Redução do conjunto de dados de treinamento para melhorar a eficiência do classificador SVM

Peterson L. Sarmento[1], Luciano V. Dutra[2], Guaraci J. Erthal[2]

[1]Programa de Mestrado em Computação Aplicada – CAP
Instituto Nacional de Pesquisas Espaciais – INPE

[2]Divisão de Processamento de Imagens
Instituto Nacional de Pesquisas Espaciais – INPE

petersarmento@homail.com , {dutra,guaraci}@dpi.inpe.br

Abstract. *The Support Vector Machine classifier (SVM), a classifier that basically uses training data near the decision boundary is often used due to its good performance. A disadvantage of this method is the increase training time of the classifier as the training set size increases. The purpose of this paper is to apply techniques to reduce the training set size preserving the classification accuracy. The techniques used are editing, multiediting and condensing of data, already applied in data reduction for k-nearest neighbor method (k-NN), which is also a classifier that operates on the data near the border separating classes. It is expected that accuracy with reduced data sets and there is gain in time to estimate the classifier parameters.*

Resumo. *O classificador Máquina de Vetores Suporte (Support Vector Machine – SVM), um classificador supervisionado que utiliza basicamente os dados das classes próximos à fronteira de decisão, é utilizado frequentemente devido ao seu bom desempenho. Uma desvantagem deste método é o aumento no tempo de treinamento do classificador à medida que o tamanho do conjunto de treinamento aumenta. A proposta deste trabalho é aplicar técnicas de redução do conjunto de treinamento mantendo a acurácia da classificação. As técnicas utilizadas serão edição, edição múltipla, e condensação dos dados, aplicados na redução dos dados para o método k-vizinhos mais próximos (k nearest neighbor – k-NN), que também é um classificador supervisionado que atua sobre os dados próximos a fronteira de separação entre as classes. Espera-se que, com a redução dos dados, seja mantida a aurácia e que haja ganho no tempo de estimativa dos parâmetros do classificador.*

Palavras-chave: *Redução de dados, SVM, KNN, Tempo, edição, edição múltipla, condensação.*

1. Introdução

Na literatura de Reconhecimento de Padrões há uma série de técnicas de classificação supervisionada bem conhecidas, podendo-se citar por exemplo, os classificadores: vizinho mais próximo, máxima probabilidade *a posteriori*, perceptron multicamadas,

árvore de decisão, e máquina de vetores suporte [Webb 2002]. Dentre esses classificadores, a máquina de vetores suporte (SVM) apresenta características desejáveis como boa capacidade de generalização, convexidade da função objetivo [Webb 2002]. Sendo um classificador supervisionado, o SVM exige, na fase de treinamento, um conjunto de dados rotulados. Apesar das suas boas qualidades, o procedimento de treinamento do SVM não é adequado para trabalhar com grandes conjuntos de dados por exigir a solução de um problema de programação quadrática para construir o modelo do classificador, o que implica em altos custos de memória e tempo de processamento [Romero 2011]. Métodos têm sido propostos para viabilizar a utilização do SVM para grandes conjuntos de dados, podendo ser divididos em dois grupos: (1) modificar algoritmo SVM de modo que ele possa ser aplicado a grandes conjuntos de dados, e (2) selecionar exemplos ou protótipos representativos do conjunto de treinamento, de modo que o SVM padrão possa tratar [Romero 2011]. A abordagem deste trabalho é direcionada ao estudo da redução do tamanho do conjunto de treinamento, com o intuito de reduzir o tempo de estimativa dos parâmetros do classificador, procurando manter a acurácia da classificação.

2. Objetivos

Os objetivos principais deste trabalho são: (1) Aplicar métodos para reduzir o tamanho do conjunto de dados a serem usados no treinamento do SVM; (2) Testar a eficiência destes métodos para seleção de dados que sejam adequados para o classificador SVM; (3) verificar o potencial de cada método para reduzir os tempos de treinamento e classificação e ao mesmo tempo manter a acurácia próxima da obtida pela utilização dos dados completos.

3. Fundamentação teórica

O modelo básico do classificador SVM consiste em resolver o problema de classificação binário (duas classes: ω_1 e ω_2) definindo, no espaço de atributos, uma superfície de decisão linear (hiperplano) dada por $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = 0$, onde $\mathbf{x} \in \mathrm{R}^p$, $\mathbf{w}$ define a orientação do hiperplano de separação e w_0 dá a posição do hiperplano. Para tanto, será utilizado um conjunto de amostras rotuladas dado por $\{(\mathbf{x}_1, y_1), ...,(\mathbf{x}_n, y_n)\}$, onde $y_i \in \{+1, -1\}$ e com $(\mathbf{x}, +1) \in \omega_1$ e $(\mathbf{x}, -1) \in \omega_2$.

O modelo do SVM para o caso linearmente separável define como solução ótima, o hiperplano que maximiza a *margem* entre as duas classes como mostra a Figura 1 (a) [Webb 2002]. Os pontos de ambas as classes localizados sobre os limites da margem (indicados pelas linhas tracejadas) são os *vetores de suporte* e serão utilizados para a definição da superfície de decisão do classificador.

A Figura 1 (b) mostra o caso em que os dados não são linearmente separáveis. Neste caso a solução (hiperplano) desejada deve ser obtida pela maximização da margem e pela penalização dos pontos dentro da margem, ou classificados erroneamente. Para tanto as seguintes restrições são impostas: $y_i\,(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 - \xi_i$, $\forall i = 1,..., n$ e onde $\xi_i \geq 0$ é a variável que quantifica a pena associada ao exemplo $\mathbf{x}_i$. A solução é obtida pela minimização da seguinte função objetivo:

$$\frac{1}{2}\|\mathbf{w}\| + C\sum_{i=1}^{n}\xi_i \quad \text{sujeito a: } \xi_i \geq 0 \text{ e } y_i\,(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 - \xi_i \text{ , } \forall i = 1,..., n \qquad (1)$$

onde o parâmetro $C > 0$ controla o compromisso entre a margem de separação e o número de erros. Tal problema de otimização pode ser resolvido com a introdução da

função Lagrangiana e posterior solução do problema dual, que implica em maximizar as variáveis α_i e minimizar $\mathbf{w}$ e w_0. O problema pode ser posto como

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \text{ sujeito a } \sum_{i=1}^{n} \alpha_i y_i = 0 \text{ e } 0 \le \alpha_i \le C, \forall i = 1,...,n \quad (2)$$

onde α_i são os multiplicadores de Lagrange e a solução será dada por

$$\mathbf{w} = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i \text{ e } w_0 = \frac{1}{|SV|} \sum_{i \in SV} \alpha_i (y_i - \mathbf{w}^T \mathbf{x}_i) \quad (3)$$

onde *SV* é o conjunto de índices para os vetores de suporte e |*SV*| é a cardinalidade de *SV*. Finalmente o classificador será dado pela seguinte regra:

$$\text{Se } f(\mathbf{x}) \ge 0 \text{ então } \mathbf{x} \in \omega_1 \text{ senão } \mathbf{x} \in \omega_2 \quad (4)$$

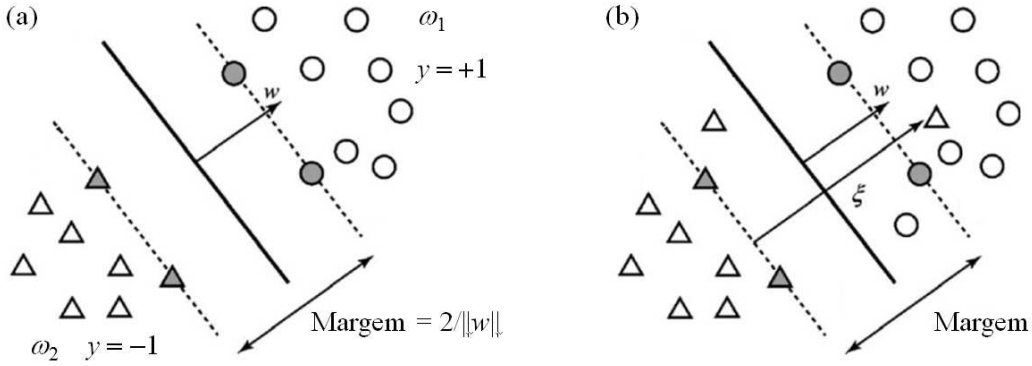


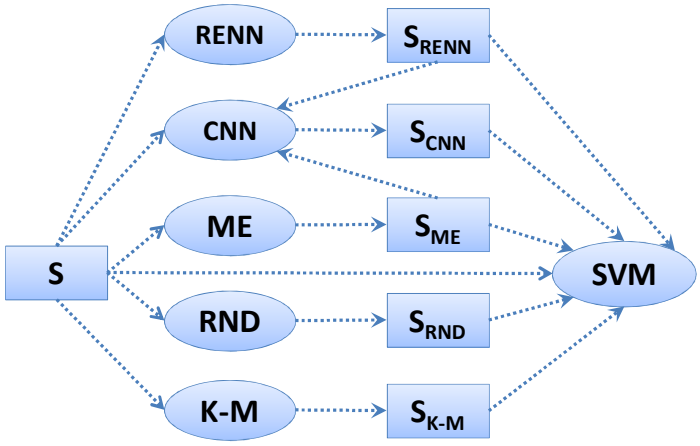
Figura 1 – Modelo linear para o SVM: (a) caso separável e (b) caso não separável [adaptado de Foody 2006].

O problema apresentado em (2) mostra claramente a dependência com relação ao número de amostras (*n*). Dentre as alternativas existentes para reduzir o tamanho deste problema, está o algoritmo SMO (*sequential minimal optimization*) [Platt 1998] que divide o problema de tamanho *n* numa sequência de problemas de tamanho dois que são resolvidos analiticamente.

Uma vez que apenas os vetores de suporte participam da construção do modelo de classificação, como é mostrado em (3), métodos de redução de dados são usados alternativamente para reduzir o custo computacional. Vários métodos são conhecidos na literatura [Wilson e Martinez 2000] para reduzir o custo do classificador vizinho mais próximo e tem mostrado sua utilidade ao serem aplicados aos dados de treinamento do SVM [Romero 2011], eliminando pontos que não participam da geração do modelo.

4. Módulos de redução

No desenvolvimento do trabalho pretende-se implementar e aplicar os algoritmos de redução em processos que podem ser definidos como módulos de execução. O diagrama descrito na Figura 2 mostra os módulos de redução de dados e as possíveis combinações, pretendidas neste trabalho, para realizar a redução do conjunto de amostras inicial *S* e utilizá-lo como conjunto de treinamento para construir o classificador SVM. As setas indicam o fluxo de processamento.



RENN – Edição de Wilson Repetida
CNN – Condensação de Hart
ME – Multiedição de Devijver e Kittler
RND – Seleção aleatória
K-M – Seleção de protótipos pelo *k*-Médias
SVM – Classificador SVM
S – Conjunto de amostras
S_X – Conjunto obtido pelo alg. X

Figura 2 – Processos de redução do conjunto de amostras S

Os métodos RENN, CNN e ME utilizam o classificador *k*-NN na redução de dados. A descrição do *k*-NN é feita na Seção 5. O RND é um método de seleção aleatória e o K-M um algoritmo de classificação não supervisionada (*k*-médias). A descrição dos métodos é feita na Seção 6.

5. Classificador vizinho mais próximo

O algoritmo *k* vizinhos mais próximos (*k nearest neighbor* – *k*-NN) é um método para classificar objetos com base em exemplos de treinamento mais próximos a ele no domínio do espaço de atributos [Webb 2002]. Os passos principais do *k*-NN são mostrados na Figura 3.

ALGORITMO *k*-NN
Entrada: conjuntos S_T (treinamento) e S_C (a classificar)
Saída: conjunto S_C classificado
Passo 1 – Para cada instância **x** do conjunto S_C:
Encontrar os *k* vizinhos mais próximos de x, em S_T.
Classificar **x** para a classe majoritária entre os k pontos selecionados.
Passo 2: Retornar o conjunto S_C classificado.

Figura 3 – Algoritmo *k*-NN

A medida de proximidade comumente utilizada é a distância Euclidiana. O valor de *k* deve ser escolhido de forma a maximizar a acurácia de classificação. Os algoritmos de redução dos dados do conjunto de treinamento adotados neste artigo utilizam o método do vizinho mais próximo.

6. Algoritmos de Redução

Para este trabalho serão apresentados os métodos de redução por edição de Wilson repetida (RENN), edição múltipla de Devijver e Kittler (ME) e condensação de Hart (CNN). Além destes métodos, serão mostrados também os resultados para seleção aleatória de dados do conjunto de treinamento.

6.1 – **Edição de Wilson Repetida - RENN**

O algoritmo de edição Wilson (*repeat editing nearest neighbor* – RENN) é um dos primeiros métodos de edição propostos na literatura, a fim de reduzir o conjunto de treinamento pela regra do vizinho mais próximo, através da eliminação de instâncias classificadas erroneamente.

O modelo proposto por Wilson [Wilson 1972] pode ser resumido como: se uma instância é erroneamente classificada pelo método *k*-NN, onde *k* representa o número de vizinhos mais próximos, ela será removida do conjunto de treinamento.

O método de estimativa de erro utilizado neste algoritmo corresponde ao *leaving-one-out*: um ponto é separado do conjunto para teste e o restante é utilizado para gerar o modelo. O algoritmo RENN é descrito na Figura 4.

ALGORITMO RENN
Entrada: conjunto de treinamento *S*, número de vizinhos *k*.
Saída: conjunto editado $S_{RENN} \subseteq S$
Passo 1 – Classificar cada instância **x** do conjunto S com o *k*-NN.
Passo 2 – Eliminar de *S* todas as instâncias classificadas incorretamente.
Passo 3 – Repetir os Passos 1 e 2 até que não haja eliminação de instâncias.
Passo 4 – Retornar $S_{RENN} = S$

Figura 4 – Algoritmo *k*-NN

Mesmo sendo considerado simples, o algoritmo de Wilson tem complexidade $O(n^2)$ onde $n = |S|$.

6.2 – **Condensação de Hart - CNN**

O método CNN (*condensed nearest neighbor*) proposto por Hart [Hart 1968] baseia-se na ideia de que se um objeto é classificado incorretamente, ele está próximo da fronteira de decisão, logo deverá permanecer no subconjunto condensado.

Inicialmente uma amostra é transferida do conjunto de treino *S* para um conjunto vazio S_C. Cada amostra de *S* é então classificada pelo método 1-NN, usando S_C como referência. Somente as amostras classificadas incorretamente vão para o conjunto condensado S_C. Desta maneira o conjunto final condensado S_C terá somente as amostras classificadas incorretamente, ou seja, aquelas que estão próximas da fronteira de decisão. É fácil perceber que o algoritmo é dependente da ordem em que os padrões são analisados. O algoritmo CNN é descrito na Figura 5.

ALGORITMO CNN
Entrada: Conjunto de treinamento *S*
Saída: Subconjunto condensado $S_{CNN} \subseteq S$
Passo 1 - Criar o conjunto S_{CNN} vazio.
Passo 2 - Transferir aleatoriamente um padrão do conjunto *S* para S_{CNN}.
Passo 3 - Para cada **x** de S, fazer:
Classificar **x** pela regra 1-NN usando o conjunto S_{CNN} .
Se x é classificada incorretamente então transferir **x** de S para S_{CNN}.
Passo 4 - Repetir o passo 3 até que: $S = \{\}$, ou não há transferências para S_{CNN}.

Figura 5 – Algoritmo *k*-NN

A desvantagem deste método é que quando os dados são excluídos, a característica estatística dos dados também é eliminada. Uma vantagem do modelo é o alto índice de redução.

6.3 – **Edição múltipla de Devijver e Kittler - ME**

No caso do algoritmo de Wilson, é incorreto assumir que a estimativa feita em cada uma das instâncias do conjunto de treino é estatisticamente independente, portanto, não será possível realizar uma análise do comportamento assintótico correspondente ao conjunto editado. Para resolver esta dificuldade, Devijver e Kittler (1986) propõem o algoritmo de edição múltipla (*Multiedit* - ME) baseado na edição de Wilson, mas mudando o método de estimar a participação de uma instância.

Segundo este modelo, o método de estimativa consiste em particionar o conjunto de treinamento em N blocos e, após o particionamento, fazer uma estimativa para cada bloco i, usando o bloco ((i +1) módulo N) para projetar o classificador.

Assim, a técnica ME permite fazer uso repetido da ideia de edição de dados para cada bloco. O algoritmo ME é descrito na Figura 6:

ALGORITMO ME
Entrada: Conjunto de treinamento S e tamanho da partição N.
Saída: Conjunto $S_{ME} \subseteq S$
Passo 1. Particionar aleatoriamente S em N subconjuntos, $S_{(0)}...S_{(N-1)}$, com $N \geq 3$.
Passo 2. Classificar $S_{(i)}$ utilizando o método 1-NN usando $S_{((i+1) \bmod N)}$ como um conjunto de treino, para todo i = 0, ..., N–1.
Passo 3. Descartar todas as amostras erroneamente classificadas no passo 2.
Passo 4. Constituir um novo conjunto S, a partir conjuntos editados $S_{(0)}$, ..., $S_{(N-1)}$.
Passo 5. Encerramento: Se as últimas I iterações não produzirem nenhuma edição, então sair com o conjunto $S_{ME} = S$. Caso contrário, retornar ao passo 1.

Figura 6 – Algoritmo ME

6.4 – **Redução por seleção aleatória - RND**

Adotou-se o método de seleção aleatória, pela rapidez e facilidade de implementação. Este método será utilizado como referência para comparação com os outros métodos.

O método consiste em selecionar aleatoriamente uma fração do conjunto de dados de treinamento e utilizá-la para treinamento do classificador SVM.

Espera-se que este método seja um limitante inferior no tempo de processamento e um limitante superior para o erro de classificação.

7. Resultados

A primeira análise a ser feita envolve a identificação dos pontos que são considerados como vetor de suporte. Testes efetuados com uma imagem com três atributos (RGB) e duas classes, mostraram que os pontos que podem ser utilizados para serem vetores de suporte são na grande maioria, os mesmos. Para tanto, foram criados dois conjuntos de amostras de treinamento A e B de diferentes tamanhos. Através do algoritmo de treinamento do SVM (Seção 3) foram obtidos os vetores de suporte de A e B, representados, respectivamente, pelos conjuntos A-SVs e B-SVs.

A Tabela 1 apresenta os tamanhos de amostra para cada classe e para cada conjunto de treinamento, para os classificadores SVM linear (núcleo linear) e não linear (núcleo RBF). São também apresentados os tempos de treinamento e classificação (em segundos) para cada conjunto e para cada modelo de classificador.

Tabela 1 – Vetores de Suporte X Tempo de Treinamento

	Kernel Linear				Kernel RBF	
AMOSTRAS	A	A - SVs	B	B - SVs	B	B - SVs
Tamanho Classe 1	158	5	847	10	847	14
Tamanho Classe 2	208	6	812	10	812	12
Tempo Treino (s)	2,63±1,92	0,03±0,01	136,2±149,28	0,15±0,08	71,37±178,13	0,16±0,07
Tempo Classif. (s)	12,93±0,03	12,91±0,03	23,24±0,03	23,26±0,02	71,97±0,07	64,44±2

Com base na Tabela 1, podemos verificar que o tempo de treinamento do SVM reduz consideravelmente (de 9 a 10 vezes para o conjunto maior) quando analisamos o conjunto completo A e o seu respectivo conjunto reduzido A-SVs. O mesmo acontece para o conjunto completo B e seu respectivo conjunto reduzido B-SVs.

A segunda análise considera a redução por seleção aleatória de amostras. Este método não faz discernimento entre as amostras que podem ser vetores de suporte e as demais, o que pode levar à exclusão de pontos que podem ser importantes para a construção da superfície de decisão entre as classes. Para analisar a acurácia da classificação para este método, foram utilizados percentuais de seleção entre 10% e 90%, com variação de 10%, em uma imagem com desvio padrão menor que a da utilizada na terceira análise. Para cada conjunto foram realizadas 30 simulações e os resultados são apresentados na Figura 7.

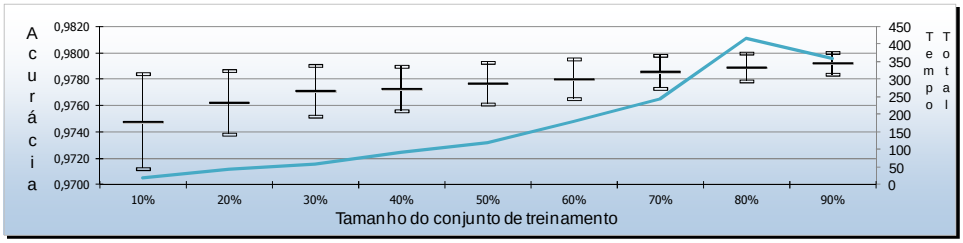


Figura 7 – Acurácia de classificação para o RND

A análise da figura 7 mostra o crescimento da acurácia e a redução do desvio padrão com aumento no tamanho do conjunto de amostras selecionadas. Este resultado mostra a relação entre a quantidade de amostras do conjunto de treinamento e a acurácia da classificação.

Na análise seguinte, objeto principal deste trabalho, utilizou-se uma imagem sintética de 3 canais, com tamanho de 512x512 píxeis e contendo 2 classes. O tamanho do conjunto de amostras de treinamento foi de 681 para a primeira classe e 657 para a segunda classe. Os parâmetros do classificador SVM utilizados são: núcleo linear e C =90. Foram executados testes de redução de dados pelos algoritmos RENN, CNN, ME e RND (percentual de seleção = 50%). Também foram realizadas reduções em cascata, com as combinações: RENN+CNN e ME+CNN.

Para cada módulo de redução, foram realizados 50 experimentos. Os valores de k utilizados nos módulos RENN, ME e CNN, foram respectivamente 3, 1 e 1 vizinhos mais próximos e N=3 para o módulo ME. Os resultados estão sintetizados na Tabela 2 abaixo, que apresenta, para cada método, a acurácia de classificação (exatidão global), os tempos de redução de dados, treinamento e classificação, o tamanho dos conjuntos de dados, os números de vetores de suporte médios e os tempos totais (redução, treino e classificação).

Tabela 2 – Resultados para os Módulos de Redução

Método	SVM	RENN	ME	CNN	RENN+CNN	ME+CNN	RND
Acuracia (%)	92,440	93,839	93,932	93,995	93,991	93,974	93,977
Tempo Redução (s)	0,0	4,6	1,8	0,9	5,0	2,0	0,0
Tempo Treino (s)	15,5	24,5	28,1	11,2	3,7	0,7	69,5
Tempo Classificação (s)	105,8	75,1	47,9	140,6	50,1	24,6	95,8
Nro de amostras	1338	1235	1191	269	105	55	669
Nro de V.S.	119	84	54	159	56	27	110
Tempo Total	121,3	104,2	77,8	152,8	58,7	27,3	165,4

8. Conclusões

Como conclusão inicial é possível afirmar que, para os experimentos realizados, os processos, RENN, ME, RENN+CNN e ME+CNN permitem reduzir o tempo total de processamento em relação ao procedimento convencional. O tempo de classificação da imagem também foi reduzido, incentivando a aplicação destes métodos. Dentre as características de cada algoritmo de redução, pode-se dizer que o RENN e o ME buscam eliminar amostras classificadas incorretamente, eliminando amostras próximas ou distantes a fronteira de decisão. O algoritmo CNN realiza o processo inverso, mantendo justamente as amostras classificada erroneamente. As reduções RENN e ME eliminam pontos, mas não o suficiente para que a redução no tempo de treinamento seja realmente significativa. A redução CNN elimina uma grande quantidade de amostras, mais ainda assim o conjunto obtido gera um grande número de V.S. aumentando o tempo de classificação. A utilização conjunta dos módulos RENN+CNN e ME+CNN, mostrou-se mais eficaz que os demais, reduzindo o tempo de treinamento, tempo de classificação e mantendo a acurácia da classificação próxima a do classificador SVM sem utilização destes redutores. Em relação à acurácia, verificou-se que todos os métodos de redução ficaram com valor superior ao classificador SVM sem redutores.

A continuidade deste trabalho abrangerá imagens com mais de duas classes e tamanhos de amostras superiores a 3.000 pontos. Também pretende-se implementar métodos de redução seguindo a abordagem da geração de protótipos artificiais (os algoritmos *k*-médias e do líder são alternativas possíveis).

Referências

Devijver, P., Kittler, J. (1986) *Pattern Recognition-A Statistical Approach,* Prentice Hall.

Foody, G.M., Mathur, A. The use of small training sets containing mixed pixels for accurate hard image classification: Training on mixed spectral responses for classification by a SVM. *Remote Sensing of Environment*, 103 (2): 179-189, 2006.

Hart, P. E. The Condensed Nearest Neighbour Rule. *IEEE Transactions on Information Theory*, 14 (3): 515-516, 1968.

Platt, J. (1998), Fast training of support vector machine using sequential minimal optimization, in: *Advances in Kernel Methods: Support Vector Machine*, MIT Press.

Romero, E. Using the Leader Algorithm with Support Vector Machines for Large Data Sets. *ICANN* 2011, Part I, LNCS 6791, p.225–232, Espoo, Finland, 14-17 june 2011.

Theodoridis, S., Koutroumbas, K. (2008) *Pattern Recognition*, Academic Press, 4th edition.

Webb, A.R. (2002), *Statistical Pattern Recognition*, John Wiley & Sons, 2nd edition.

Wilson, D.L. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Mans and Cybernetics*, 2 (3): 408-421, 1972.

Wilson, D.R., Martinez, T.R. Reduction Techniques for Instance-Based Learning Algorithms. *Machine Learning*, 38 (3): 257-286, 2000.