Departamento de Engenharia da Computação

CLASSIFICAÇÃO TEXTUAL VIA MÁQUINAS DE VETORES SUPORTE (SVM)

Aluno: Breno Alberti Faria
Orientador: Ruy Luiz Milidiú

Introdução

Quando tratamos de um pequeno corpus, a categorização manual de documentos ainda é viável, apesar de demorada. A construção de regras para uma classificação automatizada é limitada pelo fato de ser cara e muito difícil. Portanto, a enorme quantidade de informação online e a alta taxa de crescimento da Internet[1] tornam necessária uma classificação automática e inteligente, com custo computacional aceitável e sem o uso de regras predefinidas.

O uso de um método de classificação por machine learning é uma possível solução para esse problema. Dentre os diferentes algoritmos de aprendizado, o algoritmo de classificação por Máquinas de Vetores Suporte (Support Vector Machines) tem tido o melhor desempenho em tarefas de classificação textual[3].

O presente relatório visa a documentar o projeto de implementação de um classificador textual por SVM's e de experimentos realizados com o corpus Reuters-21578, um corpus bastante utilizado por pesquisadores da área de machine learning.

Objetivos

Pesquisar uma técnica de classificação textual no estado da arte que ofereça material para aprofundamento teórico, e ao mesmo tempo imediata aplicação prática. Conhecer o problema de classificação textual e compreender a importância do desenvolvimento de técnicas que explorem esse ramo de pesquisa. Entender o algoritmo de classificação baseado em SVM. Ser capaz de aplicar a técnica estudada e analisar os resultados obtidos, entendendo as vantagens e limitações do método de classificação textual usando SVMs.

Implementação

Antes da implementação do algoritmo houve um período de engenharia de software para desenvolver um framework não só capaz de tratar grandes quantidades de texto, como também suficientemente generalizado para possibilitar a implementação de outros algoritmos, tanto de classificação, como algoritmos auxiliares para a tarefa de classificação em geral, como tokenizadores e parsers. A preocupação de desenvolver um framework generalizado foi a de possibilitar reúso de código e facilidade para manutenção. Além disso foram levados em consideração os problemas de desempenho encontrados ao se lidar com grandes quantidades de informação e por isso foi escolhida a implementação de um framework utilizando a estruturação de documentos como "bag of words", onde o documento é representado por palavras ocorrentes e suas respectivas freqüências. Essa estruturação é consideravelmente menos custosa que o simples uso de vetores para a representação de documentos, pois é uma característica inerente de informação textual que o número de palavras de um corpus é muito maior do que o número de palavras em um documento e por isso o espaço gasto com um vetor (cuja dimensão é o tamanho do léxico) é desnecessariamente grande. A linguagem utilizada na implementação foi Python por ser uma linguagem propícia para o tratamento de texto além de ser uma linguagem de fácil aprendizado.

Para reduzir o tamanho do espaço (originalmente o tamanho do léxico) foram codificadas algumas heurísticas, como a remoção de "stopwords" (palavras que ocorrem em grande quantidade e que não influenciam fortemente na classe do documento) e a remoção de palavras de baixíssima freqüência (document frequency thresholding). Para auxiliar nos experimentos foram codificados módulos para validação cruzada (sorteio aleatório de documentos a serem utilizados para treino e teste) e para análise de desempenho, onde eram computadas as métricas ("precision", "recall" e "F1") para os experimentos.

O algoritmo de SVM's se baseia na idéia de que dados dois conjuntos de pontos linearmente separáveis no espaço, ao se encontrar um hiperplano que separe esses dois conjuntos, a classificação de um novo ponto torna-se trivial, pois basta verificar o sinal do resultado de se inserir esse ponto na equação do hiperplano encontrado. No entanto, é quase imediata a percepção de que podem existir infinitos hiperplanos que separam dois conjuntos de pontos linearmente separáveis no espaço. Demonstra-se que o hiperplano cuja margem para os pontos mais próximos (vetores suporte) é a maior, é o hiperplano que minimiza o risco de se classificar erroneamente um novo ponto[4]. O desafio do algoritmo de SVM's é, portanto, o de encontrar o hiperplano com a maior margem para os pontos mais próximos a ele.

Esse problema pode ser formulado como um problema de programação quadrática para a maximização da margem. Para a resolução dos problemas de programação quadrada foi utilizada uma biblioteca de otimização convexa[5]. A formulação dual desse problema nos permite deslinearizar o algoritmo com a ajuda de funções de núcleo[6]. Com isso o algoritmo passa a ser capaz de classificar conjuntos que não são linearmente separáveis. Foram implementadas três funções de núcleo: linear, polinomial e gausseana.

Foram realizados inúmeros experimentos, dos quais vale ressaltar os seguintes: Dependência do desempenho do classificador da fração do corpus usado para treino; Desempenho em relação à função de núcleo usada; Relação entre o desempenho do classificador e o método de redução do espaço, Documento Frequency Thresholding. Do primeiro experimento pudemos constatar que ao se utilizar aproximadamente 8% do corpus para treino, chega-se a um desempenho bastante próximo ao desempenho máximo. Isso é um resultado bastante interessante, pois mostra que com poucos documentos classificados, que são geralmente difíceis de obter, chega-se a resultados muito bons. Do segundo pudemos confirmar que problemas de classificação textual são de natureza linear (a função linear de núcleo teve um desempenho excelente). Isso mostra que categorias textuais são, geralmente, facilmente distinguíveis. Do último experimento pudemos constatar que para obter desempenhos próximos do máximo precisamos apenas de aproximadamente 1.5% do léxico, o que reduz consideravelmente o gasto de memória e de tempo de CPU, o que também é bastante animador, pois esses são os dois grandes problemas do algoritmo de SVM's.

Conclusões

O estudo teórico aliado à aplicação direta do algoritmo permitiu uma compreensão profunda do problema de classificação textual e também do problema de tratamento de informação em geral. A implementação do algoritmo foi bastante satisfatória, assim como os resultados experimentais, que confirmaram a importância do método de classificação por SVM's. Ficou clara também, a aplicabilidade do algoritmo em outros problemas de classificação.

Referências

1 - LYMAN, Peter and Hal R. Varian. **How Much Information**. Disponível em: http://www.sims.berkeley.edu/how-much-info-2003. Acesso em: 20/04/2006

2 - HAYES, P. & WEINSTEIN, S. Construe/tis: a system for content-based indexing of a database of news stories. em **Annual Conference on Innovative Applications of AI**. 1990.

3 - JOACHIMS, T. **Learning to classify text using Support Vector Machines: methods, theory and algorithms**. Kluwer Academic Publishers. Dordrecht, NL. 2002.

4 - VAPNIK, V. **Statistical Learning Theory**. Wiley. Chichester, GB. 1998.

5 - DAHL, J. & VANDENBERGHE L. CVXOPT: **A Python Package for Convex Optimization.** Disponível em: http://www.ee.ucla.edu/~vandenbe/cvxopt/. Acesso em 03/02/2006.

6 - CRISTIANINI, N. & SHAWE-TAYLOR, J. **An Introduction to Support Vector Machines and Other Kernel-based Learning Methods**. Cambridge University Press. Cambridge, GB. 2000.