Teoria ML

Desigualdad de Hoeffding:

$P(z_i = 1) = \phi \qquad P(z_i \neq 1) = 1 - \phi$

↳ $P(|\phi - \hat{\phi}| > \gamma) \leq 2\exp(-2\gamma^2 m)$ ↳ Conc. Valores

↳ se acerca al valor real de ϕ

II) Error de entrenamiento

$$\hat{\varepsilon}(h) = \frac{1}{m} 1\{h(x_i) \neq y_i\}$$

Error de generalización

$$\varepsilon(h) = P(h(x) \neq y) \longrightarrow x \in D$$

III) Underfitting → $\hat{\varepsilon}$ alto

Overfitting → $\hat{\varepsilon}$ bajo y ε alto

IV) h def por ϕ

$\hat{\phi} = \text{argmin}\ \hat{\varepsilon}(h_\phi)$ → minimizar $\hat{\varepsilon}$

) $H = \{h_1, \ldots, h_k\}$ hipotesis $h_i(x) \begin{cases} 1 \\ 0 \end{cases}$

$\hat{\varepsilon}$ $\qquad z_j = 1\{h_i(x_j) \neq y_j\}$

$\downarrow$

$\hat{\varepsilon}_{(h_i)} = \frac{1}{m} \sum_{j=1}^{m} z_j$ $\qquad \varepsilon \quad z = 1\{h_i(x) \neq y\}$

$\downarrow$

$\varepsilon = P(z = 1)$

$P(|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma) \leq 2\exp(-2\gamma^2 m)$

VI) Convergencia Uniforme

$$P(\exists h \in H / |\varepsilon(h) - \hat{\varepsilon}(h)| > \gamma)$$

$$P(A_1 \cup A_2 \cdots \cup A_k)$$

$$\leq \sum_{i=1}^{k} P(A_i)$$

$$\leq \sum_{i=1}^{k} 2\exp(-2\gamma^2 m) = 2k\exp(-2\gamma^2 m)$$ → Prob $\hat{\varepsilon}$ y $\varepsilon \geq \gamma$

VII) Despejamos m

$$m \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$$

↳ Complejidad de la muestra.

VIII)

$$h^* = \underset{h \in H}{\operatorname{argmin}}\ \varepsilon(h)$$ → Hipotesis Optima

$$\varepsilon(\hat{h}) \leq \hat{\varepsilon}(\hat{h}) + \gamma$$

$$\hat{\varepsilon}(\hat{h}) \leq \hat{\varepsilon}(h^*)$$

↳ $\varepsilon(\hat{h}) \leq \hat{\varepsilon}(h^*) + \gamma$

$$\varepsilon(h^*) - \hat{\varepsilon}(h^*) \leq \gamma$$

↳ $\hat{\varepsilon}(h^*) \leq \varepsilon(h^*) + \gamma$

↳ $\boxed{\varepsilon(\hat{h}) \leq \varepsilon(h^*) + 2\gamma}$

IX) $$\boxed{\varepsilon(\hat{h}) \leq \min_{h \in H} \varepsilon(h) + 2\sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}}$$

↳ TEOREMA DE LA GENERALIZACIÓN

X) Dimension VC

$VC(H)$

$VC(\mathcal{O}_x + \mathcal{O}_1) = 3$ → Cardinalidad maxima de puntos en 2D

Teo Radon: Si tenemos $d+2$ puntos en $\mathbb{R}^d$ podemos particionarlos en 2 ejes disjuntos tal que sus fronteras convexas se intersecten → no separables linealmente

↳ $VC = d+1$

TEOREMA HART

↳ Si tenemos M puntos en d dims. Existe con Prob alta una f que proyecta $\mathbb{R}^d$ en $\mathbb{R}^k$ $k >>> d$ de forma tal que los puntos sean linealmente inde separables.

TEOREMA VAPNIK: Sea H una clase de Hipotesis $VC(H) = d \Rightarrow$

con Prob al menos $1-\delta$ $\forall h \in H$

$$|\varepsilon(h) - \varepsilon(\hat{h})| \leq O\sqrt{\frac{d}{m}\log\frac{m}{d} + \frac{1}{m}\log\frac{1}{\delta}}$$

Cor: $|\varepsilon(h) - \hat{\varepsilon}(h)| \leq \gamma$

$\forall h \in H$ con prob $1-\delta$ es suficiente que $m = O(d)$

$$d \approx |\Theta|$$

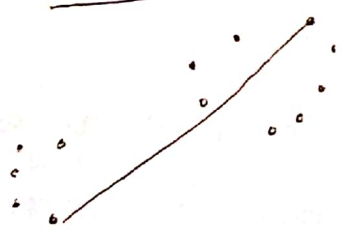
CLUSTERIWG

- Clustering Jerarquico:
  1) Partimos de que cada punto es un cluster.
  2) En cada paso, unimos los dos cluster mas cercano.
  3) Repetir hasta que quede uno solo o la cantidad buscad

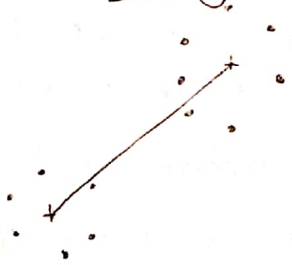
- Discancias

- Single-linkage

- Complete-linkage



- Average-Link



- Medion-Link

- OPTIMIZACIÓN LSH Single Lineage
  - ↳ Apromixación
  - ↳ En cada bucket habra como maximo un punto de cada cluster
  - ↳ Los clusters en un mismo bucket son candidatos

K-MEANS

↳ Dado un set de datos en $\mathbb{R}^d$, Particionar en k sets ($k < D$) de forma tal de minimizar la sumatoria de las distancias al centroide.

$$\sum_{i=1}^{k} \sum_{X_i \in S_i} \|X - \mu_i\| \rightarrow \text{NP-HARD}$$

algoritmo

1) Elegir k puntos, uno para cada cluster, centroides iniciales
2) Asignarle a cada punto el cluster con centroide mas cercano
3) Recalcular el centroide como el centro de los puntos
4) Volver a 2

K-MEANS ++

1) Elegir un punto al azar como centroide
2) Calcular la distancia de cada punto contra los centroides y quedarse con la mínima.
3) Calcular la probabilidad de cada punto como la distancia dividida por la suma de todas las distancias de los puntos
4) Elegir un punto al azar usando la nueva proba.
5) Repetir.

$$\|X_1 - Y\|^2 + \|X_2' - Y\|^2 + \|X_3 - Y\|^2 = 0$$

$$\frac{X_1 - C_1}{\|X_1 - Y\|} + \frac{X_2 - C_1}{\|X_2 - Y\|} = \text{max } X_1, X_2 \|X_1 - Y\|$$

STREAMING ⇝ FLUJO DE DATOS SIN FIN

Reservoir Sampling
- Guardar muestra de tamaño constante K
- n cantidad de datos vistos
- Probabilidad de que un dato este en la muestra
  - $\frac{K}{n}$
- Genero un $X \in [0, 1]$ aleatorio
  - Si $X < \frac{K}{n}$, reemplazo uno aleatorio del array

MOMENTOS DE STREAM
- Decimos que la frecuencia de cada dato i es M_i → cant. veces visto
  - Momento de orden K del stream
    - $\sum M_i^K$

K=0 → CANTIDAD DE ELEM DIFERENTES VISTOS

K=1 → CANTIDAD DE ELEM OBSERVADOS

K=2 → Factor Sorpresa → indicador de distribución pareja.

FLAJOLET-MARTIN → K=0
- APLICA FUNCIÓN DE HASH DE M bits, $M > \log_2 n$
- OBSERVA CANTIDAD DE 0 a izq
- $r = MAX(\uparrow)$
- CANT ELEM DIFERENTES $\simeq 2^r$
- Requiere buen hash → USAMOS VARIOS HASH EN GRUPOS.

HyperLogLog

↳ USAMOS UN HASH DE 64 bits
↳ Los primeros n bits son numero de estimadores
↳ 64-n bits contamos 0 a izquierda y actualizamos el estimador si es mayor.

↳ La estimación es el promedio armonico de los estimadores

AMS. → K=2
↳ se define un numero k de estimadores
↳ c/u con 2 campos → VALOR, CANTIDAD

↳ Por cada elem
↳ Si esta en el valor de algun K estimador → cantidad +1
↳ Si no se sortea si reemplaza a un estimador.
↳ Reservoir Sampling.

→ EL MOMENTO DE ORDEN 2 es el PROMEDIO DE $n(2C_i - 1)$
↳ cantidad est. R_i

BLOOM FILTERS → SABER SI UN ELEM PERTENECE A UN SET
↳ Vector binario de m-bits y K funciones de hash de 0 a m-1
↳ Para construir el filtro, por cada elem del set le aplico los hash y enciendo en 1 las posiciones indicadas
↳ Para consultar, se aplican los hash y me fijo si todas las pos estan en 1 → Pertenece con proba $(1-\gamma)$ con γ P(FALSO POS)
↳ Si no esta encendido alguno, definitivamente no pertenece.

→ El valor OPTIMO de Hashes K para el vec de M bits en un universo de n elementos
↳ $K = \left(\frac{m}{n}\right) \log 2$

→ SABIENDO EL K, LA CANTIDAD DE M bits (P(FALSO POS))

$$m = \frac{n \ln p}{(\ln 2)^2}$$

→ ESTIMAR CUANTOS ELEM EN EL FILTRO (bits, bits en 1)

$$E = \frac{-\left(M \ln\left(1 - \frac{X}{M}\right]\right)}{K}$$

Counting Filter

↳ en vez de bitmap, tengo un contador
↳ Nos permite eliminar elementos del filtro.
↳ Construimos como bloom, pero aumentando en 1 el contador
↳ Consulta como bloom, > 0
↳ USA MAS ESPACIO

COUNT MIN SKETCH

↳ Basado en Counting Filter
↳ Se usan d filtros de w bits asociado a una func. de hash.
↳ Observo elem, aplico d func. hash e incremento la pos en cada vec.
↳ Para estimar frecuencia, hasheo y me quedo con el contador min.
↳ Para mantener los i elementos mas frec, por cada elem actualizo y hago count-min, si supera al menor de los i elementos lo reem
↳ guardo los top-i en memoria

HEAVY HITTER PROBLEMS

↳ Encontrar valor mas repetid

CUCKOO FILTERS

↳ USA CUCKOO HASHING
↳ + RAPIDOS PARA CONSULTA
↳ NO ALMACENA CLAVES SI NO FINGERPRINT (6 a 8 bits)
↳ ALMACENADOS TABLA DE CUCKOO
↳ M buckets
↳ b fingerprint por bucket.

SISTEMAS DE RECOMENDACIÓN ⇝ Recomendar Contenido al usuario

- ↳ PRECISIÓN
- ↳ Serendipicy → Cosas Nuevas
- ↳ Diversidad

SISTEMAS NO-PERSONALIZADOS ⇝ Para todos, no específicos.

↳ Recomendación de Comentarios
- ↳ UPVOTES − DOWNVOTES → NO PROPORCIÓN
- ↳ $\frac{\text{UPVOTES}}{\text{UP + DOWN}}$ ⇝ ECUACIÓN MAS PELIGROSA

↳ Intervalos de Confianza
- ↳ Dado los UP ↑ y los down ↓ queremos saber con confianza del x%. cual es el limite inferior para la proba de que el comentario sea POSITIVO.
  - ↳ ORDENAMOS POR ESA PROBA.

↳ Siendo U los UP y D los down, proba upvote (proporción) $P = \frac{U}{U+D}$ y $n = U+D$,

USAMOS La formula de Wilson para intervalos de confianza:

↳ No sirve para noticias

$$\left(P + \frac{z^2}{2n} \pm z\sqrt{\left[P(1-P) + \frac{z^2}{4n}\right]/n}\right) \Big/ \left(1 + \frac{z^2}{n}\right)$$

z-VALUE PARA INTERVALO

ORDENANDO NOTICIAS

↳ función de utilidad U, P proba que le guste la noticia, q proba que no sea nueva

$$u(p,q) = a * pq + (1-p)q * b + p(1-q) * c + (1-p)(1-q) * d$$

Siendo $p = \frac{U+1}{U+D+2}$ y $q = e^{-\lambda\tau}$

↳ $u(U, D, A, B) = \log(U-D) + \frac{A-B}{45000}$

SISTEMAS BASADOS EN CONTENIDOS

↳ Recomendar basado en gustos previos
↳ Por cada item mantengo un profile → Calcular interes: $\cos\theta = \frac{xy}{\|x\| \|y\|}$
↳ Por cada usuario mantengo un profile

↳ Ej:

	1	2
ANIMATED	✓	x
MARVEL	x	✓
BECHDEL TEST	✓	✓

→ Perf. I.

Collaborative Filtering

↳ Suponiendo n usuarios y m items, construyo matriz de utilidad

	I1	I2	I3
U1	2	?	4
U2	?	3	4
U3	2	4	?

→ Valores a estimar

Dado el usuario X, busco los n usuarios mas similares y estimo base a eso

Calculo la semejanza entre usuarios como

$$sim(x,y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}}$$

	P1	P2	P3	P4	P5	P6	PROMEDIO
U1	2		2	4	5		13/4
U2	5		4			1	10/3
U3			5		2		7/2
U4		1		5		4	16/3
U5			4			2	6/2
U6	4	5		1			10/3

→ Resto los promedios a las calificaciones

Para estimar la calificación de un usuario a un item uso los n vecinos mas cercanos

$$r_{xi} = \frac{\sum_{y \in N} s_{xy} r_{yi}}{\sum_{y \in N} s_{xy}}$$

Calculo por desviación:

↪ $r_{ij} = \mu + \delta_i + \delta_j + \delta_{ij}$ donde:

μ = promedio de todas las calificaciones de todas las peliculas

δ_i = desviación usuario i respecto a μ

δ_j = desviación item j respecto a μ

δ_{ij} = desviación usuario i para pelicula j calculado como:

$$\delta_{ij} = \frac{\sum_{j \in N} s_{ij} (r_{ij} - b_{xj})}{} \quad \text{y} \quad b_{xj} = \mu + \delta_x + \delta_j$$

Factorization Machines

General Consulate of the Republic of Sierra Leone

Page Rank

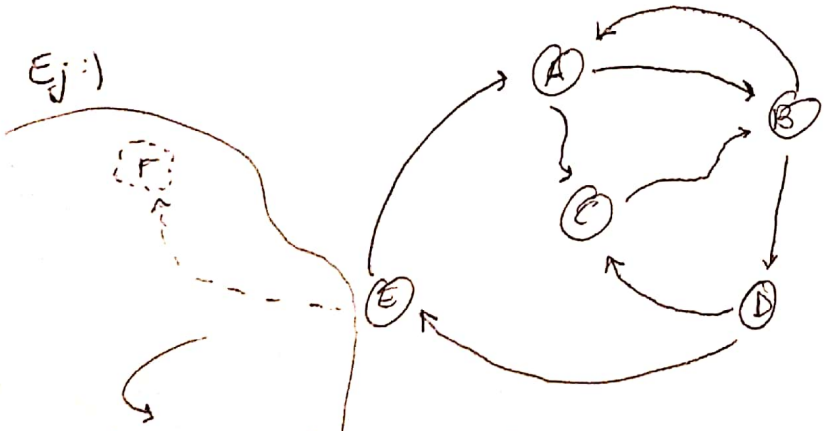
↳ Independiza el resultado del contenido del mismo
↳ Basado en la estructura de links de la web
↳ La importancia de un nodo viene de la importancia de lo que lo apuntan

Algoritmo

↳ Asignar a cada nodo $PR = \frac{1}{n}$

Realizar k veces:
↳ Cada nodo reparte su PR en partes iguales a los que apunta
↳ El nuevo PR es la suma de los pedazos de PR recibidos

Ej :)



	A	B	C	D	E
1	1/5	1/5	1/5	1/5	1/5
2	3/10	3/10	2/10	1/10	1/10
	5/20	7/20	4/20	3/20	1/20
	⋮				
	6/25	8/25	5/25	4/25	2/25

Solución para grafos con dead-end

Repartimos en partes iguales la proba del nodo entre todos los demas
(Sumo 1/n de la Probabilidad del nodo a cada nodo) → le reparto a todos, incluyendo

Otro Problema → Ciclos entre nodos que reparten entre ellos
↳ Introduzco Parametro β tal que con proba β elegimos un link al azar y co
Proba $(1-\beta)$ se teletransporta a cualquier pagina al azar

En gral, $0,8 < \beta < 0,9$

TRUST RANK

↳ APLICAR TELETRANSPORTACIÓN UNICAMENTE HACIA PAGINAS CONFIABLES

↳ CALCULO PR y TR

↳ MASA SPAM = (PR - PT)/PR

↳ ELIMINO LAS QUE SUPEREN UN LIMITE