Multinomial Naive Bayes

Input $x^{(i)}$ that is a list of words with length d_i.

Y	X
+1	great acting and score
−1	terrible directing
+1	great execution
−1	terrible
+1	amazing

NB assumption: $P(x^{(i)}|y^{(i)}) = \prod_{j=1}^{x_i} P(x_j^{(i)}|y^{(i)})$

e.g. $P(\text{"great directing"}|+1)$

$= P(\text{"great"}|+1)\, P(\text{"directing"}|+1)$

Additional assumption: word position doesn't matter

Step 1: Parameter Estimation (a.k.a. training)

$P(Y)$

$P(Y=+1) = \frac{3}{5}$

$P(Y=-1) = \frac{2}{5}$

$P(X|Y)$

$Y=+1$

$P(\text{"great"}|+1) = \frac{2+\lambda}{7+8\lambda}$

$P(\text{"acting"}|+1) = \frac{1+\lambda}{7+8\lambda}$

$\vdots$

$P(\text{"directing"}|+1) = \frac{0+\lambda}{7+8\lambda}$

$Y=-1$

$P(\text{"terrible"}|-1) = \frac{2+\lambda}{3+8\lambda}$

$P(\text{"directing"}|-1) = \frac{1+\lambda}{3+8\lambda}$

$\vdots$

$P(\text{"great"}|-1) = \frac{0+\lambda}{3+8\lambda}$

Step 2: Inference

Given input $x=$ "great directing", compute $P(Y|X=\text{"great directing"})$

$$= \frac{P(Y)\,P(X=\text{"g d"}|Y)}{\underbrace{P(X=\text{"g d"})}_{\text{normalizing constant}}}$$

$Y=+1$: $\underbrace{\frac{3}{5}}_{P(Y=+1)} \cdot \underbrace{\frac{3}{15}}_{P(\text{"great"}|+1)} \cdot \underbrace{\frac{1}{15}}_{P(\text{"directing"}|+1)} = 0.008$

$y=-1: \frac{2}{5} \cdot \frac{1}{11} \cdot \frac{2}{11} = 0.0066$

$$P(y=+1 \mid \text{"g d"}) = \frac{0.008}{0.008 + 0.0066} = 0.55$$

If documents are long, you have to multiply many small probabilities together.

Numerical underflow i.e. on computer, everything $\to 0$

Solution: work in log space

$y=+1$: compute $\log(p(y=-1) \cdot P(\text{"g d"} \mid +1))$

$$= \log\left(\tfrac{3}{5}\right) + \log\left(\tfrac{3}{15}\right) + \log\left(\tfrac{1}{15}\right) \approx -2.1$$

$y=-1$: $\log\left(\tfrac{2}{5}\right) + \log\left(\tfrac{1}{11}\right) + \log\left(\tfrac{2}{11}\right) \approx -2.2$

To get probabilities, compute max los score $= -2.1$

subtract that from everything

$y=+1 = 0$

$y=-1 = -0.1$

then exponentiate

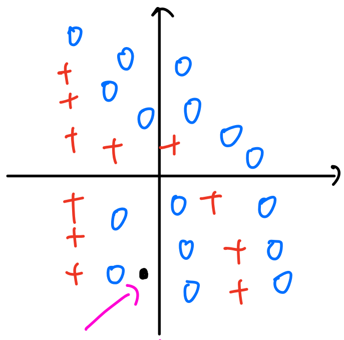
$y=+1 = 1$

$y=-1 = e^{-0.1} \approx 0.9$

finally normalize

$p(y=+1) = \frac{1}{1+0.9}$

	model $p(y \mid x)$ Discriminative	$p(y)\,p(x \mid y)$ Generative
Parametric learn some params, afterward can throw away training data	logistic regression w softmax regression $w^{(1)}, \ldots, w^{(c)}$	naive bayes $P(y)$ a.k.a. π $p(x \mid y)$ τ Puk's
non-parametric use training set when making decisions	K-Nearest Neighbors	



test input

closest is 0

predict 0

1-NN : predict the same class as $\boxed{\text{closest}}$ training example

e.g. Euclidean distance

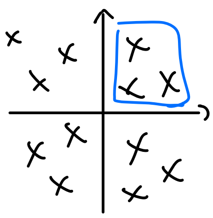
Generalisation : K-NN

1) find K-nearest neighbors of test input x

2) predict label that's most common among neighbors
"majority vote"

Curse of Dimensionality

In high dimensions, you very rarely have nearby neighbors.



← prob in the same quadrant $= \frac{1}{4}$

If $d = 1000$, $P = (\frac{1}{2})^{1000}$

No close neighbors to test data ⇒ using nearest neighbors to predict may not be accurate