Implementação e Aplicação de CB-SVM

Ivalbert dos Santos Pereira [1], **Anderson Ara** [1] **e Ricardo Rocha** [1]

[1] Departamento de Estatística, Universidade Federal da Bahia, Brasil

Resumo

O OBJETIVO desse trabalho é aplicar uma variação do método de classificação SVM chamado Clustering-Based SVM (YU at all, 2005). Essa variante propõe-se a tornar possível a utilização de máquinas de vetor suporte em grandes bases de dados. Assim, são aplicados o CB-SVM e SVM em uma base de dados grande e comparados os resultados em termos de tempo de processamento e poder preditivo.

Keywords: *Métodos de classificação; SVM; CB-SVM; Big Data.*

1. Introdução

NESTE trabalho de conclusão de curso aplicou-se o método denominado CB-SVM (Clustering Based SVM) proposto por (YU et al., 2005). Esta abordagem efetua refinamentos sucessivos no modelo gerado através da seleção apropriada de amostras do conjunto de treinamento. Segundo os autores, o modelo final gerado mantém boa acurácia mesmo fazendo uso de um número reduzido de observações do conjunto de treinamento. Esta metodologia requer a aplicação prévia de um tipo de agrupamento hierárquica no conjunto de dados para gerar uma árvore de *features* denominada CF Tree.

2. A proposta

DENOMINADO de **BIRCH** (*Balanced Iterative Reducing and Clustering Using Hierarchies*), esse método apresentado por (ZHANG; RAMAKRISHNAN; LIVNY, 1996) foi concebido para resolver o problema de identificar áreas densamente populadas, ou *clusters*, em conjuntos de dados multi-dimensionais massivos. Assim, os autores definiram o problema como: **É desejável que se leve em conta a quantidade de tempo que o usuário deseja esperar pelo resultado do algoritmo**.
Esse método cria tipicamente uma boa representação dos dados (*cluster*) em um único *scan* no conjunto de dados e pode, opcionalmente, refinar o resultado com processamento adicional na árvore criada.

Cada *clusters* possui as segintes propriedades:

- X0: Centroide $\frac{\sum_{i=1}^{N} \vec{X}_i}{N}$
- R: Raio do cluster $\sqrt{\frac{\sum_{i=1}^{N} (\vec{X}_i - \vec{X}_0)^2}{N}}$
- D: Diâmetro do cluster $\sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (\vec{X}_i - \vec{X}_j)^2}{N}}$

Essas propriedades podem ser calculadas pelo vetor de features (N, LS, SS), onde:

- N Nº de pontos no cluster
- LS Soma linear dos pontos no cluster $\sum_{i=1}^{N} \vec{X}_i$
- SS Soma quadrática dos pontos $\sum_{i=1}^{N} \vec{X}_i^{\,2}$

O agrupamento ocorre em fases distintas a saber:

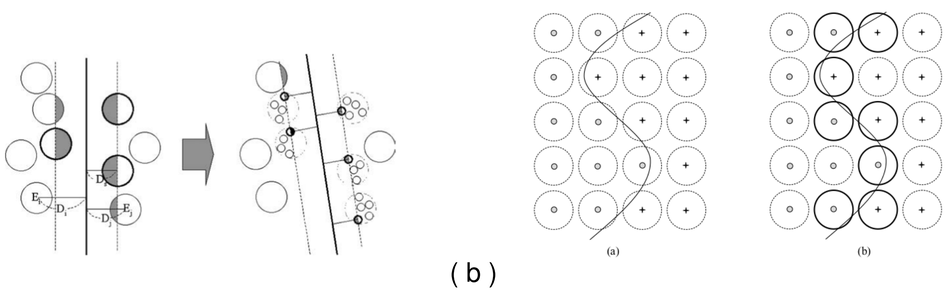
- Fase 1: Constroi uma árvore inicial ajustada à memória;
- Fase 2 (opcional): Varre os nós folhas e constroi uma árvore menor removendo outliers e agrupando clusters muito populosos em clusters maiores;
- Fase 3: Aplica refinamento na árvore para gerar um conjunto de clusters que descreve melhor os padrões de distribuição dos dados;
- Fase 4: Redistribui os clusters para centroides mais próximos e obtém um conjunto de novos clusters. Pode ser extendida para uma fase extra que escaneia o conjunto de dados novamente, remove outliers e pode rotular os dados indicando em qual cluster esse ponto pertence

O resultado do método é uma árvore de *clusters* (não de pontos de dados) que será utilizada para a construção do modelo SVM.

CB-SVM foi projetado para lidar com conjuntos de dados muito grandes dadas limitações de recursos de sistema, como por exemplo, memória.

A ideia fundamental dessa abordagem é utilizar a técnica de micro-agrupamento hierárquico para obter uma descrição mais precisa dos dados próximos às fronteiras de classificação e outra descrição menos precisa dos dados mais afastados dessas fronteiras. São construídas árvores de micro *clusters*, uma para cada classe, onde cada nó é a representação resumida dos nós filhos.

Em conjuntos de dados linearmente separáveis, é definida uma distância D_s do plano de separação à uma margem. Então, os *clusters* que satisfazem a restrição $D_i - R_i < D_s$ são considerados *low margin clusters* e terão seus *subclusters* utilizados na construção do hiperplano de separação g da próxima iteração. Onde D_i e R_i são respectivamente a distância do centroide ao plno de separação e o raio de um *cluster* E_i.



(a) (b)

Figure 1: *Processo de desagrupamento: (a) dados linearmente separáveis, (b) dados não-linearmente separáveis.* ***Fonte:*** *(YU at all, 2005)*

No caso de um conjunto de dados não-linearmente separável, o processo de criação das árvores é o mesmo, o que muda é a forma como os *subclusters* são escolhidos a cada iteração. Para contornar essa dificuldade os autores do método propuseram mais dois passos simples:

- São gerados dados artificiais na fronteira dos *clusters*;
- Faz-se a classificação nos dados artificiais e escolhem-se os *clusters* que obtiverem erros de classificação.

3. Aplicação

O AMBIENTE escolhido para os testes foi o RStudio e a implementação do algoritmo está na linguagem R.
Foram realizados testes iniciais com os dados artificiais para descobrir como os parâmetros afetam o desempenho do CB-SVM.

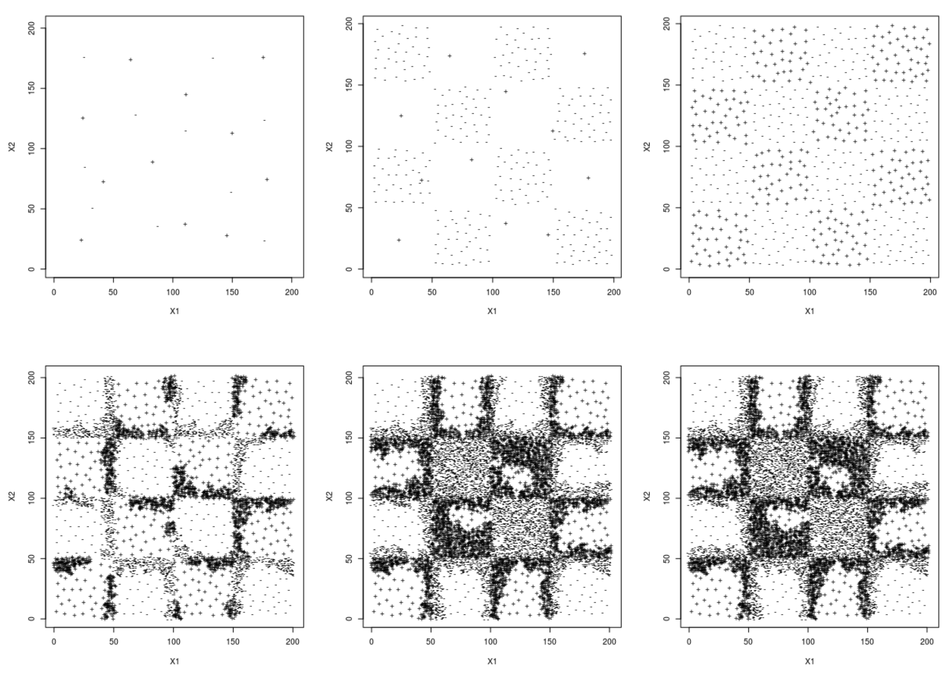
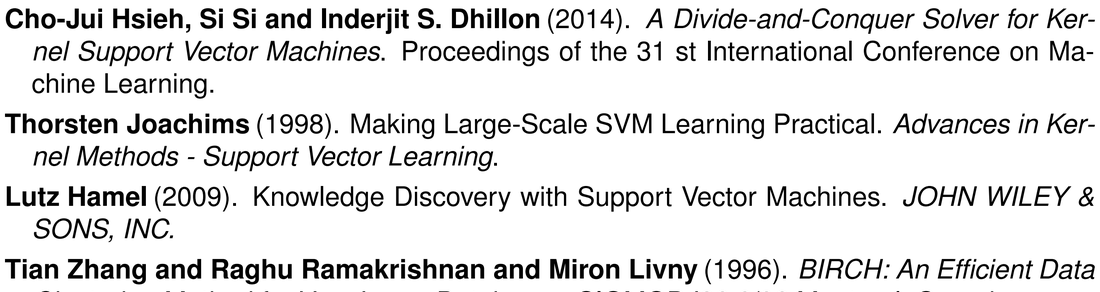


Figure 2: *Evolução da escolha de pontos do dataset.*

CB-SVM processou uma base de aproximadamente 247.000 observações e 116 *features* com 10 *folds* em cerca de 23 minutos e obteve bons resultados em termos numéricos (acima de 90%). Tentou-se aplicar o mesmo procedimento com o SVM (10 *folds*), mas após 11 horas de processamento, apenas os *folds* de 1 a 5 haviam finalizado. Por isso, o processo foi interrompido e uma amostra de 50.002 (cinquenta mil e duas) observações foi extraída e os dois métodos foram aplicados (*K-fols* com k=2).

Table 1: *Métricas associadas a tempo de execução.*

Classificadores	BAcc	Acc	Sens	Espec	VPP	VPN	F1Score	Tempo(s)
CB-SVM	0,994	0,993	0,991	0,998	0,999	0,978	0,995	61,44
SVM	0,991	0,987	0,981	1	1	0,956	0,991	297,19



(a) (b)

Figure 3: *Comparativo: (a) predição, (b) tempo de execução*

Conclução̧es

CONSTATAMOS que os resultados dos testes com um *dataset* relativamente grande revelaram o poder de processamento e assertividade do método frente a uma implementação clássica de SVM. Apesar disso, existem algumas oportunidades de melhoria que tornarão o algoritmo implementado ainda mais rápido, como a paralelização da construção da árvore de *features* e melhoria do algoritmo de varredura de árvore.

Referências

HWANJO YU and JIONG YANG and JIAWEI HAN and XIAOLEI LI (2005). *Making SVMs Scalable to Large Data Sets using Hierarchical Cluster Indexing*. Data Mining and Knowledge Discovery

Cho-Jui Hsieh, Si Si and Inderjit S. Dhillon (2014). *A Divide-and-Conquer Solver for Kernel Support Vector Machines*. Proceedings of the 31 st International Conference on Machine Learning.

Thorsten Joachims (1998). Making Large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*.

Lutz Hamel (2009). Knowledge Discovery with Support Vector Machines. *JOHN WILEY & SONS, INC.*

Tian Zhang and Raghu Ramakrishnan and Miron Livny (1996). *BIRCH: An Efficient Data Clustering Method for Very Large Databases*. SIGMOD ’96 6/96 Montreal, Canada.