

Análise de Sentimentos Aplicada à Política

Lucas Romão Silva

Prof Dr. Roberto Hirata Jr.

12 de setembro de 2017

Sumário

| | | |
|----------|-------------------------------|----------|
| 1 | The First Chapter | 1 |
| 2 | Materiais e Métodos | 3 |
| 2.1 | Considerações | 3 |
| 2.2 | Contextualização | 3 |
| 2.3 | Logistic Regression | 4 |

Capítulo 1

The First Chapter

Capítulo 2

Materiais e Métodos

2.1 Considerações

Ao longo deste capítulo, se usará n para se referir à quantidade de elementos fornecidas ao nosso modelo, cada entrada i é um vetor $x_i \in \mathcal{R}^m$. Para cada i associaremos duas variáveis t_i e y_i que se referem ao valor esperado e ao valor obtido através do treinamento, respectivamente.

2.2 Contextualização

Os problemas tratados por *Machine Learning* classificam-se de forma geral em três tipos:

- Aprendizado supervisionado: nesse caso tem-se os elementos de entrada e para cada um desses elementos, tem-se associado um rótulo t_i . Nesse caso o modelo deve ser treinado com base nos elementos dados para que se possa prever o rótulo de uma nova entrada;
- Aprendizado não-supervisionado: nesse caso tem-se apenas os elementos de entrada. O objetivo deste tipo de problema é tentar modelar uma distribuição ou estrutura comum entre os dados para que se possa entendê-los melhor;
- Aprendizado semi-supervisionado: nesse último caso alguns elementos possuem um rótulo associado. Problemas desse tipo aplicam técnicas tanto de aprendizado supervisionado como de não-supervisionado.

Neste trabalho será tratado um problema de aprendizado supervisionado que é o da classificação.

Na classificação temos k classes e cada elemento i da entrada é associado a uma classe $t_i = \{1..k\}$. O objetivo do problema da classificação é dado entrada $X = (x_1, x_2, \dots, x_n)$ e $t = (t_1, \dots, t_n)$ treinar um modelo capaz de prever classes para um x qualquer.

Há diversos algoritmos na literatura que se propõem a resolver o problema da classificação. Bishop (2006)[1] enuncia diversos dos algoritmos comumente utilizados para a classificação, cada algoritmo possui seus prós e contras e utiliza diferentes abordagens.

Para este trabalho escolheu-se implementar os algoritmos *Logistic Regression* e *Support Vector Machines*, que será chamado simplesmente de SVM por facilidade.

Tanto para o *Logistic Regression* quanto SVM será explicado a princípio o problema será inicialmente abordado a partir da classificação binária e, a partir dela, será descrito como estender para o problema com mais de duas classes, que será o caso deste trabalho.

2.3 Logistic Regression

Para classificar um dado elemento x entre as possíveis classes C_1 e C_2 , é utilizado um discriminante linear da forma $y(x) = w^T x + w_0$ sendo w o vetor de pesos associado. A classe atribuída a um vetor x é baseado no sinal de $y(x)$, se $y(x) \geq 0$ ele é designado à classe C^1 , caso contrário é designado à classe C^2 .

Nesse caso, diz-se que uma superfície de decisão é definida pelo hiperplano $y(x) = 0$ onde sua posição é determinada pelo elemento w_0 que chamaremos de viés. Uma vez que tanto nosso vetor de pesos w quanto nossos vetores x do conjunto de treino possuem m elementos, iremos criar vetores novos $w' = (w_0, w)$, $x' = (1, x)$.

O nosso modelo será construído de forma probabilística, uma vez que o objetivo será obter um vetor w de modo que possamos estimar as probabilidades condicionais $P(C^1|x)$ e consequentemente $P(C^2|x) = 1 - P(C^1|x)$, isto é, a probabilidade de um vetor x pertencer à uma determinada classe.

Para utilizarmos nosso discriminante $y(x)$ para atribuir as probabilidades, utiliza-se a função sigmóide definida por:

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \quad (2.1)$$

Com \exp sendo a função exponencial. Aplicando ao nosso modelo obtêm-se a expressão:

$$P(C^1|x) = y(x) = \sigma(w^T x) \quad (2.2)$$

Importante notar que apesar de utilizarmos o vetor x nas equações, é possível aplicarmos uma transformação linear $\phi : \mathcal{R}^m \rightarrow \mathcal{R}^d$ à entrada x para obtermos $\phi(x)$. O uso de transformação linear no nosso conjunto de entrada nos permite transformar o domínio para que se obtenha uma separação melhor entre as classes ou até mesmo fazer a redução da dimensão do domínio.

Referências Bibliográficas

- [1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1 edition, 2006.