DOI: http://dx.doi.org/10.29069/forscience.2015v3n2.e130

REDES NEURAIS NO TRATAMENTO DE DADOS PARA MAXIMIZAÇÃO DE MARGEM

Emerson Maurício de Almeida Alves[1]
Francisco Heider Willy dos Santos[2]

RESUMO

Este trabalho propõe uma nova técnica para a maximização de margem, através do tratamento de dados. As várias abordagens de tratamento de dados serão utilizadas no treinamento do *Perceptron* Espanhol. As distâncias entre os vetores de suporte e a superfície de separação foram utilizadas como métrica para a avaliação da máxima margem. Os vetores de suporte são fornecidos através da *Support Vector Machine* (SVM), após treinamento com a base de dados sem tratamento. A base utilizada é sintética, linearmente separável, com duas classes, aleatória de distribuição normal. Os resultados sugerem que a nova técnica proporciona uma melhoria na margem gerada com o perceptron espanhol, quando o erro de treinamento é maior.

Palavras-chave: Maximização de margens. Perceptron Espanhol. SVM.

NEURAL NETWORKS IN THE DATA PROCESSING FOR MAXIMIZING MARGINS

ABSTRACT

This paper proposes a new technique for maximizing margin, through the processing of data. The various data processing approaches will be used in Spanish Perceptron training. The distances between the support vectors and the separation surface was used as a metric to evaluate the maximum margin. The support vectors are provided by the Support Vector Machine (SVM) after training database without treatment. The base used is synthetic linearly separable, with two classes, random normal distribution. The results suggest that the new technique provides an improvement in margin generated with Spanish Perceptron, when the training error is higher.

Keywords: Maximizing margins. Spanish Perceptron. SVM.

[1] Mestre em Engenharia Elétrica pela Universidade Federal de Minas Gerais (UFMG) e professor de Eletrônica no Departamento de Engenharia e Computação do Instituto Federal de Minas Gerais (IFMG) - Campus Bambuí. E-mail: emerson.alves@ifmg.edu.br.

[2] Mestre em Engenharia de Sistemas e Automação pela Universidade Federal de Lavras (UFLA) e professor de Automação e Controle no Departamento de Engenharia e Computação do Instituto Federal de Minas Gerais (IFMG) - Campus Bambuí. E-mail: francisco.santos@ifmg.edu.br.

1 INTRODUÇÃO

Maximizar margem melhora a separação do hiperplano; isso proporciona maior acurácia e diminuição da probabilidade do classificador errar. Objetiva-se que classificadores encontrem a máxima margem, uma vez que é esperado que o classificador seja capaz de rotular corretamente os dados que não foram apresentados a ele durante o treinamento. Os parâmetros considerados para maximizar margem são a maximização da acurácia e minimização da norma. Norma menor proporciona uma resposta suave ao classificador (SMOLA *et al*, 2000).

O trabalho propõe uma nova técnica para maximização de margem, baseada no tratamento dos dados de entrada. A técnica será aplicada no treinamento do *perceptron* espanhol (FERNANDEZ-DELGADO, 2011). *Support Vector Machine* (SVM) é utilizada nesse trabalho para o fornecimento não só dos vetores de suporte, mas também do vetor com os pesos da reta separadora. Os pesos gerados pela SVM foram utilizados como referencia nos gráficos, que mostram as retas separadoras. As distancias entre os vetores de suporte e as retas separadoras serão utilizadas como métrica para comparação das superfícies separação.

Perceptron Espanhol é um método proposto por Fernandez-Delgado *et al* (2011). O método utiliza uma abordagem analítica para o treinamento de *perceptrons*, que maximiza a margem e minimiza o erro. Segundo os autores, o algoritmo proposto funciona como uma SVM, em que todos os padrões de treinamento são vetores de suporte (HORTA, 2013). Fundamentado nessa ideia, a SVM foi utilizada como referência de comparação ao *perceptron* espanhol.

O tratamento de dados é realizado pela remoção dos pontos distantes da superfície separadora. A superfície separadora referenciada é gerada através do *bagging* (*bootstrap aggregating*). *Bagging* foi escolhido como método fornecedor da superfície base, ou seja, a superfície que determina quais pontos deverá permanecer e quais serão removidos. *Bagging* é fundamentado na estatística, no teorema central do limite. Pelo teorema, a média das superfícies separadoras de várias repetições, considerando os padrões de treinamento amostrados de forma aleatória, representa o valor mais provável da melhor superfície de separação, ou seja, a margem maximizada (BREIMAN, 1994). *Bagging* utiliza o treinamento do *perceptron* simples para a geração dos pesos da reta separadora.

2 SVM

SVM, ou Máquinas de Vetores de Suporte, é uma técnica para classificação e regressão proposta por *Vapnik* e sua equipe nos laboratórios *AT&T*®. Tem se mostrado mais poderosa que máquinas de aprendizado (VAPNIK; CORTES, 1995).

SVM utiliza *Kernel* de vetores de suporte para mapear os dados de entrada para espaço de característica de alta dimensão. Vetores de suporte são vetores com pontos mais próximos do hiperplano separador, e que realmente definem a qualidade da reta separadora Figura 1.

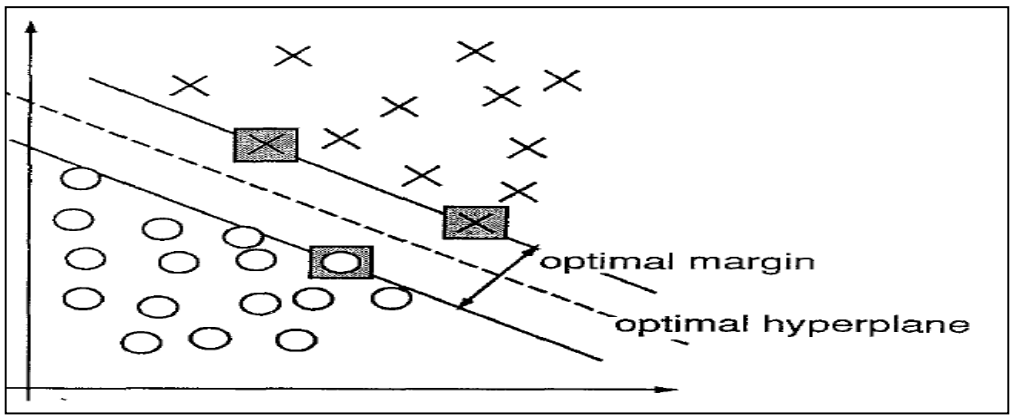

Figura 1 – Distribuição com hiperplano separador, vetores de suporte destacados nos quadrados
Fonte: Vapnik e Cortes (1995).

Kernel é a função que realiza o mapeamento não linear, através do produto escalar entre vetores. O produto é o novo mapeamento do espaço de característica linearmente separável, que maximiza a margem e minimiza o erro.

O aprofundamento sobre *kernel* é disponível em Vapnik e Cortes (1995). Para simplificar a ideia de mapeamento do espaço de entrada para o espaço de característica, podemos considerar, um espaço de entrada não linearmente separável com $X=\{x1,x2\}$, mas se os padrões de entrada forem elevados ao quadrado, com $X^2=\{x1^2,x2^2\}$ se tornam linearmente separáveis. A Figura 2 mostra a transformação dos espaços, realizadas pelo *kernel*, para este caso particular.

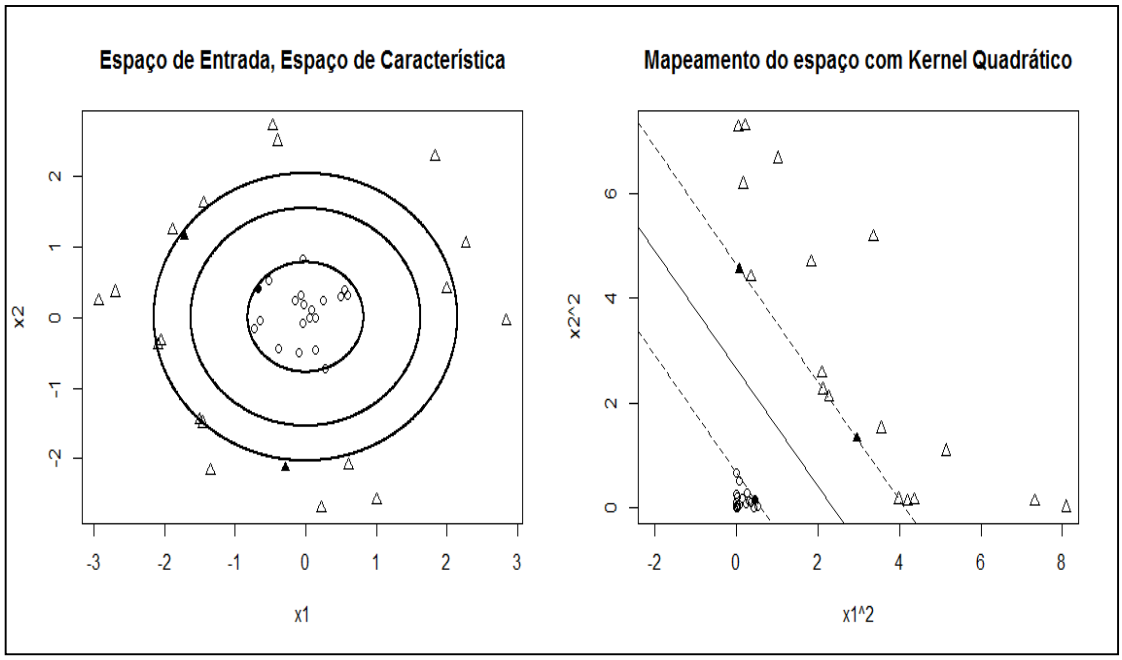

Figura 2 - Demonstração de transformação de espaço de entrada em espaço de característica
Fonte: Autores.

O *kernel* utilizado foi RBF (*Radial Basis Function*) definido pela equação (1), em que k é o *kernel,* x_i e x_j, são duas amostras que representam vetores de características e σ o desvio padrão. A separação demonstrada na Figura 2 utilizou desvio padrão de 0.1.

$$k(x_i, x_j) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right) \quad (1)$$

A utilização do SVM nesse trabalho foi através do pacote *kernelab* (KARATZOGLOU; SMOLA; HORNIK, 2004), integrante do software R, e está descrito no Algoritmo 1.

Algoritmo 1 – SVM

```
#------ SVM ----------------------
library('kernlab')
# kernelk RBF
rbf <- rbfdot(sigma = 0.1)

#Prepara matriz X
sz<-ncol(xAll)
x<-xAll[,2:sz];

#Prepara os rótulos, transforma 0 em -1
y<-ifelse(yAll<=0,-1,1) # ajuste 0 para -1

# treina SVM para capturar vetores de suporte e pesos
svp<- ksvm(x,y,type="C-svc",kernel=rbf,C=10,prob.model=TRUE)

#Rótulos do vetores de suporte
ymat <- ymatrix(svp)

#Vetores de suporte
xsv=x[SVindex(svp),1:sz-1]

#Prepara vetor com -1 para bias
C=nrow(xsv)
Bias=matrix(rep(-1),nrow=c,ncol=1)

#Cria as matrizes com vetores de suporte e rótulos
xsvb=cbind(bias,xsv)
ysv=ymat[SVindex(svp)]

#Calcula o vetor de pesos
wsv <- colSums(coef(svp)[[1]] * x[SVindex(svp),])
b <- b(svp)  # bias

# Vetor de pesos W, para montagem da reta separadora
wsvsw<-(c(b,wsv))
```

3 PERCEPTRON ESPANHOL

O *perceptron* espanhol utiliza todos os padrões disponíveis para o treinamento, não levando em consideração o erro da classificação. Este *perceptron* utiliza a soma dos pesos gerados por cada padrão, cada padrão se comporta como vetor de suporte, desta forma o peso é determinado pela multiplicação do padrão pelo rótulo, que minimiza o erro e maximiza a margem (HORTA, 2013).

No algoritmo proposto, os pesos são ajustados conforme equação (2), que se assemelha ao SVM, em que todos os padrões de treinamento são vetores de suporte com multiplicadores de *Lagrange* iguais a 1 (HORTA, 2013).

$$W0 = \frac{\sum_{k=1}^{N} y_k * x_k}{\left\| \sum_{k=1}^{N} y_k * x_k \right\|} \qquad (2)$$

Os rótulos ou a saída de cada padrão, que deve ser 1 ou -1. O rótulos são y_k , e os vetores x e w são vetores expandidos onde: $x = (1\ x_1\ x_2\ x_n)'$ e $w = (bias\ w_2\ w_n)'$ (1). O Algoritmo 2, contém o *perceptron* espanhol, foi programado no *software* de estatística R.

Algoritmo 2 – Perceptron Espanhol

```
trainPerceptronEspanhol<- function(x,y)
{
  #Extrai quantidades de coluna e linhas
  np=nrow(x)
  nc=ncol(x)

  #Cria a matriz que vai armazenar W
  wTemp<-matrix(c(rep(0)),nrow=np,ncol=nc)
  for(j in 1:np)
  {
  #Calcula W para cada padrão e armazena
  wTemp[j,]= x[j,]*y[j]
  }

     for(i in 1:nc)
      {
        temp=sum(wTemp[,i])
         if(i==1)
        {
          #Inicializa WA
         wa=temp;
        }
          else
        {
        #Aciona as outras colunas em wa
         wa=cbind(wa,temp)
        }
      }
#Norma Euclidiana
  normaE=euclidean.norm(wa)
  w<-wa/normaE

  return(w)
}
```

4 BAGGING

Bagging é um método baseado na inferência estatística (BREIMAN, 1994). A parte dos dados submetida a treinamento tem que ser selecionada de forma aleatória. Várias replicações deste treinamento são realizadas, em cada replicação, é gerado um vetor de peso que definem uma reta separadora. Com o aumento da replicação do treinamento e escolha aleatorizada dos dados, a distribuição amostral se aproxima da distribuição normal (MONTGOMERY; RUNGER, 2008), ou seja, a média de todas as retas expressa o valor mais provável da reta, que maximiza a margem. O método tem se mostrado robusto em situações onde pequenas perturbações geram grandes alterações nos resultados (BREIMAN, 1994).

Leo Breiman utilizou em seu trabalho 25 replicações que geraram resultados satisfatórios (BREIMAN, 1994). A estatística indica que mais de 30 replicações não há uma contribuição significativa nos resultados, considerando o atendimento às premissas que definem ao enunciado do teorema central do limite (MONTGOMERY; RUNGER, 2008). Nesse trabalho o *bagging* é replicado 40 vezes.

O treinamento do *perceptron* simples pode levar a 100% de acurácia, ou seja, a geração de uma reta que proporciona separação total das classes Figura 3 (a). Várias retas podem separar completamente as classes com 100% de acurácia, Figura 3 (b) na cor verde, entretanto existe uma reta que melhor divide a superfície, ou seja, fica no centro das duas classes. A reta de cor preta Figura 3 (b), é a reta que maximiza a margem, sendo mais robusta a ruídos.

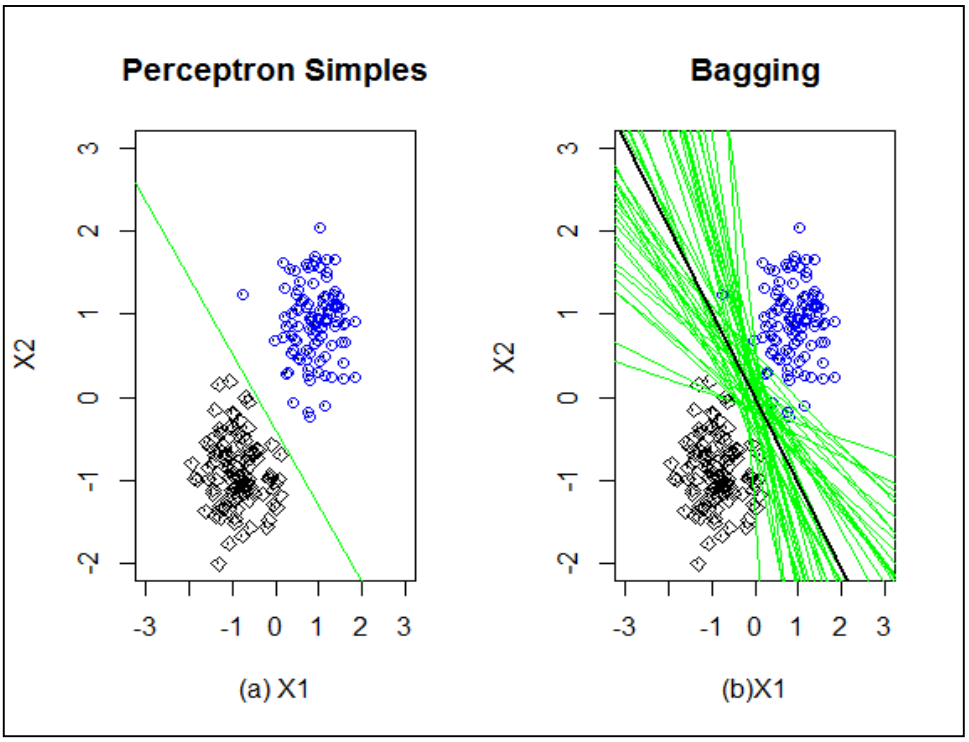

Figura 3 - Comparação entre retas classificadoras para o perceptron simples e *bagging*
Fonte: Autores.

5 MÉTODO

O novo método, com tratamento de dados, consiste em filtrar os dados de entrada de forma que somente os padrões mais relevantes sejam apresentados para o treinamento.

A primeira etapa do método consiste em gerar uma superfície de separação, ou seja, uma reta que será utilizada como referência para determinação das distâncias dos pontos ou padrões. Essa reta é gerada com todos os padrões, utilizando o *bagging*. Nessa etapa, o *perceptron* espanhol é treinado com os dados sem tratamento.

A segunda etapa consiste em realizar o treinamento do *perceptron* espanhol com os dados selecionados, denominados dados tratados. A seleção é realizada após determinar a distância entre os pontos do padrão e a reta gerada pelo *bagging* na primeira etapa. A partir da distância dos pontos, é calculada a distância média de cada classe. Os pontos com distância menor que 70% da distância média são selecionados. Nessa etapa, o *bagging* é aplicado aos dados selecionados para gerar outra reta separadora. Esse método elimina os pontos mais distantes da superfície separadora, portanto ficando os padrões de maior relevância para definir uma reta separadora. A maximização da margem é conseguida eliminando os pontos mais distantes.

Como medida da distância do padrão ao classificador, foi utilizada a distância calculada através da equação (3), onde a distância *d* é resultado da multiplicação do padrão *x* pela matriz transposta do peso *w,* desta forma é determinada a distância para cada padrão.

$$d = x * w^t \tag{3}$$

Na terceira etapa ocorre nova eliminação de pontos distantes, ou seja, uma nova seleção de pontos próximos. A seleção é realizada dos pontos já selecionados na segunda etapa, ou seja, ocorre uma seleção em cadeia. O critério de seleção continua em manter os padrões que possuem a distância menor que 70% da distância média dos padrões analisados, esses denominados dados retratados. Então o *perceptron* espanhol é treinado com os dados desta nova seleção.

6 EXPERIMENTO

Foi utilizada no experimento uma base aleatória com distribuição normal, com duas classes, sendo 100 padrões e duas variáveis. Esta base é simétrica e linearmente separável. Os dados foram normalizados, os parâmetros abaixo mantiveram constantes em todos os experimentos:

1. Número máximo de iterações: 100
2. Taxa de aprendizado: 0.1
3. Número de repetições para *bagging*: 40

A taxa de aprendizado é utilizada para a atualização dos pesos do perceptron simples, utilizado pelo bagging. O parâmetro erro de treinamento, ou seja, erro admitido no treinamento do perceptron simples é alterado. Nesse trabalho foram realizados experimentos com erros de: 0,01 e 0,1.

Para cada parâmetro de erro foram realizados quatro experimentos. Os experimentos consistiram em usar a base completa (A), base com dados selecionados (B), base com dados selecionados a partir da última seleção (C) e dados selecionados pela SVM (D). Os dados selecionados foram denominados de vetores de suporte. Em cada experimento foram aplicados os métodos *bagging*, *perceptron* espanhol e SVM, Figura 4.

A Figura 4 mostra os quatro gráficos para o parâmetro de erro de 0,01. Os pontos acima das retas na cor azul pertencem a classe 1 e os pontos abaixo das retas na cor verde pertencem a classe 2. As retas contínuas na cor verde indicam as retas separadoras determinada em cada execução do *bagging* e a reta tracejada na cor vermelha é a reta resultante do *bagging*. A reta do *perceptron* espanhol é indicada pela linha pontilhada azul e a reta da SVM pela linha contínua na cor preta.

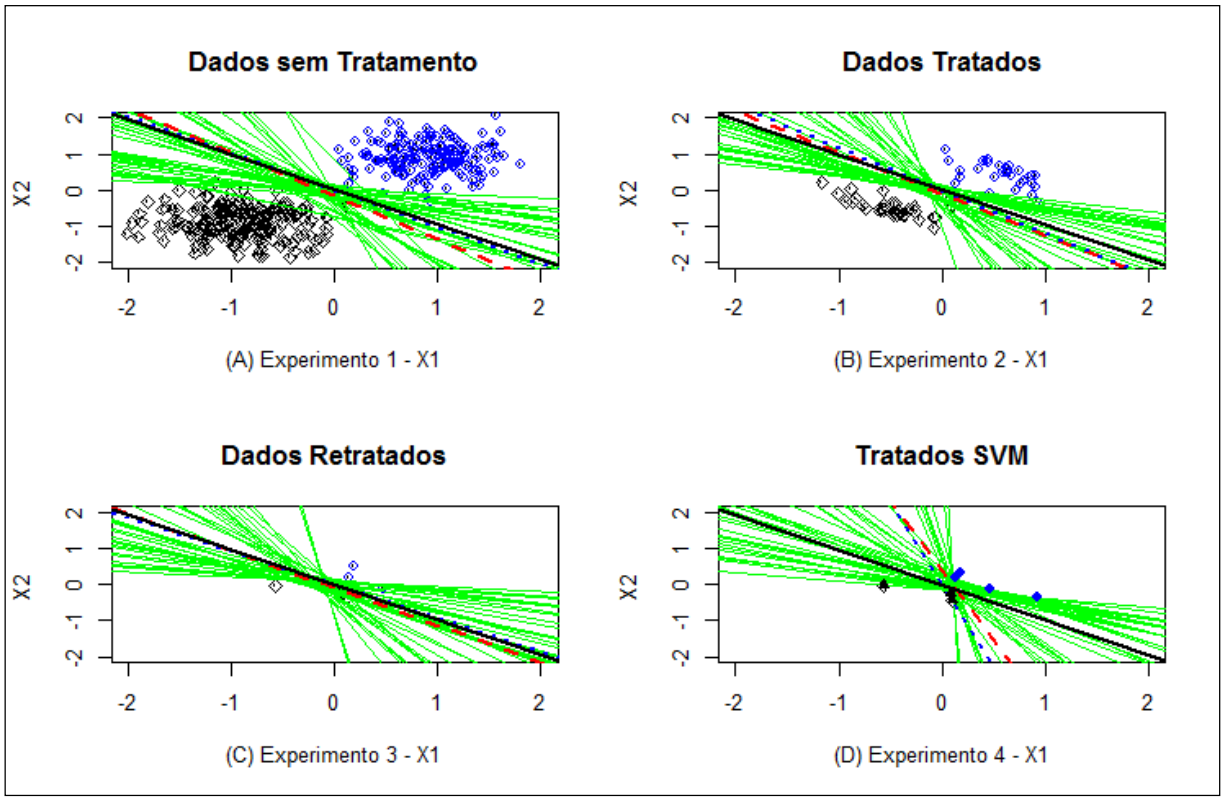

Figura 4 - Comparação dos métodos com parâmetro de erro de treinamento de 0,01
Fonte: Autores.

A Figura 5 mostra os quatro gráficos para o parâmetro de erro de 0,1. As considerações sobre os pontos e as retas são as mesmas realizadas para a Figura 4.

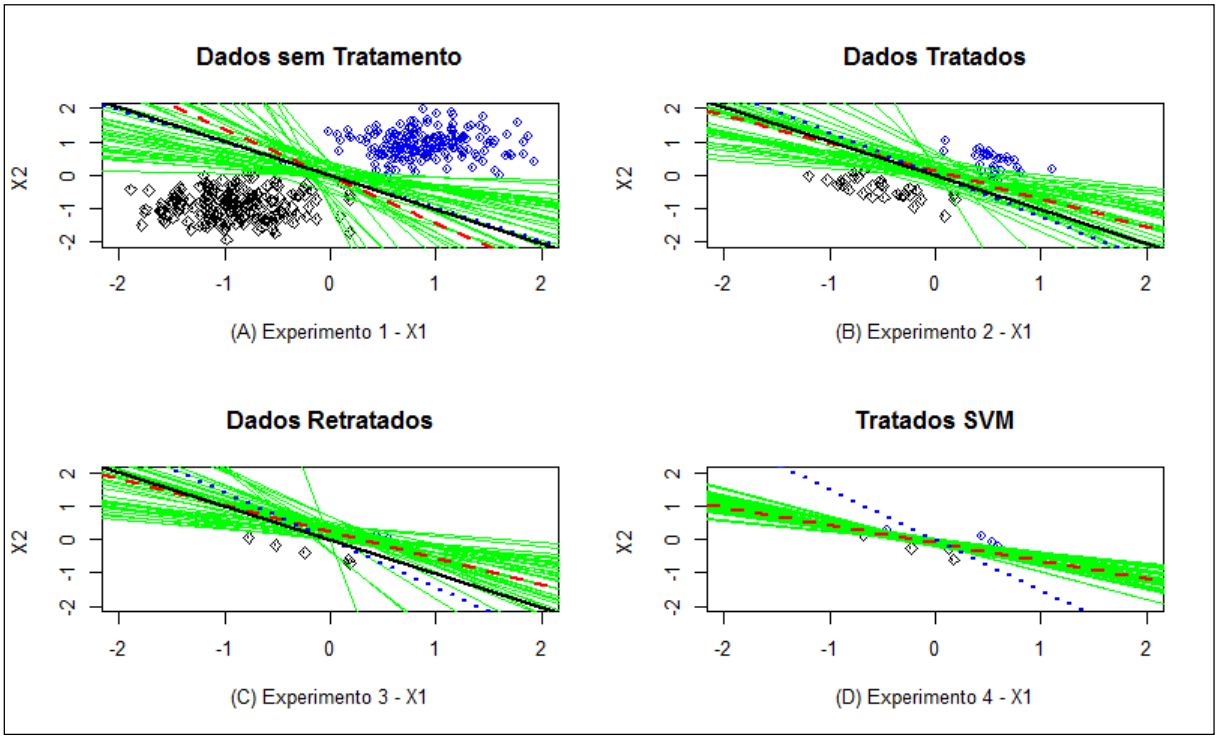

Figura 5 - Comparação dos métodos com parâmetro de erro de treinamento de 0,1
Fonte: Autores.

7 AVALIAÇÃO

Uma avaliação visual pode ser realizada verificando o comportamento das retas separadoras geradas pelo *perceptron* espanhol nos quatro experimentos. Na Figura 4, onde o parâmetro do erro de treinamento foi de 0,01 a reta do *perceptron* espanhol ficou muito próxima reta da SVM em todos os experimentos, exceto onde os dados foram selecionados pela SVM. O mesmo comportamento não ocorreu com os experimentos com parâmetro de erro de 0,1, Figura 5. Ainda na Figura 5, observa-se que a reta do *perceptron* espanhol é próxima a reta do *bagging* e afasta um pouco da reta da SVM.

Outra avaliação dos resultados do método proposto para maximização de margem com o *perceptron* espanhol foi realizada através comparação das distâncias entre os vetores de suporte e a reta separadora. As distâncias estão descritas em gráficos de *boxplot,* Figura 6. O gráfico de *boxplot* mostra que com o aumento do erro de 0,01 para 0,1, ocorre uma diminuição da distância. A diminuição da distância é perceptível no gráfico da classe 1 com erro de 0,1, Outra observação que os *boxplot* dos experimentos com parâmetro de erro de 0,01 apresentaram maior variabilidade. Pode verificar uma pequena melhoria da distância dos experimentos 2 em relação ao experimento 1, entretanto estas distancias são muito

semelhantes.

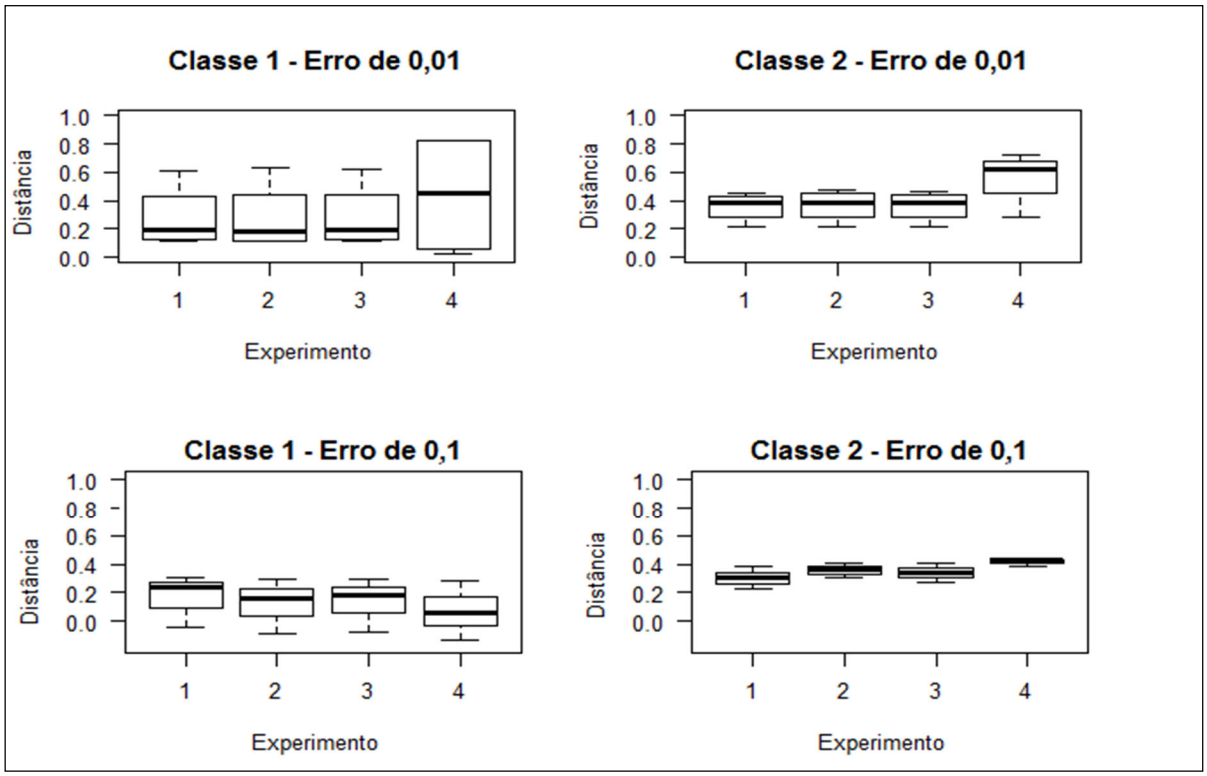

Figura 6 - Grafico com boxplot das distância entre os pontos e a reta separadora.
Fonte: Autores.

A análise dos *boxplot* sugere que com o aumento do erro o novo método proporciona melhor distância com menor variabilidade.

8 CONCLUSÃO

Com a metodologia aplicada, o experimento sugere que o novo método proporciona uma melhoria na margem gerada com o *perceptron* espanhol, quando o erro de treinamento é maior. Este comportamento é interessante, pois abre a possibilidade da aplicação prática deste método. As etapas envolvidas, no método com tratamento de dados, têm um custo computacional alto, devido ao processamento do *bagging*, processo de cálculo de distância e seleção dos padrões. À medida que o erro de treinamento aumenta, diminui o tempo de processamento do *bagging*, consequentemente podemos diminuir o custo computacional do novo método. A métrica utilizada no trabalho, ou seja, a distância entre os pontos dos padrões e o hiperplano separador foi satisfatória, uma vez que todos os métodos proporcionaram 100% de acuraria.

Podemos perceber nesse trabalho que o *perceptron* espanhol teve seu desempenho comprometido quando houve um excesso de remoção dos padrões para treinamento. Desta forma, deve se avaliar qual é o limite para tratamento dos dados, em métodos que se baseiam na remoção dos dados menos significativos. O melhor desempenho do *perceptron* espanhol, com o tratamento dos dados, foi após a segunda seleção dos dados e erro maior de treinamento. Esse desempenho foi superior ao da SVM, mesmo a reta do *perceptron* espanhol estando um pouco distante da reta de referência da SVM.

Considerando que foi utilizada somente uma base sintética, as conclusões sobre esse novo método ficam restrita a esse trabalho e deve ser analisada de forma conservadora. A sugestão é que este método seja submetido a uma análise com utilização de outras bases, bases reais, conhecidas e citadas pela literatura, assim pode se ter uma conclusão com um poder maior de generalização.

REFERÊNCIAS

BREIMAN, L. Bagging predictors. **Technical Report**, Berkeley, n. 421, Sept. 1994.

FERNANDEZ-DELGADO, M. et al. Direct parallel perceptrons (DPPs): fast analytical calculation of the parallel perceptrons weights with margin control for classification tasks. **IEEE transactions on neural networks**, v. 22, n. 11, p. 1837-1838, Nov. 2011.

HORTA, E. G. **Aplicação de máquinas de aprendizado extremo ao problema de aprendizado ativo**. Belo Horizonte: Centro de Desenvolvimento e Planejamento Regional, Universidade Federal de Minas Gerais, 2013.

KARATZOGLOU, A.; SMOLA, A.; HORNIK, Kurt. Kernlab: an S4 Package for Kernel Methods in R. **Journal of Statistical Software**. v. 11, Apr. 2004.

MONTGOMERY, D. C.; RUNGER, G. C. **Estatística aplicada e probabilidade para engenheiros**. 8. ed. Rio de Janeiro: LTC, 2008.

SMOLA, A. et al. **Advances in Large Margin Classifiers**. Cambridge, Massachusetts: MIT, 2000. 422 p.

VAPNIK V; CORTES C. **Support-Vector Networks**. Machine Learning, v. 20, n. 3, p. 273-297, Sep. 1995.

Recebido em: 17/04/2015

Aprovado em: 17/08/2015

Publicado em: 15/01/2016