Um algoritmo de otimização *online* para a solução do problema SVM baseado na topologia de uma rede *Perceptron* dual

Raul Fonseca Neto, Samuel Belini Defilippo
Departamento de Ciência da Computação – UFJF
Coordenação de Matemática Aplicada e Computacional – LNCC
raulfonsecaneto@ig.com.br, samueldefilippo@yahoo.com.br

Resumo

Este trabalho consiste no desenvolvimento e implementação de um algoritmo para a solução do problema relacionado ao treinamento de um classificador denominado máquina de vetores suportes (SVM). Trata-se da solução de um problema dual de otimização quadrática colocado sob a forma de Wolfe. O método apresentado tem como embasamento teórico o desenvolvimento de uma rede Perceptron dual cuja topologia está relacionada a uma representação dependente dos dados que computa uma função de saída responsável pela maximização da margem de separação dos dados em um problema de classificação binária. O algoritmo, denominado KPDS, Kernel Perceptron Dual SVM, utiliza uma forma mais estável, porém, com uma menor taxa de convergência, de correção dos multiplicadores, com base no gradiente da função lagrangeana.

1. Introdução:

O desenvolvimento de classificadores *kernel*, Smola e Scholkopf [1] e, em particular, das máquinas de vetores suportes, Cortes e Vapnik [2], representam um grande avanço, e, até mesmo, uma mudança de perspectiva, no campo da aprendizagem de máquinas, no contexto, sobretudo, do aprendizado supervisionado.

Vários problemas de classificação, como exemplo, o reconhecimento de padrões complexos de estrutura não rígida ou de formações variáveis como a escrita, fala e objetos deformáveis, envolvendo os processos de categorização e generalização do ser humano, passaram a ser resolvidos com maior eficiência, atingindo níveis de performance bem acentuados, quase comparados aos processos decisórios de especialistas.

A aplicação destes algoritmos de aprendizagem, não se resumem, entretanto, a simulação dos processos de percepção. Um grande número de problemas reais pode ser solucionado com a utilização destas técnicas.

Neste particular, podemos destacar os problemas nas áreas de Medicina, Biologia, Engenharia, Economia e Computação, incluindo a tomada de decisão em diagnósticos médicos; a análise de dados provenientes de experimentos de *microarray* e a análise de seqüências biológicas; o controle de processos em plantas industrias, o reconhecimento de imagens e o processamento de sinais; a mineração de dados relacionados a comercialização, estoques e lucratividade de empresas, e, finalmente, a mineração de textos na *web*.

Inicialmente, vamos abordar algumas das principais técnicas utilizadas para o treinamento de máquinas de vetores suportes, ou seja, aplicadas a solução do problema de otimização quadrática na forma dual de Wolfe, conhecido como problema SVM, Cortes e Vapnik [2], e descrito como:

$$\begin{aligned} &Max\ \Lambda.\mathbf{1} - 1/2\,.\Lambda^T.H.\Lambda \\ &Sujeito\ a: \\ &\Lambda^T.Y = 0 \\ &0 \le \Lambda \le C \end{aligned} \qquad (1)$$

considerando o vetor $\Lambda = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ o vetor de multiplicadores, $Y = (y_1, y_2 \ldots, y_m)$, o vetor de rótulos de valores binários e H a matriz Hessiana simétrica, positiva semi-definida com todos autovalores não negativos, na forma:

$H = [h_{i,j}]$, onde $h_{i,j} = y_i.y_j.K_{i,j}$,
estando associada a um conjunto de treinamento $Z = \{(x_i, y_i)\}$ e a uma função *kernel* $k : \Re^d \times \Re^d \rightarrow \Re$, sendo $K_{i,j} = k(x_i,x_j)$.

Se a matriz H for positiva definida, a função objetiva do problema tem a forma estritamente convexa e a sua solução ótima global relativa a um ponto de máximo que satisfaça as condições de KKT é única, podendo ser obtida, segundo Fletcher [3], por algum método de otimização quadrática convexa.

Entretanto, mesmo se a matriz Hessiana for positiva semi-definida a solução obtida pode ser global e única. No caso mais geral, a solução não será única se dado alguma solução Λ, escolhemos um vetor Λ' que pertence ao espaço nulo da Hessiana, sendo o vetor Λ' ortogonal ao vetor unitário, derivando uma solução $\Lambda + \Lambda'$ também ótima. Porém, a solução encontrada será sempre uma solução ótima global, em contraste com as técnicas de

Redes Neurais Artificiais, que empregam o algoritmo *backpropagation*, onde muitas soluções de máximos locais poderão existir.

2. Métodos de solução:

A elaboração de um método para a solução do problema de otimização quadrática relacionado ao treinamento de uma máquina SVM depende, essencialmente, de três fatores. Em primeiro lugar, como devemos considerar as informações utilizadas da função objetiva do problema, ou seja, informações de primeira ordem ou informações de segunda ordem. Em segundo lugar, se o método de treinamento, relacionado a técnica de otimização, deve ser implementado de forma o*nline*, ou através de soluções *batch*. Finalmente, em terceiro lugar, se a solução do problema será feita no espaço de variáveis primais ou duais.

A dificuldade maior na solução do problema de programação quadrática, associado a este treinamento, está no tamanho da matriz Hessiana, que é quadrática em relação ao tamanho do conjunto de treinamento e extremamente densa, não permitindo a utilização de técnicas eficientes de fatoração no cômputo de sua inversa.

Entretanto, devemos considerar que na solução ótima do problema, somente alguns pontos ou vetores, chamados de vetores suportes, participam do conjunto ativo, possibilitando o emprego de técnicas de otimização baseados na redução de variáveis e em métodos de decomposição que utilizam um sub-conjunto de trabalho a cada iteração.

Atualmente, existem várias técnicas e métodos de otimização, aplicados ao treinamento de uma máquina SVM. Estes métodos, de certa forma, utilizam-se de uma análise apropriada dos fatores mencionados, desenvolvendo, assim, determinadas estratégias específicas.

O primeiro, tradicionalmente mais utilizado, está relacionado ao uso de estratégias de conjunto de trabalho, sendo considerado um método de decomposição. Utiliza um *solver* de otimização não linear, que retém a cada iteração um sub-conjunto de variáveis, associados a pedaços do conjunto de treinamento, para a formação da matriz Hessiana. Esta técnica foi empregada por Osuna, Freund e Girosi [4] no problema de reconhecimento de imagens deformáveis ou faces.

O segundo, desenvolvido recentemente por John C. Platt [5], pode ser considerado como uma técnica de decomposição na sua forma mais extrema. Neste caso, os sub-problemas de programação quadrática envolvem a cada iteração, somente dois multiplicadores, podendo ser solucionados de forma analítica, sem a necessidade de um *solver* de otimização não linear. Recebeu o nome de Otimização Mínima Seqüencial (SMO).

O terceiro, desenvolvido por Friess, Cristianini e Campbell [6], é uma técnica simples, mas bastante eficiente conhecida como *Kernel Adatron*. Foi desenvolvido com a introdução de funções *Kernel* no algoritmo *Adatron*, proposto por Anlauf e Biehl [7]. Consiste de um método de otimização *online* que utiliza somente informações de primeira ordem da função objetiva do problema. A sua formulação clássica, descrita por Campbell e Cristianini [8], utiliza a base do algoritmo *Adatron* realizando, entretanto, uma análise mais detalhada na avaliação do bias da equação do hiperplano.

3. Problema SVM com *soft margin*:

Na utilização de um hiperplano ótimo como separador em um problema de classificação binária podem ocorrer problemas onde o conjunto de pontos não seja perfeitamente linearmente separável. Neste caso, pode existir alguma forma de *overlap* entre as classes, provocando uma violação das restrições de classificação do sistema.

A forma mais usual de corrigir este problema consiste na introdução de variáveis de folga não negativas, ou de relaxação das restrições, segundo Cortes e Vapnik [2], que permitirão que o conjunto de treinamento seja separado linearmente com um número mínimo de erros relacionado ao controle da capacidade do classificador.

3.1 Flexibilização da margem:

Seja a introdução das variáveis de relaxação e_i, *i=1,...,m*, onde $e_i \geq 0$. Devemos minimizar a função de erro: $\sum_i \epsilon_i^s$, sujeito às restrições de classificação na forma relaxada:

$$wx_i + b \geq 1 - e_i \ para \ y_i = +1$$
$$wx_i + b \leq e_i - 1 \ para \ y_i = -1,$$

permitindo, desta forma, segundo a Figura 1, que alguns pontos ultrapassem a barreira dos hiperplanos segundo os valores de e_i.

Claramente, os valores de e_i estão associados aos erros de treinamento. Se o valor de ϵ_i estiver contido no intervalo, $0 < e_i < 1$, os vetores associados ultrapassam a margem de segurança, sem, no entanto, ocorrerem erros de classificação. Por outro lado, se $e_i > 1$, então o vetor associado x_i está sendo classificado na classe contrária, ocorrendo um erro de classificação.

Se os vetores associados aos erros de classificação puderem ser separados do restante do conjunto de treinamento, então os dados remanescentes poderão ser treinados de forma a definir um hiperplano separador ótimo. Este problema pode ser tratado de maneira formal através da minimização de um funcional Φ que inclui uma medida de capacidade do classificador, associado a maximização da margem, e uma medida de penalidade dos erros de treinamento.

Considerando a formulação primal do problema SVM, temos:

Min ½ w^t . w + C. Φ ($\sum_i \epsilon_i^{\sigma}$) *(2)*

Sujeito a:

$wx_i + b \geq 1 - e_i$ *para* $y_i = +1$

$wx_i + b \leq e_i - 1$ *para* $y_i = -1$,

$e_i \geq 0$,

onde o parâmetro C é uma constante positiva que controla a penalidade do erro e o vetor w se refere ao vetor normal do hiperplano separador.

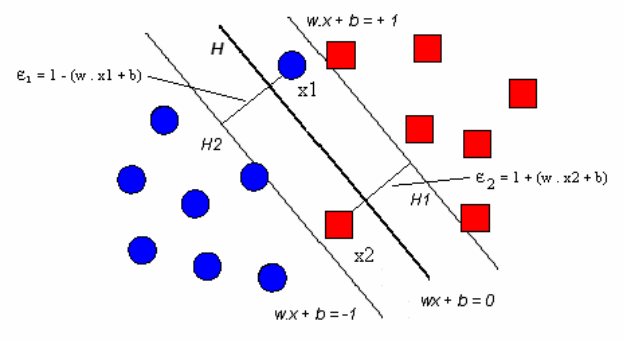

Figura 1: SVM com *soft margin.*

3.2 Formulação SVM:

Tomando $\Phi(u) = u$ e $\sigma = 1$, temos, segundo Cortes e Vapnik [2], a primeira forma de solução do classificador com *soft margin* relacionada a minimização da norma linear ou normal L_1:

$$Min\ \frac{1}{2} w^T .w + C.\sum_i e_i \qquad (3)$$

Sujeito a:

$yi \cdot (w \cdot x_i + b) + e_i - 1 \geq 0$, *para* $i = 1,...,m$

$e_i \geq 0$

Introduzindo os multiplicadores a e m , temos:

$$Min \frac{1}{2} w^t w + C\sum e_i - \sum a_i y_i (wx_i + b) + \sum a_i - \sum a_i e_i - \sum m_i e_i$$

ou

$$Min \frac{1}{2} w^t w - \sum a_i y_i (wx_i + b) + \sum a_i + \sum e_i (C - a_i - m_i)$$

$a_i, m_i \geq 0$

Fazendo uso do gradiente da função lagrangeana em relação a ***w, b e ϵ***, e igualando a zero de modo a satisfazer as condições de otimalidade de primeira ordem, temos:

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w - \sum a_i y_i x_i = 0$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow a_i y_i = 0$$

$$\frac{\partial L}{\partial e_i} = 0 \Rightarrow C - a_i - m_i = 0$$

Substituindo o valor de ***w*** na expressão da função lagrangeana e introduzindo as demais equações como restrições do problema, anulamos a dependência da função objetiva em relação aos parâmetros ***w, b e ϵ***, estabelecendo o problema na forma dual de Wolfe, ou seja:

$$Max\ L(a) = \sum a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j \langle x_i, x_j \rangle \qquad (4)$$

Sujeito a:

$\sum_i y_i a_i = 0$

$C - a_i - m_i = 0$

$m_i >= 0$, $a_i >= 0$

Considerando $m_i = C - a_i$ e $m_i \geq 0$, temos $\alpha_i \leq C$, possibilitando reescrever o problema SVM na forma quadrática conforme apresentado em (1):

$$Max\ L(a) = \sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j \langle x_i, x_j \rangle \qquad (5)$$

Sujeito a:

$\sum_i y_i a_i = 0$

$0 \leq a_i \leq C$

A viabilidade da solução do problema primal e dual, associada às equações de complementaridade, estabelecem as seguintes condições de otimalidade, conhecidas como condições de KKT (Karush-Kuhn-Tucker):

$y_i \cdot (w \cdot x_i + b) + e_i - 1 \geq 0$

$e_i, a_i, m_i \geq 0$

$a_i (y_i (wx_i + b) + e_i - 1) = 0$

$m_i e_i = 0$

Podendo ser feita a seguinte análise de viabilidade considerando a flexibilização na classificação dos dados:

1º caso: O vetor esta fora das margens.

Se $a_i = 0 \Rightarrow m_i = C \Rightarrow e_i = 0$

derivando: $y_i (wx_i + b) - 1 \geq 0$

2º caso: O vetor ultrapassa as margens.

Se $a_i = C$ => $\mu_i = 0$ => $\epsilon_i >= 0$

derivando: $y_i (wx_i + b) - 1 \leq 0$.

3º caso: O vetor está sobre a margem.

Se $0 < a_i < C$ => $\mu_i > 0$ => $\epsilon_i = 0$

derivando: $y_i (wx_i + b) = 1$

3.3 Funções *kernel*:

Quando o conjunto de dados não é linearmente separável aplicamos uma transformação não-linear dos dados, do espaço de entrada original, para um espaço de mais alta dimensão, denominado espaço de características. Esta transformação pode ser obtida através do uso de várias funções de mapeamento. Após esta transformação, os dados são separados de forma linear no espaço de características através da construção de um hiperplano separador por um tipo de classificador como as máquinas SVM, conforme o exemplo da Figura 2, que mostra uma projeção de pontos de um espaço de entrada R^2 para um espaço de características R^3.

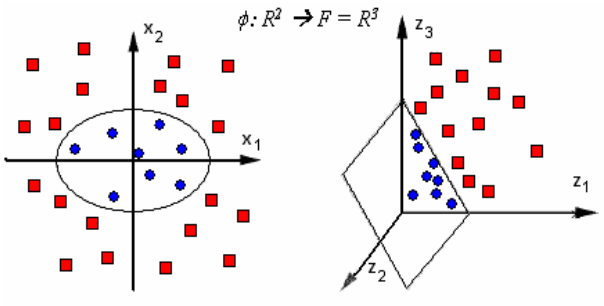

Figura 2: Espaço de entrada e de características.

De forma simplificada, podemos resumir a utilização de funções *kernel* apresentando o seguinte problema: Seja um conjunto de pontos ou dados pertencentes ao espaço euclidiano R^D, definido como espaço de entrada, seja F um espaço de mais alta dimensão definido como espaço de características. Definindo uma função de mapeamento $f:R^D \rightarrow F$, $x \rightarrow f(x)$, podemos estabelecer uma função *kernel* k, $k(x,x_i) = <f(x).f(x_i)>$, na forma de um produto interno do mapeamento de dois vetores associados a função característica ϕ, sendo x e $x_i \in R^D$, considerando que a função k atende as condições estabelecidas por Mercer [9].

É importante ressaltar que o algoritmo de treinamento de um classificador *kernel* como uma máquina de vetores suportes, depende, somente, do produto interno dos vetores no espaço de entrada, seguido da avaliação da função *kernel* k, a fim de determinar uma superfície de decisão linear ou hiperplano separador no espaço de características.

Neste sentido, ao projetarmos os pontos no espaço de características, através do mapeamento obtido pela função f, necessitamos definir, somente, a função *kernel*, não precisnado avaliar a função f explicitamente e nem mesmo conhecê-la. Utilizando a função k no algoritmo de treinamento obtemos uma superfície de decisão linear no espaço de características F, a qual corresponde a uma superfície de decisão não linear no espaço de entrada.

4. Algoritmo KPDS:

Neste trabalho desenvolvemos e implementamos um método *online* para a solução do problema SVM. O algoritmo tem como objetivo a determinação do hiperplano separador ótimo no espaço dual das variáveis, estabelecendo um *Perceptron* de larga margem.

O processo de aprendizagem é baseado no cômputo do gradiente da função lagrangeana, apresentada na formulação (5), em relação a cada multiplicador, ou seja, determinamos $\partial L/\partial a_i$, associado a uma taxa de aprendizagem η.

Consideramos em nossa implementação a utilização de funções *kernel*, a adoção do parâmetro C como um parâmetro de regularização e de flexibilização da margem e, por fim, a manutenção do conjunto de multiplicadores diferentes de zero, formando o conjunto SV, que acelera o cômputo da função f discriminante.

O *loop* principal do algoritmo, como veremos, testa todo os exemplos do conjunto de treinamento, a exemplo do algoritmo *Perceptron*, determinando uma época ou etapa no processo de aprendizagem.

4.1 Topologia de rede:

A principal diferença em relação ao algoritmo *Kernel Adatron* está na forma de atualização dos multiplicadores e de avaliação do bias da equação.

Os métodos *online*, em sua maioria, atualizam todos os multiplicadores a cada época, ou seja, a cada passagem do conjunto de treinamento, seguindo a forma de correção do algoritmo *Adatron*.

Em nossa implementação, optamos pela correção de um único multiplicador a cada época, em uma forma de aprendizado caracterizada pela correção do vetor ao padrão mais informativo, associada a escolha de um vencedor, como ocorre no processo de aprendizado competitivo. Escolhemos para atualizar sempre aquele multiplicador associado ao maior valor da correção. Ou seja, atualizamos α_k sendo $k = Arg\ Max_i \{\partial L/\partial a_i\}$, mantendo inalterados os valores dos demais multiplicadores. Este método segue a forma de correção existente no algoritmo *MinOver*, Kinzel [10], que apesar de apresentar uma menor taxa de convergência, em relação ao algoritmo *Adatron*, produz resultados mais estáveis.

Como funções *kernel* implementamos a função produto interno, para a solução de problemas linearmente separáveis, a função de Gauss para a solução de problemas no espaço de características e algumas funções relacionadas ao computo de similaridades entre seqüências biológicas para a solução de problemas na área de Bio-Informática. A matriz *kernel*, caso haja memória suficiente, é computada na fase inicial do algoritmo, no sentido de tornar mais eficiente a avaliação da função f.

A estrutura do processamento pode ser representada pela topologia de rede do *Perceptron* Dual, descrita na Figura 3, fornecendo a seguinte função discriminante:

$$f(x_i) = \sum_{j=1}^{m} a_j y_j k(x_i, x_j) + l \,. \qquad (6)$$

De outra forma, esta equação pode ser reescrita considerando os valores de a_i não associados a vetores suporte iguais a zero, resultando em:

$$f(x_i) = \sum_{j \in SV} a_j y_j k(x_i, x_j) + l \qquad (7)$$

$$g(x_i) = y_i f(x_i),$$

determinando a seguinte função de decisão:

$z(x_i) = j\,(g(x_i))$, para uma função sinal j .

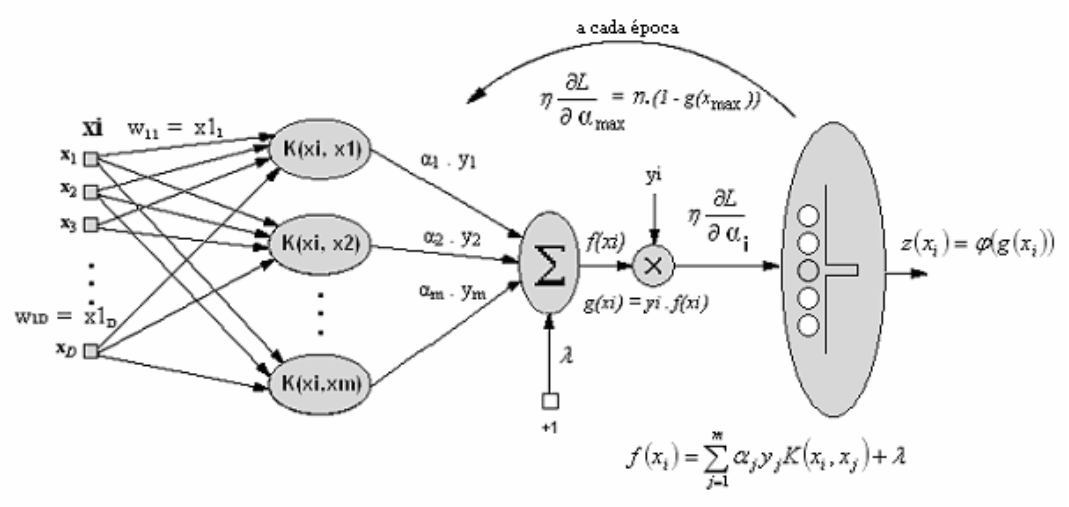

Figura 3: Topologia de rede do algoritmo KPDS.

4.2 Base teórica e formulação:

Para um problema SVM que considera a relaxação de restrição de igualdade, na forma:

$$Max\ L(a,l) = \sum_{i=1}^{m} a_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} a_i a_j y_i y_j K(x_i, x_j) - l\sum_{i=1}^{m} y_i a_i$$

$0 \le a_i \le C$, l irrestrito *(8)*

computamos a derivada parcial em relação a cada multiplicador α_i:

$$\frac{\partial L}{\partial a_i} = 1 - y_i \sum_{j=1}^{m} a_j y_j K(x_i, x_j) - l.y_i .$$

Tomando o valor máximo:

$$\frac{\partial L}{\partial a_k} = 1 - y_k f(x_k), \text{ sendo } k = Arg\ Max_i \left\{\frac{\partial L}{\partial a_i}\right\},$$

atualizamos o multiplicador vencedor α_k com base na expressão:

$$\Delta a_k = h[1 - g(x_k)],$$

sendo o novo valor de α_k dado por:

$$a_k^t = 0 \text{ se } a_k^{t-1} + \Delta a_k \le 0$$

$$a_k^t = a_k^{t-1} + \Delta a_k \text{ se } 0 < a_k^{t-1} + \Delta a_k < C$$

$$a_k^t = C \text{ se } a_k^{t-1} + \Delta a_k \ge C$$

Na atualização do valor do bias, utilizamos um processo iterativo de ajustamento com base no gradiente da função lagrangeana *L* da equação (8) em relação ao parâmetro λ, ou seja, considerando:

$$\frac{\partial L}{\partial l} = \sum_{i=1}^{m} y_i a_i ,$$

atualizamos o bias λ, ao final de cada época, em função do único multiplicador modificado, segundo a equação:

$$l^t = l^{t-1} + y_k \Delta a_k \text{ se } a_k^{t-1} + \Delta a_k > 0 , \qquad (9)$$

$$l^t = l^{t-1} - y_k \Delta a_k \text{ se } a_k^{t-1} + \Delta a_k \le 0 .$$

O critério de parada é definido pelo cálculo da semi-margem, cujo valor deve convergir para o valor unitário, ou seja, ao final de cada época computamos:

$$g = \frac{1}{2}\left(Min\left(f^+(x_i)\right) - Max\left(f^-(x_i)\right)\right),$$

comparando seu valor ao intervalo:

$1 - \epsilon < \gamma < 1 + \epsilon$, para uma constante de erro ϵ.

O algoritmo, por atualizar somente um multiplicador a cada época, converge, segundo Kinzel [10], a uma taxa polinomial. Esta forma de otimização é equivalente na teoria da matemática aos métodos conhecidos como *row-action*, tendo como técnicas similares o método de projeção de Bregman ou o método iterativo de Hildreth para programação quadrática, Bregman [11], Hildreth [12].

4.3 Análise das Condições de otimalidade:

A satisfação das condições de KKT, resultante da obtenção da máxima margem, pode ser analisada da seguinte forma:

Se $0 < \alpha_i < C$ e $y_i \,.\, f(x_i) \approx 1$, então não ocorre correção do multiplicador e o ponto associado permanece na margem como um vetor suporte, atendendo a condição de KKT.

Se $0 < \alpha_i < C$ e $y_i \,.\, f(x_i) > 1$ ou $y_i \,.\, f(x_i) < 1$, então o valor do multiplicador é alterado, ou seja, diminuído ou aumentado, no sentido de satisfazer a condição de KKT.

Se $\alpha_i = 0$ e $y_i \,.\, f(x_i) > 1$, então o valor de α_i é diminuído, porém as restrições de canalização fazem com que o valor de α_i permaneça zero.

Se $\alpha_i = 0$ e $0 < y_i \,.\, f(x_i) < 1$, então o valor de α_i é aumentado, descolando-se do valor zero, e o ponto associado se torna um vetor suporte.

Se $\alpha_i = C$ e $y_i \,.\, f(x_i) > 1$, então o valor de α_i é diminuído e o ponto desloca-se de dentro da margem tornando-se um vetor suporte.

Finalmente, se $\alpha_i = C$ e $0 < y_i \,.\, f(x_i) < 1$, então o valor de α_i é aumentado, porém as restrições de canalização fazem com que o valor de α_i permanece igual a *C* e o ponto dentro da margem.

5. Resultados e Aplicações:

Para comprovar a corretude do algoritmo KPDS apresentamos a solução de um problema artificial de classificação binária relacionado à disposição de um

conjunto de pontos no espaço de entrada R^2, sendo o mesmo, denominado problema da espiral, de natureza não linear, exigindo, para a sua solução o emprego de uma função *kernel* Gaussiana.

Atualmente, estamos trabalhando junto ao núcleo de Bio-Informática do LNCC, utilizando o algoritmo KPDS com funções *kernel* relacionadas ao computo da medida de similaridade de seqüências biológicas em problemas de predição de estruturas secundárias de famílias de proteínas e no reconhecimento de regiões promotoras.

5.1 Problema de separabilidade não linear:

Seja, o conjunto de pontos em R^2 rotulados pelas respectivas classes, conforme o conjunto de treinamento apresentado pela Figura 4.

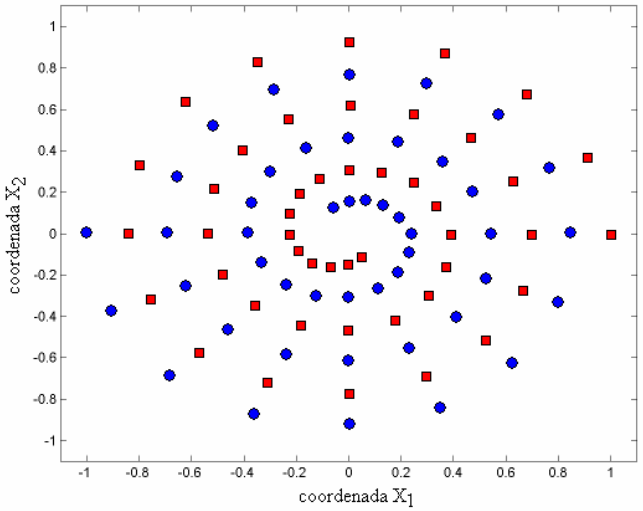

Figura 4: Conjunto de treinamento.

Os seguintes resultados foram apresentados pelo algoritmo KPDS, utilizando como *kernel* a função gaussiana com variância $\sigma^2 = 1$.

Valor da margem: 0.99023

Vetores suportes: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 85, 86, 90, 91

Bias (λ): 0.00395

6. Conclusões:

Neste trabalho descrevemos, de forma sucinta, o desenvolvimento das máquinas de vetores suportes, embasado na teoria de programação matemática, incluindo a utilização de uma margem flexível, chamada de s*oft margin*, que permite a efetiva realização do controle da capacidade do modelo e de seu poder de generalização.

Considerando a classe de hipóteses formadas por hiperplanos ou funções lineares incompatíveis com a solução de problemas não linearmente separáveis descrevemos, também, como este problema pode ser resolvido de maneira eficiente com a utilização de funções *Kernel*.

Finalmente, mostramos a implementação de um algoritmo *online* de treinamento de uma máquina SVM, denominado KPDS, que utiliza uma forma mais estável, porém com uma menor taxa de convergência, de correção dos multiplicadores, com base no gradiente da função lagrangeana. Achamos, pelos testes realizados, que este algoritmo apresenta uma robustez superior ao algoritmo *Kernel-Adatron* de Friess, Cristianini e Campbell [6].

Agradecimentos:

O autor Raul Fonseca agradece ao CNPq pelo apoio prestado a realização de seu Pós-Doutorado.

Referências:

[1] A. Smola e B. Scholkopf**.** *Learning with Kernels.* MIT Press, 2001.

[2] C. Cortes e V. Vapnik, *Support Vector Networks*. Machine Learning, 20: 273-297, 1995.

[3] R. Fletcher**.** *Practical Methods of Optimization.* John Wiley and Sons, 1987.

[4] E. Osuna, R. Freund e F. Girosi, *Support Vector Machines: Training and Applications.* A. I. Memo. n. 1602, CBCL, AIL, MIT, 1997.

[5] J. C. Platt, *Sequencial Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.* Technical Report n. MSR-TR-98-14, Microsoft Research, Seattle, 1998.

[6] T.-T Friess, N. Cristianini e C. Campbell**.** *The Kernel-Adatron Algorithm: A Fast and Simples Learning Procedure for Support Vector Machines.* In Machine Learning Proc. of the 15TH Conf., San Francisco, Morgan Kauffman Pub., 1998.

[7] J. K. Anlauf e M. Biehl, *The Adatron: An Adaptive Perceptron Algorithm.* Europhys. Letters., 10:687-692, 1989.

[8] C. Campbel, e N. Cristianini, *Simple Learning Algorithms for Training Support Vector Machines.* Technical Report, Dept. of Engineering Mathematics, University of Bristol, UK, 1998.

[9] J. Mercer, *Functions of positive and negative type and their connection with theory of integral equations. Philosophical Transactions of the Royal Society*, London A 209, 415-446, 1909.

[10] W. Kinzel**,** *Statistical Mechanics of the Perceptron with Maximal Stability*. Lecture Notes In Physics, Springer Verlarg 368: 175-188, 1990.

[11] L. M. Bregman, *The relaxation methodof finding the common point of convex sets and its application to the solution of problems in convex programming.* U.S.S.R. Computational, Math and Math. Phys., 7:200-217, 1967.

[12] C. Hildreth, *A quadratic programming procedure*. Naval Res. Logist. Quart. 4: 79-85, 1957.