Selección de Atributos

Aprendizaje Automático INF-398 II-2021

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.

Selección de Atributos:

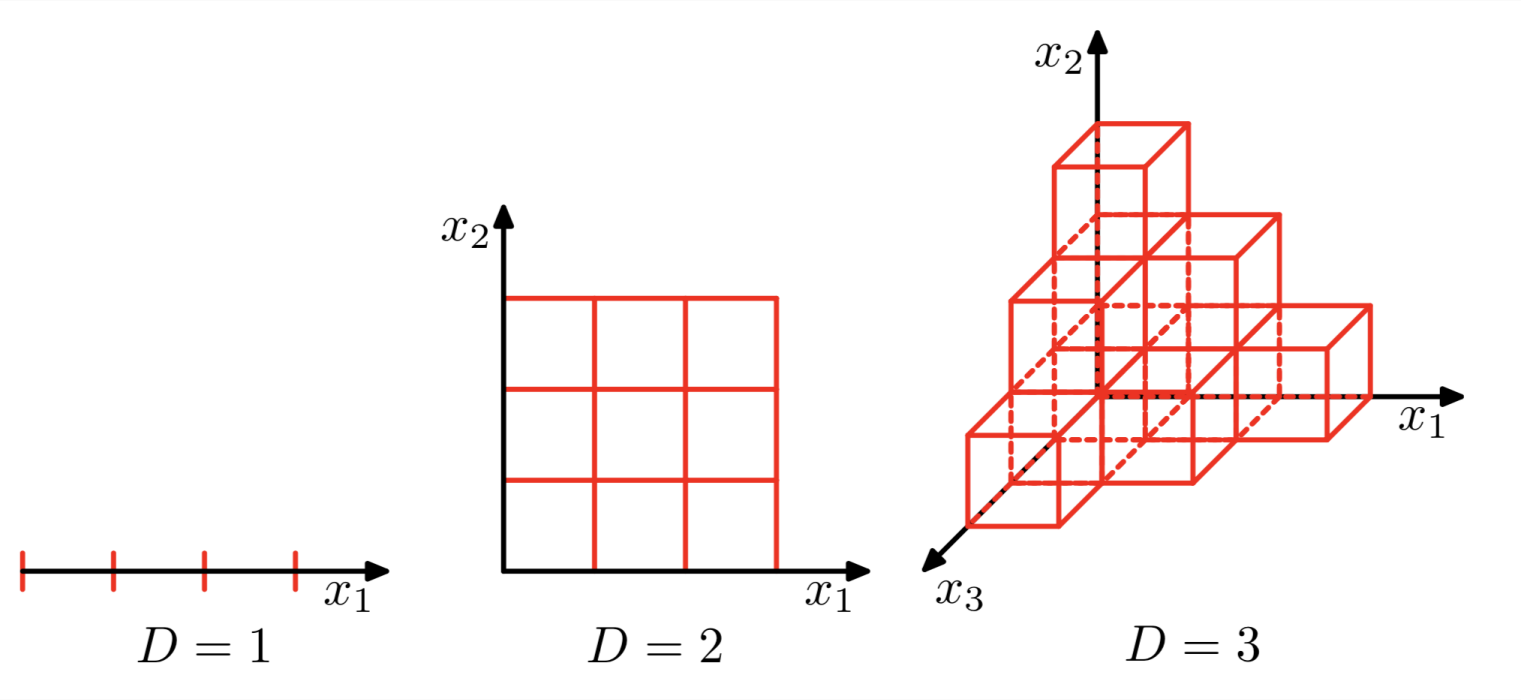
1. Encontrar las variables verdaderamente relevantes para la tarea (e.g. predecir/explicar Y).
2. Encontrar el mejor subconjunto de atributos de tamaño $B \ll d$ para la tarea (e.g. predecir/explicar Y).

Beneficios de la Selección de Atributos

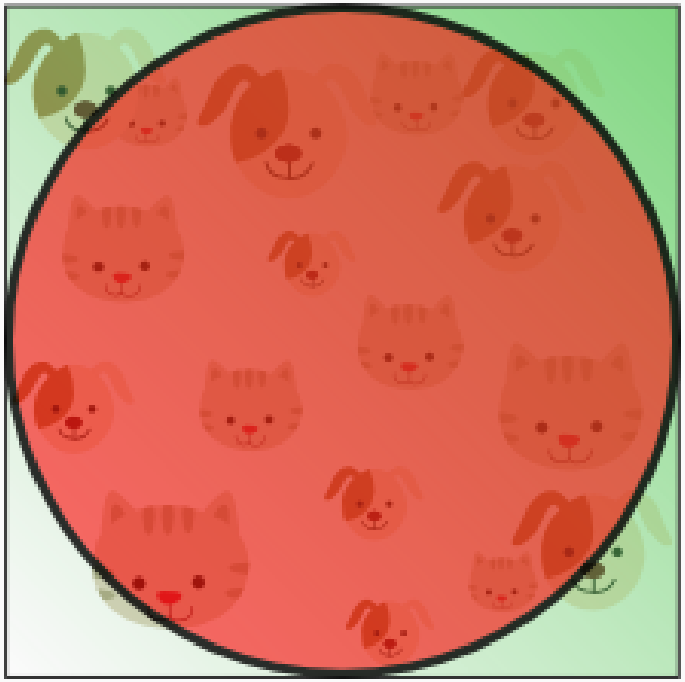
En cualquier caso el resultado tiende a ser una representación "más compacta" de los datos, que puede ser importante por muchos motivos:

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.

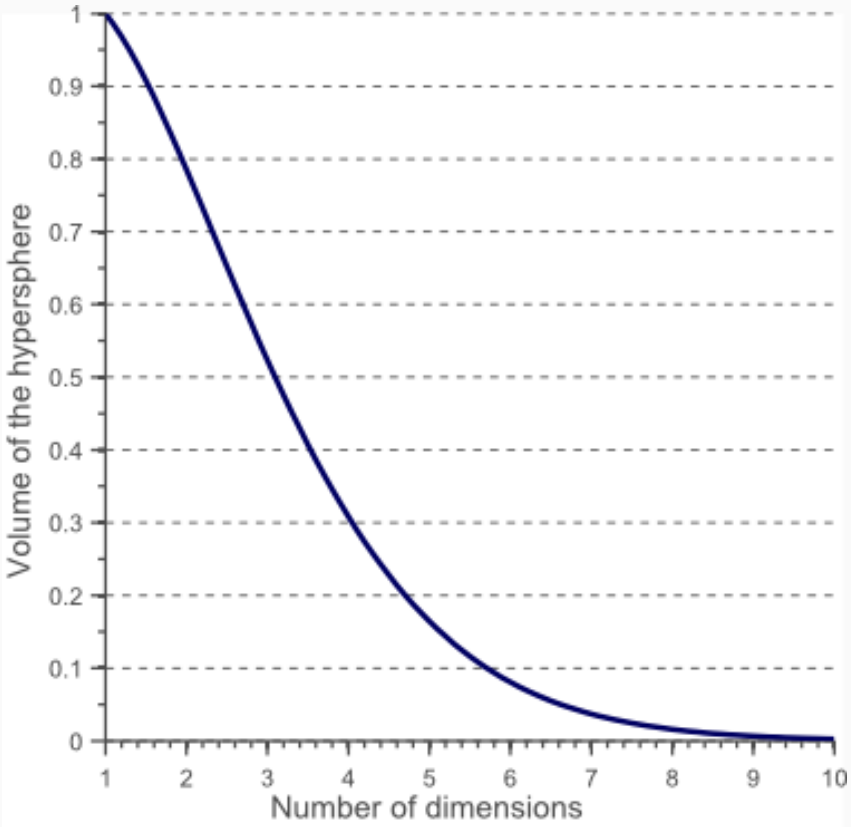
Selección de Atributos & Maldición de la Dimensionalidad



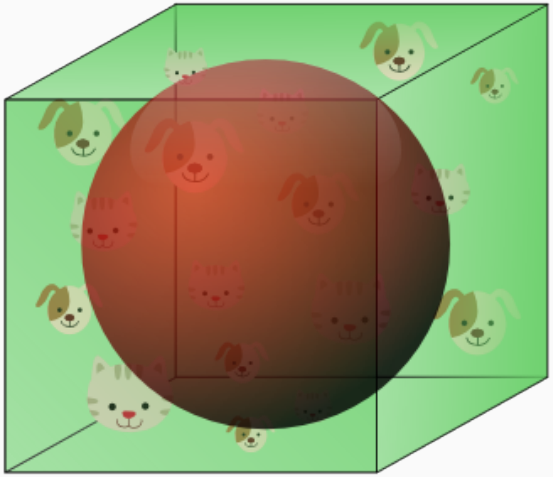
Selección de Atributos & Maldición de la Dimensionalidad

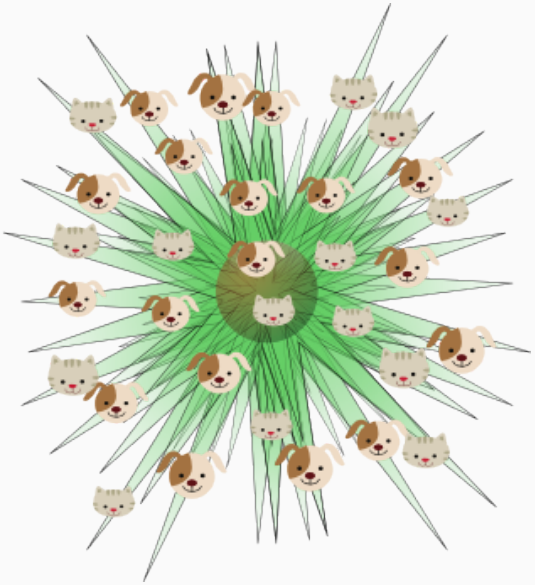


Selección de Atributos & Maldición de la Dimensionalidad



Selección de Atributos & Maldición de la Dimensionalidad



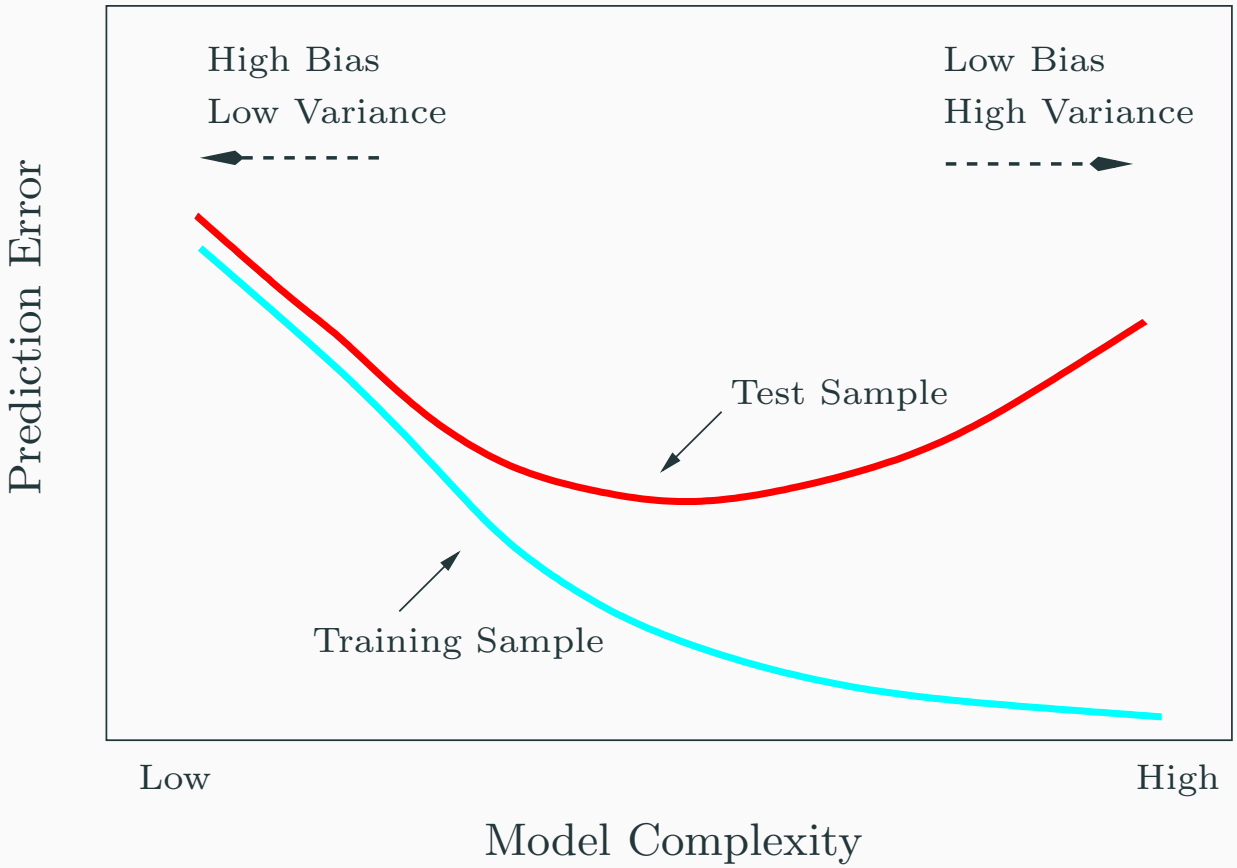


Propósitos de la Selección de Atributos

Encontrar una 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.

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 tipo Wrapper (wrapper methods): Métodos que utilizan el modelo o técnica de aprendizaje como caja negra para evaluar el efecto de una variable.
2. Métodos Embedidos (embedded methods): Métodos diseñados específicamente para un tipo de modelo o técnica de aprendizaje.
3. Métodos de Filtrado (filter methods): Métodos independientes del modelo o técnica de aprendizaje.

Naturalmente también hay híbridos que por ejemplo incluyen criterios de filtrado en métodos tipo Wrapper.

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

Métodos tipo Wrapper

Propósitos de la Selección de Atributos

Selección de Atributos:

1. Encontrar las variables verdaderamente relevantes para la tarea (e.g. predecir/explicar Y).
2. Encontrar el mejor subconjunto de atributos de tamaño $B \ll d$ para la tarea (e.g. predecir/explicar Y).

Si tenemos d atributos candidatos, existen 2^d subconjuntos posibles.

Resolver el problema exhaustivamente implicaría entrenar 2^d modelos y estimar su error de predicción.

El problema de búsqueda es NP-hard.

Generalidades

1. Un método tipo wrapper ataca el problema utilizando algoritmos que reducen la complejidad de explorar el espacio de búsqueda.
2. La función que evalúa la relevancia de un determinado subconjunto de atributos para el problema se basa explícitamente en re-entrenar y evaluar el desempeño del modelo.

Generalidades

Ventajas/Limitaciones:

1. Se utilizan casi siempre heurísticas de búsqueda sin garantías explícitas de optimalidad.
2. Respecto a los métodos embedidos son técnicas de uso más general, pudiendo acomodar cualquier técnica.
3. Respecto a muchos métodos de filtrado, permiten decidir fácilmente cuántas variables incluir.
4. Si bien de manera limitada, consideran el efecto conjunto que pueden tener las variables (a diferencia de muchos métodos de filtrado).
5. Como la función de relevancia requiere re-entrenamiento, exhiben un gran costo computacional.

Forward Stepwise Selection

Heurística *greedy* que parte con un conjunto vacío de atributos $\mathcal{F}$ e itera agregando un atributo a $\mathcal{F}$ hasta que $|\mathcal{F}| = K$ o la mejora de la función de relevancia $< \epsilon$.

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: $\mathcal{F}$

Backward Stepwise Selection

Heurística *greedy* que parte con un conjunto completo de atributos $\mathcal{F}$ e itera removiendo un atributo a la vez, hasta que $|\mathcal{F} = K$ o el empeoramiento de la función de relevancia $> \epsilon$.

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
```

FSS y BSS en el Modelo Lineal

Para el modelo lineal de regresión $f(\boldsymbol{x}; \boldsymbol{\beta}) = \sum_i \beta_i x_i + \beta_0$, es posible determinar si un cambio de $\mathcal{F}$ durante FSS ó BSS cambia significativamente el error del modelo.

Sea M_0 un modelo lineal con p_0 atributos y M_1 uno con los mismos p_0 atributos más $p_1 - p_0$ predictores adicionales. Si denotamos por SSE_i la suma de errores de entrenamiento de M_i, y asumimos que M_0 es correcto,

$$F = \frac{\frac{SSE_0 - SSE_1}{p_1 - p_0}}{\frac{SSE_1}{n - p_1 - 1}} \sim \mathcal{F}_{p_1 - p_0, n - p_1 - 1}\,, \tag{2}$$

donde $\mathcal{F}_{p_1 - p_0, n - p_1 - 1}$ denota la distribución de Fisher con $p_1 - p_0$ grados de libertad en el numerador y $n - p_1 - 1$ grados de libertad en el denominador.

FSS y BSS en el Modelo Lineal

- El resultado anterior se puede utilizar para decidir si agregar/eliminar un conjunto de predictores 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.

Atención: Decidir que una variable X_i tiene un efecto estadísticamente significativo en la respuesta no implica que su "peso" o relevancia en el modelo sea "grande" o que sea "más grande" que la de otras variables. Sólo significa que $\beta_i \neq 0$.

Métodos Avanzados de Búsqueda

Por supuesto, en la literatura se han investigado muchos otras estrategias de búsqueda:

- (Tu et al. 2007[3]) propone el uso de *particle swarm optimization*.
- (Nakariyakul & Casasent, 2009[4]) 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[5]) propone el uso de estrategias evolutivas.

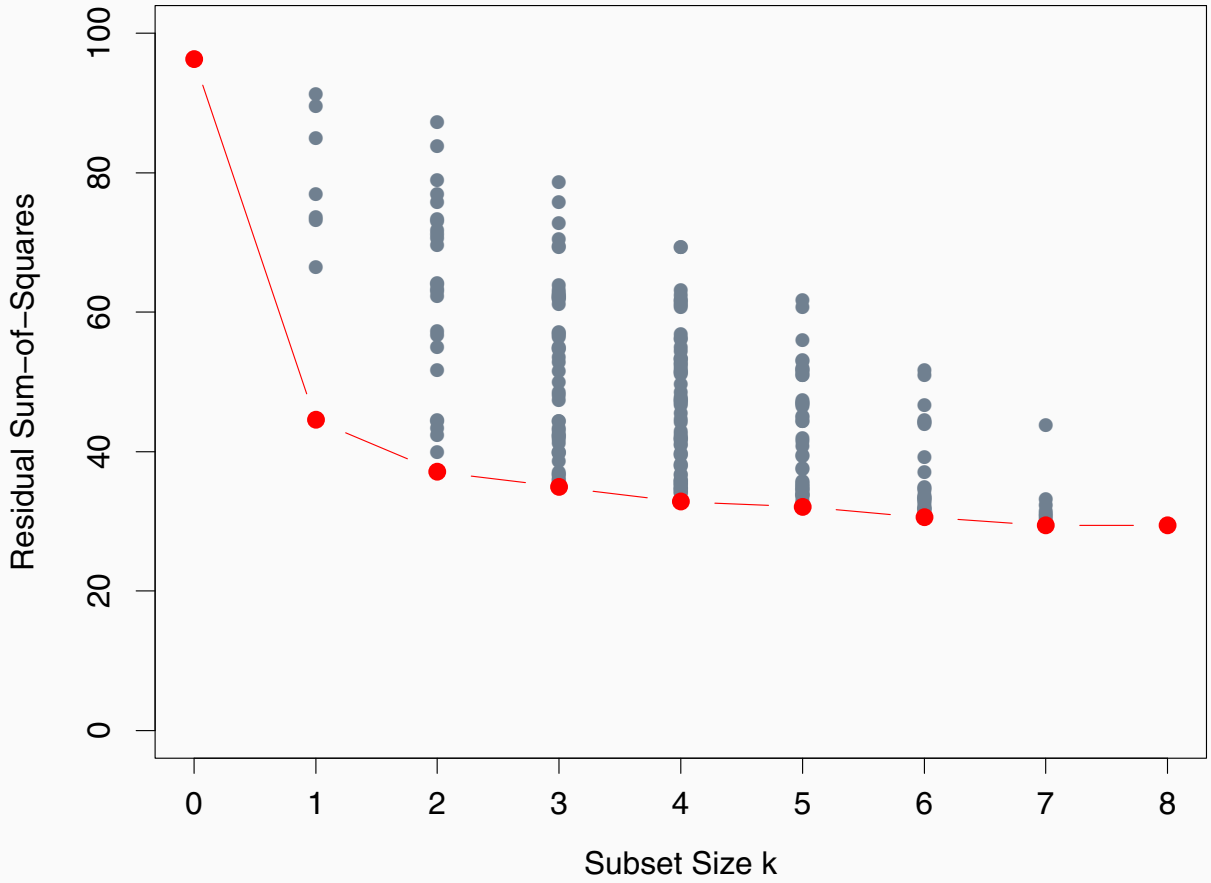
[3] Feature selection using PSO-SVM.

[4] *An improvement on floating search algorithms for feature subset selection.*

[5] *A two-stage evolutionary algorithm for variable selection in the development of RBF neural network models.*

Función de Relevancia

Por defecto, el procedimiento para evaluar un conjunto de atributos consiste en entrenar el modelo con esa representación y estimar el error obtenido. En general, el error de entrenamiento disminuye usando un subconjunto de atributos más grande.



Función de Relevancia

Es preferible utilizar otros estimadores del error de predicción.
Las opciones típicas son:

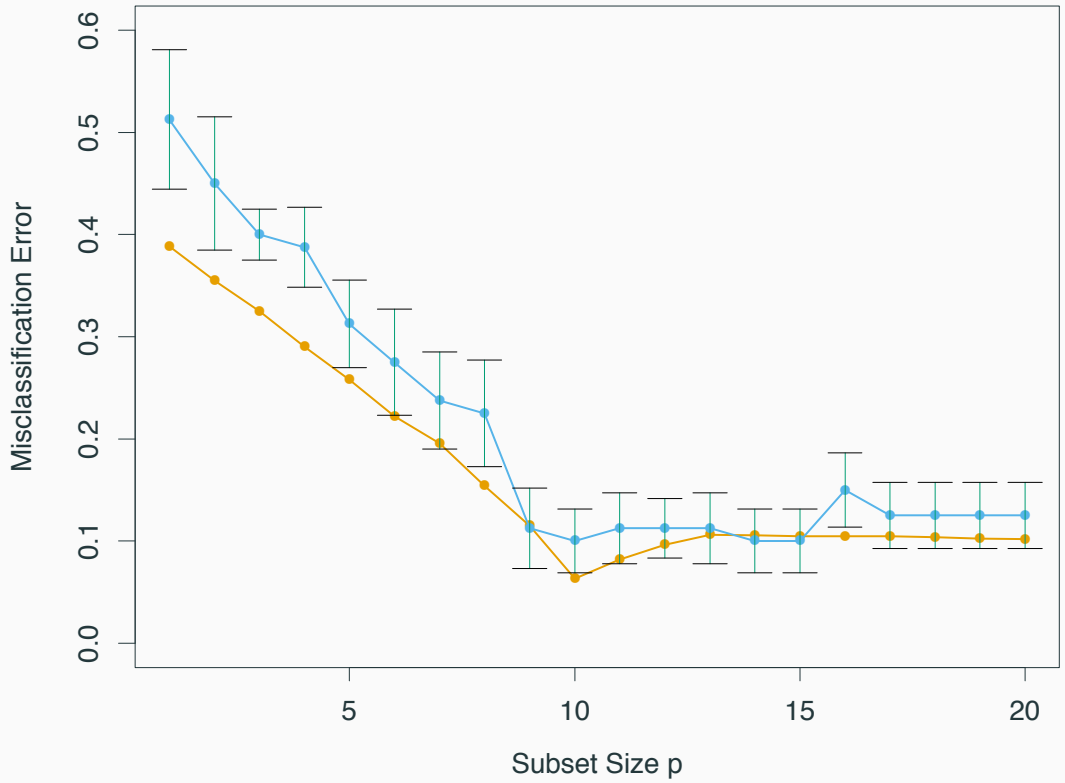
- Usar un conjunto de validación

- Usar validación cruzada (cross-validation)

1	2	3	4	5
Train	Train	Validation	Train	Train

Funciones de Relevancia

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.



Métodos Embedidos

Generalidades

Un método embedido busca encontrar el subconjunto de atributos que son, en conjunto, óptimos para un modelo específico, explotando su forma o propiedades.

1. Son menos universales, lo que por ejemplo dificulta su implementación y adopción en librerías.
2. Al combinar selección de atributos con entrenamiento suelen ser mucho más eficientes que los métodos tipo wrapper.
3. Conocer la forma o propiedades del modelo podría permitir hacer una mejor selección o al menos facilitar el análisis teórico (mejores garantías).

Penalties

Para modelos $f(\boldsymbol{x}; \boldsymbol{w})$ cuya función de decisión es de la forma

$$f(\boldsymbol{x}) = \sum\nolimits_i w_i x_i + b\,, \tag{3}$$

la forma de selección embedida más utilizada consiste en incluir un penalty ($\sim$ regularizador) en la f.o. que prefiera soluciones dispersas de $\boldsymbol{w} = (w_1, \ldots, w_d)$.

$$J(\boldsymbol{w}) = \sum_{\ell=1}^{n} L(f(x^{(\ell)}), y^{(\ell)})) + P(\boldsymbol{w})\,, \tag{4}$$

Para nosotros esto aplica al perceptrón, el regresor logístico y el modelo lineal de regresión.

Penalties

$$J(\boldsymbol{w}) = \sum_{\ell=1}^{n} L(f(x^{(\ell)}), y^{(\ell)})) + P(\boldsymbol{w}), \tag{5}$$

Ejemplos:

1. Lasso: $P(\boldsymbol{w}) = \lambda \sum_i |w_i|$.
2. Elastic Net: $P(\boldsymbol{w}) = \lambda_1 \sum_i |w_i| + \lambda_1 \sum_i w_i^2$.
3. Group Lasso: $P(\boldsymbol{w}) = \lambda \sum_i |w_i| + \sum_{g \in G} \sum_{ij} |w_i - w_j|$, $G \subset [d]^2$.
4. Bridge: $P(\boldsymbol{w}) = \lambda \sum_i |w_i|^p$ con $p < 1$.
5. Adaptive Lasso: $P(\boldsymbol{w}) = \sum_i \lambda_i |w_i|$ con $\lambda_i \sim N(0, \hat{w})$[6].
6. Automatic Relevance Determination.

Dependiendo de la complejidad de $P(\boldsymbol{w})$ y/o del modelo $f(\boldsymbol{x}; \boldsymbol{w})$ puede requerirse una modificación importante del algoritmo de entrenamiento.

[6] $\hat{w}$ es típicamente la solución no regularizada.

Penalties

Por ejemplo, para resolver bridge:

$$J(\boldsymbol{w}) = \sum_{\ell=1}^{n} L(f(x^{(\ell)}), y^{(\ell)})) + \lambda \sum_{i} |w_i|^p \ (p < 1)\,, \tag{6}$$

se ha propuesto utilizar una aproximación de segundo orden del penalty

$$|w_i|^p = |w_{0i}|^p + \frac{p}{2}|w_{i0}|^{p-2}(w_i^2 - w_{0i}^2)\,, \tag{7}$$

con w_{0i} una estimación inicial del valor del coeficiente.

La aproximación puede tratarse como una regularización L_2. En (Park 2011[7]) se propone combinar esta idea con una eliminación prematura de las variables tan pronto $|w_i| \leq \epsilon$.

[7] Cheolwoo Park et al. "Bridge regression: adaptivity and group selection", Journal of Statistical Planning and Inference, 2011.

Penalties

En el caso del modelo lineal de regresión, es posible transformar el penalty de la Elastic Net en un penalty Lasso[8]

$$J(\boldsymbol{w}) = \| Y^* - X^* \beta^* \|_2^2 + \lambda^* \sum_i |\beta_i^*| \ , \tag{8}$$

definiendo

$$X^* = (1+\lambda_2)^{-1/2} \begin{pmatrix} X \\ \lambda_2^{1/2} I_{p\times p} \end{pmatrix} , Y^* = \begin{pmatrix} Y \\ 0_{p\times 1} \end{pmatrix} \tag{9}$$

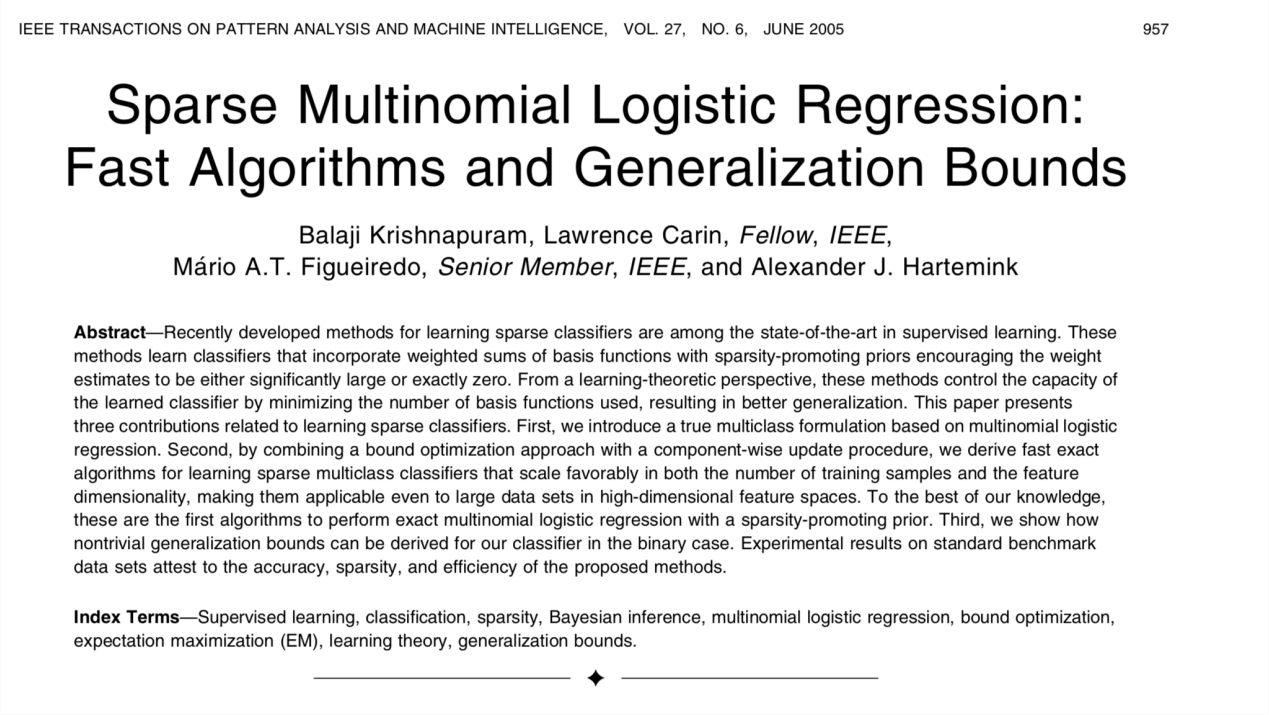
$$\beta_i^* = (1+\lambda_2)^{1/2} \beta_i \, , \lambda^* = \lambda_1 (1+\lambda_2)^{-1/2} \tag{10}$$

y aplicar algoritmos tradicionales para este problema.

[8] Hui Zou & Trevor Hastie. "Regularization and variable selection via the elastic net." Journal of the Royal Statistical Society, 2005.

Penalties

Como un ejemplo en donde el penalty es estándar, pero se requiere una cuidadosa ingeniería del algoritmo de entrenamiento por la forma del modelo $f(\boldsymbol{x}; \boldsymbol{w})$, podemos citar (Krishnapuram et al., 2005)



IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 27, NO. 6, JUNE 2005 957

Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds

Balaji Krishnapuram, Lawrence Carin, *Fellow*, *IEEE*,
Mário A.T. Figueiredo, *Senior Member*, *IEEE*, and Alexander J. Hartemink

Abstract—Recently developed methods for learning sparse classifiers are among the state-of-the-art in supervised learning. These methods learn classifiers that incorporate weighted sums of basis functions with sparsity-promoting priors encouraging the weight estimates to be either significantly large or exactly zero. From a learning-theoretic perspective, these methods control the capacity of the learned classifier by minimizing the number of basis functions used, resulting in better generalization. This paper presents three contributions related to learning sparse classifiers. First, we introduce a true multiclass formulation based on multinomial logistic regression. Second, by combining a bound optimization approach with a component-wise update procedure, we derive fast exact algorithms for learning sparse multiclass classifiers that scale favorably in both the number of training samples and the feature dimensionality, making them applicable even to large data sets in high-dimensional feature spaces. To the best of our knowledge, these are the first algorithms to perform exact multinomial logistic regression with a sparsity-promoting prior. Third, we show how nontrivial generalization bounds can be derived for our classifier in the binary case. Experimental results on standard benchmark data sets attest to the accuracy, sparsity, and efficiency of the proposed methods.

Index Terms—Supervised learning, classification, sparsity, Bayesian inference, multinomial logistic regression, bound optimization, expectation maximization (EM), learning theory, generalization bounds.

donde se expone un método para entrenar el modelo logístico con el penalty L_1 (Lasso).

Automatic Relevance Determination

ARD[9] es una técnica de *Aprendizaje Bayesiano* y como tal maneja todos los parámetros entrenables como variables aleatorias.

De manera similar a los métodos de regularización que hemos visto, la idea es introducir un prior sobre el vector de parámetros $\boldsymbol{w}$ del modelo $f(x; \boldsymbol{w})$. Normalmente

$$w_i \sim \mathcal{N}(0, \lambda_i^{-1}), \tag{11}$$

Dos diferencias importantes son: (i) en vez de fijar λ_i externamente, la variable se aprende!, (ii) el prior para cada parámetro $\boldsymbol{w}_i$ es independiente de los demás (desacoplamiento).

[9] Michael Tipping, "Sparse bayesian learning and the relevance vector machine". Journal of machine learning research, 2001.

Automatic Relevance Determination

El aprendizaje de λ se hace introduciendo un hyper-prior, que (i) introduzca mucha masa de probabilidad en soluciones dispersas y que (ii) permita derivar analíticamente el la distribución a-posteriori de λ. Si D denota el conjunto de observaciones, esta viene dada por

$$p(\lambda|D) \propto p(D|\lambda)p(\lambda) \tag{12}$$

$$p(D|\lambda) = \int p(D|\boldsymbol{w}, \lambda)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}\,, \tag{13}$$

El hyper-prior usual es la distribución Gamma no informativa

$$\lambda_i \sim \text{Gamma}(\epsilon, \epsilon)\,, \tag{14}$$

con $\epsilon \approx 0$.

Automatic Relevance Determination

El método funciona optimizando $p(\lambda|D)$ en λ, vía por ejemplo *Gradient Descent*[10], y usando (como en cualquier método Bayesiano) $p(\boldsymbol{w}, \lambda|D)$ para hacer predicciones,

$$p(y|\boldsymbol{x}, \boldsymbol{w}, \lambda) \propto \int p(y|\boldsymbol{w}, \lambda, \boldsymbol{x}) p(\boldsymbol{w}, \lambda|D) d\boldsymbol{w}\, d\lambda \tag{15}$$

$$p(\boldsymbol{w}, \lambda|D) \propto p(D|\boldsymbol{w}, \lambda) p(\boldsymbol{w}|\lambda) p(\lambda)\,, \tag{16}$$

Si se desean estimadores puntuales de $\boldsymbol{w}$, se suele tomar la media/moda de $p(\boldsymbol{w}|\lambda, D)$.

[10] Samuel Rudy et al. "Sparse Methods for Automatic Relevance Determination" arXiv preprint arXiv:2005.08741, 2020.

Automatic Relevance Determination

Como consecuencia del entrenamiento de ADR se obtienen muchos $w_i = 0$.

¿Porqué sucede esto? (1) A diferencia de Ridge, los niveles de regularización λ_i están desacoplados entre sí. (2) El hyper-prior hace que la mucha de la masa de probabilidad en $p(\boldsymbol{w}, \lambda|D)$ esté concentrada en soluciones dispersas.

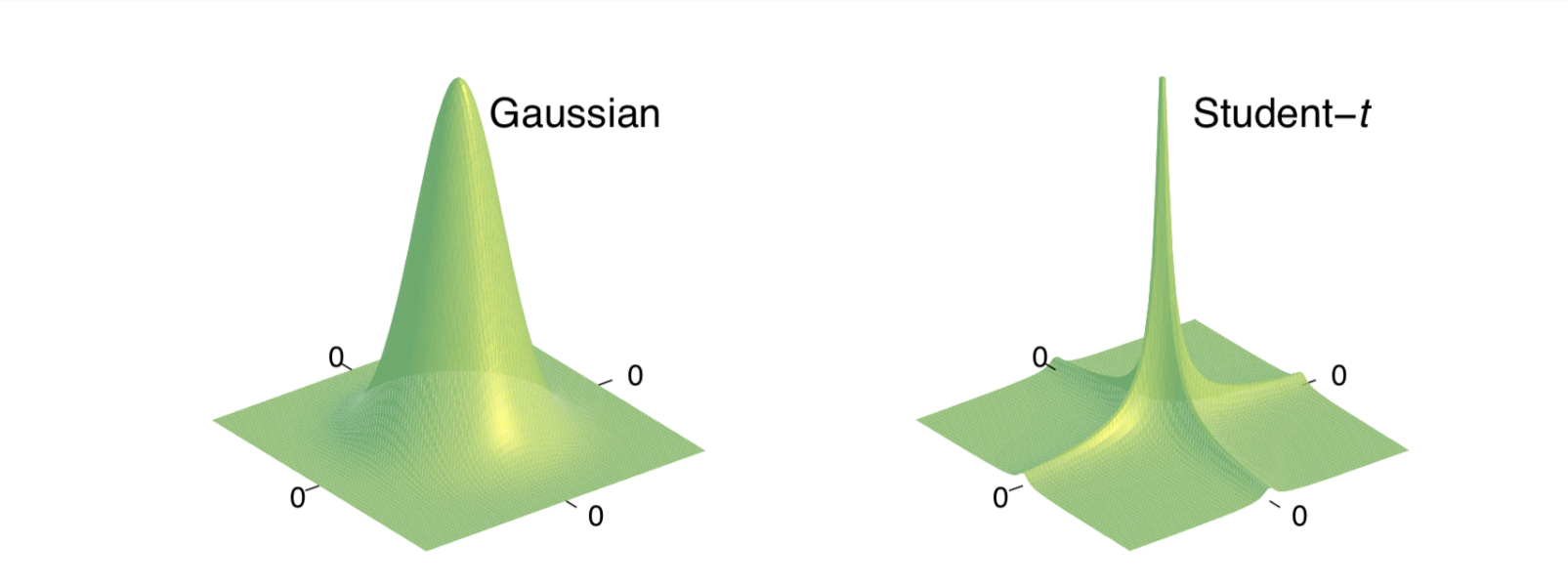
Para ver esto, uno puede calcular la marginal de $\boldsymbol{w}$

$$p(w_i) = \int p(w_i|\lambda_i)p(\lambda_i)d\lambda_i \,, \tag{17}$$

y notar que para un hyper-prior Gamma no informativo, $p(w_i)$ es una t-Student!

Automatic Relevance Determination

Gráficamente, es fácil notar la fuerte preferencia de $p(w_i)$ por soluciones dispersas, situación mucho menos clara cuando miramos $p(w_i, \lambda)$.



Para más detalles ver (Rudy 2020) o (Tipping, 2001).

Métodos Embedidos para LDA/NB

¿Selección de atributos para LDA/NB?

En el contexto de LDA, se ha propuesto regularizar el clasificador modificando la estimación de Σ como

$$\hat{\Sigma}_\lambda = (1-\lambda)\hat{\Sigma} + \lambda D\,, \tag{18}$$

donde D es la matriz diagonal con las varianzas de cada variable y $\hat{\Sigma}$ es el estimador MV estándar. Análogamente, QDA se puede regularizar combinando el estimador MV con el estimador de LDA.

Estos métodos no hacen selección de características.

Métodos Embedidos para LDA/NB

En (Cheng, 2013[11]) se ha propuesto seleccionar características en LDA (binario) resolviendo un problema de optimización que promueve la dispersión de $\boldsymbol{w} = 2\Sigma^1 (\mu_+ - \mu_-)$, que corresponde al vector que transforma la función de decisión de LDA a la forma $f(\boldsymbol{x}) = \sum_i w_i x_i + b$. El problema minimiza la norma L1 sobre $\boldsymbol{w}$ con una restricción sobre el error que se origina de sustituir el $\boldsymbol{w}$ original por su versión dispersa.

Métodos de selección de atributos para NB se han estudiado muy recientemente en (Askari, 2020[12]).

---

[11] Cheng Wang et al., "Optimal feature selection for sparse linear discriminant analysis and its applications in gene expression data", Computational Statistics & Data Analysis 2013.

[12] Armin Askari et al. "Naive feature selection: Sparsity in naive Bayes" International Conference on Artificial Intelligence and Statistics, 2020.

Métodos de Filtrado

Generalidades

Un método de filtrado intenta asignar un puntaje a cada atributo que de manera independiente del modelo o técnica de aprendizaje y sin entrenamiento.

1. En general se trata de técnicas más eficientes que las anteriores.
2. En general se trata de técnicas de amplia aplicabilidad, fáciles de entender e implementar.
3. La selección se basa en "proxies", no un estimador del error del modelo, lo que puede conducir a aumentar el error de predicción.
4. La mayoría de las técnicas populares son medidas de asociación individuales que ignoran los efectos conjuntos de las demás variables candidatas.

Métodos de Filtrado

Métodos de Filtrado Individual:

1. Correlación Lineal, Z-score, Coeficiente de Determinación, y F-Score.
2. Fisher-score F_i, χ^2-score.
3. Información Mutual $I(X_i, Y)$.

Métodos que Toman en Cuenta Interacciones:

1. RELIEF.
2. Métodos basados en Información Mutual $I(X_i, Y)$.
3. Eigenvector Centrality.

Correlación Lineal

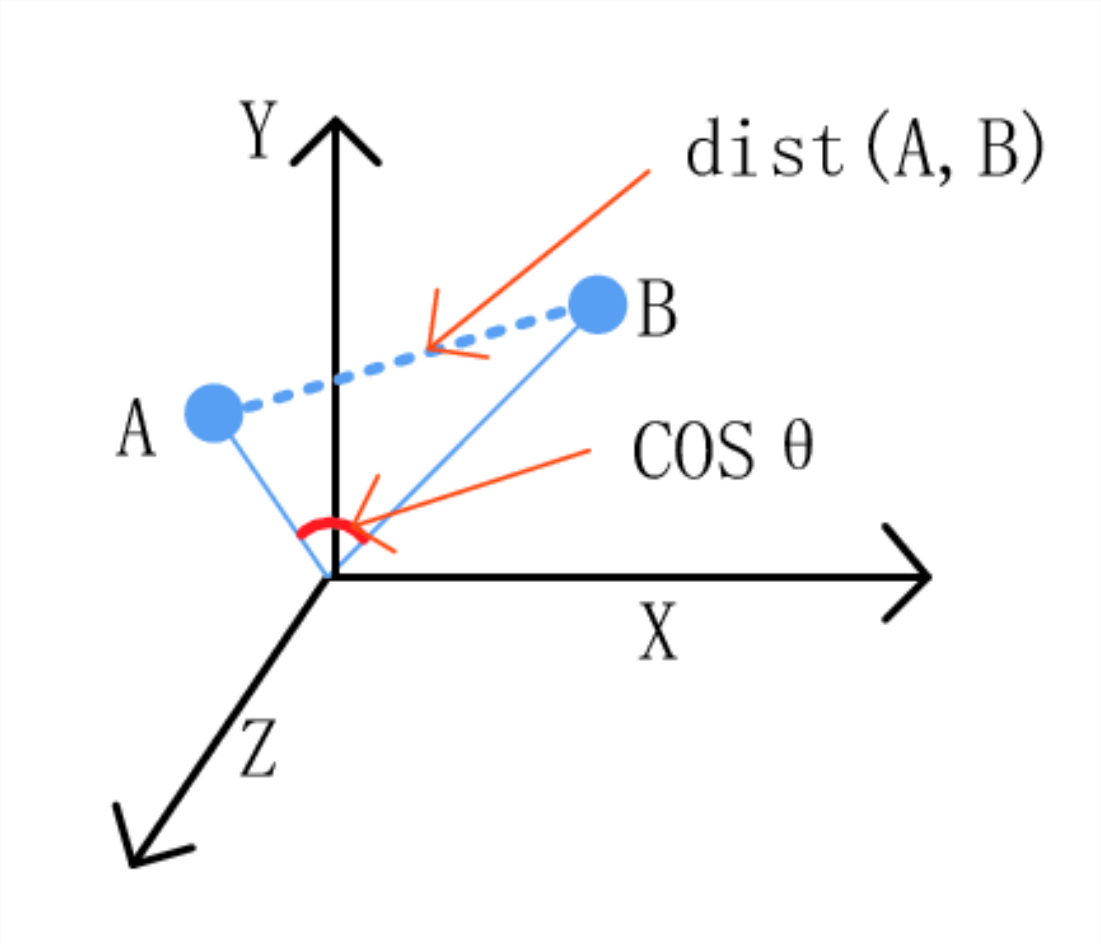
Asumiendo que Y es continua, un criterio simple y popular para medir la relevancia de una característica X_i es el coeficiente de correlación de Pearson

$$\rho(i) = \frac{cov(X_i, Y)}{\sqrt{\text{var}(X_i)\text{var}(Y)}} \,. \tag{19}$$

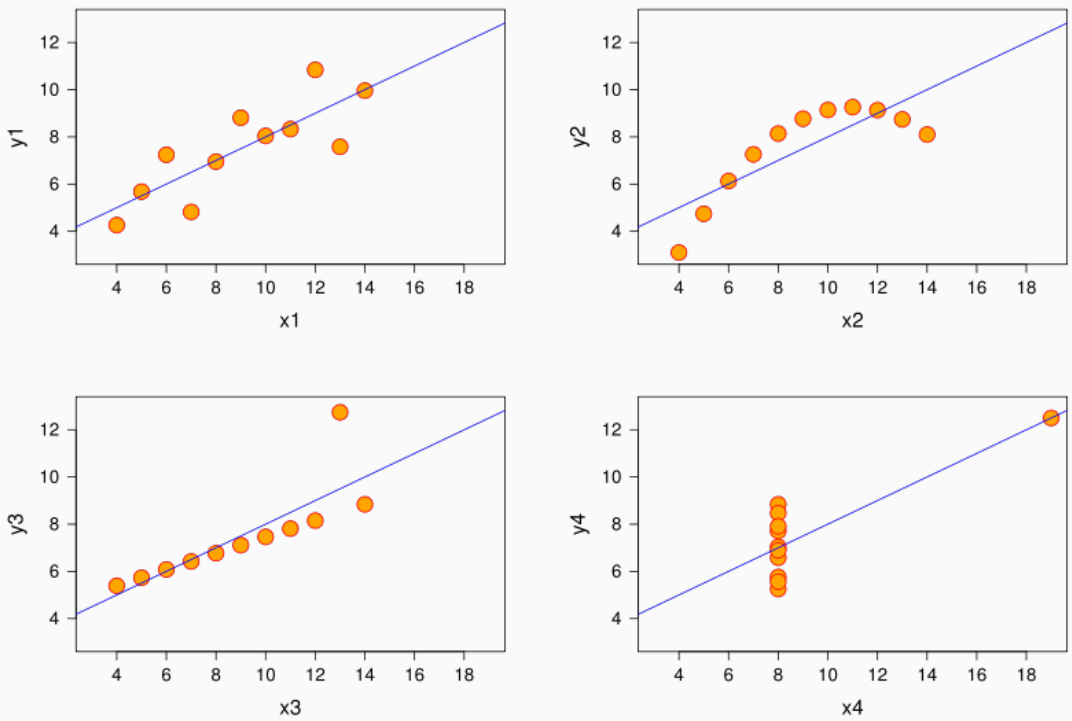
Dados n ejemplos, $\{(x^{(\ell)}, y^{(\ell)})\}_\ell$, es posible estimar $\rho(i)$ como

$$\hat{\rho}(i) = \frac{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)(y^{(\ell)} - \bar{y})}{\sqrt{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2}\sqrt{\sum_\ell (y^{(\ell)} - \bar{y})^2}} \,. \tag{20}$$

Interpretación Geométrica



Interpretación Geométrica



Z-score

Otra métrica individual ampliamente utilizada es el Z-score

$$Z(i) = \frac{\hat{a}_i}{\mathrm{STD}(\hat{a}_i)} \,. \tag{21}$$

donde $\hat{a}_i$ resulta de entrenar un modelo lineal uni-dimensional de Y sobre X_i, es decir, de ajustar el modelo $Y = a_i X_i + b_i + \epsilon_i$ usando mínimos cuadrados,

$$\hat{a}_i = \frac{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)(y^{(\ell)} - \bar{y})}{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2} \,. \tag{22}$$

Bajo el supuesto $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, y bajo la hipótesis $H_0 : a_i = 0$, $Z(i) \sim \mathcal{T}_{n-2}$ permite efectuar un contraste contra $H_1 : a_i \neq 0$ y obtener un p-valor.

Z-score

Asumiendo $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ obtenemos que

$$\mathrm{STD}(\hat{a}_i) = \frac{\sigma}{\sqrt{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2}} \tag{23}$$

De este modo, un ranking de los atributos en base a $Z(i)$ esencialmente equivalente a hacerlo en base a $\rho(i)$

$$\begin{aligned} Z(i) = \frac{\hat{a}_i}{\mathrm{STD}(\hat{a}_i)} &= \frac{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)(y^{(\ell)} - \bar{y})}{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2} \frac{\sqrt{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2}}{\hat{\sigma}^2} \\ &= \hat{\sigma}^2 \frac{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)(y^{(\ell)} - \bar{y})}{\sqrt{\sum_\ell (x_i^{(\ell)} - \bar{x}_i)^2}} \propto \rho(i)\,. \end{aligned} \tag{24}$$

Z-score

Como vimos en el capítulo anterior, una forma de medir la importancia de un atributo X_i en el modelo lineal es el Z-score multi-variado,

$$Z(i) = \frac{\hat{a}_i}{\text{STD}(\hat{a}_i)} = \frac{\hat{a}_i}{\sigma(X^T X)^{-1}_{ii}}\,, \tag{25}$$

donde $\hat{a}_i$ es el valor de a_i después del entrenamiento (con todas las variables).

Atención: Si bien la versión multi-variada considera el efecto de las demás variables en la evaluación de un atributo, esto (i) se hace en el contexto del modelo lineal y (ii) no tiene en cuenta la eliminación conjunta de variables.

Una métrica estrechamente relacionada con las anteriores es el coeficiente de determinación (uni-variado):

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

donde $SSR(i), SST$ aparecen de la descomposición

$$\sum_\ell (y^{(\ell)} - \bar{y})^2 = \sum_\ell (\hat{y}_i^{(\ell)} - \bar{y})^2 + \sum_\ell (\hat{y}_i^{(\ell)} - y^{(\ell)})^2 \qquad (26)$$

$$SST = SSR(i) + SSE(i),$$

con $\hat{y}_i^{(\ell)} = \hat{a}_i x^{(\ell)} + \hat{b}$ resultante de una regresión lineal uni-variada de Y sobre X_i. Es fácil ver que $R^2 = \rho(i)^2$ y que por lo tanto, un ranking de los atributos en base a R_i^2 es equivalente a un ranking en base a $\rho(i)$.

F-score

La descomposición de la varianza también motiva el denominado F-score (univariado) de la variable i

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

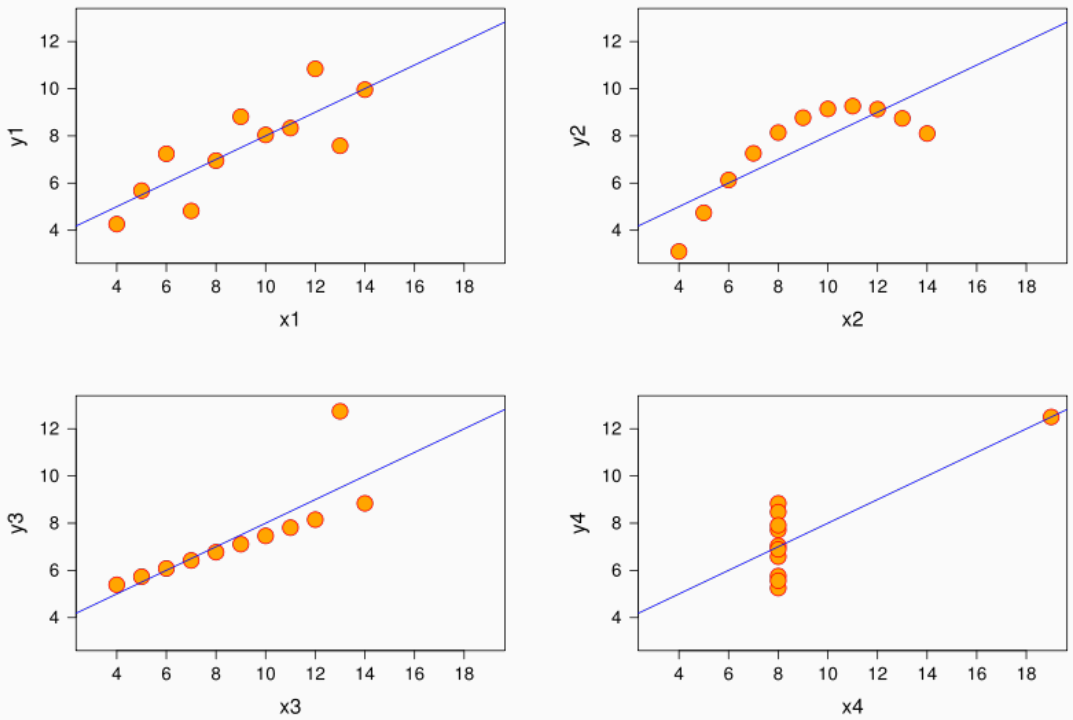
Bajo el supuesto $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, y bajo la hipótesis $H_0 : a_i = 0$, $F(i) \sim \mathcal{F}_{1,n-2}$ permite efectuar un contraste equivalente a aquel obtenido en base a $Z(i)$. Es fácil ver que

$$F(i) > F(j) \Rightarrow \frac{\rho(i)^2}{1-\rho(i)^2} > \frac{\rho(j)^2}{1-\rho(j)^2} \Rightarrow \rho(i)^2 > \rho(j)^2\,, \qquad (27)$$

es, decir, un ranking de los atributos en base a $F(i)$ es equivalente a un ranking en base a $Z(i)$.

Límites de $\rho(i)$, $Z(i)$, $F(i)$, R_i^2

Todos estos criterios de selección permiten detectar sólo relaciones de dependencia lineal entre la variable X_i y la respuesta.



Además, si Y (ó X_i) es categórica, el uso del coeficiente de correlación no es teóricamente correcto.

χ^2-score

Cuando Y es discreta con K posibles valores y X_i es binaria (e.g. presencia de una palabra en un texto), un criterio muy utilizado es el estadístico χ^2.

Sea $F_k(i)$ el número de veces que $X_i = 1$ en la clase k y $O_k = n/KV$. Bajo la hipótesis de independencia de X_i e Y,

$$\chi^2(i) = \sum_k \frac{(F_k(i) - O_k)^2}{F_k(i)}\,, \tag{28}$$

se distribuye aproximadamente χ^2_{K-1}. Si $\chi^2(i) \gg \chi^2_{\alpha,k-1}$, se rechaza la hipótesis de independencia. Por ese motivo, es posible medir la asociación del atributo y la clase usando la magnitud de $\chi^2(i)$.

χ^2-score

Cuando Y es discreta con K posibles valores y X_i es discreta con V posibles valores, es posible generalizar el estadístico χ^2.

Sea $F_{k,v}(i)$ el número de veces que $X_i = v$ en la clase k y $O_{kv} = n/KV$. Bajo la hipótesis de independencia de X_i e Y,

$$\chi^2(i) = \sum_{k,v} \frac{(F_{k,v}(i) - O_{k,v})^2}{F_{k,v}(i)}\,, \tag{29}$$

se distribuye aproximadamente $\chi^2_{(K-1)(V-1)}$. Si $\chi^2(i) \gg \chi^2_{\alpha,k-1}$, se rechaza la hipótesis de independencia. Por ese motivo, es posible medir la asociación del atributo y la clase usando la magnitud de $\chi^2(i)$.

Fisher Score

Cuando Y es discreta con K posibles valores y X_i es continua, una medida de asociación popular es el Fisher-score (relacionado con LDA).

Sean $\hat{\mu}_k(i)$ y $\hat{\mu}(i)$ la media de los valores de la variable X_i en la clase k y la media global de la variable X_i respectivamente.

Sea además $\sigma^2(i)$ la varianza muestral de de los valores de la variable X_i en el conjunto de ejemplos. Entoncess

$$\text{Fisher}(i) = \sum_k n_k \left(\frac{(\hat{\mu}_k(i) - \hat{\mu}(i))^2}{\sigma^2(i)} \right) , \tag{30}$$

donde n_k es el número de datos de la clase k.
Claramente Fisher(i) es una medida de cuánto X_i permite separar las clases.

Información Mutua

Criterios más flexibles se pueden obtener recurriendo a la 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

$$I(X, Y) = \mathbb{E}_{x,y} \log \left(\frac{f(x, y)}{f(x) f(y)} \right) = \int \int f(x, y) \log \left(\frac{f(x, y)}{f(x) f(y)} \right) dxdy \quad (31)$$
$$= \mathbb{E}_y \mathbb{E}_{x|y} \log \left(\frac{f(x|y)}{f(x)} \right) = \mathbb{E}_x \mathbb{E}_{y|x} \log \left(\frac{f(y|x)}{f(y)} \right),$$

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**,

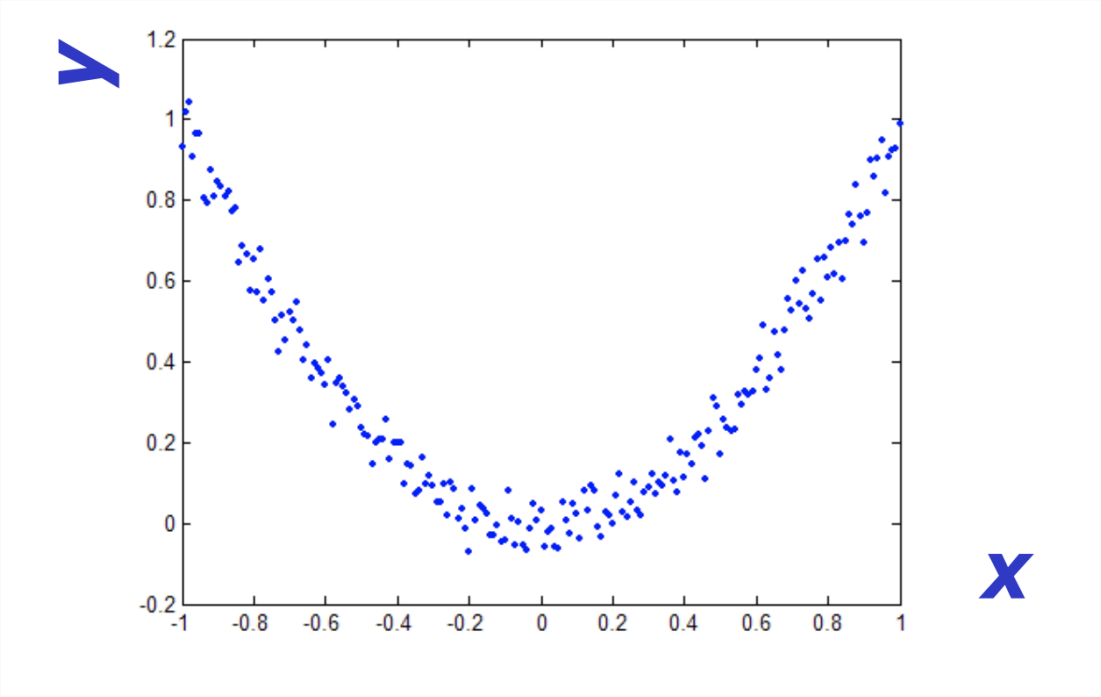
$$D_{KL}(P||Q) = \mathbb{E}_z \log\left(\frac{P(z)}{Q(z)}\right) = \int_z P(z) \log\left(\frac{P(z)}{Q(z)}\right) dz \tag{32}$$

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 $f(y|x)$.**

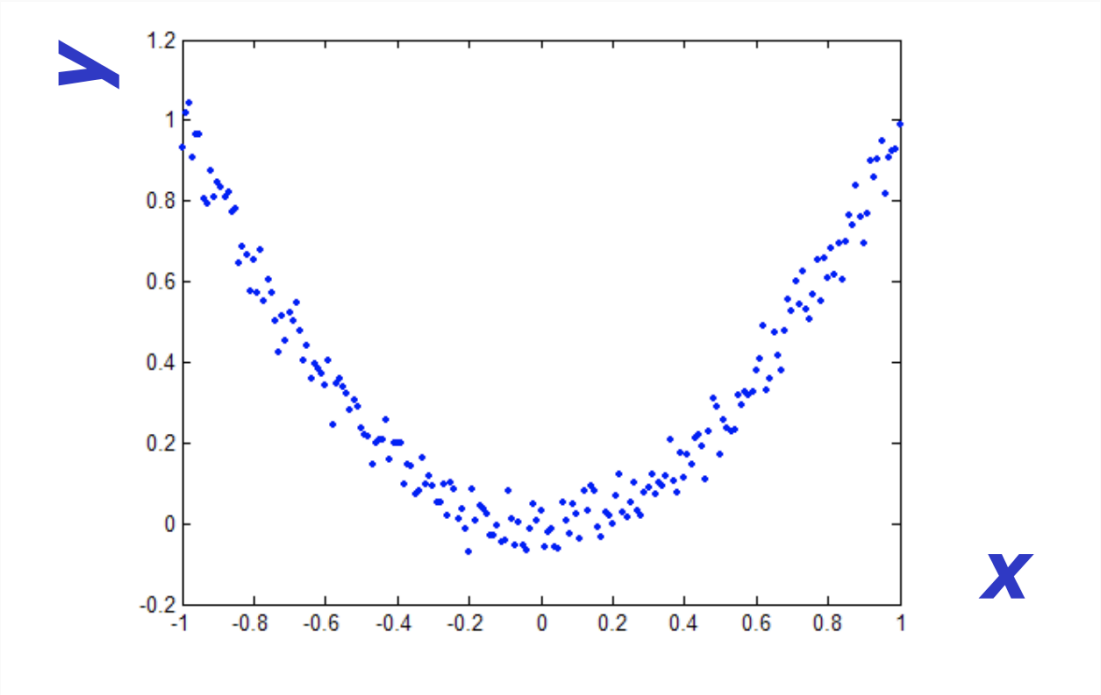
Información Mutua & No-Linealidad

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

$$\hat{I}(X_i, Y) = \sum_{x,y} \hat{f}(x_i, y) \log \left(\frac{\hat{f}(x_i, y)}{\hat{f}(x_i)\hat{f}(y)} \right) . \tag{33}$$

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

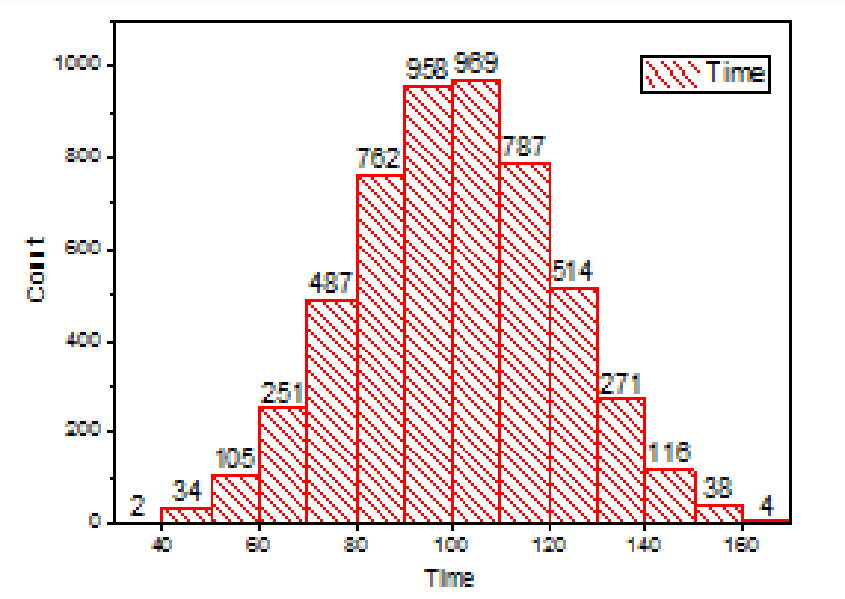
$$\hat{I}(X_i, Y) = \sum_{j,k} p(v_j, c_k) \log \left(\frac{p(v_j, c_k)}{p(v_j)p(c_k)} \right) , \tag{34}$$

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 [13].

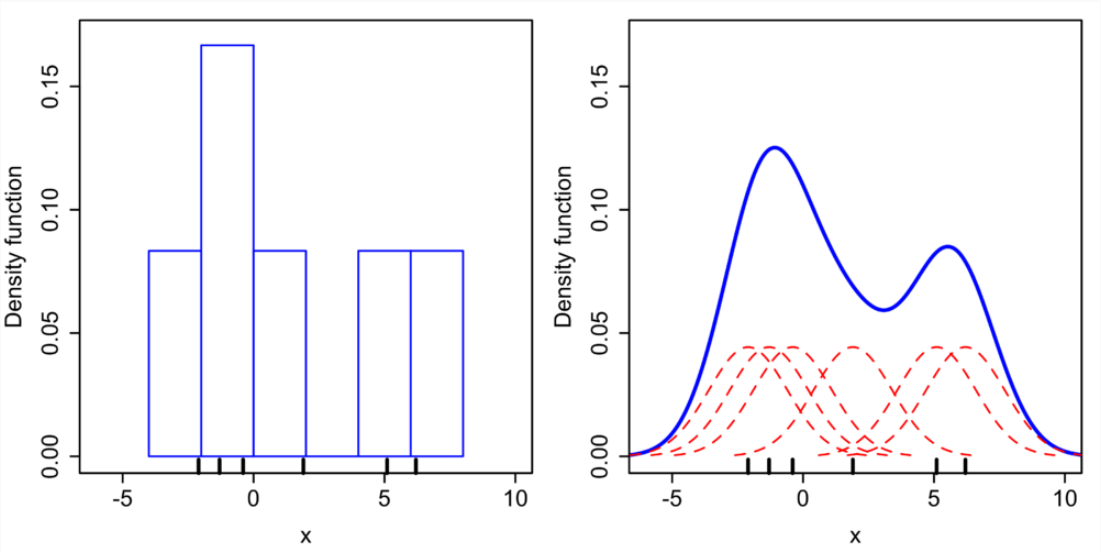


[13] Ver Kraskov et al. *Estimating Mutual Information*, Physical Review E, 2004.

Estimadores Basados Superposición de Kernels

Una alternativa más sofisticada consiste en usar estimadores basados en kernel (kernel density estimators)[14], 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.



[14]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

$$\int_V p(x)dx \approx |V|p(x)\,, \tag{35}$$

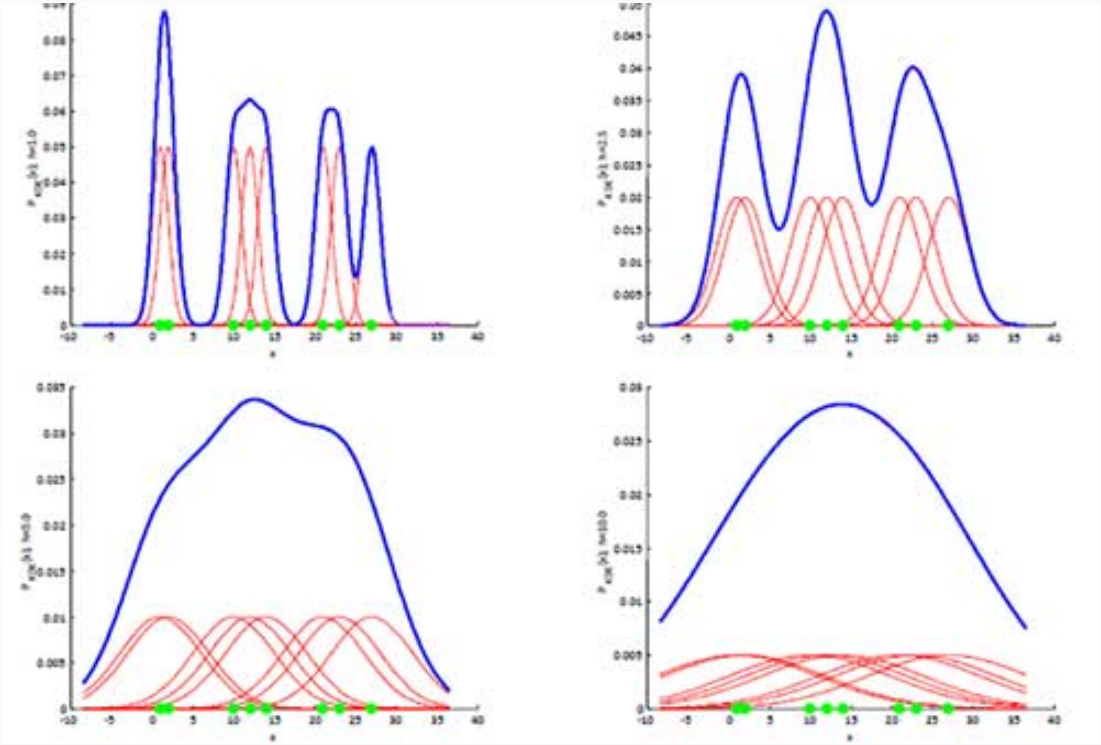
Además, si tenemos una muestra, podemos contar la fracción de datos k_n/n que caen dentro del volumen V, en modo de obtener

$$|V|p(x) \approx \frac{k_n}{n} \;\Rightarrow p(x) \approx \frac{k_n}{n|V|} = \frac{1}{n}\sum_{\ell} \frac{I(x_i - x)}{|V|}\,, \tag{36}$$

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

Efecto del Bandwidth

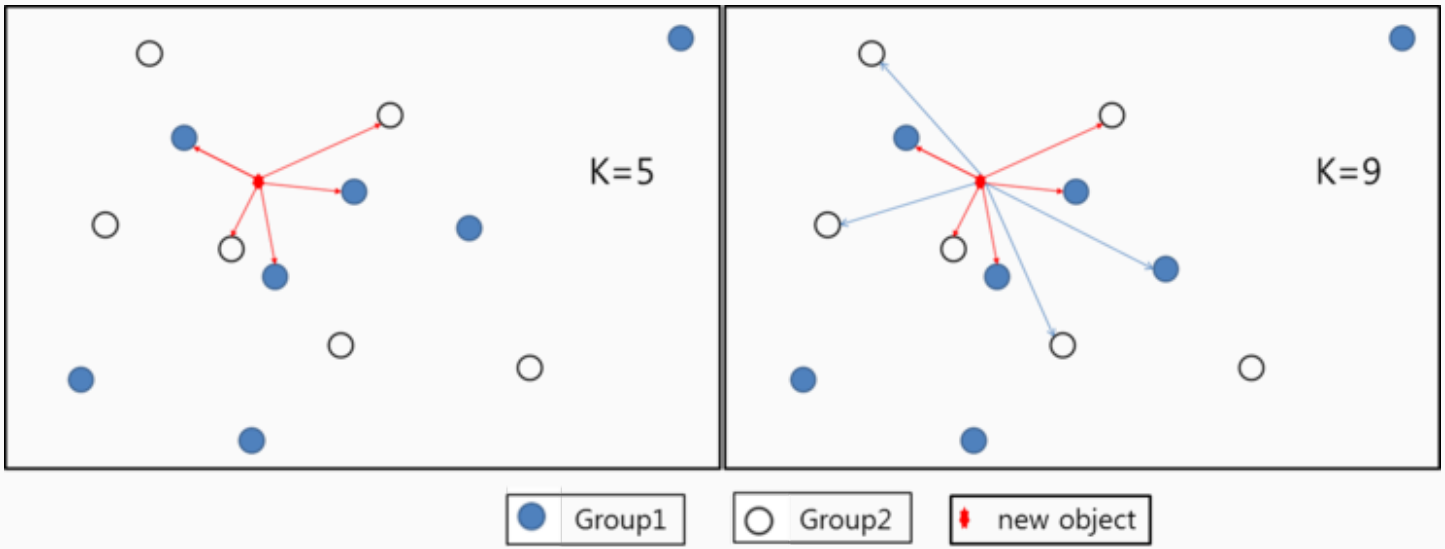
Sin embargo, los resultados de este método pueden resultar muy sensibles a la elección de *bandwidth* (tamaño del volumen), que define el grado de "influencia" de cada kernel.



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

$$I(X,Y) = H(X) + H(Y) - H(X,Y)\,, \tag{37}$$

donde

$$H(Z) = -\mathbb{E}_z \log(f(z)) = \int f(z)\log(f(z))dx\,, \tag{38}$$

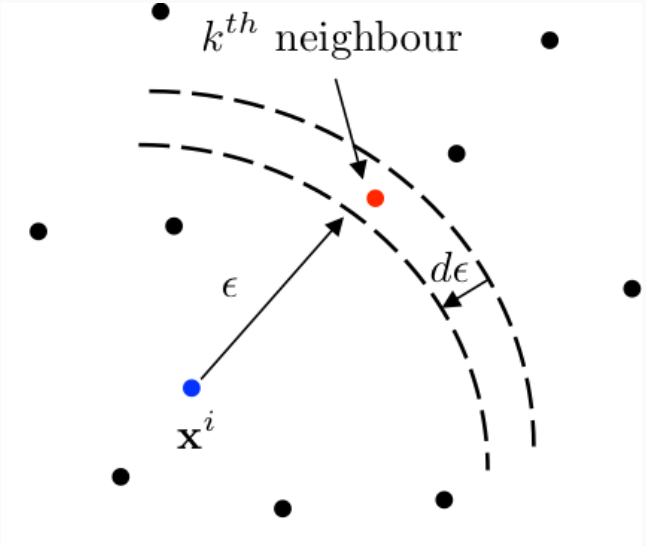
es la entropía de Z.

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

$$\hat{H}(Z) = -\frac{1}{n}\sum_\ell \log(f(z^{(\ell)})) = \frac{1}{n}\sum_\ell \log\left(\frac{1}{f(z^{(\ell)})}\right)\,. \tag{39}$$

Estimación Basada en KNN

Idea (Kraskov, 2003)[15]: 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).

[15] 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 $n-k-1$ a una distancia mayor (sino, ϵ no sería la distancia al k-ésimo vecino más cercano). Es decir,

$$p(\epsilon)d\epsilon = \binom{n-1}{1}\binom{n-2}{k-1}\frac{dP_z(\epsilon)}{d\epsilon}d\epsilon\,(P_z(\epsilon))^{k-1}\,(1-P_z(\epsilon))^{n-k-1}\,, \tag{40}$$

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

$$P_z(\epsilon) = \int_{B(z,\epsilon)} f(\tilde{z})d\tilde{z} \tag{41}$$

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,

$$\mathbb{E}_\epsilon \log P_z(\epsilon) = \int \log P_z(\epsilon) p(\epsilon) d\epsilon \tag{42}$$

Usando la ecuación (40) tenemos que

$$\begin{aligned}\mathbb{E}_\epsilon \log P_z(\epsilon) &= \int \log P_z(\epsilon) \binom{n-1}{1} \binom{n-2}{k-1} \left(P_z(\epsilon)\right)^{k-1} \left(1 - P_\ell(\epsilon)\right)^{n-k-1} dP_z(\epsilon) \\ &= \psi(k) - \psi(n)\,, \end{aligned} \tag{43}$$

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

Estimación Basada en KNN

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

$$P_z(\epsilon) = \int_{B((z,\epsilon)} f(\tilde{z}) d\tilde{z} \approx |B(z,\epsilon)| f(z) \tag{44}$$

Obtenemos que

$$\begin{aligned}
\log P_z(\epsilon) =& \approx \log |B(z,\epsilon)| f(z) && (45)\\
\mathbb{E}_\epsilon \log P_\epsilon(\epsilon) &\approx \mathbb{E}_\epsilon \left(\log |B(z,\epsilon)| + \log f(z)\right) \\
\psi(k) - \psi(n) &\approx \mathbb{E}_\epsilon \left(\log |B(z,\epsilon)|\right) + \left(\log f(z)\right) \\
-\mathbb{E}_z \left(\log f(z)\right) &\approx \psi(N) - \psi(k) + \mathbb{E}_z \mathbb{E}_\epsilon \left(\log |B(z,\epsilon)|\right) \\
H(Z) &\approx \psi(N) - \psi(k) + \frac{1}{n} \sum_\ell \log\, \epsilon_\ell^d && (46)
\end{aligned}$$

donde ϵ_ℓ es dos veces la distancia de un punto de la muestra $(z^{(\ell)})$ a su k-ésimo vecino más cercano.

Estimación Basada en KNN

- 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 Multivariados

Los métodos descritos anteriormente asignan un puntaje individual a las variables, lo que les dona sus principales ventajas:

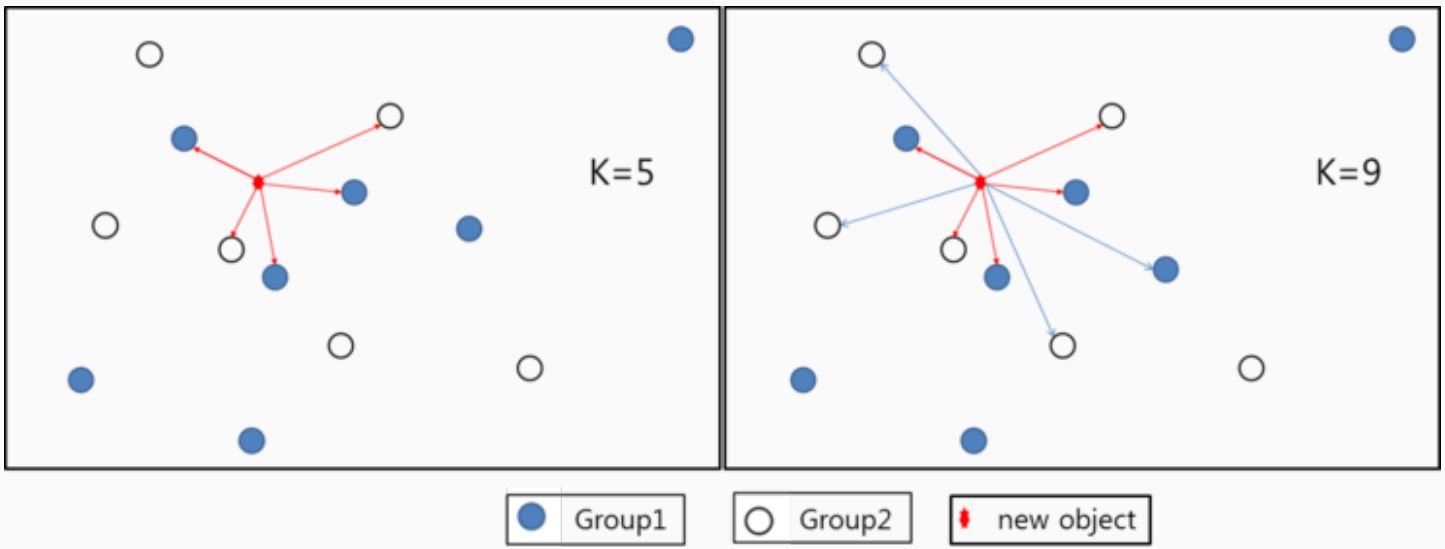
1. Son por lejos los más simples y eficientes.
2. Permiten hacer un ranking rápido e intuitivo de las variables antes del entrenamiento.

Sin embargo, ese enfoque ignora el efecto conjunto de las variables:

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

RELIEF

Una heurística muy conocida y popular, que implícitamente considera interacciones entre atributos, es RELIEF[16]. Si tenemos dos clases, este método calcula el *score* $W(i)$ del atributo X_i iterando m veces las siguientes operaciones ...



[16]K. Kira & L. Rendell, "A practical approach to feature selection" Machine Learning Proceedings, 1992.

RELIEF

Pasos:

1. Elegir aleatoriamente un punto $\boldsymbol{x}$ del dataset.
2. Encontrar su vecino más cercano que es de la misma clase $\boldsymbol{s}$.
3. Encontrar su vecino más cercano que es de otra clase $\boldsymbol{d}$.
4. $W(i) = W(i) - \frac{1}{m}\|\boldsymbol{x}_i - \boldsymbol{s}_i\|^2 + \frac{1}{m}\|\boldsymbol{x}_i - \boldsymbol{d}_i\|^2$.

La extensión a múltiples clases puede hacerse desde una óptica OVR u OVO (calcular $W(i)$ para cada clase y promediar). El segundo método suele funcionar mejor en la práctica (Kokonenko 1994[17])

La "distancia" $\|\boldsymbol{x}_i - \boldsymbol{s}_i\|^2$ suele sustituirse por una binaria y la búsqueda de vecinos suele extenderse a los k más cercanos para ganar robustez.

[17] Igor Kononenko, "Estimating attributes: analysis and extensions of RELIEF" European Conference on Machine Learning, 1994.

RELIEF

Como comenta por ejemplo (Urbanowicz 2018[18]), RELIEF logra detectar la relevancia de un atributo considerando las otras variables del problema, situación que han reportado muchos papers en el área (existen más de 20 extensiones documentadas). Ejemplo: XOR.

Instances	A_1	A_2	A_3	$Class$
R_1	1	0	1	1
R_2	1	0	0	1
R_3	0	1	1	1
R_4	0	1	0	1
R_5	0	0	1	0
R_6	0	0	0	0
R_7	1	1	1	0
R_8	1	1	0	0

[18] Ryan Urbanowicz et al. "Relief-based feature selection: Introduction and review", Journal of Biomedical Informatics, 2018.

RELIEF

La razones teóricas de esta capacidad de RELIEF no son claras.

Se ha mostrado (Robnik 2003[19]) que si la condición de vecino más cercano se elimina o si se extiende (vía media) a un número muy grande de vecinos, se obtiene que el puntaje de RELIEF para la variable i está altamente correlacionado con medidas clásicas de información uni-variadas como el Gini-gain

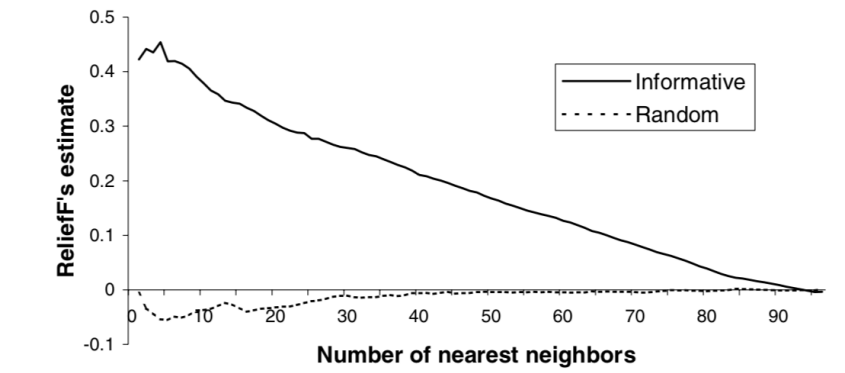
$$GG(i) = \sum_v p(\boldsymbol{x}_i = v) \sum_c p(y = c|\boldsymbol{x}_i = v)^2 - \sum_c p(y = c)^2 \,, \quad (47)$$

y por lo tanto, la capacidad de detectar la utilidad de una variable en el contexto de las demás se pierde.

[19] M. Robnik and I. Kononenko. "Theoretical and empirical analysis of ReliefF and RReliefF." Machine Learning, 2003.

RELIEF

Es necesaria investigación para entender mejor esta propiedad.



Atención: Se ha reportado en muchos trabajos la inhabilidad de RELIEF para detectar variables redundantes. Además, (Urbanowicz 2018) sugiere que la habilidad para detectar interacciones podría limitarse a 2 variable (candidata + otra).

Métodos Multivariados

Muchos métodos se basan (nuevamente) en teoría de información para encontrar conjuntos óptimos de características.

Si K es la dimensionalidad deseada de la representación, el objetivo de estos métodos puede definirse como el de encontrar un conjunto $\mathcal{F}$ de cardinalidad K que maximiza

$$I(\mathcal{F}, Y) = \int \int \int \ldots p(x_1, x_2, \ldots, x_k, y) \log \frac{p(x_1, x_2, \ldots, x_k, y)}{p(x_1, x_2, \ldots, x_k)p(y)} \, dx_1 dx_2 \ldots dx_m \,,$$

es decir, la información conjunta de todo el conjunto de características con la respuesta Y.

Desafortunadamente, estimar $I(\mathcal{F}, Y)$ es extremadamente costoso y requiere de muchos datos, de modo que los métodos no sólo deben ingenierizar la forma de navegar el espacio de búsqueda sino también la de sustituir $I(\mathcal{F}, Y)$ por un subrogado que se pueda estimar de modo más eficiente.

Maximum Relevancy Minimum Redundancy (MRMR)

Tal como los métodos tipo wrapper, la gran mayoría de estos métodos son iterativos. Por ejemplo **MRMR**, propuesto originalmente por (Battiti 1994[20]), y re-propuesto por (Peng et al. 2005[21]), adopta como algoritmo de búsqueda FSS. En cada iteración, se busca resolver el siguiente problema de optimización

$$\max_{i \notin \mathcal{F}} I(Y, X_i) - \beta \sum_{j \in \mathcal{F}} I(X_i, X_j) \tag{48}$$

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 incluidos.

[20] Roberto Battiti "Using mutual information for selecting features in supervised neural net learning". IEEE Transactions on Neural Networks, 1994.

[21] Hanchuan Peng et al. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005.

Maximum Relevancy Minimum Redundancy (MRMR)

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

Con $\beta = 0$, el método se denomina Mutual Information Maximisation (MIM). Con esta elección, el método podría incluir atributos redundantes.

En la versión de (Peng et al. 2005), $\beta = 1/|\mathcal{F}|$, es decir la redundancia se mide como un promedio sobre el conjunto ya seleccionado.

El nombre actual (Minimum-Redundancy Maximum-Relevance, MRMR) es de Peng et al. 2005.

Quadratic Programming Feature Selection (QPFS)

En 2010, (Rodriguez et al.[22]) proponen un método centrado en atacar la sub-optimalidad de FSS como método de búsqueda. Concretamente se propone resolver el problema de optimización:

$$\max_{\boldsymbol{z}} \boldsymbol{z}^T \boldsymbol{H}^{(1)} - \beta \boldsymbol{z}^T \boldsymbol{H}^{(2)} \boldsymbol{z} \tag{49}$$

$$\text{s.t. } \boldsymbol{z}^T \mathbf{1} = 1 \, \boldsymbol{z}_i \geq 0 \tag{50}$$

donde $\boldsymbol{H}^{(1)} = (I(Y, X_1), I(Y, X_2), \ldots, I(Y, X_d))^T$ y
$\boldsymbol{H}^{(2)}_{ij} = I(X_i, X_j) \, \forall i \neq j$.

La variable $\boldsymbol{z}_i$ se interpreta como una medida de la "relevancia" de la variable X_i para explicar Y.

[22] Irene Rodriguez et al. "Quadratic programming feature selection", Journal of Machine Learning Research, 2010.

Quadratic Programming Feature Selection (QPFS)

Si se agrega la restricción $\|\boldsymbol{z}\|_0 = k$, el problema

$$\max_{\boldsymbol{z}} \boldsymbol{z}^T \boldsymbol{H}^{(1)} - \beta \boldsymbol{z}^T \boldsymbol{H}^{(2)} \boldsymbol{z} \tag{51}$$

$$\text{s.t. } \boldsymbol{z}^T \mathbf{1} = 1 \, \boldsymbol{z}_i \geq 0 \,, \tag{52}$$

es equivalente a encontrar el subconjunto de atributos $\mathcal{F}$ de cardinalidad K que maximiza

$$\sum_{X_i \in \mathcal{F}} I(Y, X_i) - \beta \sum_{X_i, X_j \in \mathcal{F}, i \neq j} I(X_i, X_j) \,, \tag{53}$$

problema que como hemos mencionado tiene complejidad exponencial en K. Sin la restricción $\|\boldsymbol{z}\|_0 = k$, $\boldsymbol{z}_i$ el problema se puede resolver eficientemente.

Información Mutua Condicional

Nuevas (y mejores) aproximaciones al problema, se obtienen al observar, como ya había hecho (Yang & Moody 1999[23]), la siguiente propiedad de la información mutua conjunta $I(\mathcal{F}^+, Y)$ de un conjunto de características $\mathcal{F}^+ = \mathcal{F} \cup \{X_i\}$:

$$I(\mathcal{F}^+; Y) = I(Y, \mathcal{F}) + I(X_i; Y|\mathcal{F}) \tag{54}$$

donde $I(X_i; Y|\mathcal{F})$ se denomina información mutua condicional. Nuevamente, $I(X_i; Y|\mathcal{F})$ es muy difícil de estimar, pero la descomposición sugiere que nuevos métodos podría obtenerse aproximando $I(X_i; Y|\mathcal{F})$ ...

[23] Howard Yang & John Moody "Data visualization and feature selection: new algorithms for nongaussian data", Advances in Neural Information Processing Systems, 1999.

Información Mutua Condicional

De hecho, se han explorado varios métodos basados en información mutua condicional que miden

$$I(X_i; Y|X_j) = \mathbb{E}_{x_i,x_j,y} \log \left(\frac{f(x_i, y|x_j)}{f(x_i|x_j)f(y|x_j)} \right) \tag{55}$$
$$= \int \int \int f(x_i, y|x_j) \log \left(\frac{f(x_i, y|x_j)}{f(x_i|x_j)f(y|x_j)} \right) dx_i dy dx_j$$

para cada variable $X_j \in \mathcal{F}$ separadamente y luego agregan los resultados de alguna forma. Por ejemplo,

$$I(X_i; Y|\mathcal{F}) \approx \beta \sum_{j \in \mathcal{F}} I(X_i; Y|X_j) \tag{56}$$

$$I(X_i; Y|\mathcal{F}) \approx \min_{j \in \mathcal{F}} I(X_i; Y|X_j) \tag{57}$$

A diferencia de la agregación aditiva, el último criterio asegura que $I(X_i; Y|X_j)$ es grande $\forall X_j \in \mathcal{F}$.

Global Mutual Information Based Selection

Como buen ejemplo, (Nguyen et al. 2014[24]) proponen modificar el método de Peng y el método de Rodriguez removiendo la información mutua marginal e incorporando la información mutua condicional con una aproximación aditiva. En cada iteración, el algoritmo greedy busca la variable X_i a incorporar resolviendo

$$\max_{i \notin \mathcal{F}} I(Y, X_i) + \beta \sum_{j \in \mathcal{F}} I(X_i; Y | X_j) \tag{58}$$

La formulación de Rodriguez se adapta como

$$\max_{\boldsymbol{z}} \boldsymbol{z}^T \boldsymbol{H} \boldsymbol{z} \quad \text{s.t. } \|\boldsymbol{z}\|^2 = K,\ \boldsymbol{z}_i \in [0, 1] \tag{59}$$

con $H_{ij} = I(X_i; Y | X_j)\, \forall i \neq j$ y $H_{ii} = I(Y, X_i)$.

[24] Xuan Nguyen et al. "Effective Global Approaches for Mutual Information based Feature Selection" ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014.

Extended/Conditional MRMR

No es difícil demostrar que la información mutua condicional $I(X_i; Y|X_j)$ se relaciona con los términos $I(X_i; Y)$ (relevance) e $I(X_i, X_j)$ (redundancy) característicos de MRMR, mediante a siguiente ecuación

$$I(X_i; Y|X_j) = I(X_i; Y) - I(X_i, X_j) + I(X_i, X_j|Y) \tag{60}$$

Esto sugiere que en vez de sustituir $I(X_i, X_j)$ por $I(X_i; Y|X_j)$ en la formulación de MRMR, podemos agregar el término que falta ($I(X_i, X_j|Y)$).

Extended/Conditional MRMR

En un algoritmo greedy, esto se traduciría en elegir la variable X_i a incorporar en cada paso maximizando

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

método que se suele denominar Extended MRMR ó Conditional MRMR (Cai 2018[25]).

Naturalmente, esta formulación también se puede usar en una formulación cuadrática en el estilo de Rodriguez (Nguyen et al. 2014[26]) .

[25] Cai, Jie, et al. "Feature selection in machine learning: A new perspective." Neurocomputing, 2018.

[26] Xuan Nguyen et al. "Effective Global Approaches for Mutual Information based Feature Selection" ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014.

Información Mutua versus Información Mutua Condicional

Existen muchos otras métodos que se obtienen aprovechando otras descomposiciones/identidades de la Información Mutua Condicional.

Algunos tienen ventajas interpretativas. Por ejemplo, (Bennasar, 2015[27]) propone usar FSS con el criterio

$$\max_{i \notin \mathcal{F}} I(Y, X_i) + \min_{j \in \mathcal{F}} I(X_j; Y | X_i) \tag{61}$$

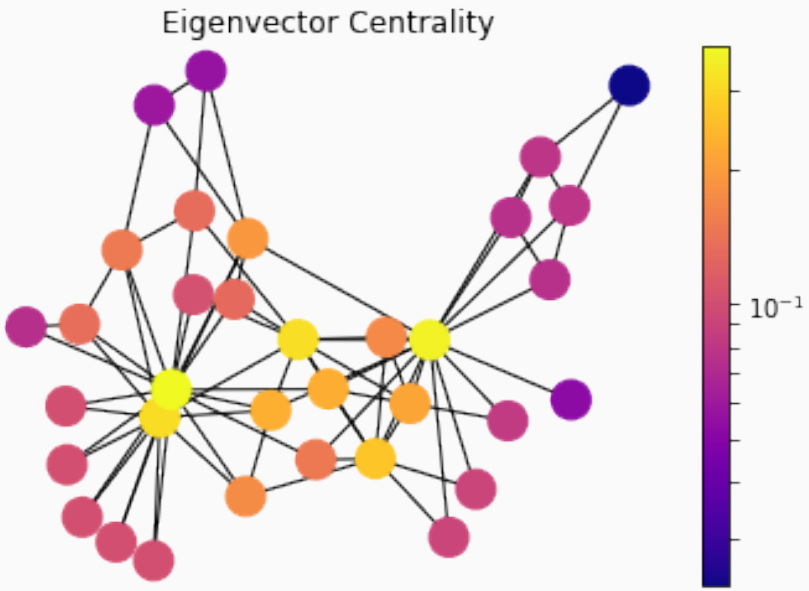
en cada paso. En esta formulación, el término $I(X_i, X_j | Y)$ mide los "aportes indirectos" de X_i sobre Y a través de las variables ya incluidas en el modelo.

Aunque es difícil decir que uno de estos métodos sea superior a los demás, una constante en la literatura más reciente es usar IM condicional en vez de solamente IM incondicional.

[27] Mohamed Bennasar et al. "Feature selection using joint mutual information maximisation" Expert Systems with Applications, 2015.

Teoría de Grafos

Para concluir, vale la pena mencionar en una interesante corriente de trabajos está proponiendo utilizar medidas basadas en Teoría de Grafos para estimar la relevancia de un atributo.



Por ejemplo, un método propuesto recientemente por (Roffo & Melzi, 2016 [28]), representa los atributos como nodos de un grafo de afinidad.

La afinidad de dos atributos se construye usando medidas clásicas como la información mutua y el Fisher-score. Una vez obtenida la matriz de adyacencia $\boldsymbol{A}$, el puntaje de los atributos se obtiene de las entradas del primer eigen-vector $\boldsymbol{v}_0$ de la matriz de adyacencia. Este vector satisface:

$$\boldsymbol{v}_0 = \lim_{m \to \infty} A^n \mathbf{1} \,. \tag{62}$$

Con la normalización apropiada, $\boldsymbol{v}_0$ puede verse entonces como la distribución de probabilidad estacionaria del grafo. Es más probable terminar en atributos con mayor score y por lo tanto los atributos se ordenan en términos de su centralidad en el grafo.

[28] Giorgio Roffo and Simone Melzi. "Features selection via eigenvector centrality", New Frontiers in Mining Complex Patterns, 2016

Teoría de Grafos

Un aspecto interesante de un método basado en Teoría de Grafos es que puede ser usarse en escenarios no-supervisados y semi-supervisados, como se ha hecho recientemente en (He 2006[29]) o en (Luo 2018[30]).

[29] Xiaofei He et al. "Laplacian score for feature selection" Advances in Neural Information Processing Systems. 2006.

[30] Tingjin Luo et al. "Semi-supervised feature selection via insensitive sparse regression with application to video semantic recognition" IEEE Transactions on Knowledge and Data Engineering, 2018.