Big Data:Volume;Velocity;Variety;Veracity;Value

Data;Information;Knowledge;Wisdom

Raw;Meaning;Context;Applied

Table;barchart;pie chart; stem and leaf plot

Stem	Leaf
6	0 1 1 4 4 4 4 6 6 8 8 8 9

Mode: The value appears most often

Range: Min to max

pth: p% are smaller

Interquartile range: First quartile=25%

LSF:

$$b = \frac{\sum_{i=1}^{n}(x_i - \overline{X})(y_i - \overline{Y})}{\sum_{i=1}^{n}(x_i - \overline{X})^2}$$ covariance / variance

Least square fit

$$a = \overline{Y} - b\overline{X}$$ Fit a line $Y = a + bX$ such that it minimizes the error S

Correlation Coefficient:>0 正 <0 负 ≈0不相关

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{X})(y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{X})^2 \sum_{i=1}^{n}(y_i - \overline{Y})^2}}$$

$Var(X) = E((X-\mu)^2) = E(X^2) - (E(X))^2$ σ^2

$$= \sum_{i=1}^{n}(x_i - \mu)^2 \times p(x_i)$$

Variance of a sample: $\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2$

Standard deviation of a sample: $\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2}$

Binomial Probability Model 二项分布 P53

$P(X=x) = C_n^x p^x (1-p)^{n-x}$ $X \sim bin(n, p)$

mean: $\mu = n \cdot p$ n次独立重复的伯努利实验

standard deviation: $\sigma = \sqrt{n \cdot p \cdot (1-p)}$

Poisson Probability Model 泊松分布 P56

$P(X=x) = e^{-\lambda}\frac{\lambda^x}{x!}$ $X \sim Po(\lambda)$

λ:单位时间内事件的平均发生次数

$\mu = \lambda$ 平均值(也等于期望)

$\sigma = \sqrt{\lambda}$ 标准差 单位时间内随机事件发生的次数

个离散：对应离分布函数

1连续：对应概率密度函数

Expectation and Variance of Continuous Random Variables 定义和离散都不同

期望值: $E(X) = \mu = \int_{-\infty}^{\infty} x f_X(x)dx$

方差: $Var(X) = \sigma^2 = \int_{-\infty}^{\infty}(x-\mu)^2 f_X(x)dx$

Normal Probability Model / Gaussian Distribution 正态分布

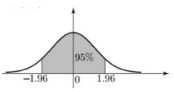
$y = f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\cdot\sigma^2}}$ $X \sim N(\mu, \sigma^2)$

Standard Normal Distribution 标准正态分布 P63

$y = f_X(x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{x^2}{2}}$ $X \sim N(0, 1)$

如果 X 服从 $N(\mu, \sigma^2)$，则 $Z = \frac{X-\mu}{\sigma}$ 服从 $N(0,1)$ 标准正态分布

Confidence Intervals 置信区间 P71



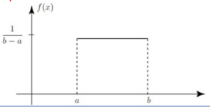
95%的 confidence intervals 是 (-1.96, 1.96)

计算真实值: $\mu \pm Z\frac{\sigma}{\sqrt{n}}$ → 样本数

Uniform Distribution 均匀分布

$X \sim U(a, b)$

$$y = f_X(x) = \begin{cases} \frac{1}{b-a}, & a \le x \le b \\ 0, & otherwise \end{cases}$$



概率论中期望即均值

Mean: 代入E(X)通式 $= \frac{b+a}{2}$

Variance: 代入 $= \frac{(b-a)^2}{12}$

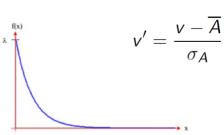
Exponential Distribution 指数分布 $X \sim Exp(\lambda)$

$$y = f_X(x) = \begin{cases} \lambda \cdot e^{-\lambda x}, & x > 0 \\ 0, & x \le 0 \end{cases}$$

λ: rate of the distribution

mean: $E(X) = \frac{1}{\lambda}$

$Var(X) = \frac{1}{\lambda^2}$



$v' = \frac{v - \overline{A}}{\sigma_A}$

贝叶斯公式

Bayes Theorem

- In many situations, you will know one conditional distribution $P(x|y)$ and $P(x)$ but you are really interested in the other conditional distribution $P(y|x)$.

$$P(y|x) = \frac{P(x,y)}{P(x)} = \frac{P(x|y)P(y)}{P(x)}$$

$P(y|x) \cdot P(x) = P(x|y)P(y)$ $= P(x,y)$

- Let $A_1, A_2, \ldots, A_n$ be a set of mutually exclusive events that together form the sample space S.
  Let B be any event from the same sample space, such that $P(B) > 0$. Then

$$P(A_k|B) = \frac{P(B|A_k)P(A_k)}{\sum_{i=1}^{n} P(B|A_i)P(A_i)} = P(B)$$

Example

- For a magazine, the probability that the reader is male given that the reader is at least 35 years old is 0.3. The probability that a reader is male, given that the reader is under 35, is 0.65. If 75% of the reader are under 35, what is the probability that a randomly chosen reader is
  a) Male
  b) Female
  c) Under 35 and it is given the reader is a female
- Solution:
- (a) Let A_1 be the event of the reader being at least 35 years old,
  A_2 the event of the reader being under 35 years old,
  M be the event of the reader is being a male, and
  F be the event of the reader is being a female.

$P(A_2) = 0.75, P(A_1) = 1 - 0.75 = 0.25$

$P(M|A_1) = 0.3, P(M|A_2) = 0.65$

$P(F|A_1) = 0.7, P(F|A_2) = 0.35$

$P(M) = P(A_1, M) + P(A_2, M)$ 同时满足两个条件

$= P(A_1)P(M|A_1) + P(A_2)P(M|A_2) = 0.25 \times 0.3 + 0.75 \times 0.65 = 0.5625$

- (b) $P(F) = 1 - P(M) = 1 - 0.5625 = 0.4375$
- (c) $P(A_2|F) = \frac{P(F|A_2)P(A_2)}{P(F|A_1)P(A_1)+P(F|A_2)P(A_2)} = \frac{0.35\times0.75}{0.7\times0.25+0.35\times0.75} = 0.6$

Quadratic Transformation

可都助分离数据

$Z = r_{1,1}B_1^2 + r_{1,2}B_1B_2 + r_{m-1,m}B_{m-1}B_m + r_{m,m}B_m^2$

Non-polynomial Approximations of Transformations 少数情况：如判断三角形是否全等

$A = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$

Rank Transformation 用于将值分布到高斯曲线上

$$y = \Phi^{-1}\left(\frac{r_i - \frac{3}{8}}{m + \frac{1}{4}}\right)$$

r_i the rank of the observation i

φ the cumulative normal function

Box-Cox Transformation

修正偏离量转变为近乎正态分布

$$y = \begin{cases} (x^\lambda - 1)/\lambda, & \lambda \neq 0 \\ log(x), & \lambda = 0 \end{cases}$$

The value of lambda is iteratively found by

- testing different values in the range from -3.0 to 3.0 in small steps until the resulting attribute is as close as possible to the normal distribution

All linear, inverse, quadratic and similar transformations are special cases of the Box-Cox transformations.

$$y = \begin{cases} ((x+c)^\lambda - 1) \div g\lambda, & \lambda \neq 0 \\ log(x+c) \div g, & \lambda = 0 \end{cases}$$ 有负数时使用

The parameter g is used to scale the resulting values, and it is often considered as the geometric mean of the data

Nominal to Binary Transformation

This simple approach has two big drawbacks:

- An ordering of the attribute values is assumed
- The integer values can be used in operations as numbers, whereas the nominal values cannot

If N is the number of different values the nominal attribute has, we will substitute the nominal variable with a new set of binary attributes, each one representing one of the N possible values.

For each instance, only one of the N newly created attributes will have a value of 1, while the rest will have the value of 0

This transformation is also referred in the literature as 1-to-N transformation.

Data Preprocessing

Data Cleaning; Transformation;Integration
Normalization;Missing Data Imputation;
Noise Identification;Feature Selection;
Instance Selection;Discretization;
Feature Extraction/Instance Generation

Categorical: Nominal & Ordinal

Numerical: Discrete & Continuous

Finding Redundant Attributes:

- χ^2 Correlation Test quantifies the correlation among two nominal attributes contain c and r different values each:

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

where o_{ij} is the frequency of (A_i, B_j) and:

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{m}$$

The larger the χ^2value, the more likely the variables are related

Data Reduction

- When the data set is very large, performing complex analysis and DM can take a long computing time
- Data reduction techniques
  - To reduce the size of the data set
  - At the same time, to maintain the integrity and the information of the original data set as much as possible
- The main strategies to perform data reduction are Dimensionality Reduction (DR) techniques
  - Aim to reduce the number of attributes or instances
  - Wavelet transforms
  - Principal Component Analysis (PCA).
  - Many techniques can be found for reducing the dimensionality in the number of instances, like the use of clustering techniques, parametric methods and so on.
  - The use of binning and discretization techniques is also useful to reduce the dimensionality and complexity of the data set.
  - They convert numerical attributes into nominal ones, thus drastically reducing the cardinality of the attributes involved.

卡方相关检验：适用于 nominal attributes

Chi-Square Correlation Test: An Example

(开)

	Play Chess	Not Play Chess	Sum (row)
Like science fiction	250	200	450
Not like science fiction	50	1000	1050
Sum	300	1200	1500

- Consider "Play Chess" and "Like science fiction", the expected value = $\frac{300\cdot450}{1500} = 90$
- The expected values for the above:

	Play Chess	Not Play Chess
Like science fiction	90	360
Not like science fiction	210	840

实际-期望/期望

$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$

=507.93

- It shows that "Like science fiction" and "Play Chess" are correlated.

Finding Redundant Attributes

Pearson's product moment coefficient 适用numerical attributes

$$r_{A,B} = \frac{\sum_{i=1}^{m}(a_i - \overline{A})}{m\sigma_A\sigma_B} = \frac{\sum_{i=1}^{m}(a_ib_i) - m\overline{AB}}{m\sigma_A\sigma_B}$$

covariance 适用于两个变量同时变化

$$Cov(A,B) = E((A-\overline{A})(B-\overline{B})) = \frac{\sum_{i=1}^{m}(a_i - \overline{A})(b_i - \overline{B})}{m}$$

$$r_{A,B} = \frac{Cov(A,B)}{\sigma_A\sigma_B}$$

Binning: Sort the data, partition into bins

Equal-depth bins:
- Bin1: 4, 8, 9, 15
- Bin2: 21, 21, 24, 25
- Bin3: 26, 28, 29, 34

Smooth by bin means:
- Bin1: 9, 9, 9, 9
- Bin2: 23, 23, 23, 23
- Bin3: 29, 29, 29, 29

Min-Max Normalization

$$v' = \frac{v - min_A}{max_A - min_A}(newmax_A - newmin_A) + newmin_A$$

For example, income range $12,000 to $98,000 to be normalized to [0.0, 1.0]

- The value $73,000 is transformed to the new value:

$$\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0.0) + 0.0 = 0.0716$$

Z-score Normalization 不知道最大最小值时用

For example, $\bar{A} = 54{,}000$ and $\sigma_A = 16{,}000$

$$v' = \frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$$

Decimal-scaling Normalization 减小绝对值用

$$v' = \frac{v}{10^j}$$

Linear Transformation 使用原始数据制造新数据

Normalization无法提升性能时使用

$Z = r_1B_1 + r_2B_2 + \ldots + r_mB_m$

Similarity

Manhattan distance: $m = 1,\ d = \sum_{i=1}^{n}|x_i - y_i|$

Euclidean distance: $m = 2,\ d = \sqrt{\sum_{i=1}^{n}|x_i - y_i|^2}$

Cosine similarity: $s = \frac{\sum_{i=1}^{n} x_iy_i}{\|x\|_2\|y\|_2}$

Pearson correlation: $s = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}}$

Similarity Measures: Sequence-based

Hamming Distance:d("peter", "pedro") = 3

Edit(Levenshtein):每次删除/插入/替换就+1

x = dva, y = dave

		y0	y1	y2	y3	y4
			d	a	v	e
x0		0	1	2	3	4
x1	d	1				
x2	v	2				
x3	a	3				

x变y

字母相同为copy:+0 不同为Substitute:+1	Delete:+1
Insert:+1	三个里面 min填这里

	y0	y1	y2	y3	y4
		d	a	v	e
x0	0	1	2	3	4
x1 d	1	0	1	2	3
x2 v	2	1	1	1	2
x3 a	3	2	1	2	2

The edit distance is 2.

To transform x to y: substitute 'a' with 'e' and insert 'a' (after 'd').

x = d - v a

y = d a v e

找一条连通的路

方向倒推回去就是步骤

相似性计算方法：

$$s(x,y) = 1 - \frac{d(x,y)}{max(length(x).length(y))}$$

$$s(\text{"dva"}, \text{"dave"}) = 1 - \frac{2}{max(3,4))} = 0.5$$

Needleman-Wunch计算相似度分数而不是距离值,Edit改进版

	d	v	a	e
d	2	-1	-1	-1
v	-1	2	-1	-1
a	-1	-1	2	-1
e	-1	-1	-1	2

+左边的分数矩阵	-空隙惩罚cg
-空隙惩罚cg	三个里面 max填这里

		d	e	v	e
	0	-1	-2	-3	-4	-5
d	-1	2	1	0	-1	-2
v	-2	1	1	0	2	1
a	-3	0	0	0	1	1 ←这个就是s(x,y)

The Needleman-Wunch score is 1 and the corresponding alignment is:

d - - v a

d e e v e

s(i, 0) = -i*c_g , s(0, j) = -j*c_g Cg为给定空隙惩罚

Affine Gap Measure 用于匹配项有很长的间隙,N-W的扩展

c_o is the cost of opening a gap,

c_r is the cost of continuing a gap, ←这三题目都会给

$c(x_i, y_j)$ is the score for corresponding character x_i with y_j in the score matrix.

Initialization:

- $M(0, 0) = 0$, $I_x(0, 0) = -c_o$, $I_y(0, 0) = -c_o$
- $I_x(i, 0) = - c_o - c_r * (i - 1)$
- $I_y(0, j) = - c_o - c_r * (j - 1)$
- Other cells in top row and leftmost column = $-\infty$

$$M(i,j) = max \begin{cases} M(i-1,j-1) + c(x_i, y_j) \\ I_x(i-1,j-1) + c(x_i, y_j) \\ I_y(i-1,j-1) + c(x_i, y_j) \end{cases}$$

$$I_x(i,j) = max \begin{cases} M(i-1,j) - c_o \\ I_x(i-1,j) - c_r \end{cases}$$

$$I_y(i,j) = max \begin{cases} M(i,j-1) - c_o \\ I_y(i,j-1) - c_r \end{cases}$$

例子：$c_o = 4$, $c_r = 1$

$$c(x_i, y_j) = \begin{cases} 1 & \text{if } x_i = y_j \\ -1 \text{ if } x_i \neq y_j \end{cases}$$

			A	C	A	C	T
	M(0,0) = 0		-∞	-∞	-∞	-∞	-∞
	Ix(0,0) = -4		-∞	-∞	-∞	-∞	-∞
	Iy(0,0) = -4		-4	-5	-6	-7	-8
	-∞		1	-5	-4	-7	-8
A	-4		-∞	-∞	-∞	-∞	-∞
	-∞		-∞	-3	-4	-5	-6
	-∞		-3	0	-2	-5	-6
A	-5		-3	-3	-9	-11	-12
	-∞		-∞	-7	-4	-5	-6
	-∞		-6	-4	-1	-3	-4
T	-6		-4	-4	-6	-9	-10
	-∞		-10	-8	-5	-6	-6

		A	C	A	C	T
M(0,0) = 0		-∞	-∞	-∞	-∞	-∞
Ix(0,0) = -4		-∞	-∞	-∞	-∞	-∞
Iy(0,0) = -4		-4	-5	-6	-7	-8
-∞		1	-5	-4	-7	-8
A	-4	-∞	-∞	-∞	-∞	-∞
-∞		-∞	-3	-4	-5	-6
-∞		-3	0	-2	-5	-6
A	-5	-3	-3	-9	-11	-12
-∞		-∞	-7	-4	-5	-6
-∞		-6	-4	-1	-3	-4
T	-6	-4	-4	-6	-9	-10
-∞		-10	-8	-5	-6	-6

涂红色块线：

Alignment 1:	Alignment 2:	Alignment 3:
ACACT __AAT	ACACT AA__T	ACACT A__AT

Smith-Waterman Measure 允许从任一位置匹配,而不是全局,N-W匹配改进版

		d	a	v	e
	0	0	0	0	0
a	0	0	2	1	0
v	0	0	1	4	3
d	0	2	1	3	3

s(i, 0) = 0, s(0, j) = 0

$$s(i,j) = max \begin{cases} 0 \\ s(i-1,j) - c_g \\ s(i,j-1) - c_g \\ s(i-1,j-1) + c(x_i, y_j) \end{cases}$$

C(xi,yj)是分数矩阵会给，cg是间隙惩罚也会给

Similarity score: $max_{i,j} s(i, j)$

Jaro Measure 短字符匹配,如姓氏

jaro(x, y) = 1/3 [c/|x| + c/|y| + (c – t/2)/c]

C是共同字符的数量，t是共同字符需转置的次数

x = jon, y =john

- 3 common character ('j', 'o' and 'n'), i.e. c = 3
- t = 0
- jaro(x, y) = 1/3 (3/3 + 3/4 + 3/3) = 0.917

x = jon, y =ojhn

- c = 3
- t = 2 jon vs ojn 需2次 transpose
- jaro(x, y) = 1/3 (3/3 + 3/4 + 2/3) = 0.81

Jaro-Winkler Measure Jaro分低但有共同前缀

*Jaro-winkler(x, y) = (1 - PL * PW) * jaro(x, y) + PL * PW*

PL is length of the longest common prefix, and

PW is a weight given to the prefix

Similarity Measures: Set-based

Overlap Measure

Let B_x = set of tokens generated for string x

Let B_y = set of tokens generated for string y

$$O(x,y) = |B_x \cap B_y|$$

E.g., *x* = dave, *y* = dav, considering 2-grams:

- B_x = { #d, da, av, ve, e# }, B_y = { #d, da, av, v# }

$$O(x,y) = 3$$

Jaccard Measure

$$J(x,y) = \frac{|B_x \cap B_y|}{|B_x \cup B_y|}$$

E.g., x = dave, y = dav, considering 2-grams

- B_x = { #d, da, av, ve, e# }, B_y = { #d, da, av, v# }

$$J(x,y) = \frac{3}{6}$$

TF/IDF Measure 重复字段长,真正识别段短

Term Frequency 术语频率

- *tf(t, d)* = number of times a term *t* appears in document *d*

Inverse Document Frequency 包含该术语的文档数

- *idf(t) = N / N_d*,

idf(t) is often defined as *log(N / N_d)*

For Example, assume $v_d(t) = tf(t, d) * idf(t)$

x = aab → B_x = { a, a, b }

y = ac → B_y = { a, c }

z = a → B_z = { a }

tf(a, x) = 2, *tf(b, x)* = 1, *tf(c, x)* = 0,

tf(a, y) = 1, *tf(b, y)* = 0, *tf(c, y)* = 1,

tf(a, z) = 1, *tf(b, z)* = 0, *tf(c, z)* = 0

idf(a) = 3/3 = 1, *idf(b)* = 3/1 = 3, *idf(c)* = 3/1 = 3

文档数/包含该术语的文档数

Feature vector of string x

	a	b	c
v_x	2	3	0
v_y	1	0	3
v_z	1	0	0

Hybrid Similarity Measures

Generalized Jaccard Measure 适用于有拼写错误

Let $B_x = x_1, x_2, \ldots, x_n$, $B_y = y_1, y_2, \ldots, y_m$

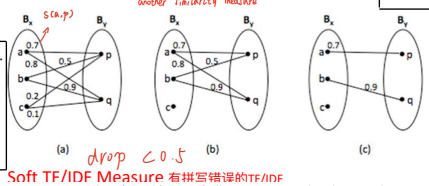
Step 1: find token pairs that will be in the "softened" overlap set

- apply a similarity measure s to compute sim score for each pair (x_i, y_j)
- keep only those score ≥ a given threshold , this forms a bipartite graph *G*
- find the maximum-weight matching *M* in *G* 二分图

Step 2: return normalized weight of *M* as generalized Jaccard score

$$GJ(x,y) = \sum_{(x_i,y_j) \in M} s(x_i, y_j) / (|B_x| + |B_y| - |M|)$$

another similarity measure



drop 0.5

Soft TF/IDF Measure 有拼写错误的TF/IDF

Step 1: compute *close(x, y, k)*: set of all terms $t \in B_x$ that have at least one close term $u \in B_y$, i.e., *s'(t, u)* ≥ *k*

- s' is a basic sim measure (e.g., Jaro-Winkler) and k prespecified

Step 2: compute s(x, y) as in traditional TF/IDF score, but weighing each TF/IDF component using s'

$$s(x,y) = \sum_{t \in close(x,y,k)} v_x(t) \cdot v_y(u_*) \cdot s'(t, u_*)$$

$u_* \in B_x$ maximizes $s'(t, u) \forall u \in B_y$

x = abcd; y = aa'b'cd'

$B_x = \{a, b, c, d\}$

1 0.7 0.8 1 0.6

$B_y = \{a, a', b', c, d'\}$

close(x, y, 0.75) = {a, b, c}

K 预先设定的门槛

$$s(x,y) = v_x(a) \cdot v_y(a) \cdot 1 + v_x(b) \cdot v_y(b') \cdot 0.8 + v_x(c) \cdot v_y(c) \cdot 1$$

Monge-Elkan 分解后再匹配

x = "peter christen", y = "christian pedro"

Using Jaro measure as the secondary similarity measure

- s'("peter", "christian") = 0.3741
- s'("peter", "pedro") = 0.7333
- s'("christen", "christian") = 0.8843
- s'("christen", "pedro") = 0.4417

s("peter christen", "christian pedro") = ½ (0.7333 + 0.8843) = 0.8088

Phonetic Similarity Measures

用于匹配人名，soundex最常用

From Sets to Boolean Matrices

Shingles	Documents	S_1	S_2	S_3	S_4
	A	1	0	1	0
	B	1	0	0	1
	C	0	1	0	1
	D	0	1	0	1
	E	0	1	0	1
	F	1	0	1	0
	G	1	0	1	0

For example, sim(S_1, S_2) =?

- Size of intersection = 0; size of union = 7,
- Jaccard Similarity (not distance) = 0/7 = 0,
- Distance: d(S_1, S_2) = 1 - (Jaccard Similarity) = 1

sim(S_3, S_4) =?

- Size of intersection = 3; size of union = 4,
- Jaccard similarity = 3/4 = 0.75,
- Distance: d(S_3, S_4) = 1/4

随机置换1

所有位置置换1

LSH Analysis – An Example

Case 1: Documents S_1 and S_2 are 80% similar

- Find pairs of documents with similarity ≥ t where t = 0.8, set b = 20, r = 5
- Assume: sim(S_1, S_2) = 0.8
- Since sim(S_1, S_2) ≥ t, we want S_1 and S_2 to be a candidate pair:
  - We want them to hash to at least 1 common bucket (at least one band is identical)
- Probability that S_1 and S_2 are identical in one particular band: $(0.8)^5 = 0.328$
- Probability that S1 and S2 are not similar in all of the 20 bands: $(1 - 0.328)^{20} = 0.00035$
- i.e., about 1/3000th of the 80%-similar column pairs are false negatives (we miss them)
- We would find 99.965% pairs of truly similar documents 判断为错，但实际上为对

Case 2: Documents S_1 and S_2 are 30% similar 小,不配对

Find pairs of documents with similarity ≥ t where t = 0.8, set b = 20, r = 5

Assume: sim(S_1, S_2) = 0.3

Since sim(S_1, S_2) < t,

- We want them to hash to NO common bucket (all bands should be different

Probability that S_1 and S_2 are identical in one particular band: $(0.3)^5 = 0.00243$

Probability that S_1 and S_2 are identical in at least 1 of the 20 bands: $1 - (1 - 0.00243)^{20} = 0.0474$

In other words, approximately 4.74% pairs of docs with similarity 0.3 end up becoming candidate pairs

They are false positives since we will have to examine them (they are candidate pairs) but then it will turn out their similarity is below threshold t

LSH Analysis – b Bands, r Rows per Band

Columns S_1 and S_2 have similarity s

Pick any band (r rows)

- Probability that all rows in band equal = s^r
- Probability that some row in band unequal = 1 - s^r

Probability that no band identical = $(1 - s^r)^b$

Probability that at least 1 band identical = $1 - (1 - s^r)^b$

Example: b = 20, r = 5

Similarity threshold t

Probability that at least 1 band is identical

t	$1 - (1 - s^r)^b$
0.2	0.006
0.3	0.047
0.4	0.186
0.5	0.470
0.6	0.802
0.7	0.975
0.8	0.9996

增长迅速

Three types of machine learning

- **Supervised learning**
  - The training set given for supervised learning is the labeled dataset.
  - It tries to find the relationships between the feature set and the label set.
- **Unsupervised learning**
  - The training set given for unsupervised learning is the unlabeled dataset.
  - It aims at clustering, probability density estimation, finding association among features.
- **Reinforcement learning**
  - It is used to solve problems of decision making (usually a sequence of decisions), e.g., automatic chess player and automatic vehicle driving.

Confusion Matrix

		Predicted Classes: Yes	Predicted Classes: No
Actual Classes	Yes	True Positive (TP)	False Negative (FN)
	No	False Positive (FP)	True Negative (TN)

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Error rate} = \frac{FP + FN}{TP + TN + FP + FN}$$

$$\text{F measure} = \frac{2\ Recall\ Precision}{Recall + Precision}$$

$$\text{specificity} = \frac{TN}{TN + FP}$$

$$\text{fpr} = 1 - \text{specificity}$$

Overfitting and Underfitting

Generalization

- The ability of an algorithm to perform well on unseen samples

Training error:

- Error on training dataset

Generalization error:

- Test error
- Expected value of the error on a new input, i.e. test dataset

Machine learning algorithm should

- Reduce the training error
- Reduce the gap between training error and generalization (test) error

Overfitting

- an analysis that fits too closely to a particular set of data, and
- thus may fail to fit additional data (future observations) reliably
- Low training error yet high test error
  - The gap between the training error and test error is too large

Underfitting

- occurs when the model or algorithm does not fit the data well enough
- Not able to obtain a sufficiently low error value on the training dataset
- Both high training error and test error
- Often caused when model is simpler compared with data

Variance: the variability of model prediction for a given data point

意味:Pay a lot of attention to training data and does not generalize on the unseen data

High variance: model fits random noise of data, rather than valid information (overfitting)

High bias: high error on training and testing set

意味:Pay little attention to training data and oversimplifies the model

Errors: Bias and Variance

Expect error = Bias² + Variance + Noise

Bias: difference between the average prediction of our model and the correct value which we are trying to predict

What is a "complex" model?

Complex models are those "flexible" models that can fit the data as well as possible

For examples:

- Polynomial models with higher degrees, they can fit data that lower degree model cannot do.
- Models with larger parameters, they can fit data that model with small parameters cannot do.

Complex models have low bias and high variance:

- Low bias: complicated models capture a lot of feature in the training data, perform well on training
- High variance: testing set may not have the same feature
- Overfitting!

Simple models have low variance and high bias:

- Just the other way round for complicated models
- Underfitting!

Naïve bayes 计算会四条件是否打球

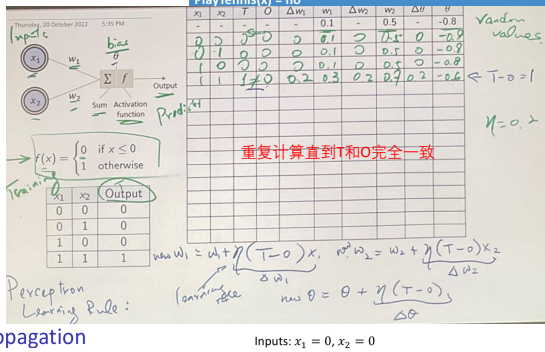
Answer:

P(yes)P(Sunny|yes)P(Cool|yes)P(High|yes)P(Strong|yes)
= 0.64 × 0.22 × 0.33 × 0.33 × 0.33 = 0.0051

P(no)P(Sunny|no)P(Cool|no)P(High|no)P(Strong|no)
= 0.36 × 0.60 × 0.20 × 0.80 × 0.60 = 0.0207

PlayTennis(x) = no

Artificial Neural Network计算



h(s1)->sig(s1) s1,s2的sig相同,则sim(s1,s2)高

An example of computing Minhash signatures

Consider the following matrix of four sets:

S1 = {a, d}, S2 = {c}, S3 = {b, d, e}, and S4 = {a, c, d}.

Element	row #	Sets S1	S2	S3	S4
a	0	1	0	0	1
b	1	0	0	1	0
c	2	0	1	0	1
d	3	1	0	1	1
e	4	0	0	1	0

1.根据字符串绘制矩阵,存在就填1,否则填0

row #	$h_1(r) = (r + 1) \bmod 5$	$h_2(r) = (3r + 1) \bmod 5$
0	1	1
1	2	4
2	3	2
3	4	0
4	0	3

2.根据行号计算替换用的哈希矩阵

	S1	S2	S3	S4
h_1	∞	∞	∞	∞
h_2	∞	∞	∞	∞

3.初始化签名矩阵,准备开始计算

4.看row0,只有S1和S4为1，所以替换这两;看S1,替换矩阵是1,1,都比无穷小,所以填1,1,S4同理

例如：

SIM(S1, S4) = 2/2 = 1.0.

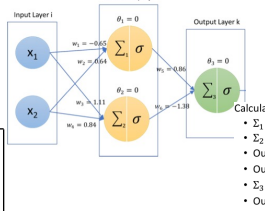
而原版的jaccard simi.为2/3 = 0.67,原版计算只看1,哈希后的1,0都要看

	S1	S2	S3	S4
h_1	1	∞	∞	1
h_2	1	∞	∞	1

5.依次计算完所有5行,得到如下的矩阵,即为signature matrix,可用于计算Jaccard similarities

	S1	S2	S3	S4
h_1	1	3	0	1
h_2	0	2	0	0

Forward Propagation



Inputs: $x_1 = 0, x_2 = 0$

Actual Output: $T_k = 0$

Learning rate: $\eta = 0.5$

Weights: $w_1 = -0.65, w_2 = 0.64, w_3 = 1.11, w_4 = 0.84, w_5 = 0.86, w_6 = -1.38$

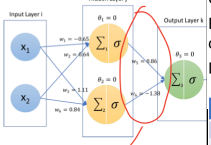
Biases: $\theta_1 = 0, \theta_2 = 0, \theta_3 = 0$

$$\sigma(\Sigma + \theta) = \frac{1}{1 + e^{-(\Sigma + \theta)}}$$

Calculations:

- $\Sigma_1 + \theta_1 = w_1 x_1 + w_2 x_2 + \theta_1 = (-0.65)0 + (0.64)0 + 0 = 0$
- $\Sigma_2 + \theta_2 = w_3 x_1 + w_4 x_2 + \theta_2 = (1.11)0 + (0.84)0 + 0 = 0$
- Output $O_{j1} = \sigma(\Sigma_1 + \theta_1) = \sigma(0) = 0.5$
- Output $O_{j2} = \sigma(\Sigma_2 + \theta_2) = \sigma(0) = 0.5$
- $\Sigma_3 + \theta_3 = w_5 O_{j1} + w_6 O_{j2} + \theta_3 = (0.86)0.5 + (-1.38)0.5 + 0 = -0.26$
- Output $O_{k1} = \sigma(\Sigma_3 + \theta_3) = \sigma(-0.26) = 0.435364$

Example – Round 1, Step 1, Backward Propagation



Calculations:

- Output $O_{j1} = \sigma(\Sigma_1 + \theta_1) = \sigma(0) = 0.5$
- Output $O_{j2} = \sigma(\Sigma_2 + \theta_2) = \sigma(0) = 0.5$
- Output $O_k = \sigma(\Sigma_3 + \theta_3) = \sigma(-0.26) = 0.435364$
- $\delta_k = (O_k - T_k)O_k(1 - O_k)$
  $= (0.435364 - 0)0.435364(1 - 0.435364)$
  $= 0.107022$

Update the weights and bias:

- $New\ w_5 = old\ w_5 - \eta\ \delta_k\ O_{j1} = 0.86 - 0.5(0.107022)(0.5) = 0.833245$
- $New\ w_6 = old\ w_6 - \eta\ \delta_k\ O_{j2} = -1.38 - 0.5(0.107022)(0.5) = -1.406756$
- $New\ \theta_3 = old\theta_3 - \eta\ \delta_k = 0 - 0.5(-0.107022) = -0.053511$

$\delta_{j1} = O_{j1}(1 - O_{j1}) \sum_{k \in K} \delta_k\ w_{jk}$
$= 0.5\ (1 - 0.5)(0.86)(0.107022) = 0.023010$

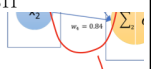
Update the weights and biases:

- $New\ w_1 = old\ w_1 - \eta\ \delta_{j1}\ x_1 = -0.65 - 0.5(0.023010)(0) = -0.65$
- $New\ w_2 = old\ w_2 - \eta\ \delta_{j1}\ x_2 = 0.64 - 0.5(0.023010)(0) = 0.64$
- $New\ \theta_1 = old\ \theta_1 - \eta\ \delta_{j1} = 0 - 0.5(0.023010) = -0.011505$

$\delta_{j2} = O_{j2}(1 - O_{j2}) \sum_{k \in K} \delta_k\ w_{jk} = 0.5\ (1 - 0.5)(-1.38)(0.107022) = -0.036923$

Update the weights and biases:

- $New\ w_3 = old\ w_3 - \eta\ \delta_{j2}\ x_1 = 1.11 - 0.5(-0.036923)(0) = 1.11$
- $New\ w_4 = old\ w_4 - \eta\ \delta_{j2}\ x_2 = 0.84 - 0.5(-0.036923)(0) = 0.84$
- $New\ \theta_2 = old\ \theta_2 - \eta\ \delta_{j2} = 0 - 0.5(-0.036923) = -0.018462$



Constraint Enforcer 附加规则选匹配结果

Combiner produces match combinations

Enforcer is to prune the match combinations domain integrity constraints

Domain integrity constraints

- Domain knowledge

Average-combiner:
name = <name: 0.6, title: 0.5>
releaseInfo = <releaseDate: 0.6, releaseCompany: 0.25>
classification = <rating: 0.3>

Combination	Aggregated Score
M1 **name** = name, **releaseInfo** = releaseDate, **classification** = rating, **price** = basePrice,	0.6*0.6*0.3*0.5 = 0.054
M2 **name** = title, **releaseInfo** = releaseDate, **classification** = rating, **price** = basePrice	0.5*0.6*0.3*0.5 = 0.045
M3 **name** = name, **releaseCompany**, **classification** = rating, **price** = basePrice	0.6*0.25*0.3*0.5 = 0.0225
M4 **name** = name, **releaseCompany**, **classification** = rating, **price** = basePrice	0.5*0.25*0.3*0.5 = 0.0188

In M1, name of AGGREGATOR matches with name of DVD-VENDOR

In M2, name of AGGREGATOR matches with title of DVD-VENDOR

A constraint:

- If an attribute A matches name, then in any random sample of 100 data values of A, at least 10 values must contain four words or more.

M1 does not satisfy the constraint, M2 is selected

Match Selector 大于阈值则匹配

Produces matches from the similarity matrix

Thresholding

- Simplest strategy
- For example, threshold = 0.5

name = <title: 0.5>
releaseInfo = <releaseDate: 0.6>
classification = <rating: 0.3>
price = <basePrice: 0.5>

Matches selected:
name = title
releaseInfo = releaseDate
price = basePrice

Learning the CPTs Given No Training Data

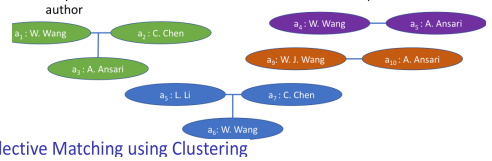
(a_4, b_4)	<?, $s_1(a_4, b_4)$, $s_2(a_4, b_4)$, $s_3(a_4, b_4)$ >
(a_5, b_5)	<?, $s_1(a_5, b_5)$, $s_2(a_5, b_5)$, $s_3(a_5, b_5)$ >
(a_6, b_6)	<?, $s_1(a_6, b_6)$, $s_2(a_6, b_6)$, $s_3(a_6, b_6)$ >

- Assume (a4,b4), (a5,b5), (a6,b6) are tuple pairs to be matched
- Convert these pairs into training data with missing values
- Now apply EM algorithm to learn both the CPTs and the missing values at the same time
- once learned, the missing values are the labels (i.e., "matched", "not matched") that we want to see

Collective Matching 集体匹配:Matching are often correlated

	First initial	Middle initial	Last name
a_1	W		Wang
a_2	C		Chen
...	...	...	...
a_9	W	J	Wang
a_{10}	A		Ansari

- nodes = authors, hyperedges connect co-authors
- Suppose we have matched a_3 and a_5
  - then intuitively a_1 and a_4 should be more likely to match
  - they share the same name and same co-author relationship to the same author



Rule-based Matching

Linearly Weighted Combination Rules 简单但容易收益递减

Table X

	Name	Phone	City	State
x_1	Dave Smith	(608) 3959462	Madison	WI
x_2	Joe Wilson	(408) 123 4265	San Jose	CA
x_3	Dan Smith	(608) 256 1212	Middleton	WI

Table Y

	Name	Phone	City	State
y_1	David D. Smith	395 9426	Madison	WI
y_2	Daniel W. Smith	256 1212	Madison	WI

$sim(x, y) = 0.3sim_{name}(x, y) + 0.3sim_{phone}(x, y) + 0.1sim_{city}(x, y) + 0.3\ sim_{state}(x, y)$

- $sim_{name}(x, y)$: based on Jaro-Winkler
- $sim_{phone}(x, y)$: based on edit distance between x's phone (after removing area code) and y's phone
- $sim_{city}(x, y)$: based on edit distance
- $sim_{state}(x, y)$: based on exact match; yes → 1, no → 0

Logistic Regression Rules 解决收益递减

address the diminishing-returns problem

$sim(x, y) = 1/(1 + e^{-z})$, where $z = \sum_{i=1}^{n} \alpha_i * sim_i(x, y)$

- here α_i is not constrained to be in [0, 1] and sum to 1
- so z goes from $-\infty$ to $+\infty$, in which case $sim(x, y)$ gradually increases, but minimally so after z has exceeded a certain value
- → ensuring diminishing returns

Learning-based Matching

Applying model *M* to match new tuple pairs

- given pair (x, y), transform it into a feature vector
  - $v = < f_1(x, y), \ldots, f_m(x, y) >$

apply *M* to *v* to predict whether *x* matches *y* Example: Learning a Linearly Weighted Rule

<a_1 = (Mike Williams, (425) 247 4893, Seattle, WA), b_1 = (M. Williams, 247 4893, Redmond, WA), yes>
<a_2 = (Richard Pike, (414) 256 41257, Milwaukee, WI), b_2 = (R. Pike, 256 1237, Milwaukee, WI), yes>
<a_3 = (Jane McCain, (206) 111 4215, Renton, WA), b_3 = (J. M. McCain, 112 5200, Renton, WA), no>

match names | match phones | match cities | match states | check area code against city

$v_1 = <[s_1(a_1, b_1), s_2(a_1, b_1), s_3(a_1, b_1), s_4(a_1, b_1), s_5(a_1, b_1), s_6(a_1, b_1)], 1>$ yes
$v_2 = <[s_1(a_2, b_2), s_2(a_2, b_2), s_3(a_2, b_2), s_4(a_2, b_2), s_5(a_2, b_2), s_6(a_2, b_2)], 1>$ yes
$v_3 = <[s_1(a_3, b_3), s_2(a_3, b_3), s_3(a_3, b_3), s_4(a_3, b_3), s_5(a_3, b_3), s_6(a_3, b_3)], 0>$ no

s_1 and s_2 use Jaro-Winkler and edit distance

s_3 uses edit distance (ignoring area code of a)

s_4 and s_5 return 1 if exact match, 0 otherwise

s_6 encodes a heuristic constraint

Goal: learn rule:

- $s(a, b) = \sum_{i=1}^{6} \alpha_i s_i(a, b)$

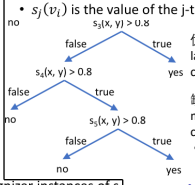
Perform a least-squares linear regression on training data c_j

- $v_1 = < [s_{1(a_1,b_1)}, s_{2(a_1,b_1)}, s_{3(a_1,b_1)}, s_{4(a_1,b_1)}, s_{6(a_1,b_1)}], 1 >$
- $v_2 = < [s_{1(a_2,b_2)}, s_{2(a_2,b_2)}, s_{3(a_2,b_2)}, s_{4(a_2,b_2)}, s_{6(a_2,b_2)}], 1 >$
- $v_3 = < [s_{1(a_3,b_3)}, s_{2(a_3,b_3)}, s_{3(a_3,b_3)}, s_{4(a_3,b_3)}, s_{6(a_3,b_3)}], 0 >$

to find weights α_i that minimizes the squared error:

$\sum_{i=1}^{3} \left(c_i - \sum_{j=1}^{6} \alpha_j s_j(v_i)\right)^2$

- c_i is the label associated with v_i
- $s_j(v_i)$ is the value of the j-th element of v_i



优点:can automatically examine a large number of features and use complex rules.

缺点: still require training examples, in many cases a large number of them, which can be hard to obtain

- clustering addresses this problem

Collective Matching using Clustering

- sim(A, B) = $sim_{attributes}(A, B) + (1 - \alpha) sim_{neighbors}(A, B)$
- α is pre-defined weight
- $sim_{attributes}(A, B)$
  - uses only attributes of A and B, examples of such scores are single link, complete link, average link, etc.
- $sim_{neighbors}(A, B)$
  - considers correlations

we define a Jaccard measure as

$JaccarSim_{coauthors}(a, b) = \frac{|coAuthors(a) \cap coAuthors(b)|}{|coAuthors(a) \cup coAuthors(b)|}$

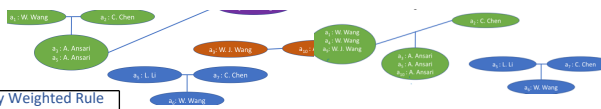
Contrast that to

$sim_{\{neighbors\}}(A, B) = Jaccard(N(A), N(B)) = \frac{|N(A) \cap N(B)|}{|N(A) \cup N(B)|}$

In the former, we assume two co-authors match if their "strings" match

In the latter, two co-authors match only if they have the same cluster ID

示意图：



Scaling up Rule-based Matching 扩大匹配规模

- Two goals:
  1. minimize # of tuple pairs to be matched
  2. minimize time it takes to match each pair

For the second goal
no well-established technique as yet
tailor depending on the application and the matching approach

For the first goal:

- hashing
- sorting
- indexing
- canopies
- using representatives
- combining the techniques

Data Integration = alignment + linkage + fusion

- Schema alignment: mapping of structure
- Record linkage: matching based on content
- Data fusion: reconciliation of mismatching content

Schema Alignment → Record Linkage (调和) → Data Fusion

Schema Alignment: Three Steps

Mediated Schema	Attributes Matching	Schema Mapping
enables domain specific modelling	identifies correspondences between schema attributes	specifies transformation between records in different schemas

Mediated Schema: Simple Example 1

Possible instances of mediated schema:

- Source 1: (Director, Title, Year) with tuples
  - (Allen, Manhattan, 1979), (Coppola, GodFather, 1972)

Mediated schema: (Title, Year)

- Simple projection of Source 1
- Only one possible instance:
  - (Manhattan, 1979), (GodFather, 1972)

Schema Matching and Mapping: Problem Definition

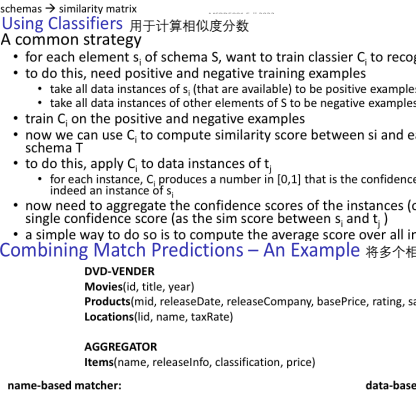
Let S and T be two relational schemas

DVD-VENDER
Movies(id, title, year)
Products(mid, releaseDate, releaseCompany, basePrice, rating, saleLocID)
Locations(lid, name, taxRate)

AGGREGATOR
Items(name, releaseInfo, classification, price)

A semantic mapping is a query expression that relates a schema S with a schema T

- the following mapping shows how to obtain Movies.title:
  SELECT name AS title
  FROM Items
- The following mapping shows how to obtain Items.price:
  - SELECT (basePrice * (1 + taxRate)) AS price FROM Products, Locations
    WHERE Products.saleLocID = Locations.lid
- The following mapping shows how to obtain an entire tuple for Items table of AGGREGATOR:
  - SELECT title AS name, releaseDate AS releaseInfo,
    rating AS classication, basePrice * (1 + taxRate) AS price
    FROM Movies, Products, Locations
    WHERE Movies.id = Products.mid AND Products.saleLocID = Locations.lid



Using Classifiers 用于计算相似度分数

A common strategy

- for each element s_i of schema S, want to train classier C_i to recognizer instances of s_i
- to do this, need positive and negative training examples
  - take all data instances of s_i (that are available) to be positive examples
  - take all data instances of other elements of S to be negative examples
- train C_i on the positive and negative examples
- now we can use C_i to compute similarity score between si and each element t_j of schema T
- to do this, apply C_i to data instances of t_j
  - for each instance, C_i produces a number in [0,1] that is the confidence that the instance is indeed an instance of s_i
- now need to aggregate the confidence scores of the instances (of t_j) to return a single confidence score (as the sim score between s_i and t_j)
- a simple way to do so is to compute the average score over all instances of t_j

Combining Match Predictions – An Example 将多个相似度合并成一个

DVD-VENDER
Movies(id, title, year)
Products(mid, releaseDate, releaseCompany, basePrice, rating, saleLocID)
Locations(lid, name, taxRate)

AGGREGATOR
Items(name, releaseInfo, classification, price)

name-based matcher:
name = <name: 1, title: 0.2>
releaseInfo = <releaseDate: 0.5, releaseCompany: 0.5>
price = <basePrice: 0.8>

data-based matcher:
name = <name: 0.2, title: 0.8>
releaseInfo = <releaseDate: 0.7>
classification = <rating: 0.6>
price = <basePrice: 0.2>

Average-combiner:
name = <name: 0.6, title: 0.5>
releaseInfo = <releaseDate: 0.6, releaseCompany: 0.25>
classification = <rating: 0.3>
price = <basePrice: 0.5>

Learning the CPTs Given Training Data 条件概率表

- Convert training examples into feature vectors
- Then apply learning with no missing values (i.e. counting) to feature vectors to learn the CPTs

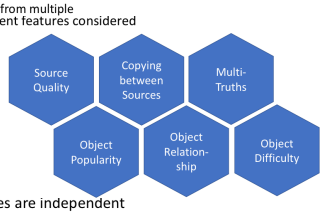
(a_1, b_1, yes)	<t, $s_1(a_1, b_1)$, $s_2(a_1, b_1)$, $s_3(a_1, b_1)$ >
(a_2, b_2, yes)	<t, $s_1(a_2, b_2)$, $s_2(a_2, b_2)$, $s_3(a_2, b_2)$ >
(a_3, b_3, no)	<f, $s_1(a_3, b_3)$, $s_2(a_3, b_3)$, $s_3(a_3, b_3)$ >

Once learned, can apply the BN to match a new pair of tuples (a_4, b_4) by comparing $P(M = t|a_4, b_4)$ and $P(M = f|a_4, b_4)$

Data Fusion

Data Fusion aims to detect and solve data conflicts from multiple sources

Different features considered



Source Quality

- Evaluates the trustworthiness of each data source according to the correctness of the values it provides.
- Measures the quality of a source by using metrics such as accuracy, exactness, freshness, reputation, precision, and recall.

Copying between Sources

- Many methods assume that data sources are independent
- However, in the real world, copying exists between data sources
- There are two types of data source:
  - Independent sources
    - provide values independently
  - Copier sources
    - copy part (or all) of data from other sources
- Detecting copying between data sources
- Discounted vote count can be assigned to a copied value in voting

Object Relationship

- Many approaches assume that objects are independent of each other
- However. objects can have relationships

Object Popularity

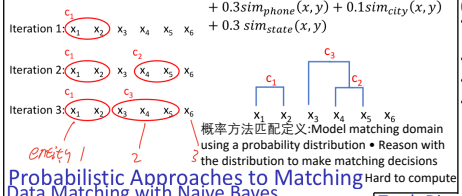
- More people will be misled if a popular object has a false value than when a less popular object has a false value
- Sources that provide false values for popular objects can be penalized more heavily

Object Difficulty

- Objects are usually treated equally, and the characteristics of each object are not considered
- However, objects have different levels of difficulty.
- For some objects, it is easier to find the truth than it is for others
- The difficulty of objects is modeled as uncertainty
  - If the object is extremely hard (i.e., if it is challenging to infer the real true information of the object), then the object has a high uncertainty value
  - If few sources provide claims on the object, then, since there is insufficient data, estimating the truth becomes more difficult, and so the object also has a high uncertainty value

Matching by Clustering

Matching by Clustering - Example



概率方法匹配定义:Model matching domain using a probability distribution • Reason with the distribution to make matching decisions Hard to compute

Probabilistic Approaches to Matching

Data Matching with Naive Bayes

Define variable M that represents whether a and b match

Our goal is to compute $P(M|a, b)$

- declare a and b matched if $P(M = t|a.b) > P(M = f|a.b)$

$P(M|S_1, \cdots, S_n) = \frac{P(S_1, \cdots, S_n|M)\ P(M)}{P(S_1, \cdots, S_n)}$

Assume $S_1, \cdots, S_n$ are independent given M

$P(S_1, \cdots, S_n|M) = \prod_{i=1}^{n} P(S_i|M)$

We only need to know:
$P(S_1|M), \ldots, P(S_n|M)$ *and* $P(M)$

Truth Discovery Problem Definition

- An observation is a tuple of the following:
  - o - an object of interest
  - s - a source where the object of interest is collected from
  - v_o^s - a value provided by s about o
- v_o^* - the identified truth for the object o
- w_s - source weight
  - The higher the weight, the more reliable s is.
- For a set of objects O, where related information are collected from a set of sources S
- Find the truth v_o^* for each object o by resolving the conflicts among the information from different sources $\{v_o^s\}_{s \in S}$.
- Truth discovery methods estimates source weights $\{w_s\}_{s \in S}$ that will be used to infer truths.

Category of Approaches

- Rule-based
- Probability-based
- Optimization-based
- Machine learning

Rule-based Approaches
- Early Approaches
  - Using the observed value from the most recently updated source
  - Taking the average, maximum, or minimum for numerical values
  - Majority voting
- Advantage of these approaches
  - The result is generally easier to debug and to understand

	S1	S2	S3
Jagadish	UM	ATT	UM
Dewitt	MSR	MSR	UW
Bernstein	MSR	MSR	MSR
Carey	UCI	ATT	BEA
Franklin	UCB	UCB	UMD

	S1	S2	S3
Jagadish	UM	ATT	UM
Dewitt	MSR	MSR	UW
Bernstein	MSR	MSR	MSR
Carey	UCI	ATT	BEA
Franklin	UCB	UCB	UMD

Naïve Voting
- Supports difference of opinion, allows conflict resolution
- Works well for independent sources that have similar accuracy
- When sources have different accuracies
  - Need to give more weight to votes by knowledgeable sources
- When sources copy from other sources
  - Need to reduce the weight of votes by copiers
- Problem: the wisdom of minority

Iterative Approach
- Recent studies combine iteratively estimating the quality of the source with truth discovery
- The basic principle is:
  - Sources that provide true information are more reliable, and
  - it is more likely that the information provided by reliable sources is true
- Iteratively until converges:
  - Truth computation step
  - Source weight estimation step
- Truth computation:
  - The truth is inferred through weighted voting.
  - The votes of a value v: $\text{vote}(v) = \left(\sum_{s \in S_v} \frac{w_s}{|V_s|}\right)^g$
  - S_v - a set of sources providing the value v
  - $|V_s|$ - the number of values by source s
  - g- the growth of belief, e.g. 1.2
- Source weight estimation: $w_s = \left(\sum_{v \in V_s} \text{vote}(v) \frac{\frac{w_s}{|V_s|}}{\sum_{s' \in S_v} \frac{w'_s}{|V's|}}\right)^g$

Optimization-based Approaches
- Similar to those iterative-based methods
- Models the truth discovery as an optimization problem
- Infer the reliability of the source and reliable information and to update truths and reliability weights from sources, iteratively to convergence

$$\arg\min_{\{w_s\},\{v_o^*\}} \sum_{o \in O} \sum_{s \in S} w_s \cdot d(v_o^s, v_o^*)$$

Probability-based Approaches
- Uses probabilistic models to jointly calculate the reliability of the source and the correctness of the values
- For example:
  - The likelihood:

$$\prod_{s \in S} p(w_s|\beta) \prod_{o \in O} \left(p(v_o^*|\alpha) \prod_{s \in S} p(v_o^s|v_o^*, w_s) \right)$$

Big Data Privacy Issues
- User's privacy may be breached under the following circumstances:
  - Personal information when combined with external datasets may lead to the inference of new facts about users.
  - Personal information is collected and used to add value to business.
  - Sensitive data are stored and processed in a location not secured properly and data leakage may occur during the storage and processing phases.

Four Types of Attributes in Data
- Identifier
  - Can be used to uniquely identify a person
  - For example: name, driving license number, mobile number
- Quasi-identifier
  - Cannot uniquely identify a person by the attribute
  - May be used to re-identify a person when linked with some external dataset
  - For example: age, gender, postcode
- Sensitive attribute
  - Attributes that a person may want to conceal
  - For example: salary, disease
- Non-sensitive attribute
  - Privacy of a person will not be violated when the attribute is disclosed

Privacy Protection Methods

Anonymization
- Anonymization (De-identification) of personal records
  - The subject identity of the data records are removed, concealed or hidden
- Can be performed by either the users or data providers
  - Users may perform anonymization before doing analytics on the big datasets
  - Providers may perform anonymization before storing the datasets which then be used, sold or share

Re-identification Attacks 可能会面临重识别攻击
- Even when identifying details are removed, individuals can still be re-identified
- An intentional act to identify individuals and revealing personal details
- Correlations among datasets lead to a unique fingerprint of a single individual

Protection against Re-identification Attacks
To ensure no individual's record is unique in a given dataset

K-Anonymity
- No individual's record in the dataset released is distinguishable from at least K-1 other records

For accomplishing k-anonymity:
- Suppression
  - All or some values are replaced by '*'
- Generalization
  - Individual values of attributes are replaced by a broader category or range
  - E.g.
    The age of an individual can be replaced by an age group:
    28 can be replaced by "≤ 30",
- K-anonymity may not be able to protect against target identification attacks
- L-diversity
  - Enhanced version of K-anonymity
  - Reduce the granularity of data representation
  - For each group of records sharing a combination of quasi-identifier (key attributes) attribute values, there are at least L "well-represented" values for each confidential attribute

K-anonymity and L-diversity: An Example
4-anonymity

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

3-diversity

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	1305*	≤ 40	*	Heart Disease
4	1305*	≤ 40	*	Viral Infection
9	1305*	≤ 40	*	Cancer
10	1305*	≤ 40	*	Cancer
5	1485*	> 40	*	Cancer
6	1485*	> 40	*	Heart Disease
7	1485*	> 40	*	Viral Infection
8	1485*	> 40	*	Viral Infection
2	1306*	≤ 40	*	Heart Disease
3	1306*	≤ 40	*	Viral Infection
11	1306*	≤ 40	*	Cancer
12	1306*	≤ 40	*	Cancer

3 values

Other Privacy Protection Methods
- Data aggregation
  - Aggregating individual records within a report-based and summarized format before release
  - Similar to K-anonymity
  - Analysis at finer levels difficult
  - Create problems of ecological inferences
    - Relations deduced by aggregation are not seen without aggregation
- Data suppression
  - Not all data values are released
  - Some are removed, withheld or disclosed
  - May lead to inaccurate data mining and analysis
- Data swapping
  - Data values of selected records are swapped to hide the true owner of the records
  - A high rate of swapping destroys relationships involving the swapped and unswapped variables
- Data randomization
  - Adding noise of randomly generated numerical values to data variables
  - Distort the values of sensitive variables
  - Difficult to deduce accurate matching
  - Large variance noise distribution may introduce measurement errors and inaccurate regression coefficients
- Data synthesis
  - The values of sensitive variables are replaced with synthetic values generated by simulation

Comparison of Encryption Schemes[a]

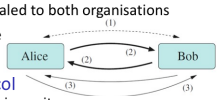
Encryption Scheme	Features	Limitations
Identify-based Encryption (IBE)	• Access control is based on the identity of a user • Complete access over all resources	• Time consuming in larger environment • Granular access control is hard to implement • Changing ciphertext receiver is not possible • Data to be processed must be downloaded and decrypted
Attribute-based Encryption (ABE)	• Access control is based on user's attribute • Granular access control is possible • More secure and flexible	• Computation overhead in handling different user categories • Updating ciphertext receiver is not possible • Data to be processed must be downloaded and decrypted
Proxy Re-Encryption	• Can be deployed in IBE or ABE scheme settings • Updating ciphertext receiver is possible	• Computational overhead • Data to be processed must be downloaded and decrypted
Homomorphic Encryption	• Computations are performed on the encrypted data • Very secure	• Computational overhead is very high

Privacy Preserving Data Processing

Privacy Preserving Collaborative Data Mining
- Distributed collaborative approaches where organizations retain their own datasets and cooperate to learn the global data mining results

Privacy Preserving Record Matching
Recent approaches include data transformation and mapping into vector spaces, and combination of secure multiparty computation and data sanitization (e.g., differential privacy and k-anonymity)

Privacy for Data Matching
Two-Party Data Matching Protocol
- Only records in the two databases that are similar are identified
- The identities of these records are revealed to both organisations
- Neither of the two parties must be able to learn anything else about the other party's confidential data



Three-Party Data Matching Protocol
- All databases are sent to a trusted matching unit
- No database owner is able to learn anything about any other databases that are being matched
- No external adversary is able to learn anything about the source databases even if they get access to any data exchanged
- Only selected attributes of matched records are revealed to the research team

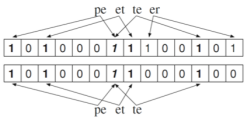
Exact Privacy-Preserving Matching Techniques
- One-way hash-encoding function
- Only allows exact matching
  - Approximate matching is not feasible
- Pre-process and standardize inputs for matching
- Three-party protocol: matching based only on hash-codes
- Two-party protocol: does not hide information from the database owners
- Add a secret key in three-party protocol

Approximate Privacy-Preserving Matching Techniques
- A two-party protocol for securely calculating edit distance
  - The distance matrix is stored in a shared fashion between the two database owners and generated iteratively
  - Based on homomorphic encryption approach
- "Blindfolded record linkage" in a three-party setting for calculating the Dice coefficient
  - Hash-encoding q-gram lists rather than full string values
  - Let B_x = set of tokens generated for string x, B_y = set of tokens generated for string y

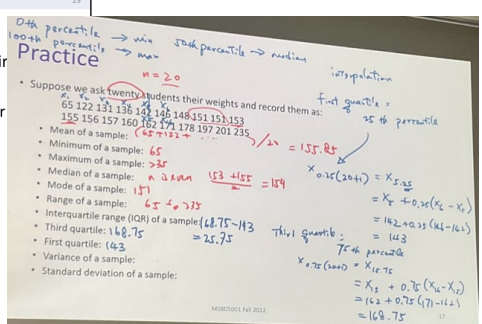
$$sim_{dice}(x, y) = \frac{2 \times |B_x \cap B_y|}{|B_x| |B_y|}$$

- Alice: record a1: "peter" and Bob: record b2: "pete",
- Using bigrams (q = 2)
  - Alice: ["pe", "et", "te", "er"], ["et", "te", "er"], ["pe", "te", "er"], ["pe", "et", "er"], ["pe", "et", "te"]
  - Bob: ["pe", "et", "te"], ["et", "te"], ["pe", "te"], ["pe", "et"]
- For matching:
  - Alice: ("7d44721f6f34", 4, "a1", 4), ("23b2dce7d424", 3, "a1", 4), ("49ee4c45fcb7", 3, "a1", 4), ("1c7e180795ad", 3, "al", 4)
  - Bob: ("1c7e180795ad", 3, "b2", 3), ("820bcd28023c7", 2, "b2", 3), ("6fa95b1427a3", 2, "b2", 3), ("34bb046d8055", 2, "b2", 3)
  - Bloom filters
    - Five bits set to 1 are in common in both Bloom filters
    - "peter" is with seven bits set
    - "pete" is with 5 bits set
    - The Dice coefficient is 2(5)/(7+5) = 10/12

Dice coefficient
= 2(3)/(3 +4)
= 6/7



Statistical Analysis
- Statistical analysis is the science of collecting, exploring and presenting large amounts of data (a.k.a. dataset) to discover underlying patterns and trends[1].
- It is used extensively in science, from physics to social sciences.
- The following are the major tasks in statistical analysis:
  - Describing and summarizing the data
  - Identifying the relationship between variables
  - Forecasting the outcomes



Mean of a sample: $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Minimum of a sample: Minimum of $\{x_1, x_2, \ldots, x_i, \ldots, x_n\}$

Maximum of a sample: Maximum of $\{x_1, x_2, \ldots, x_i, \ldots, x_n\}$

Median of a sample: Middle number of the sorted list of $\{x_1, x_2, \ldots, x_i, \ldots, x_n\}$.
If n is even, the median is the simple average of the middle two numbers

Mode of a sample: The value that appears most often in $\{x_1, x_2, \ldots, x_i, \ldots, x_n\}$

Range of a sample: Minimum to Maximum

pth percentile of a sample: The value so that roughly p% of the sample are smaller and (100 - p)% of the sample are larger

Interquartile range (IQR) of a sample: Third quartile - First quartile
- First quartile: Median of the first half of the data
- Third quartile: Median of the second half of the data

Variance of a sample: $\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2$

Standard deviation of a sample: $\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2}$