DM_1

OPEN DATA : NASA, twitter, Datos Argentina, etc (via Open-API)

DATA MINING : Knowledge Discovery from Data

↳ EXTRACCIÓN DE PATRONES INTERESANTES Y CONOCIMIENTO A PARTIR DE GRANDES CANTIDADES DE DATOS. → • No triviales • Implícitos • Previamente Desconocidos • Potencialmente Utiles

KNOWLEDGE DISCOVERY IN DATABASES (KDD)

• Proceso interactivo e Iterativo

1. ENTENDER EL DOMINIO — Donde quiero aplicarlo, Cual es el problema, Cuales los objetivos
2. Seleccionar el Conjunto de Datos — Crear el Dataset (Averiguar de los datos, obtener adicionales, Integrar con otras fuentes)
3. Preprocesamiento — Mejorar la fiabilidad de los Datos (Datos Faltantes, Eliminar Ruido, Outliers (Ojo: no todo es Ruido))
4. TRANSFORMACIÓN
   - Reducir Dimensionalidad (PCA, X², Correlación)
   - Suavizado (Discretizar, Binning, Medias)
   - Agregación, Generalización, Normalización (Z-Score)
   Construir Nuevos Atributos
5. TAREA Selección del Método
   - Se extraen patrones y tendencias
   - Predicción, Clasificación, Clustering, Asociación
6. ALGORITMO Selección del Algoritmo
   - Selección, parametros, tactica de aprendizaje. Precisión vs capacidad de explicar
7. Utilización
   - emplear el algoritmo varias veces hasta obtener Resultados satisfactorios (ajustar parametros)
8. EVALUACIÓN
   - Evaluar e Interpretar Resultados
   - Medidas Cuantitativas
9. Utilizar el Conocimiento
   - Pasar de Datos de Laboratorio al Mundo Real

//DM-2//

Atributos

- → Nominal: Categorías, estados, "nombres de cosas"
- → Binario: Simétrico (~~[illegible]~~ Género), Asimétrico (no son igual de importantes (aprobado o no aprobado))
- → Ordinal: Hay orden, ranking, pero se desconoce magnitud (grande, mediano, chico)
- → Cuantitativos: Valores enteros o reales
- → Escala de intervalos: misma unidad, hay orden, no hay cero (C° y F°)
- → Escala de razón: Hay cero, son múltiplos de otro valor (grados Kelvin, long/lat, dinero)

Discretos: un solo conjunto de valores finito o infinito contable (id. cliente, CP, profesión, etc)

Binarios caso particular de discretos

Continuos: Nros. Reales como valores de atributo (temp, altura, peso)

finito de datos

Tendencia central
  Media - Moda - Mediana - Simetría/Sesgo

Dispersión
  Cuartiles, outliers, Boxplot, Varianza, σ

Histogramas, Gráfico cuantiles, QQplot, Scatter plot, Mat. matas, caras Chernoff

Pre procesamiento

- ↳ Campos obsoletos o faltantes
- ↳ Valores faltantes
- ↳ Outliers
- ↳ Formato incorrecto
- ↳ Valores no consistentes

Calidad
↳ Precisión, Integridad, Consistencia, Puntualidad, Credibilidad, Interpretabilidad

Ruido: Error aleatorio o varianza en una variable medida

↳ Soluciones → **Regresión**: suavizando ajustando una fun. de regresión lineal a partir de otra del data set.

Clustering: Identificar outliers

Binning
- Ordeno datos y particiono en bins (igual frecuencia)
- Suavizo por media, mediana, etc.

Robustez: capacidad de construir modelos insensible a datos corruptos

Integración de Datos: Combinar
Identificar
Resolver conflictos

↓
ETL

Datos redundantes los detecto con Análisis de Correlación
Análisis de Covarianz

DM-3: Outliers

Univariado	Multivariado
• Valores atípicos si se encuentran en una simple variable.	• Se encuentran en un espacio N-dimensional
• Buenos para detectar extremos pero no en otros casos.	• Para detectarlos hay que ajustar el modelo
• BoxPlot, z-score, IQR	• Clustering • LOF • Isolation Forest (ensambles)

BoxPlot: Problema cuando los atípicos son muy grandes y muy chicos.
- Mover tamaño de Bigotes y tamaño cajas

Z-Scores: # desviaciones estándar tiene una observación

$$z_i = \frac{x_i - \mu}{\sigma}$$

Se fija un umbral

LOF: Local Outlier Factor

- Basado en densidad y utiliza vecinos cercanos.

$LOF_2(d) = 8.004$ → Valor atípico
$LOF_2(b) = 5.336$
$LOF_2(a) = 3.507$
$LOF_2(c) = 3.501$

Hay que definir umbral de conts.

DM-4: Missing-Data

Razones: Mal diseño, Negativa a Responder, Respuestas Inaplicables.

Problemas: Dificultan el analisis, introduc- sesgos, Perdida de eficiencia

Soluciones
- Solo Registros Completos: Se Recomienda cuando el faltante es Azar
- Borrar Casos Seleccionados & variables. : Problemas si disminuyo la utilidad del Dato
- Imputación: Relleno, Pueden generar Sesgo

Imputación

- Sustitución de Casos: Se reemplaza con valores no observados (Experto de Datos)
- Sustitución de Media: Reemplazo con promedio (si no son normales usar Media Mediana)
- Cold-Deck: - Selecciona valores y usar relaciones obtenidas de fuentes distintas de la base actual
  - Se sustituye un valor constante derivado de otras fuentes
  - Misma desventaja que la Media (Var atenuada, Distorsion Distribucion, Repeticion de valor)
- Hot-Deck: - Reemplaza los faltantes con los mas similares
  - ⊕ Simple, finaliza con conjunto completo
  - ⊖ Dificil determinar que es similar
- Regresiones: - Predecir valores faltantes
  - Simples o multiples
  - Se identifican Indep ∧ Dep
- Imputaciones Multiples (MICE): - Supone que el origen del faltante es Random (MAR) (la proba de q' falte depende sob de los valores observados y no de los no observados)
  - Aplican MICE a Data no MAR pueden generar Sesgo!

Mice Steps: Se Repite el ciclo varias veces (10 aprox)

↳ ① Imputacion Simple (Reemplaza por Media o Moda)

10 times
- ② Una de las Faltantes originales se las vuelve a poner como Faltante
- ③ Predice esa Faltante con el Resto
- ④ Se Reemplaza por las predicciones

⊕ / No produce sesgo
/ Se puede usar en cualquier Analisis
/ Facil de usar

⊖ / Costoso computacionalmente
/ Genera un Dataset completo por iteracion
/ Hay que pensar el Modelo de Imputacion ademas del de Analisis

DM_5: Reduccion de la Dimensionalidad

- Reducir el volumen pero aun produciendo casi los mismos Resultados Analiticos.

- Maldicion: - Mas Dimensionalidad los datos se vuelven mas Ralos.
  - El Analisis de Atipicos se vuelve mas significativo
  - Las combinaciones de subespacios crece exponencialmente.

- Eliminar columnas c/ datos faltantes: / Atributos con menos del x% de valores se eliminan
  / Se busca valor de corte
  / Aplica para numericas y categoricas

- Low Variance Filter: / Mido var de una columna
  / Valores constantes ⇒ Var=0 ⇒ no ayudan a discriminado.
  / Remueve Atributos que estan por debajo de un umbral de Var (Hay q' normalizar rangos de datos)

- X^2: / Mide Dependencia.
  / Sacamos variables con mayor probabilidad de ser independientes de la clase (irrelevantes para clasificar)
  / 1 mas Relacionado a 0 menos.

- Componentes Principales (PCA): / Encuentra una proyeccion que captura la mayor cantidad de variacion entre los datos
  / Los datos originales se proyectan en espacios mas pequeños
  / Busco Autovectores de la Mat Cov.

Reducing Highly Correlated Columns

- Atributos correlacionados introducen redundancia.
- Se pueden eliminar sin disminuir drásticamente la cant. de info
- Eliminar pares correlacionados a partir de la matriz de correlaciones.
- Variables continuas y discretas (Pearson y X^2 de Pearson)

Variable Importantes (RF)

- Derivadas de la salida de ensamble Random Forest
- Arboles ⇒ Medidas internas de importancia.
- Calcula la importancia de cada variable y con eso el promedio de importancia de cada una.

Backward Feature Elimination

- Usa algoritmo de ML para medir como disminuye el error al quitar un atributo.

⊖ Alto numero de iteraciones (⊕ computo)

Forward Feature Construction

- Conjunto vacio a va agregando el mejor atributo determinada

DM-6 : Feature Engineering

- Transformación de datos, mejoran el rendimiento. Transformando el feature space
- Construcción de variables por → Discretización, Normalización, Binning
- Evaluación de las transformaciones

Normalización

↳ Mapear los features a un rango mas pequeño

↳ Ayuda a evitar que atributos con valores grandes tengan mas peso

Min-Max : Cuanto mas grande es el valor actual del valor mínimo y escala x rango

$i \in [0, 1]$

$$X'_{mm} = \frac{X - Min(x)}{Rango(x)} = \frac{X - Min(x)}{Max(x) - Min(x)}$$

Z-SCORE: Se normaliza X en base a la media y el desvío estandard. X.

$$Z\text{-score} = \frac{X - \text{MEDIA}(X)}{Sd(X)}$$

- Útil cuando no conozco el MIN ni el MAX
- Útil cuando outliers dominan el MIN ó MAX.

Decimal Scaling: • Asegura que cada valor normalizado $\in [-1, 1]$

$$X_{decimal} = \frac{X}{10^d}$$

. d Número de dígitos en los valores de la variable con valor absoluto más grandes.

Escalados Robustos: Muchos valores atípicos, usar media y var no funcionan bien / Uso estimaciones más solidas para el centro y rango

Mediana (& algun percentil) e IQR

Discretización

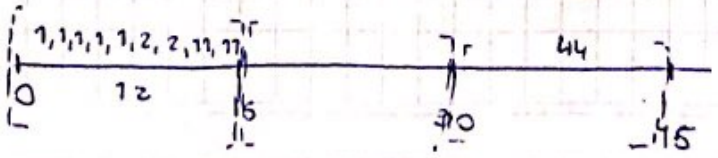
- Dividir rango continuo en intervalos
- Por Class ⇒ Supervisada (entropía
- Sin Class ⇒ No-Supervisada (Binning, R&R, Quantiles, Math

Binning

- Top-Down
- Cantidad específica de Bins
- Agrupamiento
  - Por Frecuencia: misma # de observaciones por bin
  - Igual Ancho: Defino Rango e intervalos por bin.
- No-Supervisado.
- Para cada agrupamiento puedo reemplazar por Media, Mediana, etiquetas

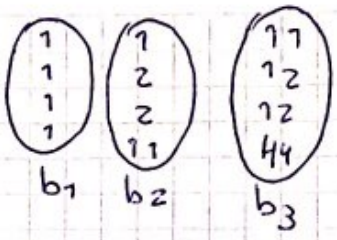
$X = \{1, 1, 1, 1, 1, 2, 2, 11, 11, 12, 44\}$ 3 categorías

Igual Ancho:

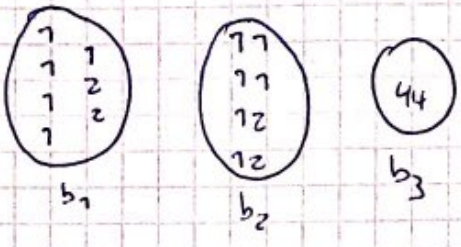


Igual Frecuencia

$m = 12$
$bins = 3$
$\frac{m}{bins} = 4$



K-Means: Identifican lo que parecen ser intuitivamente correcto



Rank

- El ranking de un número es su tamaño relativo a otros valores de una variable numérica.
- Mismos valores reciben el mismo rango, los duplicados afecta las posiciones de los posteriores.
- Solido pero el mismo valor puede tener dos rangos + en diferentes listas.

Quantiles

- Utiles pero como rank puede un X tener distintos cuantil si la lista cambia.

Math Function: Floor(LOG(x)): efectivo binning para las diamantes sesgados

Entropia

- Supervisada y separacion Top-Down
- Explora y determina split-point

Entropia Class $E(S) = \sum -p_i \ln p_i$

Entropia Class a Split-Point: $E(S,A) = \sum \frac{|S_v|}{|S|} E(S_v)$

Information-Gain $= E(S) - E(S,A)$

Variable Flag

- Regresion Requiere numericos.
- Si tengo categoricos lo puedo recodificar en una o mas Dummy o Flag.

If sex = female then sex_flag = 0
If sex = male then sex_flag = 1

DM-7: No-SQL

- Heterogeneidad de los datos y la cantidad llevaron al límite los RDBMS
- Conceptos que permitan el procesamiento poniendo foco en la performance confiabilidad y agilidad.

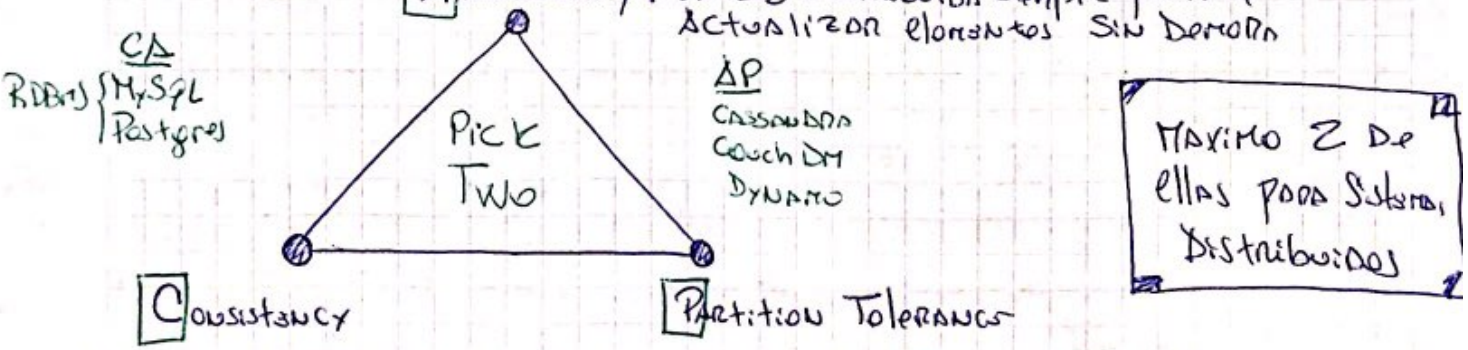
RDBMS: controla las tx a través del uso de la Atomicidad, consistencia, independencia y persistencia.

- **A**tomicity: La tx no se puede separar, se hace o no se hace.
- **C**onsistency: La tx pasa la BD de un estado consistente a otro consistente
- **I**solation: La tx se ejecuta independientemente.
- **D**urability: Los cambios son permanentes, se persisten.

ACID hace foco en Consistencia e Integridad. Bloquea Recursos

- **B**asic **A**vailability: Permite que los sistemas temporalmente sean inconsistentes para que las tx sean manejables.
- **S**oft-state: Permite inconsistencias temporales, los datos pueden cambiar mientras se usan.
- **E**ventual Consistency: Cuando toda la lógica se ejecuta se llega a un estado consistente.

BASE prioriza el no-bloqueo ante la consistencia a Disponibilidad



Tener una versión única actualizada y legible de sus datos disponible para todos los clientes

CP
Mongo
Redis
Memcache

Sigue respondiendo incluso si hay una falla entre las partes de la DB

Key-Values → -Cache de objetos, -Imagenes → MemCache, Dynamo, Redis
⊕ Escalable
⊖ No se consulta por el contenido de value

Colum Family → -Big Data, -Resultados de Web Crawls → Cassandra, HBase
⊕ Escalable y Disponibilidad
⊖ Diseño crítico. No se puede consultar un blob

Grafos → -Social Network, -Detección de Fraudes → Neo4j, InfiniteGrap
⊕ Rapidez para búsquedas en red
⊖ Poco escalable

Document Stores → -Alta disponibilidad, -Búsqueda de documentos → MongoDB, CouchDB
⊕ No hay mapeo (schemaless)
⊖ Incompatible con SQL

DM-8: Datos No Estructurados

Text Mining

- **Lexicon:** Conjunto completo de las distintas palabras para definir el corpus
- **Corpus:** Conjunto de datos correspondiente a una colección de documentos.
- **Bolsa de palabras** → Terminos / Totales = Frecuencia
  - Cual es la forma de sacar = frecuencia una palabra (se tratan como dimensiones o features)
- **Stop-words:** Palabras frecuentes que no ayudan y se pueden sacar.
- **Matriz Termino Documento**
  - Filas son los documentos
  - Columnas terminos para analisis
  - matriz llena de ceros. Alta dimensión, dispersa y no negativa.

	Hola	Mundo	Manola
Hola Mundo	1	1	
Hola Manolas	1		1

- **Stemming:** extracción de raiz morfológica

Studies → Stem + Sufijo

efectua	efectu
efectuaba	"
efectuar	"
⋮	

- **Lematización**: usa la parte específica del habla para determinar la raíz.
  - Dependen del idioma.

Forma	Info	Lemma
niñas	Femenino plural de niño	niño

- **NER**: Named Entity Recognition
  - Identificar y clasificar expresiones de un texto que hacen referencia a personas, lugares, marcas, fechas, horas, "entidades"
- **Part-of-speech Tagging**: Averigua que son sustantivos, verbos, etc.

DM-9: Reglas De Asociación

- Dado un conjunto de Tx encontrar reglas que puedan predecir la ocurrencia de un item basado en la presencia de otros
- La implicación indica co-ocurrencia (no casualidad).

itemset: Colección de 1 o más items

K-itemset: un itemset con k elementos

Support Count (σ): Cantidad de ocurrencias de un itemset.

Support (s): Fracción de tx que contienen a un itemset

$$s = \frac{\sigma(x)}{|T|}$$

Itemset Frecuente: un itemset cuyo Support(s) es mayor o igual al umbral de Minsup.

Tid	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

$\sigma(\{Milk, Diaper\}) = 3$

$S(\{Milk, Diaper\}) = \frac{\sigma(\{Milk, Diaper\})}{|T_r|} = \frac{3}{5}$

$\sigma(\{Beer\}) = 2 \Rightarrow S(\{Beer\}) = \frac{2}{5}$

Supongo minSup = 0.6

$S(\{Beer\}) = 0.4$ ⊗

$S(\{Milk, Diapan\}) = 0.6$ ←

$S(\{Milk\}) = \frac{4}{5} = 0.8$ ←

$S(\{Bread\}) = \frac{4}{5} = 0.8$

$S(\{Coke\}) = \frac{2}{5} = 0.4$ ⊗

Itemset Frecuente

$S(\{Milk, Diaper, Beer\}) = \frac{2}{5} = 0.4$ ⊗

Regla De Asociacion: Expresion $X \Rightarrow Y$ Donde X e Y son Itemset frecuentes.

Metricas

Support: S = Fraccion de transacciones que contienen a X e Y

Confidence: C : Mide con que frecuencia Y aparece en las transacciones en las que tambien aparece X

$C(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$

$\{Milk, Diaper\} \rightarrow \{Beer\}$

$S = \frac{\sigma(\{Milk, Diaper, Beer\})}{|T|} = \frac{2}{5} = 0.4$

$$C = \frac{\sigma(\{Milk, Diaper, Beer\})}{\sigma(\{Milk, Diaper\})} = \frac{\frac{2}{5}}{\frac{3}{5}} = \frac{2}{3} = 0.67 = \frac{0.4}{0.6}$$

Reglas creadas a partir del mismo itemset tienen mismo soporte pero pueden tener distinta confianza.

Encontrar todas las Reglas / Soporte $\geq$ MinSup, Confianza $\geq$ MinConf } Encontrar todas (fuerza bruta) es muy Costoso!

1. Generar todos los itemsets / support $\geq$ MinSup
2. Generar a partir de los itemsets Subconjuntos con las Reglas / confianza $\geq$ MinConf

} Dados d items hay 2^d itemsets candidatos

Fuerza bruta $\sim O(NMw)$, $M = 2^d$, $N = t_x$

- Reducir candidatos (M) mediante podas
- Reducir comparaciones (MN), utilizando estructuras eficientes
- Reducir # de T_x : ej DHP

Principio Apriori : Reduce candidatos

"Si un itemset es frecuente $\Rightarrow$ todos sus subsets deben serlo"

Propiedad Antimonotonia

$$\forall X, Y : (X \subseteq Y) \Rightarrow Support(X) \geq Support(Y)$$

el Soporte de Y nunca supera el de sus subsets

- Si X itemset no satisface MinSupport $\Rightarrow$ X no es Frecuente

Si Agrego X_2 a X : $(X \cup X_2)$ $\Rightarrow$ el Resultado no puede ser mas Frecuente

luego $S(X \cup X_2) < MinSupp$

APRIORI: FACTORES que AFECTAN la COMPLEJIDAD

- Elegir MIN-Support
  - Bajarlo INCREMENTA la CANTIDAD de ITEMSETS frecuentes.
    (PUEDE INCREMENTAR CANDIDATOS y la longitud MAXIMA de ITEMSETS frecuentes)
- DIMENSIONALIDAD del DATASET (# ITEMS)
  - MAS espacio para ALMACENAR el COUNT Supp de CADA ITEM!
    - COMPUTACIONALMENTE MAS COSTOSO
- TAMAÑO de la BD (# TRANSACCIONES)
  - APRIORI hace VARIAS PASADAS, el TIEMPO de EJECUCION AUMENTA con el NUMERO de TR.

TIPOS ESPECIALES:

ES UTIL SELECCIONAR UN CONJUNTO REPRESENTATIVO DE ITEMSETS Y A PARTIR DE ELLOS DERIVAR CONJUNTOS DE ITEMSETS FRECUENTES

MAXIMAL FREQUENT ITEMSET (MFI)

- UN ITEMSET ES MAXIMAL SI NINGUNO DE SU SUPERSET ES FRECUENTE.
- PROPORCIONAN UNA REPRESENTACIÓN COMPACTA DEL CONJUNTO DE ELEMENTOS FRECUENTES.

CLOSED FREQUENT ITEMSET (CFI)

- UN ITEMSET ES CLOSED SI NINGUNO DE SUS INMEDIATOS SUPERSET TIENEN EL MISMO SUPPORT QUE EL ITEMSET.
- REPRESENTACIÓN MINIMA DE LOS ITEMSET SIN PERDER INFO DE SOPORTE.
- CFI PERMITE REMOVER REGLAS REDUNDANTES.
- SE PUEDE USAR PARA DETERMINAR el SUPPORT DE UN ITEM QUE NO ES CLOSED.

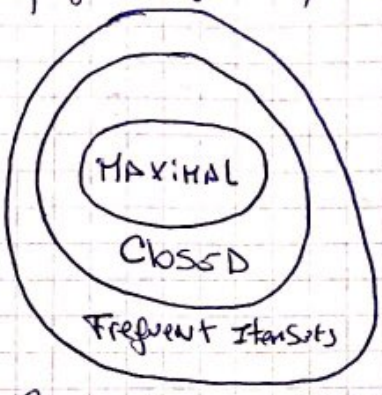
REGLAS REDUNDANTES

$X \Rightarrow Y$ es REDUNDANTE SI $\exists\ X' \rightarrow Y',\ X \subseteq X' \wedge Y \subseteq Y'$ / SUPPOR CONFIANZA SON IGUALES.

CFI las evita

Todos los Max Frecuent Itemsets son closed, porque ningun superset tieme el mismo soporte.

Maximal es Closed



La confianza es limitada, Reglas con alta confianza pueden ocurrir de casualidad.

Lift

- Asumiendo independencia
- $X \Rightarrow Y$, el Lift es la confianza de la Regla dividido la confianza esperada.
- Mide que tan lejos de la independencia estan X e Y

$$Lift(X \Rightarrow Y) = \frac{Conf(X \Rightarrow Y)}{Supp(Y)}$$

Cercanos a 1 implican que X e Y son indep x por lo tanto la Regla es poco interesants.

DM-10: Frequent Pattern Growth (FP-Growth) - Algorithm

Apriori	FP-Growth
• Utiliza un enfoque de generar y probar: Genera itemsets candidatos y prueba si son frecuentes. • Generar itemsets candidatos es costoso • El conteo de soporte es costoso	• Descubrir itemsets sin generar itemsets candidatos • Paso-1: Crea estructura compacta: FP-Tree ↳ Haciendo 2 pasadas sobre el conjunto de datos • Paso-2: Extrae itemsets frecuentes directamente desde el FP-Tree. A través del recorrido por el FP-Tree

DM_11: Sistemas de Recomendación

Matriz de Utilidad

Utilidad: Los valores representan grados de preferencia de los usuarios sobre los items.

Usuario \ Items

	i_1	i_2	i_3	i_4	i_5	i_6
U_1	?	1	?	5	?	?
U_2	4	?	2	3	?	?
U_3	?	4	?	8	5	3
U_4	?	?	5	?	3	?

Explicita

Usuario \ Items

	i_1	i_2	i_3	i_4	i_5	i_6
U_1	0	1	0	1	0	0
U_2	1	0	1	1	0	0
U_3	0	1	0	0	1	1
U_4	0	0	1	0	1	0

Implicita

Los valores representan grados de preferencia de los usuarios sobre los items.

Etapas Recomendación para usuario "U"

E1: Predicción

↳ El sistema asigna un score a cada item "i" desconocido por "U".

E2: Recomendación

↳ Se genera una lista de items ordenada por valor de score y se recomiendan los primeros k elementos de esta lista.

Clasificación de Algoritmos

- Popularidad
- Basados en Contenido
- Asociación de Productos
- Filtrado Colaborativo
- Hibrido y Ensambles

Asociación de Productos

- Cada User es una Tx
- Se calculan Reglas entre todos los pares de items
- Se utiliza única metrica de Asociación (Soporte, Lift, etc)
- Se genera matriz cuadrada de tamaño $|I| \times |I|$
- Si la metrica es un indice de similitud ⇒ Matriz simetrica y es Mat. de Similitud

[Usuario U interactua con i_4]

Prediccion

Score$(i_1) = 0.1$
Score$(i_2) = 0.2$
Score$(i_3) = 0.6$
Score$(i_5) = 0$
Score$(i_6) = 0.9$

Items

	i_1	i_2	i_3	i_4	i_5	i_6
i_1	1	.2	.5	.1	.2	.9
i_2	.2	1	.05	.2	.4	0
i_3	.5	0.5	1	.6	.2	0
→ i_4	.1	.2	.6	1	0	.9 ←
i_5	.2	.4	.2	0	1	.7
i_6	.9	0	0	.9	.7	1

↑ Items

Recomendacion:
$[i_6, i_3, i_2, i_1, i_5]$

Similitud Jaccard

↳ Los items se Representan por Conjuntos de usuarios

$$i_1 = \{u_1, u_2, u_5\}$$

$$i_2 = \{u_1, u_6\}$$

$$Sim(A,B) = \frac{Supp(A \& B)}{Supp(A) + Supp(B) - Supp(A \& B)}$$

$$Sim(i_1, i_2) = \frac{|\{u_1\}|}{|\{u_1, u_2, u_5, u_6\}|} = \frac{1}{4} = 0.25$$

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Similitud Coseno

↳ Los items se Representan como vectores con tantas dimensiones como usuarios existan.

$$i_1 = \{u_1, u_2, u_5\} \rightarrow [1, 1, 0, 0, 1, 0]$$

$$i_2 = \{u_1, u_6\} \rightarrow [1, 0, 0, 0, 0, 1]$$

$$Sim(A,B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

$$Sim(A,B) = \frac{Supp(A \& B)}{\sqrt{Supp(A) * Supp(B)}}$$

$$Sim(i_1, i_2) = \frac{1}{\sqrt{(2 * 3}} = 0.41$$

Filtrado Colaborativo

└ Predicciones

USR to USR

↳ Dado un usuario U
↳ Se calculan similitudes contra todos los usuarios
↳ Se eligen a los k vecinos mas cercanos
↳ Por cada item "i" desconocido por "U" se calcula como score a la suma de todos los indices de similitud de los k vecinos que contienen a "i".

Ítems to items

- Generalización de las recomendaciones de asociación de productos.
- Para las predicciones se suman todas las similitudes a los items conocidos por U.

$$U_1 = \{i_1, i_3, i_4\}$$

$$Score(ij) = Sim(ij, i_1) + Sim(ij, i_3) + sim(ij, i_4)$$

$$Score(i_2) = 0.2 + 0.05 + 0.2 = \boxed{0.45}$$

$$Score(i_5) = 0.2 + 0.2 + 0 = \boxed{0.4}$$

$$Score(i_6) = 0.9 + 0 + 0.9 = \boxed{1.8}$$

Factorización de Matrices

- Metodo de Reducción de Dimensionalidad
  - SV
  - Gradients Descendents.
- Se descomponer la matriz de utilidad en nuevas dimensiones
- Estas dimensiones latentes/ocultas captan ≠ caracteristicas de los items y usuarios
- Los ítems y los usuarios quedan representados en este espacio latente.
- Al estar representados en un mismo espacio pueden calcularse directamente la similitud/distancia entre un usuario y un item.