Universidade Federal de Goiás
Instituto de Informática

Introdução à Programação 2018-1

Trabalho Final de Curso
Implementação de um classificador de documentos

1 Objetivo

Este projeto tem como objetivo a implementação de um classificador automatizado de documentos. O problema de classificação de documentos é explicado a seguir. Dada uma coleção de documentos $\mathcal{D}$, denomina *coleção de treino* e, um conjunto de classes $\mathcal{C}$, onde cada documento d de $\mathcal{D}$ pertence a uma e somente uma classe de $\mathcal{C}$, o problema consiste em obter um classificador γ que "apreende" as características de um documento d que o fazem pertencer a uma dada classe c de $\mathcal{C}$. No caso de documento, as características utilizadas são as palavras (termos) que compõem os documentos. Uma vez que o classificador γ foi desenvolvido, dado um documento d_t não pertencente ao conjunto de treino $\mathcal{D}$, denominado *documento de teste*, o classificador γ atribui uma classe c de $\mathcal{C}$ ao documento d_t.

Nesse trabalho, deverá ser implementado o classificador Naive Bayes. Esse classificador apreende a classificar automaticamente documentos apreendendo probabilidades de ocorrência dos termos em cada classe. O classificador Naive Bayes a ser implementado classifica um documento de teste d_t atribuindo a d_t a classe que possui o maior valor de $\hat{P}(c|d)$, cujo valor é computado com base na Equação 1.

$$\hat{P}(c|d) = \log \hat{P}(c) + \sum_{t \in d_t} \log \hat{P}(c|t). \tag{1}$$

O valor $\log \hat{P}(c)$ é uma estimativa da probabilidade da classe c ocorrer. Esse valor é computado utilizando-se informações obtidas do conjunto de treino $\mathcal{D}$ e é dada pela Equação 2.

$$\hat{P}(c) = \frac{N_c}{|\mathcal{D}|} \tag{2}$$

onde $N_c = |\{d \in \mathcal{D} | c \text{ é a classe de } d|$ é o número de documentos de treino que pertencem a classe c e $|\mathcal{D}|$ é o total de documentos do conjunto de treino.

O valor $\hat{P}(c|t)$ é uma estimativa da probabilidade do termo t do documento de teste d_t pertencer à classe c e é dado pela Equação 3:

$$\hat{P}(c|t) = \frac{N_{t,c}}{(\sum_{t' \in \mathcal{V}} N_{t',c}) + |\mathcal{V}|} \tag{3}$$

onde $N_{t,c}$ é o número acumulado de ocorrências do termo t em todos os documentos de treino que são da classe c, $\mathcal{V}$ é o *vocabulário* do conjunto de treino, isto é, o conjunto de termos distintos que ocorrem nos documentos do conjunto de treino $\mathcal{D}$ e $\sum_{t' \in \mathcal{V}} N_{t',c}$ é a soma de todas as ocorrências de todos os termos do vocabulário $\mathcal{V}$ que aparecem em documentos da classe c no conjunto de treino $\mathcal{D}$.

2 Módulos do Programa

O programa deve ter dois módulos: *módulo de treinamento* e *módulo de classificação*. No módulo de treinamento o programa calcula para cada classe c em $\mathcal{C}$ o valor estimado de ocorrência dessa classe, isto é, $\hat{P}(c)$. Também nesse módulo são computadas as probabilidades estimadas de cada termo que ocorre nos documentos de treino de pertencerem a cada classe c, ou seja, calcula-se $\hat{P}(c|t)$ para cada termo t e cada classe c.

No módulo de classificação o programa não trabalha mais com os documentos de treino, mas sim, com documentos de teste. O programa deve ler um arquivo contendo documentos de teste e deve classificar cada documento desse arquivo de acordo com a Equação 1, utilizando os valores de $\hat{P}(c)$ e $\hat{P}(c|t)$ computados durante a execução do módulo de treinamento.

2.1 Módulo de treinamento

O módulo de treinamento deve ler o *arquivo de treino* que corresponde ao conjunto de treino $\mathcal{D}$. O arquivo de treino é um arquivo do tipo texto e possui várias linhas. Cada linha tem o seguinte formato:

$$c\ t_1 : f_{t_1}\ t_2 : f_{t_2}\ t_3 : f_{t_3}\ \cdots\ t_n : f_{t_n}$$

Cada linha do arquivo de treino corresponde a um documento. O primeiro campo c da linha corresponde à classe do documento, em seguida há vários pares do tipo $t_i : f_{t_i}$. O primeiro elemento do par, t_i é um número inteiro e corresponde ao identificador de uma palavra no texto documento correspondente à linha. O segundo elemento do par é um número inteiro e indica quantas vezes o termo t_i ocorreu nesse documento, ou seja, t_i é a frequência absoluta do termo t_i nesse documento.

O módulo de treinamento deve:

- Abrir o arquivo de treino
- Percorrer o arquivo de treino para descobrir: quantas classes distintas há no conjunto de treino e quantos termos distintos ($|\mathcal{V}|$) há no conjunto de treino.
- Definir estrutura de dados apropriada para armazenar $\hat{P}(c)$ para todas as classes encontradas.
- Definir estrutura de dados apropriada para armazenar $\hat{P}(c|t)$ para todos os termos do conjunto de treino e todas as classes encontradas no arquivo de treino.
- Percorrer o arquivo de treino mais uma vez, computando $\hat{P}(c)$ para toda classe c do conjunto de classes $\mathcal{C}$ encontrado no arquivo (conjunto de elementos que aparecem no primeiro campo de cada linha do arquivo) e computando $\hat{P}(c|t)$ para todo termo encontrado no conjunto de treino e toda classe.

- Gravar em um arquivo binário as informações sobre $\hat{P}(c|t)$ e $\hat{P}(c|t)$ computadas durante o treino.
- Fechar os dois arquivos.

2.2 Módulo de classificação

O módulo de classificação deve ler um arquivo com um ou mais documentos de teste e utilizar as informações obtidas durante a fase de treinamento para classificar os documentos de teste. As linhas do arquivo de teste possuem o mesmo formato das linhas do arquivo de treino. Entretanto a informação sobre a classe verdadeira (primeiro campo da linha) é utilizada apenas para verificar se o classificador acertou ou não a classe verdadeira do documento de teste.

O módulo classificador deve:

- Abrir arquivo de teste para leitura.
- Ler cada linha do arquivo de teste e classificar o documento correspondente. Essa classificação é feita aplicado-se a Equação 1 para cada classe de $\mathcal{C}$ e escolhendo-se a classe c que possui maior valor de $\hat{P}(c|d)$.
- Para cada linha, o módulo deve imprimir uma tabela indicando o número da linha, sua classe verdadeira e a classe indicada pelo classificador.
- Ao terminar de processar o arquivo de entrada, o programa deve computar e imprimir a *acurácia* do classificador que é dada pela equação:

$$Ac = \frac{\text{Num. classificações corretas}}{\text{Num de documentos de teste}}$$

2.3 Módulo de opções

O programa deve possuir um módulo de opções que pode ser o módulo principal ou pode ser chamado pelo módulo principal (main()) do programa. O módulo de opções deve entrar em loop, mostrando as seguintes opções:

- Treinar um classificador Naive Bayes.
- Classificar um arquivo de teste
- Terminar o programa.

Caso o usuário opte pela primeira opção, o módulo deve pedir ao usuário que digite um nome de arquivo válido para treino. Em seguida, o módulo deve chamar o módulo de treinamento para realizar o treinamento utilizando o arquivo informado.

Caso a segunda opção seja escolhida, o módulo deve perguntar ao usuário se ele quer utilizar um classificador que já está na memória (caso a opção a primeira opção tenha sido executada antes) ou usar um classificador que está gravado em disco. No último caso, o módulo deve pedir ao usuário o nome do arquivo binário contendo os dados de treino de um treinamento realizado previamente e carregar as informações correspondentes na memória. Em seguida o módulo deve perguntar ao usuário o nome do arquivo de teste e perguntar se o

usuário quer que a saída da classificação seja na tela do computador ou em um arquivo tipo texto. Se a segunda opção for a escolhida o módulo deve pedir o nome do arquivo de saída. Após isso, o módulo deve chamar o módulo de classificação passando as informações para que esse módulo execute corretamente.