Aprendizaje Automático sobre Grandes Volúmenes de Datos

Clase 6

Pablo Ariel Duboue, PhD

Universidad Nacional de Córdoba,
Facultad de Matemática, Astronomía y Física

Material de lectura

- Clase pasada:
  - Capítulo 13 del Owen et al. (2012)
  - http://ufal.mff.cuni.cz/~zabokrtsky/courses/npfl104/html/feature_engineering.pdf
- Ésta clase:
  - Jacob Kogan: **Introduction to Clustering Large and High-Dimensional Data** (2007)
  - Wikipedia: **Cluster Analysis (Evaluation of clustering results)**
    - http://en.wikipedia.org/wiki/Cluster_analysis#Evaluation_of_clustering_results
  - Halkidi, Batistakis & Vazirgiannis: **On Clustering Validation Techniques**. Journal of Intelligent Information Systems December (2001), Volume 17, Issue 2-3, pp 107-145.
  - Everitt, Landau & Leese: **Cluster Analysis** (2001)
  - Capítulo 7 del Owen et al. (2012)

Preguntas

- Binning: una técnica de reducción del espacio de valores (de 256 valores de una byte a 3 valores)
- Feature selection: ¿cuántas features? No se sabe bien, tomar un porcentaje de las mejores es una posibilidad
- Clustering e ingeniería de features:
  - una técnica de binning muy usada es particionar los datos y usar el cluster en vez del dato original
  - Técnicas de clustering se usan mucho con lenguaje natural, para tener features más informativas que simplemente palabras
  - Pre-procesamiento no-supervisado de las features de entrada es la clave de Deep Learning

Reducción de dimensionalidad

- Selección de features es una técnica de reducción de dimensionalidad
  - Otras involucran todas las features a la vez y generan un cambio de coordenadas, de un espacio mayo a uno menor
  - Una muy utilizada es descomposición en valores singulares (SVD) o también llamada análisis de componentes principales (PCA)
- Tomamos nuestras instancias como filas y armamos una matriz M, entonces

$$M^T ME = EL$$

- donde E es la matriz de los autovectores y L es la matriz diagonal de los autovectores
- Si E_k son las primeras k columnas de E, entonces ME_k es una representación en k dimensiones de M
- Véase http://infolab.stanford.edu/~ullman/mmds/ch11.pdf

En qué van a ser evaluados

- Dos algoritmos en detalle:
  - Naive Bayes
  - Decision Trees
- Dos algoritmos a nivel conceptual:
  - SVM
  - Regresión logística
- Ingeniería de features a nivel conceptual (práctico va a ser en la competencia kaggle y/o el proyecto)

Recordatorio

- El sitio Web de la materia es http://aprendizajengrande.net
  - Allí está el material del curso (filminas, audio)
- Leer la cuenta de Twitter https://twitter.com/aprendengrande es obligatorio antes de venir a clase
  - Allí encontrarán anuncios como cambios de aula, etc
  - No necesitan tener cuenta de Twitter para ver los anuncios, simplemente visiten la página
- Suscribirse a la lista de mail en aprendizajengrande@librelist.com es optativo
  - Si están suscriptos a la lista no necesitan ver Twitter
- Feedback es obligatorio y firmado, incluyan si son alumnos de grado, posgrado u oyentes
  - El "resúmen" de la clase puede ser tan sencillo como un listado del título de los temas tratados

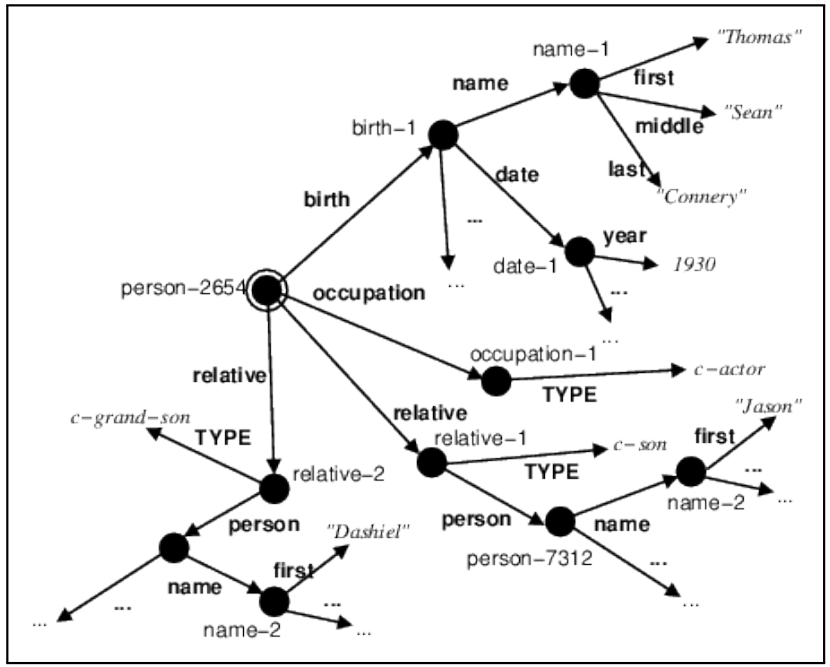
Aprendiendo lo que uno no sabe

- Un poco mágico
- Tenemos un modelo, queremos estimar sus parámetros
- Tenemos parámetros, queremos construir un modelo
- Conocer la respuesta, buscar la pregunta
- Muy ligado al concepto de distancia entre instancias

Mi tesis

- Mi tesis doctoral (http://duboue.net/thesis.html) consistió en una mezcla de aprendizaje supervisado y no supervisado
- Datos: texto e información simbólica sin alinear

Actor, born Thomas Connery on August 25, 1930, in Fountainbridge, Edinburgh, Scotland, the son of a truck driver and charwoman. He has a brother, Neil, born in 1938. Connery dropped out of school at age fifteen to join the British Navy. Connery is best known for his portrayal of the suave, sophisticated British spy, James Bond, in the 1960s. ...



Clustering

- Un tipo de aprendizaje no supervisado
- Descubre estructura presente en los datos
- Knowledge Discovery and Data Mining
  - *El objetivo de los métodos de clustering es descubrir grupos significativos presentes en los datos*
- ¿Cómo evaluar algo que no sabíamos que existía?

Una cuestión de distancias

- El concepto central en clustering es la definición de una distancia entre instancias
- Cada definición de distancias induce un agrupamiento de los datos, basado en esa métrica
- La distancia es donde se incorpora la información humana
  - Entre otros puntos, por ejemplo, el número de clusters

Ejemplos de distancias

- Datos de censo de la clase anterior:
  - age: continuous.
  - workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
  - sex: male, female.
  - education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
  - education-num: continuous.
  - marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
  - occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

Ejemplo de distancia

- Un polinomio con coeficientes *ad hoc* y distancias *ad hoc* para cada componente

$$dist(i_1, i_2) = \alpha_1 dist_{age}(age_1, age_2) + \alpha_2 dist_{workclass}(workclass_1, workclass_2) +$$

- Donde las distancias por componente pueden ser definidas matemáticamente o vía una tabla:
  - $dist_{age}(x, y) = (x - y)^2$
  - $dist_{workclass}(a, b) =$

a/b	private	self	gov't	unpaid
private	0	0.3	0.7	1.0
self		0	0.9	0.7
gov't			0	0.3
unpaid				0

Tipos de clustering

- Algoritmos por partición
- Algoritmos jerárquicos
- Algoritmos por densidad
- Algoritmos "fuzzy"

Algoritmos por partición

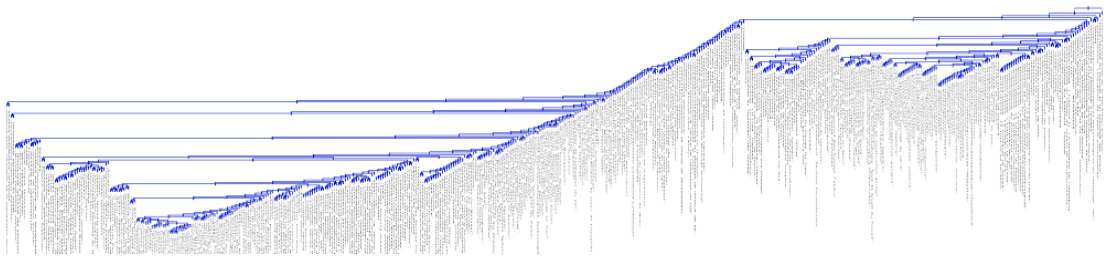
- Partir los datos en un número de particiones (clusters) donde no hay relaciones fuera de las particiones
- Las particiones optimizan una cierta función objetivo
  - La función objetivo enfatiza estructura local o global de los datos y su optimización es usualmente un proceso iterativo
- El número de clusters es normalmente un parámetro del algoritmo
- El más conocido es K-Means que vamos a ver hoy
- Otros ejemplos incluyen PAM y CLARA

Algoritmos jerárquicos

- Van uniendo clusters más pequeños en clusters más grandes o dividiendo clusters más grandes
- El resultado es un árbol de clusters (**un dendrograma**)
  - Puede ser transformado en una partición cortando el dendrograma a un nivel particular
- Aglomerativo
- Divisivo
- El algoritmo más sencillo requiere una matrix de proximidad completa
- Un algoritmo más eficiente es BIRCH (incremental, bueno con datos con ruido pero sensible al órden y limitado en tamaño)

Todos los pares

- Calcular la matrix de proximidad de manera completa es intratable para gran número de datos
- Ejemplo de dendrograma
  - Cargos laborales, distancia es la proximidad de los textos de sus perfiles en LinkedIn
  - Trabajo realizado para MatchFWD (Montreal, 2011)



Algoritmos por densidad

- Los clusters se consideran regiones del espacio de datos de alta densidad, separados por regiones de baja densidad.
- El más conocido es DBSCAN
  - La idea detrás de DBSCAN es que para cada punto en el cluster, un entorno de cierto radio debe contener un mínimo número de puntos en el cluster.
  - Soporta ruido (outliers) y puede descubrir clusters de forma arbitraria.
  - Tiene parámetros complejos de elegir, como el radio y el número de puntos en el entorno

Algoritmos "fuzzy"

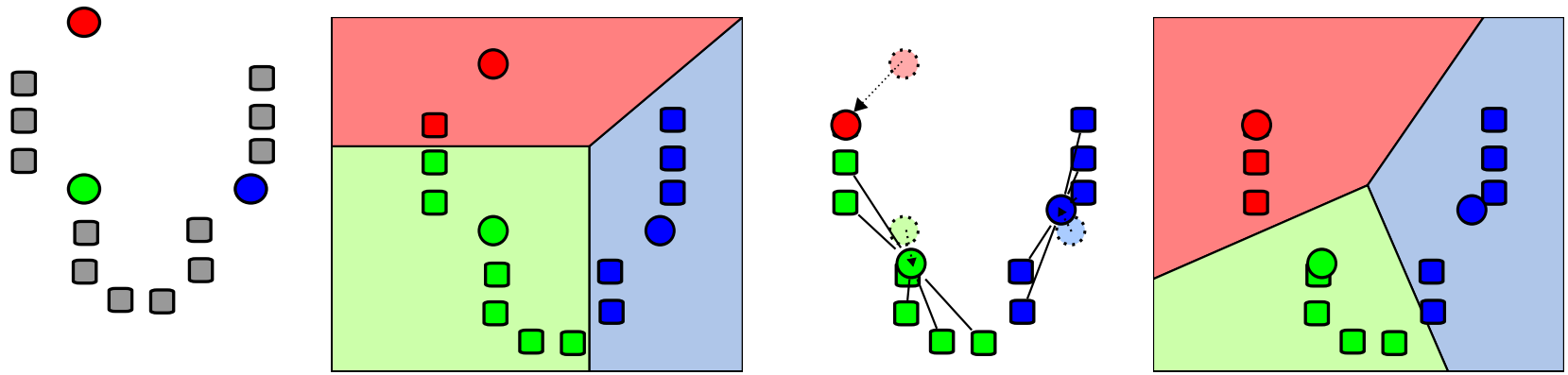
- En estos algoritmos, un punto pertenece a más de un cluster
- La membresía no es “exacta”, pero un número real entre 0 y 1.
- Los más conocidos son variantes probabilística de K-Means, que se basan en una mezcla de distribuciones Gaussianas (entrenadas usando el algoritmo EM).
  - Los centros de los clusters son la media de las distribuciones Gaussianas
  - Se estima la probabilidad de que cada punto sea generado por la Gaussiana número j (que pertenezca al cluster j)
  - La asignación de puntos a clusters asume que los puntos están generados por una distribución normal
  - La estimación se hace con el algoritmo de Expectación Maximización (EM)

K-Means

- Se basa en el concepto de elementos sintéticos:
  - cada cluster se lo representa por un centroide, un elemento ficticio
  - en vez de calcular la distancia a todos los elementos del cluster, se la calcula sólo al elemento ficticio
- El algoritmo recibe como parámetro el número K de clusters
- Al comienzo se toman como centroides K elementos al azar
- En cada paso, se re-clasifican los elementos según el centroide al que están más cerca
- Para cada cluster, se re-calcula el centroide como la media de los elementos del cluster
  - ¿Cómo calcular el centroide? Depende de los datos, igual que la distancia.

K-Means, gráficamente

- Primero y segundo paso. Los elementos línea punteado son los centroides.

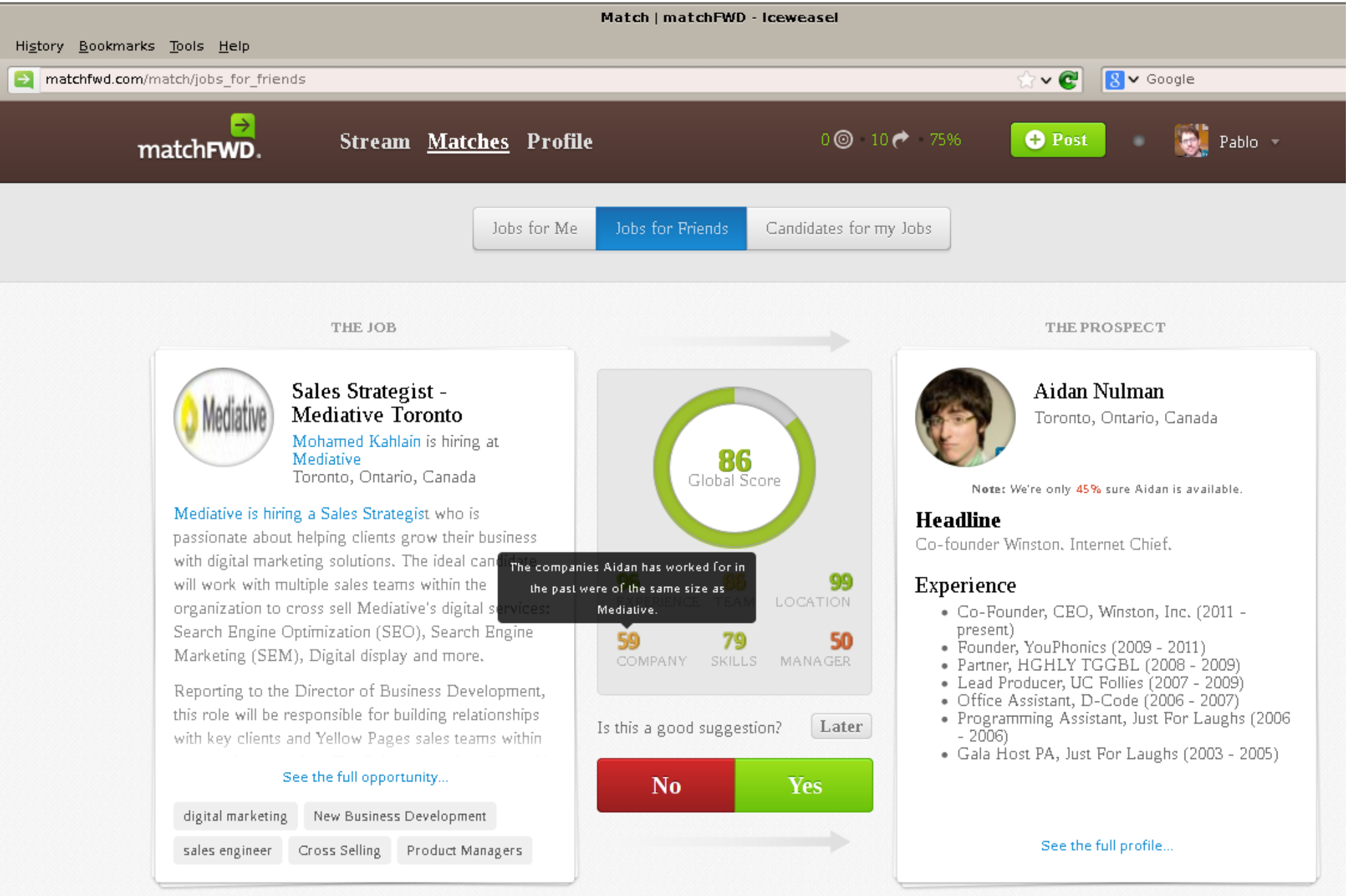


(Wikipedia)

Un cluster dominante

- Muchas veces K-Means produce un sólo cluster dominante
- En ese caso es posible volver a ejecutar K-Means en ese cluster y obtener un tipo de clustering jerárquico
- Lo más probable es que la función de distancia es pobre y no permite distinguir bien entre los elementos originales

MatchFWD: Problem



Ejemplo de Clustering

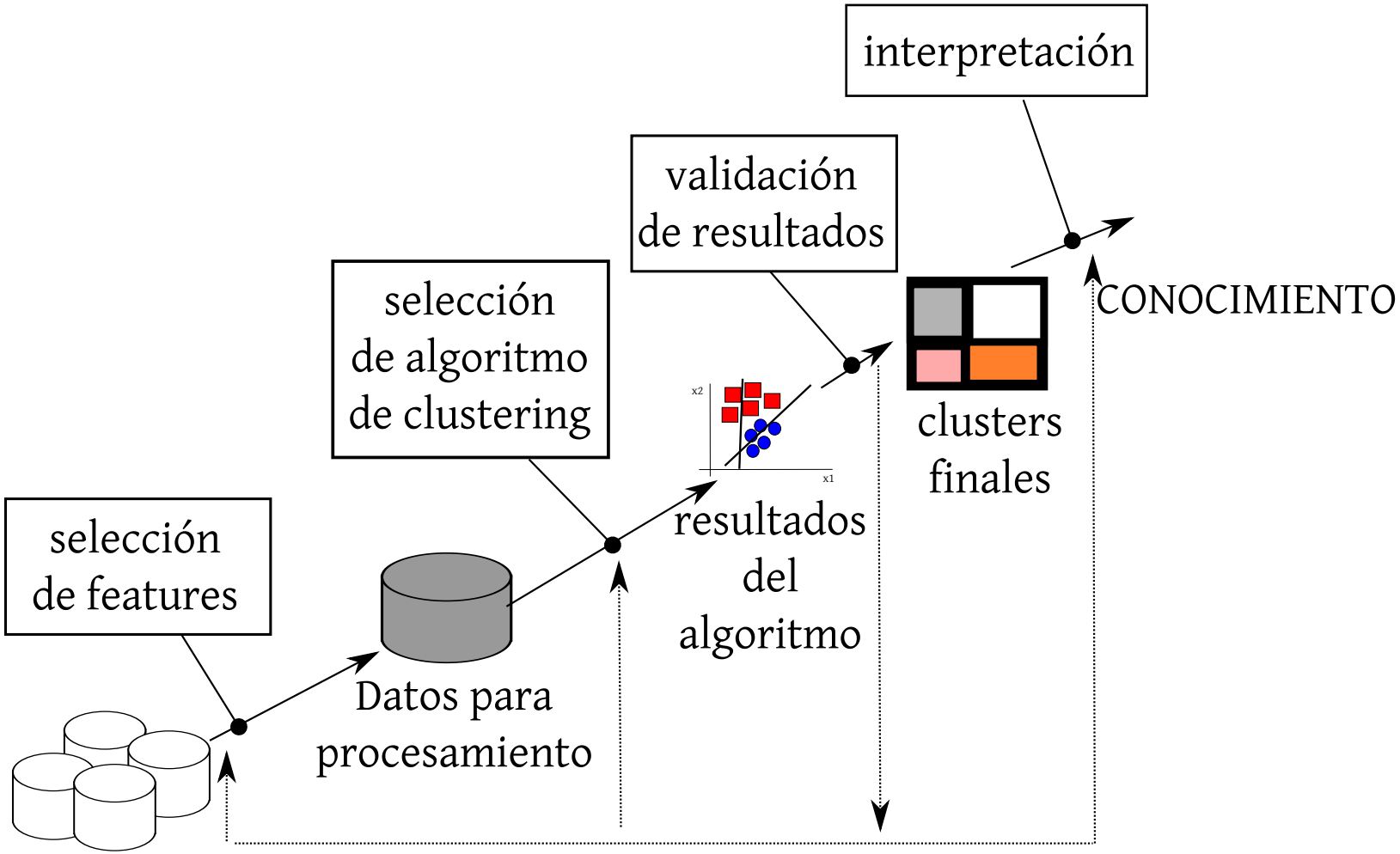
- Queremos agrupar empresas en función de que tan similares son sus culturas corporativas
- Distancia: qué tan similares son dos empresas basado en la gente que trabajo en ambas:

$$distancia(compañía_1, compañía_2) = \frac{|gente\ trabajó\ para\ ambas|}{|gente\ trabajó\ para\ cualquiera|}$$

¿Cuántos clusters?

- Canopy Cluster
  - Una técnica de fuzzy clustering
- Realizar el clustering con distintos valores de K y elegir el valor que produce mejores resultados de evaluación
- Métricas específicas
  - Ver Halkidi et al (2001)

Etapas del proceso de Clustering



Adaptado de Halkidi et al (2001), Fig. 1

Validando Clusters

- Tipos de validación
  - Validación vs. Evaluación
    - Porque estamos viendo cosas que no sabíamos antes
  - Interna: qué tan "compactos" y "robustos" son nuestros clusters
  - Externa: qué tan bien se ven comparados con clusters objetivo
  - Contra la hipótesis null: qué tan bien se ven comparados con clusters al azar

Internal Validation

- Silhouette coefficient
- David-Bouldin index
- Dunn index
- Cophenetic Correlation (for dendagrams)
- Split-sample (for robustness)

Berry and Linoff Metrics

1. *Compactness*, the members of each cluster should be as close to each other as possible.
   Metric: variance (to minimize).
2. *Separation*, the clusters themselves should be widely spaced.
   Measuring the distance between two different clusters:

   Single linkage It measures the distance between the closest members of the clusters.

   Complete linkage It measures the distance between the most distant members.

   Comparison of centroids It measures the distance between the centers of the clusters.

Silhouette coefficient

A method of *silhouette coefficient* associates a scalar $s(\mathbf{a})$ with an element $\mathbf{a}$ of the data set $\mathcal{A}$. If $\Pi = \{\pi_1, \ldots, \pi_k\}$, and $\mathbf{a} \in \pi_i$, then

1. Compute $I(\mathbf{a}) = \frac{1}{|\pi_i|} \sum_{\mathbf{x} \in \pi_i} d(\mathbf{x}, \mathbf{a})$, the average distance from $\mathbf{a}$ to other vectors in the same cluster.
2. For $j \neq i$ compute $O_j(\mathbf{a}) = \frac{1}{|\pi_j|} \sum_{\mathbf{x} \in \pi_j} d(\mathbf{x}, \mathbf{a})$, the average distance from $\mathbf{a}$ to other vectors in a different cluster and let $O(\mathbf{a}) = \min\{O_1(\mathbf{a}), \ldots, O_{i-1}(\mathbf{a}), O_{i+1}(\mathbf{a}), \ldots, O_k(\mathbf{a})\}$ ($O_i(\mathbf{a})$ is omitted).
3. Compute the silhouette coefficient $s(\mathbf{a}) = \frac{O(\mathbf{a}) - I(\mathbf{a})}{\max\{O(\mathbf{a}), I(\mathbf{a})\}}$.

- $-1 < s(\mathbf{a}) < 1$
- $s(\mathbf{a}) < 0 \Rightarrow \mathbf{a}$ might be better off in a different cluster
- $s(\mathbf{a}) \approx 1 \Rightarrow \pi_i$ is "dense"
- The coefficient for a cluster or full partition is the average for all the $s(\mathbf{a})$ in the cluster or partition

David-Bouldin index

$$DB = \frac{1}{n}\sum_{i=1}^{n} \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

where

- n is the number of clusters
- c_x is the centroid of cluster x
- σ_x is the average distance of all elements in cluster x
- $d(c_i, c_j)$ is the distance between centroids c_i and c_j
- DB should be minimized
  - Low intra-cluster and high inter-cluster distance

Dunn index

$$D = \min_{1 \le i \le n} \left\{ \min_{1 \le j \le n, i \neq j} \left\{ \frac{d(i,j)}{\max_{1 \le k \le n} d'(k)} \right\} \right\}$$

where

- $d(i,j)$ represents the distance between clusters i and j
  - may be any number of distance measures (e.g., centroids distance)
- $d'(k)$ measures the intra-cluster distance of cluster k
  - may be measured in a variety ways (e.g., maximal distance between any pair of elements)
- Algorithms that produce clusters with high Dunn index are more desirable.

Cophenetic Correlation (for dendagrams)

$$c = \frac{\sum_{i<j}(x(i,j)-x)(t(i,j)-t)}{\sqrt{[\sum_{i<j}(x(i,j)-x)^2][\sum_{i<j}(t(i,j)-t)^2]}}$$

where

- $x(i,j) = |X_i - X_j|$ is the ordinary Euclidean distance between the data points i and j
- $t(i,j)$ is the dendrogrammatic distance between the points i and j
  - The height of the first node where they are joined together
- x, t are the average for $x(i,j)$ and$t(i,j)$, respectively

Robustness: Split-Sample

1. Divide into two, by random
2. Cluster one side, then classify the second side using the centroids of the first half
3. Cluster the second side and extend with the first side as before
4. Compare the two clusterings using, for example, the Rand index

External Validation

If a goal standard with the same number of clusters and a clear correspondence with the obtained clusters exist, then the kappa statistic can be applied directly

- Confusion Matrix
- Purity
- Precision / Recall / F
  - Pair-counting F-measure (indepent of the number of clusters in each set)
- Rand / Jaccard coeffient
- Hubert's Γ

Confusion Matrix

For each cluster π_i (for $1 \leq i \leq k$) and gold cluster π_j^{min} (for $1 \leq j \leq k_m$), $c_{ij} = \left|\pi_i \cap \pi_j^{min}\right|$

We assume now that $k_m = k$. If, for example, $c_{1r_1} = \max\{c_{11}, \ldots, c_{1k}\}$, then most of the elements of π_1 belong to $\pi_{r_1}^{\min}$ and one could claim that the other elements of π_1 are "misclassified". Repetition of this argument for clusters π_i, $i = 2, \ldots, k$ leads to the numbers $c_{2r_2}, \ldots, c_{kr_k}$. The total number of "misclassifications" is, therefore, $\sum_{i,j=1}^{k} c_{ij} - \sum_{i=1}^{k} c_{ir_i} = m - \sum_{i=1}^{k} c_{ir_i}$. The fraction $0 \leq \frac{m - \sum_{i=1}^{k} c_{ir_i}}{m} < 1$ indicates a measure of "disagreement" between Π and the optimal partition $\Pi^{\min}$. We shall call the fraction $\mathrm{cm}(\Pi)$. When the partitions coincide, $\mathrm{cm}(\Pi)$ vanishes. Values of $\mathrm{cm}(\Pi)$ near 1 indicate a high degree of disagreement between the partitions.

Purity

Purity

The purity $p_i = p(\pi_i)$ of cluster π_i is given by $p_i = \max\{\frac{c_{i1}}{|\pi_i|}, \ldots, \frac{c_{ik_m}}{|\pi_i|}\}$, so that $0 \leq p_i \leq 1$ and high values for p_i indicate that most of the vectors in π_i come from the same cluster of the optimal partition. The overall purity $p = p(\Pi)$ of the partition Π is $p = \sum_{i=1}^{k} \frac{|\pi_i|}{m} p_i$, and $p = 1$ corresponds to the optimal partition. When $k_m = k$ one has $p(\Pi) = \frac{1}{m} \sum_{i=1}^{k} c_{ir_i} = 1 - \mathrm{cm}(\Pi)$, in other words $p(\Pi) + \mathrm{cm}(\Pi) = 1$.

Precision / Recall / F

- With a one-to-one mapping of computed clusters vs. golden clusters we can compute information retrieval metrics as usual
  - Same with other agreement metrics like the κ statistic
- A particularly interesting variation is **pair-counting F-measure** which is indepent of the number of clusters in each set
  - This metric might be applied to a sample of the set of points and can be of particular interest to us

Rand / Jaccard Coeffient

Consider a pair $(\mathbf{a}_i, \mathbf{a}_j)$ so that $\mathbf{a}_i \in \pi \in \Pi$, and $\mathbf{a}_i \in \pi^{\min} \in \Pi^{\min}$. The vector $\mathbf{a}_j$ may or may not belong to π and/or $\pi^{\min}$. The set $\mathcal{A} \times \mathcal{A}$ can, therefore, be divided into four subsets which we denote by $(\mathcal{A} \times \mathcal{A})_{ij}$:

$$\begin{aligned}
(\mathcal{A} \times \mathcal{A})_{00} &: \mathbf{a}_j \notin \pi \text{ and } \mathbf{a}_j \notin \pi^{\min}, \\
(\mathcal{A} \times \mathcal{A})_{01} &: \mathbf{a}_j \in \pi \text{ and } \mathbf{a}_j \notin \pi^{\min}, \\
(\mathcal{A} \times \mathcal{A})_{10} &: \mathbf{a}_j \notin \pi \text{ and } \mathbf{a}_j \in \pi^{\min}, \\
(\mathcal{A} \times \mathcal{A})_{11} &: \mathbf{a}_j \in \pi \text{ and } \mathbf{a}_j \in \pi^{\min},
\end{aligned}$$

We denote the cardinality of $(\mathcal{A} \times \mathcal{A})_{ij}$ by m_{ij}. Rand statistic and Jaccard coefficient are defined next:

$$\text{Rand statistic} = \frac{m_{00} + m_{11}}{m_{00} + m_{01} + m_{10} + m_{11}}. \tag{9.2.2}$$

$$\text{Jaccard coefficient} = \frac{m_{11}}{m_{01} + m_{10} + m_{11}}. \tag{9.2.3}$$

Hubert's Γ

$$\Gamma = \frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} X(i,j)Y(i,j)$$

$$\bar{\Gamma} = \left[\frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(X(i,j)-\mu_X)(Y(i,j)-\mu_Y)\right]/\sigma_X\sigma_Y$$

where X, Y are matrices to compare (with means and variances $\mu_x, \mu_Y, \sigma_X, \sigma_Y$,respectively), M is the total number of points and N is the total number of clusters
usual matrices to use here are:

- The proximity (distance) matrix P
- The Cluster matrix
  $C(i,j) = \{1, \text{ if } x_i, x_j \text{ belong to different clusters, } 0 \text{ otherwise}\}$
- The Distance matrix $Q(i,j) = d(v_{ci}, v_{cj})$ where v_{ci}, v_{cj} are the representative points for the clusters that x_i, x_j belong to.

Rejecting the null hypothesis

- What about if the data has no structure?
- What about if the clusters are no better than random?
- Monte Carlo simulations