Tarefa: aprender a classificar textos

- Considere o problema de aprendizagem em que as instâncias sejam documentos.
- Desejamos aprender o conceito alvo do tipo "página da Web que discutem tópicos sobre Machine Learning", para filtrar um grande volume de documentos online e apresentar ao usuário apenas os mais relevantes.
- Usaremos um classificador NB para classificar textos, com a seguinte definição:
- Considere um espaço X de documentos (todas as possíveis strings de palavras e pontuações de todos os comprimentos possíveis).
- Dispomos de um arquivo de treinamento (documentos) de uma função alvo $f(x)$ que pode assumir um valor discreto $v \in V$, por exemplo, $V = \{p$ (*interessante*), n (*não-interessante*)$\}$.
- A representação de um texto com n palavras é um vetor de n atributos, sendo cada atributo pode assumir m valores, equivalentes às palavras que podem ocorrer (vocabulário).
- Ex.: a representação da frase anterior seria um vetor com 14 posições com os valores:

["a" "representação" "do" "texto" "é" "um" "vetor" "das" "palavras" "ordenadas" "segundo" "a" "sua" posição"]

Tarefa: aprender a classificar textos

- Considere que tenhamos um conjunto de 700 documentos de treinamento que previamente tenham sido classificados como n e um outro com 300 documentos classificados como p.
- A tarefa é classificar um novo documento; p. ex., o texto com 14 palavras anterior.

  ["a" "representação" "do" "texto" "é" "um" "vetor" "das" "palavras" "ordenadas" "segundo" "a" "sua" posição"]
- Pelo classificador NB, o rótulo da classificação, $v_{NB} \in \{p, n\}$, seria calculado por:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i=1}^{14} P(a_i \mid v_j)$$

$$v_{NB} = \arg\max_{v_j \in V} P(v_j)\, P(a_1 = "a" \mid v_j) P(a_2 = "representação" \mid v_j) \cdots P(a_{14} = "posição" \mid v_j)$$

- A suposição NB implica que uma palavra numa posição no texto é independente das palavras que ocorrem nas outras posições para efeitos de classificação, o que é incorreto.
- Por exemplo: se o interesse são textos de machine learning, a probabilidade de observar "learning" numa posição é maior se a palavra precedente for "machine".

Tarefa: aprender a classificar textos

- Para calcular v_{NB} é necessário estimar as probabilidades a priori das classes $P(v_j)$ e as probabilidades que formam as verossimilhanças das classes $P(a_i = w_k \mid v_j)$, onde w_k é a k-ésima palavra do vocabulário utilizado.
- As probabilidades a priori equivalem à fração de cada classe nos dados de treinamento:

$$P(p) = 0{,}3 \text{ e } P(n) = 0{,}7$$

- Para facilitar a estimativa das prob. condicionais, assumimos que a probabilidade de encontrar uma certa palavra é independente da sua posição no texto, significando que os atributos são independentes e identicamente distribuídos (*iid*), isto é, $P(a_i = w_k \mid v_j) = P(a_m = w_k \mid v_j) = P(w_k \mid v_j)$, para todo i, j, k, m.
- Uma forma de estimar estas probabilidades condicionais é através da estimativa m com priors uniformes e com m igual ao tamanho do vocabulário (número de valores dos atributos) utilizado:

$$P(w_k \mid v_j) = \frac{n_{kj} + 1}{n_j + |vocabulário|}$$

- com n_j o número total de posições de palavras em todos os exemplos de treinamento para a classe v_j, n_{kj} é o número de vezes que a palavra w_k foi encontrada nestas n_j posições e $|vocabulário| = K$, é o número de palavras distintas encontradas nos dados de treinamento

Exemplo

- Classe 1: 700 documentos de 400 palavras (posições) cada, ou seja $n_1 = 280.000$.
- Classe 2: 300 documentos de 400 palavras cada ou seja $n_2 = 120.000$.
- Vocabulário: 1000 palavras distintas, ou seja, $K = 1000$.
- As probabilidades a priori equivalem à fração de cada classe nos dados de treinamento:

$$P(p) = 0{,}3 \text{ e } P(n) = 0{,}7$$

- Com isso, as probabilidades condicionais de encontrar uma certa palavra numa das classes, isto é, $P(w_k \mid v_j)$, são calculadas como:

$$P(w_k \mid v_j) = \frac{n_{kj} + 1}{n_j + |vocabulário|}$$

$$P(w_k \mid v_1) = \frac{n_{k1} + 1}{281000} \qquad P(w_k \mid v_2) = \frac{n_{k2} + 1}{121000}$$

LEARN_NAIVE_BAYES_TEXT(*Examples*, *V*)

Examples is a set of text documents along with their target values. V is the set of all possible target values. This function learns the probability terms $P(w_k|v_j)$, describing the probability that a randomly drawn word from a document in class v_j will be the English word w_k. It also learns the class prior probabilities $P(v_j)$.

1. collect all words, punctuation, and other tokens that occur in Examples

- *Vocabulary* ← the set of all distinct words and other tokens occurring in any text document from *Examples*

2. calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms

- For each target value v_j in V do
  - $docs_j$ ← the subset of documents from *Examples* for which the target value is v_j
  - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - $Text_j$ ← a single document created by concatenating all members of $docs_j$
  - n ← total number of distinct word positions in $Text_j$
  - for each word w_k in *Vocabulary*
    - n_k ← number of times word w_k occurs in $Text_j$
    - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

CLASSIFY_NAIVE_BAYES_TEXT(*Doc*)

Return the estimated target value for the document Doc. a_i denotes the word found in the ith position within Doc.

- *positions* ← all word positions in *Doc* that contain tokens found in *Vocabulary*
- Return v_{NB}, where

$$v_{NB} = \underset{v_j \in V}{\text{argmax}}\, P(v_j) \prod_{i \in positions} P(a_i|v_j)$$

Tarefa: aprender a classificar textos

- A partir do conjunto de dados fornecidos, construir um classificador NB.
- Para tanto, organize o conjunto de dados para permitir a validação cruzada de 10 vezes (10-fold-cv).
- Construa 10 classificadores NB para cada subconjunto de treinamento e teste o seu desempenho preditivo sobre o respectivo subconjunto de teste.
- Apresente as 10 matrizes de confusão e a matriz de confusão média correspondente, juntamente com os respectivos desvios padrões.
- A partir da matriz de confusão média, calcule as seguintes métricas:

  Precisão, Taxa de Verdadeiros Positivos (TVP), Taxa de Falsos Positivos (TFP), Medida-F