

EXTRAÇÃO DE REGRAS FUZZY DE MÁQUINAS VETOR SUPORTE PARA CLASSIFICAÇÃO EM MÚLTIPLAS CLASSES

ADRIANA CHAVES, MARLEY VELLASCO, RICARDO TANSCHKEIT

*Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro
CP 38063, 22452-970 Rio de Janeiro, RJ*

Emails: achaves@inf.puc-rio.br, marley@ele.puc-rio.br, ricardo@ele.puc-rio.br

Abstract - This paper proposes a new method for fuzzy rule extraction from trained support vector machines (SVMs) for multi-class problems. SVMs have been applied to a wide variety of application. However, SVMs are considered “black box models”, where no interpretation about the input-output mapping is provided. Some methods to reduce this limitation have already been proposed, however, they are restricted to binary classification problems and to the extraction of symbolic rules with intervals or functions in their antecedents. Hence, to improve the interpretability of the generated rules, this paper presents a new model for extracting fuzzy rules from a trained SVM. Moreover, the proposed model was developed for classification in multi-class problems. The proposed method was evaluated in four benchmark databases and results show that it is capable of generating a set of interpretable rules that explain the database, as well as the influence of the input variables in the determination of the final class.

Keywords- Support vector machines, rule-extraction, fuzzy rules, interpretability, multi-class classification

Resumo - Este trabalho propõe um novo método de extração de regras fuzzy a partir de máquinas de vetor suporte (SVMs) para problemas de classificação em múltiplas classes. Máquinas de vetor suporte são sistemas de aprendizado baseados na teoria do aprendizado estatístico, que obtiveram sucesso na solução de vários problemas. Entretanto, SVMs são modelos “caixa preta”, em que não há interpretação sobre o mapeamento entrada-saída. Alguns métodos para extração de regras a partir de SVMs treinadas já foram desenvolvidos, mas estão restritos a problemas de classificação binária e à extração de regras simbólicas com intervalos ou funções em seus antecedentes. Esse formato diminui a interpretabilidade das regras geradas e prejudica a capacidade de extração de conhecimento. Assim, para aumentar a interpretabilidade lingüística das regras geradas, propõe-se uma metodologia para extrair regras fuzzy a partir de SVMs treinadas, em que os antecedentes das regras estão associados a conjuntos fuzzy. Além disso, o modelo proposto foi desenvolvido para problemas de classificação em múltiplas classes. O método proposto foi avaliado em quatro bancos de dados e os resultados obtidos demonstram sua capacidade de gerar um conjunto de regras interpretáveis que explique os bancos de dados, bem como a influência das variáveis de entrada na determinação da classe final.

Palavras-chave - Máquinas de vetor suporte, extração de regras, regras fuzzy, interpretabilidade, classificação em múltiplas classes

1 Introdução

Máquinas de vetor suporte (SVMs) são paradigmas de aprendizado baseados na teoria do aprendizado estatístico (Cristianini and Shawe-Taylor, 2000), (Gunn, 1998), (Haykin, 1999), (Scholkopf and Smola, 2002), (Vapnik, 1998), (Vapnik, 1999), que têm sido aplicadas com excelente desempenho a uma variedade de problemas de classificação e de regressão (Drucker, et al., 1999), (Joachims, 1998), (Brown et al., 2000), (Müller et al., 1997).

Apesar de seu excelente desempenho, SVMs, da mesma forma que redes neurais artificiais, são modelos “caixa preta”, i.e., modelos que não explicam o processo pelo qual um dado resultado é obtido. O mapeamento entrada-saída resultante é composto de uma combinação linear de funções kernel (Cristianini and Shawe-Taylor, 2000), o que é difícil de ser interpretado.

Já foram propostos alguns algoritmos para extrair conhecimento de SVMs treinadas, entre eles RulExtSVM (Fu et al., 2004) e SVM+Prototypes (Nuñez et al., 2002). O algoritmo RulExtSVM extrai regras do tipo SE-ENTÃO com intervalos, definidos

por formas hiper-retangulares nos antecedentes das regras. O método SVM+Prototype calcula elipsóides (chamados protótipos) baseados nos vetores suporte obtidos para cada classe. Esses elipsóides também são usados nos antecedentes das regras.

A desvantagem do RulExtSVM é a construção do mesmo número de hiper-retângulos que o número de vetores suporte, o que é caro computacionalmente. No método SVM+Prototype, a construção de elipsóides com eixos não paralelos aos eixos coordenados reduz a interpretabilidade. Esse problema é intensificado se o espaço de entradas tem dimensão alta, i.e., existem muitos atributos de entrada.

Deve-se destacar que as regras extraídas por ambos os métodos geram, em seus antecedentes, intervalos ou funções. Esse fato diminui a interpretabilidade das regras geradas e prejudica a capacidade de extração de conhecimento. Assim, para melhorar a interpretabilidade lingüística das regras geradas, é apresentada nesse trabalho uma nova metodologia para extrair regras fuzzy de SVMs treinadas. A idéia básica é que a utilização de conjuntos fuzzy nos antecedentes das regras torna as regras resultantes mais flexíveis e interpretáveis.

Esse artigo está dividido em quatro seções adicionais. A seção 2 descreve brevemente a teoria

das máquinas de vetor suporte. A seção 3 introduz o método proposto para a extração de regras fuzzy a partir de SVMs treinadas. O método é apresentado, primeiro, para problemas de classificação binária; em seguida, o modelo de extração de regras fuzzy é estendido para aplicações de classificação com múltiplas classes. A seção 4 expõe o estudo de casos, descrevendo os bancos de dados utilizados e o desempenho obtido com o método de extração de regras fuzzy. Finalmente, a discussão e a conclusão são apresentadas na seção 5.

2 Máquinas de Vetor Suporte

2.1 Classificação Binária

Considere-se um conjunto de treinamento $\{(x_i, y_i)\}$, $i \in \{1, \dots, N\}$, onde $x_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$ e N é o número de padrões. A formulação da máquina de vetor suporte conduz para o seguinte problema de programação quadrática:

$$\text{maximizar } \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (1)$$

$$\text{sujeito a } 0 \leq \alpha_i \leq C \text{ e } \sum_{i=1}^N \alpha_i y_i = 0. \quad (2)$$

A função $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ é uma função *kernel*, onde $\Phi(x_i)$ representa o mapeamento do vetor de entrada x_i em um espaço de dimensão maior (chamado “espaço de características”); C é uma constante de regularização – um parâmetro positivo de treinamento que estabelece um equilíbrio entre a complexidade do modelo e o erro de treinamento; os α_i são os coeficientes de Lagrange.

O problema anterior tem como solução $\alpha_i = 0$ para os padrões que não são vetores suporte. Os vetores suporte serão representados por s_i para serem distinguidos dos outros pontos. Existem várias maneiras de se resolver o problema de otimização quadrática e obter a SVM (Cristianini and Shawe-Taylor, 2000), (Haykin, 1999).

Um ponto x é classificado de acordo com o sinal da função de decisão:

$$f(x) = \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, x) + b, \quad (3)$$

onde b é o *bias* e N_s é o número de vetores suporte.

2.2 Classificação em Múltiplas Classes

Como observado na seção anterior, a SVM foi inicialmente definida para classificação binária. Para estender a SVM para classificação em k classes ($k > 2$) pode-se substituir o problema de múltiplas classes por um conjunto de problemas binários ou generalizar a SVM binária (Hsu and Lin, 2002), (Crammer and Singer, 2000). Dois métodos bastante conhecidos baseados na primeira abordagem são a decomposição um por classe (*one-against-all*) (Hsu and Lin, 2002), (Weston and Watkins, 1998), (Rifkin

and Klautau, 2004) e a separação das classes duas a duas (“one-against-one”) (Hsu and Lin, 2002), (Weston and Watkins, 1998), (Abe and Inoue, 2002), (Kressel, 1999). O método de decomposição um por classe constrói k SVMs binárias, que separam cada classe de todas as outras. As saídas de todas as SVMs são então combinadas para gerar a classificação final em k classes. O método mais comum para combinar as saídas das k SVMs, e que é o usado no estudo de casos desse trabalho, é associar o vetor de entrada à classe com o maior valor da função de decisão.

O método de separação das classes duas a duas é baseado na construção de $k(k-1)/2$ SVMs binárias, que separam cada possível par de classes. A classificação final é obtida a partir das saídas de todas as SVMs. A maior desvantagem desse método é o grande número de SVMs que devem ser treinadas, quando comparado com o método de decomposição um por classe. Por outro lado, cada SVM é treinada somente com os padrões das duas classes envolvidas, enquanto que cada uma das k SVMs no método de decomposição um por classe usa todos os pontos de treinamento do conjunto de entrada. Normalmente, a classificação final de um padrão de entrada é feita pela estratégia de voto, usando as saídas das $k(k-1)/2$ SVMs.

A segunda abordagem, baseada na generalização da SVM binária, utiliza uma maneira mais natural de resolver o problema de k -classes com a construção de uma função de decisão que envolva todas as classes. O método mais importante desse tipo é chamado de Crammer e Singer (Crammer and Singer, 2000), onde o problema de k -classes é resolvido por um único problema de otimização (generalização do problema de otimização da SVM binária). Nesse método, todos os dados de treinamento são usados ao mesmo tempo.

Como será descrito na seção 3.2, o método de extração de regras fuzzy proposto pode ser aplicado para quaisquer dos métodos acima. A discussão do desempenho desses métodos é feita na seção 4.

3 Extração de Regras Fuzzy

O método proposto, chamado FREx_SVM, é dividido em três etapas: Projeção dos Vetores Suporte, Definição dos Conjuntos Fuzzy, e Extração de Regras Fuzzy. Por simplicidade, FREx_SVM é descrito primeiramente para a SVM binária. Então, na Seção 3.2, ele é estendido para SVM multiclasse.

3.1 FREx_SVM para SVM Binária

3.1.1 Projeção de Vetores Suporte

No primeiro passo do algoritmo, os vetores suporte obtidos pela SVM são projetados nos eixos coordenados. O número de projeções é igual à

dimensão do espaço de entradas (número de atributos de entradas da base de dados analisada).

3.1.2 Definição dos Conjuntos Fuzzy

O passo seguinte consiste na definição do número de conjuntos fuzzy para cada variável de entrada. Cada conjunto fuzzy é rotulado e a ele é associada uma função de pertinência triangular, normalmente com o mesmo domínio. Os suportes dos conjuntos são definidos de modo que todos os conjuntos tenham pelo menos um elemento, isto é, uma projeção de vetor suporte. Suponha-se que cada atributo do espaço de entradas n -dimensional seja dividido em m conjuntos fuzzy. Nesse caso, o conjunto fuzzy C_{ij} é o j -ésimo conjunto definido para a i -ésima coordenada, onde $i \in \{1, \dots, n\}$ e $j \in \{1, \dots, m\}$. A Figura 1 mostra um exemplo do espaço de entradas bidimensional ($n = 2$), onde cada variável foi particionada em cinco conjuntos fuzzy ($m = 5$).

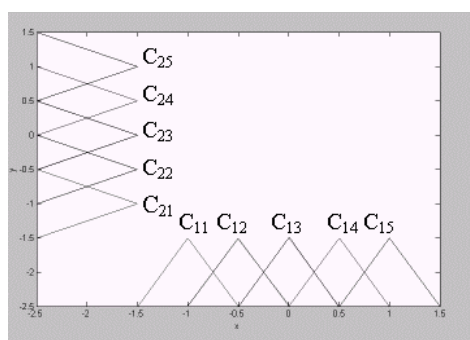


Figura 1. Definição dos conjuntos fuzzy

Cada projeção de vetor suporte obtida no passo anterior (uma projeção para cada variável de entrada) é então avaliada por todas as m funções de pertinência, identificando o conjunto fuzzy no qual a projeção do vetor suporte tem o maior grau de pertinência. Seja x_i a projeção do vetor suporte x na i -ésima coordenada e $\mu_{C_{ij}}(x_i)$ o grau de pertinência de x_i ao conjunto C_{ij} . O maior grau de pertinência é então calculado por:

$$\max_{j \in \{1, \dots, m\}} \{\mu_{C_{ij}}(x_i)\} \quad (4)$$

Como pode ser visto na Figura 2, as projeções dos vetores suporte x são $x_1 = 0,61$ e $x_2 = 0,88$. Cada uma dessas projeções ativa duas funções de pertinência, com os seguintes valores:

$$\begin{aligned} \mu_{C_{14}}(x_1) &= 0,78 & \mu_{C_{15}}(x_1) &= 0,22 \\ \mu_{C_{24}}(x_2) &= 0,24 & \mu_{C_{25}}(x_2) &= 0,76 \end{aligned}$$

Os conjuntos com grau de pertinência máximo são C_{14} e C_{25} , destacados com um quadrado. Para uma melhor visualização, os conjuntos com grau de pertinência igual a 0 não são mostrados na Figura 2.

3.1.3 Extração de Regras Fuzzy

No passo final, cada vetor suporte gera uma regra fuzzy, como descrito abaixo. Para cada vetor suporte x , seja C_{ij^i} o conjunto fuzzy com maior grau de pertinência para cada coordenada x_i , $i = 1, \dots, n$ e $j^i = 1, \dots, m$. A regra gerada pelo vetor suporte x é:

SE x_1 é C_{1j^1} e ... e x_n é C_{nj^m} ENTÃO $x = (x_1, \dots, x_n)$ pertence à classe definida pelo vetor suporte x .

O número máximo de regras é igual ao número total de vetores suporte obtidos. Deve ser observado que, no caso de um grande número de atributos de entrada, as regras geradas perdem um pouco de interpretabilidade.

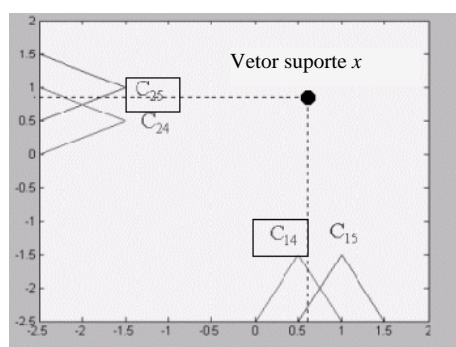


Figura 2. Graus de pertinência máximos

Para avaliar cada regra, duas métricas – *acurácia fuzzy* e *abrangência fuzzy* – são definidas (Chaves et al., 2005) para medir, respectivamente, quão bem a regra representa os dados e quantos padrões são por ela afetados. Regras conflitantes (regras com o mesmo antecedente mas com conseqüentes diferentes) são resolvidas pela escolha da regra com maior acurácia fuzzy.

3.2 FREx_SVM para SVM Multiclasse

Como o método proposto é baseado nos vetores suporte obtidos após o treinamento de todas as SVMs, a abordagem binária apresentada na seção anterior pode ser facilmente estendida para problemas multiclasse. Qualquer um dos métodos multiclasse apresentados na seção 2.2 pode ser utilizado com o FREx_SVM. O diagrama da Figura 3 descreve todo o processo de extração de regras fuzzy para problemas de classificação em múltiplas classes.

Após a escolha do método, as SVMs necessárias são treinadas com os dados pré-processados para a obtenção dos vetores suporte associados a cada classe. Cada método gera um conjunto de regras. Os conjuntos de regras resultantes são então avaliados pelas métricas acurácia fuzzy e abrangência fuzzy, de forma que os métodos de classificação em múltiplas classes possam ser comparados.

Conforme mencionado anteriormente, o método de decomposição um por classe constrói k SVMs para separar cada classe de todas as outras. Seja SVM_i a SVM que separa a classe i de todas as outras;

logo, somente os vetores suporte da classe i são considerados na geração de regras. No método de separação das classes duas a duas, uma SVM é treinada para separar a classe i da classe j ($i \neq j$). Nesse caso, todos os vetores suporte obtidos por essa SVM são utilizados no processo de extração de regras, uma vez que eles definem as classes envolvidas. O mesmo pode ser aplicado para o método de Cramer e Singer, isto é, todos os vetores suporte são usados no processo.

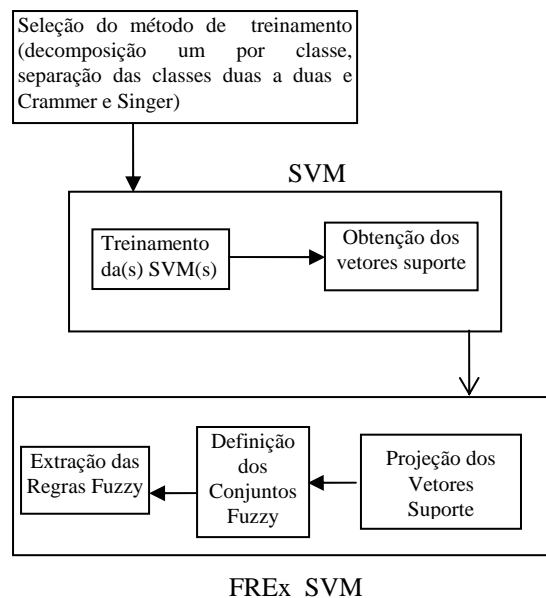


Figura 3. Processo completo de extração de regras fuzzy

4 Estudo de Casos

Para avaliar o método FREx_SVM, foram selecionados quatro bancos de dados: Bupa Liver Disorders, Wisconsin Breast Cancer, Iris e Wine de <http://www.ics.uci.edu/~mllearn/MLRepository.html>. Todas as SVMs foram treinadas com dois kernels diferentes: linear e RBF (funções Gaussianas). Três valores de constante de regularização (parâmetro C , eq. 2) e quatro valores para o desvio padrão foram empregados para treinar cada SVM: $C = 0,1$, 1 e 10 e $\sigma^2 = 1, 5, 10$ e 50 .

Nos bancos de dados de múltiplas classes Iris e Wine, todos os métodos de classificação em múltiplas classes foram avaliados para comparação de seus desempenhos.

Em todos os experimentos, os bancos de dados foram divididos em dois conjuntos disjuntos de mesmo tamanho. Esses dois conjuntos foram usados alternadamente para treinamento e teste nos dois diferentes experimentos. Os resultados apresentados nas próximas seções representam as médias dos resultados nos conjuntos de teste desses dois experimentos.

Para o treinamento das SVMs é necessário que todos os atributos de entrada tenham o mesmo peso; deste modo, os dados utilizados foram normalizados.

4.1 Bupa Liver Disorders

Esse banco de dados consiste em informações médicas relacionadas a doenças de fígado com seis atributos numéricos de entrada e uma saída binária. Para avaliar o desempenho de FREx_SVM, três configurações com 3, 5 e 7 conjuntos fuzzy, para cada coordenada, foram consideradas.

A Tabela 1 expõe a média dos desempenhos obtidos nos conjuntos de teste com a melhor configuração de parâmetros dentre todas as SVMs treinadas. As seguintes métricas foram usadas para comparar as diferentes configurações: a porcentagem dos exemplos de teste que foram cobertos pelas regras (Cobertura), a porcentagem de erros de classificação das regras geradas (Erro) e o número de regras geradas (Número de regras) para 3, 5 e 7 conjuntos fuzzy. A melhor cobertura (95,94%) foi obtida para 3 conjuntos fuzzy, com apenas 28 regras.

Tabela 1. Melhores Desempenhos do FREx_SVM e SVM - Bupa Liver Disorders.

	Kernel	Cobertura (%)	Erro (%)	Número de regras
3 conjuntos fuzzy	RBF com $\sigma^2 = 50$ e $C = 0,1$	95,94	47,25	28
5 conjuntos fuzzy	RBF com $\sigma^2 = 10$ e $C = 0,1$	85,51	40,87	74
7 conjuntos fuzzy	RBF com $\sigma^2 = 50$ e $C = 0,1$	80,58	36,52	108,5
SVM	Linear $C = 0,1$	100	38,55	---

As melhores regras obtidas, segundo a acurácia fuzzy, são:

SE x_1 é C_{16} e x_2 é C_{23} e x_3 é C_{31} e x_4 é C_{42} e x_5 é C_{51} e x_6 é C_{61} , ENTÃO Classe 1 — Acurácia: 0,7686, para 7 conjuntos fuzzy.

SE x_1 é C_{14} e x_2 é C_{21} e x_3 é C_{31} e x_4 é C_{41} e x_5 é C_{51} e x_6 é C_{61} , ENTÃO Classe 2 — Acurácia: 0,7673, para 5 conjuntos fuzzy.

4.2 Wisconsin Breast Cancer

Este banco de dados, da mesma forma que o Bupa Liver Disorders, consiste em informações médicas relacionadas ao câncer de mama. Há nove atributos numéricos de entrada e uma saída binária. Nesse caso particular, devido ao grande número de atributos de entrada, o número de conjuntos fuzzy pré-definidos para cada coordenada foi restrito a 3 e 5.

Com no exemplo anterior, a Tabela 2 apresenta as melhores configurações dentre todas as SVMs treinadas. Todos os resultados mostrados são valores médios dos desempenhos obtidos em ambos os conjuntos de teste, usando as mesmas medidas de avaliação: Cobertura, Erro e Número de regras.

Analogamente ao caso anterior, a melhor Cobertura (77,90%) foi obtida para três conjuntos fuzzy, com 131,5 regras. A melhor regra para cada classe é:

SE x_1 é C_{11} e x_2 é C_{21} e x_3 é C_{31} e x_4 é C_{41} e x_5 é C_{51} e x_6 é C_{61} e x_7 é C_{71} e x_8 é C_{81} e x_9 é C_{91} , ENTÃO
Classe 1 — Acurácia: 1, para 3 conjuntos fuzzy.

SE x_1 é C_{12} e x_2 é C_{23} e x_3 é C_{33} e x_4 é C_{43} e x_5 é C_{52} e x_6 é C_{63} e x_7 é C_{73} e x_8 é C_{83} e x_9 é C_{91} , ENTÃO
Classe 2 — Acurácia: 1, para 3 conjuntos fuzzy.

Tabela 2. Melhores Desempenhos do FREx_SVM e SVM - Wisconsin Breast Cancer.

	Kernel	Cobertura (%)	Erro (%)	Número de regras
3 conjuntos fuzzy	RBF com $\sigma^2 = 1$ e $C = 0,1$	77,90	02,49	131,5
5 conjuntos fuzzy	RBF com $\sigma^2 = 1$ e $C = 0,1$	63,98	00,73	157
SVM	Linear $C = 0,1$	100	01,27	---

4.3 Iris

Esse banco de dados é provavelmente o mais conhecido na literatura de reconhecimento de padrões. Ele é relacionado à classificação das espécies da planta íris, que estão divididas em três classes, com quatro atributos numéricos de entrada. Analogamente ao caso do *Bupa Liver Disorders*, três configurações de partição de entrada foram avaliadas: 3, 5 e 7 conjuntos para cada coordenada.

Como esse problema é de múltiplas classes (três), os métodos de classificação em múltiplas classes descritos na seção 2.2 foram aplicados no treinamento das SVMs. A Tabela 3 mostra as configurações que geraram as melhores médias de desempenho de classificação do FREx_SVM nos dois conjuntos de teste, utilizando as mesmas medidas que nos casos anteriores.

Tabela 3. Melhores Desempenhos do FREx_SVM e SVM - Íris

	Método	Kernel	Cobertura (%)	Erro (%)	Nº de regras
3 conjuntos fuzzy	Crammer e Singer	RBF $\sigma^2 = 50$ e $C = 0,1$	100	10,67	20,5
5 conjuntos fuzzy	Separação das classes duas a duas	RBF $\sigma^2 = 5$ e $C = 0,1$	100	6,00	33,5
7 conjuntos fuzzy	Crammer e Singer	RBF $\sigma^2 = 5$ e $C = 0,1$	96,67	4,67	39,5
SVM	Crammer e Singer	RBF $\sigma^2 = 10$ e $C = 10$	100	2,00	---

Como pode ser visto na Tabela 3, o melhor resultado foi obtido para cinco conjuntos fuzzy: cobertura de 100%, com somente 33,5 regras em

média. A melhor regra, para cada classe, em termos da acurácia fuzzy proposta, é:

SE x_1 é C_{11} e x_2 é C_{22} e x_3 é C_{31} e x_4 é C_{41} , ENTÃO
Classe 1 — Acurácia: 0,999, para 3 conj. fuzzy.

SE x_1 é C_{13} e x_2 é C_{22} e x_3 é C_{33} e x_4 é C_{43} , ENTÃO
Classe 2 — Acurácia: 0,8245, para 5 conj. fuzzy

SE x_1 é C_{14} e x_2 é C_{23} e x_3 é C_{34} e x_4 é C_{44} , ENTÃO
Classe 3 — Acurácia: 0,9767, para 5 conj. fuzzy

4.4 Wine

O último banco de dados testado foi o *Wine*, que também é bastante conhecido na literatura de reconhecimento de padrões. Esse banco de dados está relacionado a três tipos de vinhos produzidos em uma região específica da Itália. São 13 atributos numéricos de entrada e, como nos outros casos, uma saída de classificação.

Pela mesma razão que no caso do *Breast Cancer* (grande número de variáveis de entrada), o número de conjuntos fuzzy predefinidos para cada coordenada foi restrito a 3 e 5. Novamente, como é um problema multiclasse, todos os três métodos foram aplicados para o treinamento das SVMs.

A Tabela 4 apresenta os melhores desempenhos de classificação de FREx_SVM nos padrões de teste, usando as mesmas métricas.

Tabela 4. Melhores Desempenhos do FREx_SVM e SVM - Wine

	Método	Kernel	Cobertura (%)	Erro (%)	Nº de regras
3 conjuntos fuzzy	Separação das classes duas a duas	RBF $\sigma^2 = 10$ e $C = 0,1$	92,13	7,87	84
5 conjuntos fuzzy	Separação das classes duas a duas	RBF $\sigma^2 = 5$ e $C = 0,1$	51,69	0	86
SVM	Separação das classes duas a duas	RBF $\sigma^2 = 10$ e $C = 10$	100	1,12	---

Neste caso, o melhor resultado foi obtido com três conjuntos fuzzy: cobertura de 92,13%, com 84 regras. Finalmente, as melhores regras para cada classe, segundo sua acurácia fuzzy, são:

SE x_1 é C_{13} e x_2 é C_{22} e x_3 é C_{32} e x_4 é C_{41} e x_5 é C_{52} e x_6 é C_{63} e x_7 é C_{73} e x_8 é C_{82} e x_9 é C_{93} e x_{10} é C_{102} e x_{11} é C_{113} e x_{12} é C_{122} e x_{13} é C_{133} , ENTÃO
Classe 1 — Acurácia: 1, para 3 conjuntos fuzzy.

SE x_1 é C_{13} e x_2 é C_{22} e x_3 é C_{31} e x_4 é C_{42} e x_5 é C_{52} e x_6 é C_{63} e x_7 é C_{73} e x_8 é C_{82} e x_9 é C_{93} e x_{10} é C_{103} e x_{11} é C_{114} e x_{12} é C_{123} e x_{13} é C_{132} , ENTÃO
Classe 2 — Acurácia: 1, para 5 conjuntos fuzzy.

SE x_1 é C_{14} e x_2 é C_{25} e x_3 é C_{33} e x_4 é C_{43} e x_5 é C_{53} e x_6 é C_{62} e x_7 é C_{72} e x_8 é C_{84} e x_9 é C_{92} e x_{10} é C_{104} e x_{11} é C_{112} e x_{12} é C_{122} e x_{13} é C_{133} , ENTÃO
Classe 3 — Acurácia: 1, para 5 conjuntos fuzzy.

5 Conclusão

Esse trabalho apresentou um novo método para extrair regras fuzzy a partir de uma SVM treinada. Como foi descrito, o algoritmo de extração de regras fuzzy proposto — FREx_SVM — é baseado nos vetores suporte obtidos das SVMs treinadas. FREx_SVM compreende três fases simples, que resultam em um algoritmo de extração de regras rápido.

A principal vantagem do método FREx_SVM é que as regras geradas têm conjuntos fuzzy nos seus antecedentes, o que aumenta interpretabilidade lingüística. Além disso, o método pode ser aplicado a problemas multiclasse, alargando o escopo de aplicações possíveis.

Deve-se destacar que o objetivo maior do FREx_SVM é extrair conhecimento interpretável de uma SVM treinada. Logo, as regras geradas não são realmente usadas para classificar os padrões de entrada, mas para entender como essa classificação foi realizada pelas SVMs. Dessa forma, apesar de a percentagem de erros do FREx_SVM nos quatro banco de dados avaliados (Bupa Liver Disorders, Wisconsin Breast Cancer, Íris e Wine) ser maior do que os valores obtidos pelas SVMs, esses erros não são relevantes em termos de quão bem as regras extraídas ajudam a entender a relação entre o vetor de entrada e a saída de classificação. As métricas mais importantes são a acurácia e a cobertura obtidas por cada regra. Como foi demonstrado pelos estudos de caso, os valores resultantes dessas duas métricas foram bastante satisfatórios, confirmando as qualidades da metodologia proposta.

Como trabalho futuro, o algoritmo proposto incluirá conjuntos fuzzy adaptativos. Isso deve aumentar a acurácia e a cobertura das regras fuzzy, e possivelmente reduzir o número final de regras extraídas, melhorando ainda a interpretabilidade.

Referências Bibliográficas

- Abe, S. and Inoue, T. (2002). Fuzzy Support Vector Machines for Multiclass Problems, *10th European Symposium on Artificial Neural Networks*, Bruges, Belgium, pp. 113-118.
- Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M. and Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines, *Proc. National Academy of Sciences of The United States of America*, Vol. 97, pp. 262-267.
- Chaves, A., Vellasco, M.M.B.R. and Tanscheit, R. (2005). Fuzzy Rule Extraction from Support Vector Machines, *5th Int. Conf. on Hybrid Intelligent Systems (HIS05)*, IEEE Computer Society, Los Alamitos, California, EUA, (ISBN 0-7695-2457-5), pp. 335-340, November 6-9, Rio de Janeiro – RJ, Brazil.
- Crammer, K. and Singer, Y. (2000). On the learnability and design of output codes for multiclass problems, *Computational Learning Theory*, pp. 35-46.
- Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other kernel - based learning methods*. Cambridge University Press.
- Drucker, H. D., Wu, D. H., Vapnik, V. N. (1999). Support vector machines for spam categorization, *IEEE Transactions on Neural Networks* **10**(5): 1048-1054.
- Fu, X., Ong, C. J., Keerthi, S., Hung, G. G. and Goh, L. (2004). Extracting the Knowledge Embedded in Support Vector Machines, *International Joint Conference on Neural Networks*, Budapest, Hungary, pp. 291-296.
- Gunn, S. (1998). Support Vector Machines for Classification and Regression, *Technical Report*, ISIS.
- Haykin, S. (1999). *Neural Networks - A Comprehensive Foundation*. Macmillan College Publishing Company.
- Hsu, C.-W. and Lin, C.-J. (2002). A Comparison on Methods for Multi-class Support Vector Machines, *IEEE Transaction on Neural Networks* **13**(2): 415-425.
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features, *10th European Conference on Machine Learning*, pp.137-142.
- Kressel, U. H.-G. (1999). Pairwise classification and support vector machines, in B. Schölkopf, C. J. C. Burges and A. J. Smola (eds), *Advances in kernel methods: Support vector learning*, MIT Press, pp. 225-268.
- Müller, K. R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V. N. (1997). Predicting time series with support vector machines, *7th International Conference on Artificial Neural Networks*, Lausanne, Switzerland, pp. 999-1004.
- Núñez, H., Angulo, C. and Català, A. (2002). Rule Extraction From support vectors machines, *10th European Symposium on Artificial Neural Networks*, Bruges, Belgium, pp. 107-112.
- Rifkin, R. and Klautau, A. (2004). In Defense of One-Vs-All Classification, *Journal of Machine Learning Research*, vol. 5, pp. 101-141.
- Scholkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. The MIT Press.
- Vapnik, V. N. (1999). An Overview of Statistical Learning Theory, *IEEE Transactions on Neural Networks* **10** (5): 988-999.
- Vapnik, V. N. (1998). *Statistical Learning Theory*, John Wiley & Sons.
- Weston, J. and Watkins, C. (1998). Multi-class Support Vector Machines. *Technical report*, Royal Holloway.