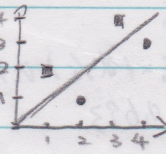
$\min_w \frac{1}{2} w \cdot w$ subject to

$y^k(w \cdot x^k + b) \geq 1, \forall k \in [1, n]$



Training Tuples: ([1,2], +1)
([2,1], -1)
([3,4], +1)
([4,3], -1)

$\min_{w_1, w_2} \frac{1}{2}(w_1^2 + w_2^2)$

subject to

$(+1)(w_1 + 2w_2) \geq 1$
$(-1)(2w_1 + w_2) \geq 1$
$(+1)(3w_1 + 4w_2) \geq 1$
$(-1)(4w_1 + 3w_2) \geq 1$

For this example:
$b = 0$, $W = [-1, +1]$,
the line is
$-x_1 + x_2 = 0$
margin $= \frac{1}{\|w\|} = \frac{1}{\sqrt{1+1}} = \frac{1}{\sqrt{2}}$

Let $\varphi: R^2 \to R^3$ such that

$\varphi(x) = \varphi([x_1, x_2]) = [z_1, z_2, z_3] = [x_1^2, \sqrt{2}x_1x_2, x_2^2]$

Let $r = [r_1, r_2, r_3]$ and $s = [s_1, s_2, s_3]$ be 2 vectors in R^3 corresponding to vectors $a = [a_1, a_2]$ and $b = [b_1, b_2]$ in R^2.

$\varphi(a) \cdot \varphi(b) = r \cdot s = r_1s_1 + r_2s_2 + r_3s_3 =$
$= (a_1b_1)^2 + 2a_1a_2b_1b_2 + (a_2b_2)^2$
$= (a_1b_1 + a_2b_2)^2 = (a \cdot b)^2$

$\min_{w,b,\xi} \frac{1}{2} w \cdot w + C \sum_{k=1}^{n} \xi_k$ …… cost 公式，越小越好

subject to

$y^k(w \cdot x^k + b) \geq 1 - \xi_k$ $\quad k \in [1, n]$

$\xi_k \geq 0$

TPR = P correctly classified / P = TP/P
FPR = P incorrectly classified / N = FP/N

*P: Positives.
N: Negatives.

Precision = TP/(TP+FP)
Recall = TP/P (same as TPR) = TP/TP+FN
Fmeasure = 2/[(1/Precision)+(1/Recall)]
= 2*Precision*Recall/[Precision+Recall]

Example: Consider now a classifier that has a recall of 90% and a precision of 40%.
Answer: The F-measure for the classifier is
$2*0.9*0.4/(0.9+0.4) = 0.55$
Conclusion: The bigger F-measure is, the classifier is better.

Actual \ Predicted	+	—
P: +	TP	FN
N: —	FP	TN

P = TP+FN
N = FP+TN

sesitivity = TP/P
Specificity = TN/N

Example: The distribution of the instance for class A: 10. remaining: 90
Random classifier Y outputs A. 50% B, 30% C. 10% D. 10%

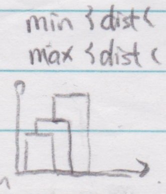
Answer: Y will say 50% of the time "A" and 50% of the time "not A".
Precision:
5/50 = 10% $\quad$ 5 = 10×50% $\quad$ 50 = 5 + 90×50% = 5+45
recall:
5/10 = 50%

---

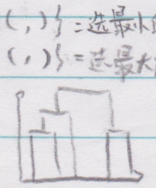
cluster analysis: merge 2 closest cluster until only a single cluster

min {dist(,) … dist(,)} = 选最小的
max {dist(,) … dist(,)} = 选最大的

$dist = \sqrt[r]{\sum_{k=1}^{n} |p_k - q_k|^r}$



min



max

Apriori Algorithm: Itemset lattice
maximal frequent sets: 连接的集合不 Frequent
closed sets: 连接的集合不相同 support number
Pruning ratio = # of N/total (不包括 null)
false alarm rate = # of I/total (不包括 null)

FP-Tree/FP Growth:
Build FP-Tree:
1. Scan data to determine the support count of item
2. Make a second pass over the data to construct
Build Conditional FP-Trees:
1. Find the paths containing on focus item
2. Read the tree to determine the new counts of the items along those paths.
— Build new header
3. Read again the tree. Insert the paths into the conditional FP-Tree according to the new order

---

Recommender System:

$Sim_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$ $\quad X = [x_1 \cdots x_n]$, $Y = [y_1 \cdots y_n]$.

$\hat{r}_{u,i} = \frac{\sum_{v \in U_i} r_{v,i} \cdot Sim_{v,u}}{\sum_{v \in U_i} Sim_{v,u}}$ $\quad$ * U_i is the set of users. this user-user

$\hat{r}_{u,i} = \frac{\sum_{j \in I_u} r_{u,j} \cdot Sim_{i,j}}{\sum_{j \in I_u} Sim_{i,j}}$ $\quad$ * I_u is the set of items rated by user u. this is item-item

Mean and biases: Gradient (with latent factors)

$e_{u,i}^2 = (r_{u,i} - \mu - b_u - b_i - p_u q_i)^2$
$b_u = b_u + \gamma(e_{ui} - \lambda_1 b_u)$ $\quad p_u \leftarrow p_u + \gamma(e_{ui} q_i - \lambda_3 p_u)$
$b_i = b_i + \gamma(e_{ui} - \lambda_2 b_i)$ $\quad q_i \leftarrow q_i + \gamma(e_{ui} p_u - \lambda_4 q_i)$

Document Retrieval

$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|V|j}\}}$

word i $\quad$ document j $\quad$ size of vocabulary

$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2} \sqrt{\sum_{i=1}^{|V|} d_i^2}}$

Query document

$idf_i = \log \frac{N}{df_i}$ $\quad$ N: total # of docs; df_i: the # of docs that t_i appears

$W_{ij} = tf_{ij} \times idf_j$

Bernoulli model:
for each class C, P(C) is estimated as the fraction of all "training" documents that are of class C.

$P(C) = \frac{N_c}{N}$

P(t|C) is estimated as the fraction of documents of class C that contains term t.

$P(t|C) = \frac{N_{c,t}}{N_c}$

Now, we can use Naive Bayes for classifying a new document d:
Estimate:
$P(C|d) = \alpha * P(C) * \prod_{t \in d} P(t|C) * \prod_{t \notin d}(1 - P(t|C))$
Produce as classification result:
$C_{map} = \arg\max_c P(C|d)$

~~To avoid number overflow, operate on logs of probabilities~~
$\log P(C|d) = \log\alpha + \log P(C) + \sum_{t \in d} \log P(t|C) + \sum_{t \notin d} \log(1 - P(t|C))$

To avoid zero-frequency: $P(t|C) = \frac{N_{c,t} + 1}{N_c + 2}$

Example:

	docID	Words in Document	in c = china?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	NO
test set	5	Chinese Chinese Chinese Tokyo Japan	?

Answer: Priors: $P(C) = 3/4$ and $P(\bar{C}) = 1/4$

Conditional probabilities:
$P(Chinese|C) = (3+1)/(3+2) = 4/5$
$P(Japanese|C) = P(Tokyo|C) = (0+1)/(3+2) = 1/5$
$P(Beijing|C) = P(Macao|C) = P(Shanghai|C) = (1+1)/(3+2) = 2/5$
$P(Chinese|\bar{C}) = (1+1)/(1+2) = 2/3$
$P(Japanese|\bar{C}) = P(Tokyo|\bar{C}) = (1+1)/(1+2) = \frac{2}{3}$
$P(Beijing|\bar{C}) = P(Macao|\bar{C}) = P(Shanghai|\bar{C}) = (0+1)/(1+2) = 1/3$

*note: there are 3 docs in C and 1 not in C.

Then we get:

$P(C|d_5) \propto P(C) \cdot P(Chinese|C) \cdot P(Japan|C) \cdot P(Tokyo|C)(1-P(Beijing|C))(1-P(Shanghai|C))(1-P(Macao|C))$
$3/4 \cdot 4/5 \cdot 1/5 \cdot 1/5 \cdot (1-2/5)(1-2/5)(1-2/5) \approx 0.005$

$P(\bar{C}|d_5) \propto P(\bar{C})$.
$1/4 \cdot 2/3 \cdot 2/3 \cdot 2/3 \cdot (1-\frac{1}{3})(1-\frac{1}{3})(1-1/3) \approx 0.022$
$0.022 > 0.005 \Rightarrow$ test result: Not China

The pearson correlation between Michael [3.0, 3.5] and Toby [4.5, 4.0] (user-user)

$Sim_{x,y} = \frac{(3.0-3.25)*(4.5-4.25)+(3.5-3.25)*(4.0-4.25)}{\sqrt{(3.0-3.25)^2+(3.5-3.25)^2} \cdot \sqrt{(4.5-4.25)^2+(4.0-4.25)^2}} = -1$

$\hat{r}_{u,i} = \frac{2.5*0.425 + 3.5*0.205 + 2.5*1.0 + 3.5*0.135}{0.425 + 0.205 + 1 + 0.135} = 2.7$ (user-user).

Only consider positive
*all user 第i 業已收到 的評分過 Sim.

Multinomial Model

$P(c) = \frac{N_c}{N}$ (same as before)

$P(t|c)$ is estimated as the relative frequency of term t in documents belonging to class C.

$P(t|c) = T_{c,t} / \sum_{t' \in V} T_{c,t'}$

V is the vocabulary,
$T_{c,t}$ is the # of occurrences of t in training documents from class C, including multiple occurrences of a term in a document.

Estimate:

$P(c|d) = \alpha * P(c) * \prod_{t \in d} P(t|C)$

$C_{map} = \text{argmax}_c P(c|d)$

Avoid number overflow:
$\log(c|d) = \log \alpha + \log P(c) + \sum_{t \in d} \log P(t|c)$

Avoid zero-frequency:
$P(t|c) = \frac{T_{c,t}+1}{\sum_{t' \in V}(T_{c,t'}+1)} = \frac{T_{c,t}+1}{(\sum_{t' \in V} T_{c,t'})+|V|}$

Same Example as before:

Priors: $P(c) = 3/4$ and $P(\bar{c}) = 1/4$

Conditional Probabilities:
$P(\text{Chinese}|C) = (5+1)/(8+6) = \frac{3}{7}$
$P(\text{Tokyo}|C) = P(\text{Japan}|C) = (0+1)/(8+6) = 1/14$
$P(\text{Chinese}|\bar{C}) = (1+1)/(3+6) = 2/9$
$P(\text{Tokyo}|\bar{C}) = P(\text{Japan}|\bar{C}) = (1+1)/(3+6) = \frac{2}{9}$

Denominators are $(8+6)$ and $(3+6)$
. because there are 8 terms in texts in Yes and 3 terms in No
. because vocabulary consists of 6 terms in training sets. (6 ✓ Chinese)

$P(c|d_5) \propto 3/4 \cdot (3/7)^3 \cdot (1/14)^2 \approx 0.0003$
$P(\bar{c}|d_5) \propto 1/4 \cdot (2/9)^3 \cdot (2/9)^2 \approx 0.0001$

the classifier assigns the test doc to C = china
the 3 occurrences of the positive outweights 2 occurrences of the negative.

---

perceptron: linear classifier for iteration:
if $\text{Sign}(W^t x^k) \neq y^k$, update $W = W + \eta \cdot y^k x^k$

linear regression:
$W \leftarrow W + K_a \frac{1}{n} \sum_{k=1}^{n} (y^k - W^t x^k) x^k$

kappa

Logistic Regression: $P(y|x) = \frac{1}{1+e^{-yw^t x}}$

$W_{t+1} = W_t + K_a \frac{1}{n} \sum_{k=1}^{n} \frac{y^k x^k}{1+e^{y^k W_t x^k}}$

Fix a threshold in $[0,1]$ to make predictions:
$P(y=1|x) \geq$ threshold Predict $y=1$
$P(y=-1|x) \leq$ threshold Predict $y=-1$

ID_3:

Example:

Outlook = Sunny entropy$(2/5, 3/5) = -2/5 \times \log(2/5) - 3/5 \times \log(3/5) = 0.971$

Outlook = Overcast $e(4/4, 0/4) = -1 \times \log 1 - 0 = 0$

Outlook = rainy $e(3/5, 2/5) = -3/5 \times \log(3/5) - 2/5 \times \log(2/5) = 0.971$

$AE = 0.971 \times (5/14) + 0 \times (4/14) + 0.971 \times (5/14) = 0.693$

Naive Beyes:

$P(Play = yes|E) = P(\text{Outlook} = \text{Sunny} | play = yes) *$
$P(\text{Temp} = \text{cool} | play = yes) *$
$P(\text{humandity} = \text{High} | play = yes) *$
$P(\text{Windy} = \text{True} | play = yes) *$
$P(play = yes) / P(E) = 2/9 \cdot 3/9 \cdot 3/9 \cdot 3/9 \cdot 9/14 / P(E)$
$= 0.0053 / P(E)$

$P(play = no|E) = P(\text{Outlook} = \text{Sunny} | play = no) *$
$P(\text{Temp} = \text{cool} | play = no) *$
$P(\text{Humandity} = \text{High} | play = no) *$
$P(\text{Windy} = \text{True} | play = no) *$
$P(play = no) / P(E) = 0.0206 / P(E)$

$P(play = yes|E) + P(play = no|E) = 1 \Rightarrow P(E)$

$0.0053 / P(E) + 0.0206 / P(E) = 1$