z(\epsilon) \binom{n-1}{1} \binom{n-2}{k-1} \frac{dP_z(\epsilon)}{d\epsilon} d\epsilon \left(P_z(\epsilon)\right)^{k-1} \left(1 - P_\ell(\epsilon)\right)^{n-k-1} \\ &= \psi(k) - \psi(n)\,, \end{aligned} \tag{21}$$

donde $\psi(\cdot)$ es la función digamma.

Estimación Basada en KNN

Consideremos ahora una aproximación alternativa de $P_z(\epsilon)$

$$P_z(\epsilon) = \int_{B((z,\epsilon)} f(\tilde{z}) d\tilde{z} \approx |B(z,\epsilon)| f(z) \tag{22}$$

Obtenemos que

$$\log P_z(\epsilon) = \approx \log |B(z,\epsilon)| + \log f(z) \tag{23}$$

$$\mathbb{E}_z \log P_z(\epsilon) \approx \mathbb{E}_z \left(\log |B(z,\epsilon)| + \log f(z)\right)$$

$$\psi(k) - \psi(n) \approx \mathbb{E}_z \left(\log |B(z,\epsilon)|\right) + \mathbb{E}_z \left(\log f(z)\right)$$

$$-\mathbb{E}_z \left(\log f(z)\right) \approx \psi(N) - \psi(k) + \mathbb{E}_z \left(\log |B(z,\epsilon)|\right)$$

$$H(Z) \approx \psi(N) - \psi(k) + \frac{1}{n} \sum_{\ell} C\, \epsilon_\ell \tag{24}$$

donde ϵ_ℓ es la distancia de un punto de la muestra $(z^{(\ell)})$ a su k-ésimo vecino más cercano y C es una constante que depende de la norma considerada para aproximar el tamaño de la bola.

Estimación Basada en KNN

Principales Ventajas

- En general se obtienen estimaciones mucho más precisas y exactas que usando métodos basados en kernel o histogramas.
- Los métodos se extienden naturalmente a múltiples dimensiones de modo que es posible estimar eficientemente la MI de un conjunto de variables con respecto a la variable a predecir.

Métodos tipo Wrapper

Generalidades

Un método tipo wrapper busca encontrar el subconjunto de atributos que son, en conjunto, óptimos para predecir la variable de salida y.

Dos componentes fundamentales:

1. Un algoritmo de búsqueda que explora el espacio de todos los 2^d posibles subconjuntos del conjunto original de atributos.
2. Una función que evalúa la relevancia de un determinado subconjunto re-entrenando y evaluando un determinado modelo para y.

Limitaciones:

1. Como el problema de búsqueda es NP-hard, se utilizan con frecuencia heurísticas sin garantías explícitas.
2. Aún usando heurísticas, la función de relevancia requiere re-entrenamiento, generando un gran costo computacional.

Forward Stepwise Selection

Heurística *greedy* que parte con un conjunto vacío de atributos [6] $\mathcal{F}$ e itera agregando un atributo a la vez para maximizar la función objetivo.

Algorithm 1: Forward Selection

Initialize $\mathcal{F} = \emptyset$
do
  for $i = 1, \ldots, d$ **do**
    if $i \notin \mathcal{F}$ **then**
      $\mathcal{F}_i = \mathcal{F} \cup \{i\}$
      Evaluar el conjunto de atributos $\mathcal{F}_i$ para obtener S_i
  $i^* = \arg\max_{i \notin \mathcal{F}} S_i$
  Set $\mathcal{F} \leftarrow \mathcal{F} \cup \{i^*\}$
while *Convergence criterion*;
Output: $\mathcal{F}$

[6] Es decir, una matriz de diseño X con sólo una columna constante

Backward Stepwise Selection

Heurística *greedy* que parte con un conjunto completo de atributos $\mathcal{F}$ e itera removiendo un atributo a la vez, buscando el menor impacto negativo sobre la función objetivo.

Algorithm 2: Backward Selection

Initialize $\mathcal{F} = \{1, 2, \ldots, d\}$
do
  for $i = 1, \ldots, d$ **do**
    if $i \in \mathcal{F}$ **then**
      $\mathcal{F}_i = \mathcal{F} - \{i\}$
      Evaluar el conjunto de atributos $\mathcal{F}_i$ para obtener S_i
  $i^* = \arg\max_{i \in \mathcal{F}} S_i$
  Set $\mathcal{F} \leftarrow \mathcal{F} - \{i^*\}$
while *Convergence criterion*;
Output: $\mathcal{F}$

Sequential Floating Forward Selection

FSS con backtracking, i.e., después de expandir $\mathcal{F}$ se intenta reducir $\mathcal{F}$ buscando no empeorar demasiado la función objetivo

Algorithm 3: SFFS

```
1 Inicializar F = ∅
2 do
3 |   Ejecutar un paso de FSS para expandir F
4 |   do
5 |   |   Ejecutar un paso de BSS para reducir F
6 |   while Switching criterion;
7 while Convergence criterion;
8 Output: F
```

Otros Métodos de Búsqueda

Por supuesto, en la literatura se han investigado muchos otros métodos

- (Tu et al. 2007[7]) propone el uso de *particle swarm optimization*.
- (Nakariyakul & Casasent, 2009[8]) proponen un método denominado *Plus-L-Minus-r* que agrega L variables simultáneamente en un paso FSS y elimina r en un paso BSS.
- (Alexadridis et al. 2005[9]) propone el uso de estrategias evolutivas.

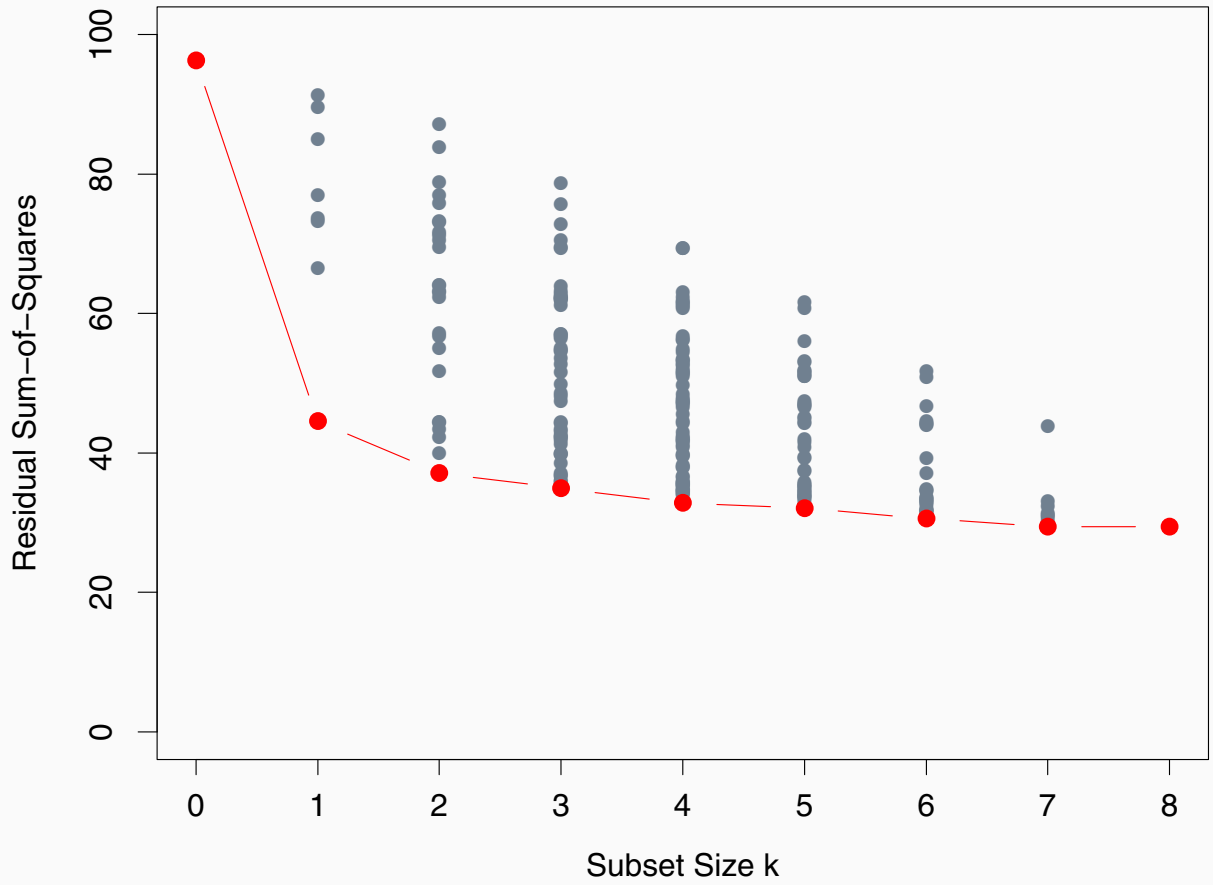
[7] Feature selection using PSO-SVM.
[8] *An improvement on floating search algorithms for feature subset selection.*
[9] *A two-stage evolutionary algorithm for variable selection in the development of RBF neural network models.*

Funciones de Relevancia

- Un algoritmo tipo wrapper usa en general una función objetivo que depende explícitamente del método de aprendizaje que se quiere optimizar.
- Por defecto, el procedimiento consiste en entrenar explícitamente el modelo con el conjunto de atributos propuesto y medir el error de predicción obtenido.
- Como el error de entrenamiento decrecerá siempre aumentando el número de atributos es preferible utilizar otros estimadores del error de predicción.
- En algunos casos, es posible usar criterios estadísticos que evalúen si una expansión/reducción de $\mathcal{F}$ aumenta/disminuye significativamente la capacidad predictiva del modelo.

Funciones de Relevancia

En general, el error de entrenamiento disminuye usando un subconjunto de atributos más grande.



Funciones de Relevancia

En la práctica, la efectividad de un sub-conjunto de atributos se mide usando un predictor de la efectividad del modelo sobre datos futuros.

Las opciones típicas son

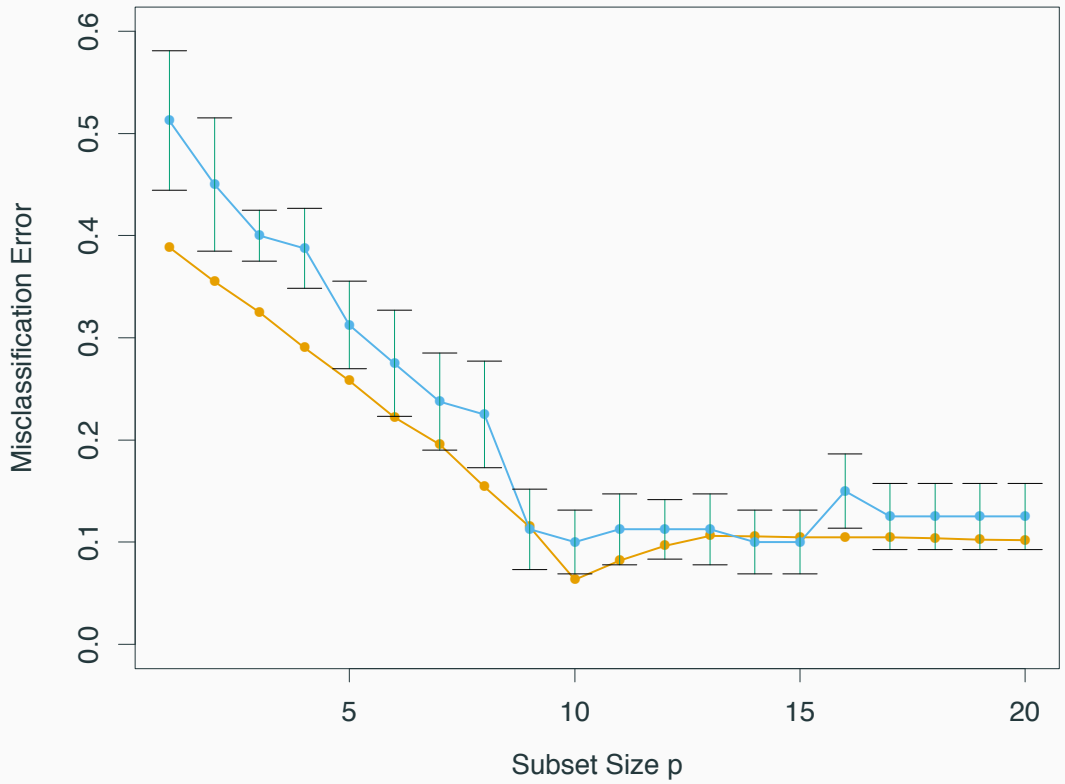
- Usar un conjunto de validación

- Usar validación cruzada (cross-validation)

1	2	3	4	5
Train	Train	Validation	Train	Train

Funciones de Relevancia

En general, el error de predicción (curva naranja en la figura) no necesariamente disminuye usando un subconjunto de atributos más grande. Si el estimador del error de predicción es bueno (validación cruzada, curva azul, en el dibujo) su comportamiento permitirá anticipar este comportamiento en el conjunto de test.



FSS y BSS en el Modelo Lineal

En el caso del modelo lineal de regresión, FSS, BSS y sus variantes suelen usar en combinación con criterios estadísticos.

Ejemplo: Consideremos un modelo M_0 construido usando p_0 atributos y otro modelo M_1 construido usando los mismos p_0 atributos más $p_1 - p_0$ predictores adicionales. Si denotamos por SSE_i la suma de errores de entrenamiento correspondientes al modelo M_i, asumimos que el modelo lineal es correcto con ruido $\epsilon \sim \mathcal{N}(0, \sigma^2)$ y bajo la hipótesis de que el modelo M_0 es correcto, es fácil demostrar que

$$F = \frac{\frac{SSE_0 - SSE_1}{p_1 - p_0}}{\frac{SSE_1}{n - p_1 - 1}} \sim \mathcal{F}_{p_1 - p_0, n - p_1 - 1} \tag{25}$$

FSS y BSS en el Modelo Lineal

- El resultado anterior se puede utilizar para decidir si agregar/eliminar un conjunto de predictores al modelo usando un p-valor. Esto no elimina la necesidad de re-entrenar el modelo.
- El resultado también permite implementar un criterio de término/convergencia basado en un p-valor.
- En el caso de problemas con muchos atributos es posible también usar los criterios de filtrado que hemos revisado para elegir las variables a agregar/eliminar. Por ejemplo el Z-score.

Métodos Embedidos

Generalidades

Un método embedido busca encontrar el subconjunto de atributos que son, en conjunto, óptimos para predecir la variable de salida y. Sin embargo, para esto utiliza una función objetivo propia, que no implica re-entrenar el modelo.

Dos componentes fundamentales:

1. Un algoritmo de búsqueda que explora el espacio de todos los 2^d posibles subconjuntos del conjunto original de atributos.
2. Una función propia que evalúa la relevancia de un determinado subconjunto.

Maximum Relevancy Minimum Redundancy (MRMR)

Propuesto en (Peng et al. 2005[10]) adopta como algoritmo de búsqueda FSS. Sin embargo en cada iteración, se busca resolver el siguiente problema de optimización

$$\max_{i \notin \mathcal{F}} I(Y, X_i) - \beta \sum_{j \in \mathcal{F}} I(X_i, X_j) \tag{26}$$

El término $I(Y, X_i)$ evalúa la *relevancia* del atributo X_i.

El término $I(X_i, X_j)$ mide la redundancia del atributo X_i con respecto a los atributos X_j ya existentes en el conjunto de características seleccionadas.

Con $\beta = 0$, el método se denomina Mutual Information Maximisation.

[10] Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy.

Joint Mutual Information (JMI)

Propuesto en (Bennasar, 2015[11]) adopta como algoritmo de búsqueda FSS, pero en cada iteración, se busca resolver el siguiente problema de optimización

$$\max_{i \notin \mathcal{F}} I(Y, X_i) - \left(\beta \sum_{j \in \mathcal{F}} I(X_i, X_j) - \alpha \sum_{j \in \mathcal{F}} I(X_i, X_j | Y) \right) \tag{27}$$

Si Y representa una clase, el nuevo término $I(X_i, X_j|Y)$ intenta determinar qué tan independiente es el atributo X_i respecto de un atributo ya seleccionado X_j en cada una de las clases del problema. Es posible que X_i, X_j sean marginalmente independientes, pero que se configure un patrón al estudiar el comportamiento de X_i, X_j en cada clase del problema.

[11] Feature selection using Joint Mutual Information Maximisation