Aplicação de Máquinas de Aprendizado Extremo ao Problema de Aprendizado Ativo

Euler Guimarães Horta
Instituto de Ciência e Tecnologia
Universidade Federal dos Vales do Jequitinhonha e Mucuri
Diamantina, MG, Brasil
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brasil
Email: euler.horta@ufvjm.edu.br

Antônio Pádua Braga
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brasil
Email: apbraga@ufmg.br

***Resumo*—As técnicas de aprendizado ativo têm por objetivo escolher os padrões mais informativos para serem rotulados por um especialista. Em geral essa rotulação tem um custo elevado, o que motiva o estudo de métodos que minimizem o número de rótulos necessários para a construção de modelos. Alguns autores demonstraram que, para classificadores lineares, escolher os padrões mais próximos ao hiperplano separador pode melhorar a capacidade de generalização dos modelos e minimizar o número de rótulos utilizados. Em geral essas abordagens fazem algumas considerações irreais quanto aos dados, exigindo separabilidade linear ou distribuição uniforme dos dados. Além disso, todos os métodos necessitam de um processo de ajuste de parâmetros livres que exige que rótulos sejam reservados para esse fim, aumentando o custo do processo. Neste trabalho será apresentado um novo método de aprendizado ativo para problemas de classificação binária que não faz nenhuma consideração quanto aos dados e que não necessita de nenhum ajuste de parâmetros livres. O algoritmo proposto é baseado em máquinas de aprendizado extremo (*Extreme Learning Machines - ELM*) e em um novo tipo de perceptron apresentado recentemente na literatura.**

Keywords*—*Aprendizado Ativo, Teorema de Convergência do Perceptron, Extreme Learning Machines

I. Introdução

Técnicas de aprendizado ativo são úteis quando muitos dados não rotulados estão disponíveis e obter o rótulo de um padrão tem um custo elevado. Isso justifica o objetivo de tentar construir modelos com o mínimo de rótulos possível. Essas técnicas utilizam algum critério capaz de dizer se um padrão é mais informativo que os outros para decidir se ele deve ou não ser rotulado.

Para realizar o aprendizado ativo, muitos autores fazem considerações que muitas vezes podem não ser realistas, como considerar que os dados são linearmente separáveis e que a distribuição deles é uniforme [1], [2], [3]. Além disso, é necessário o ajuste de parâmetros livres, o que exige a reserva de um conjunto de dados rotulados para esse fim. Isso é indesejável, uma vez que a quantidade de rótulos realmente utilizada no processo de treinamento será a soma dos rótulos de ajuste de parâmetros com os rótulos efetivamente escolhidos pelo processo de aprendizado ativo, elevando, portanto, o custo do processo [4].

Todas essas questões motivam o desenvolvimento de novas técnicas que possam trabalhar com qualquer distribuição de dados, que sejam independentes de separabilidade linear e que não necessitem de ajuste de parâmetros livres. Esses são os objetivos principais deste trabalho.

Um método que possui algumas dessas características são as máquinas de aprendizado extremo (ELM) [5]. As técnicas baseadas em ELM consistem em modelos neurais com uma camada escondida e uma camada de saída. Os pesos da camada escondida são escolhidos aleatoriamente e um separador linear é calculado na camada de saída solucionando-se um sistema de equações lineares. As ELMs exigem que o número de padrões disponíveis seja muito maior que o número de neurônios escondidos a fim de se obter uma boa capacidade de generalização, sendo, a primeira vista, inadequadas para o aprendizado ativo. Recentemente alguns autores [6] demonstraram que a camada escondida ELM pode ser usada como um *kernel* para SVMs, sendo que ao utilizar um número elevado de neurônios os dados são projetados nesse novo espaço de dimensão elevada, tendendo a se tornarem linearmente separáveis, conforme previsto pelo teorema de Cover [7]. SVMs com esse tipo de *kernel* necessitam apenas do ajuste do parâmetro de regularização C. Isso torna o *kernel* ELM um bom candidato para a construção de algoritmos de aprendizado ativo, porém a construção do separador linear baseado em SVM exige que sejam armazenados os vetores de suporte e que seja realizado o ajuste do parâmetro C.

Neste trabalho será proposto um novo algoritmo de aprendizado ativo que não necessitará de ajuste de parâmetros livres e que poderá ser aplicado a problemas não-linearmente separáveis. O método utilizará uma camada escondida ELM com um separador linear na saída. Este separador será baseado em um novo perceptron proposto por Fernandez-Delgado et al. [8] que, segundo os autores, minimiza o erro e maximiza a margem. Ao longo deste trabalho demonstraremos que o teorema de convergência do perceptron [9] poderá ser estendido para o modelo de Fernandez-Delgado et al. e que poderá ser utilizado como critério de parada para o aprendizado ativo. Demonstraremos que o modelo proposto é prático, rápido e possui capacidade de generalização similar às SVMs com *kernel* ELM.

II. Trabalhos Relacionados

Uma questão crucial no aprendizado ativo é: como rotular apenas os padrões mais informativos? Muitos autores acreditam que esses padrões são aqueles presentes na região do espaço em que há grande incerteza em relação aos rótulos. Essa região é aquela que possui diversos padrões próximos entre si e que podem ser de uma classe ou outra, sendo a região com a maior probabilidade do classificador cometer erros. Dessa forma diversos métodos de aprendizado ativo escolhem e rotulam padrões nessa região, já que eles são, teoricamente, os padrões mais informativos do problema.

Uma abordagem para escolher padrões contidos na região de incerteza foi proposta por Tong et al. [10]. Para problemas linearmente separáveis um padrão é escolhido de um conjunto de dados para ser rotulado se ele for o mais próximo de um hiperplano separador construído a priori. Quanto mais próximo um padrão estiver deste hiperplano, maior a chance de uma classificação ser incorreta, o que indica que o seu rótulo pode ser muito informativo e que deveria ser inserido no processo de treinamento. A maior desvantagem dessa abordagem é ter que calcular a distância entre todos os padrões disponíveis e o hiperplano separador, realizando um retreinamento toda vez que um novo padrão é rotulado. Outra desvantagem é que os modelos propostos são baseados em SVMs o que exige a solução de um problema de programação quadrática cada vez que um padrão é rotulado. O método pode ser aplicado em problemas não-linearmente separáveis se um *kernel* for utilizado, porém isso exige que uma quantidade de padrões rotulados seja separado para realizar o ajuste de parâmetros livres.

Outros trabalhos utilizam perceptrons para a realização de aprendizado ativo em problemas linearmente separáveis [1], [2], [3]. Essas abordagens não necessitam que um grande conjunto de dados esteja disponível sendo realizado o aprendizado *on-line*, ou seja, os padrões são avaliados à medida em que eles chegam, decidindo se deverão ser rotulados ou não. Apesar de parecerem práticos, como relatado por [3], eles têm a desvantagem de necessitar de ajuste de parâmetros livres, o que implica a reserva de uma quantidade de padrões rotulados a fim de se realizar a validação-cruzada. Como apontado por [4] isso eleva o custo do processo já que a quantidade de rótulos efetivamente utilizada no processo de aprendizagem será a quantidade de rótulos reservados para o ajuste de parâmetros somados à quantidade de rótulos utilizados na fase de aprendizado ativo. Dessa forma, a realização do aprendizado ativo sem a necessidade de ajuste de parâmetros livres é um importante desafio para novos métodos de aprendizado de máquina e é um dos objetivos deste trabalho.

III. Método Proposto

O objetivo deste trabalho é desenvolver um método de aprendizado ativo para problemas não-linearmente separáveis que não necessite de ajuste de parâmetros livres. Para tanto, propomos a utilização de um modelo neural baseado em uma camada escondida do tipo ELM para realizar a transformação necessária nos dados. A seguir apresentamos o método ELM e suas implicações para a realização do aprendizado ativo.

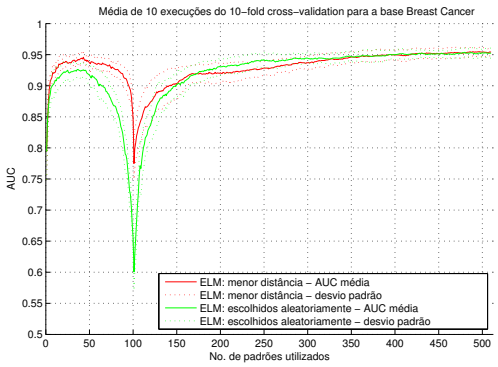

Figura 1. Queda brusca na capacidade de generalização de uma ELM quando o número de padrões se aproxima do número de neurônios escondidos.

A. O uso de uma camada escondida ELM para obter separação linear dos dados

Huang et al. [5] propuseram um algoritmo que consiste em uma rede neural com uma camada escondida em que o vetor de pesos de cada neurônio é escolhido aleatoriamente. O vetor de pesos da camada de saída é um classificador linear calculado através da solução de mínimos quadrados do sistema de equações lineares obtido após a propagação de todos os padrões pela camada escondida. Essa ideia está relacionada ao teorema de Cover [7], que diz que quando os dados são projetados em um espaço de dimensão elevada existe uma maior probabilidade de se tornarem linearmente separáveis. Para o algoritmo funcionar corretamente, o número de padrões deve ser maior ou igual ao número de neurônios escondidos [5]. Dessa forma, à medida que o número de padrões utilizados para treinamento se aproxima do número de neurônios, a capacidade de generalização das ELMs diminui. Isso aparentemente inviabilizaria o aprendizado ativo. Mesmo aplicando a heurística proposta por Tong et al. [10], escolhendo-se para rotulação os padrões mais próximos ao hiperplano separador e realizando um retreinamento, a capacidade de generalização do modelo continua sendo reduzida quando o número de padrões se aproxima do número de neurônios. A figura 1 demonstra este problema para a base Winscosin Breast Cancer Diagnostic (wdbc) do repositório UCI [11] (vide tabela I).

Nessa figura as curvas de generalização são referentes às áreas médias abaixo da curva ROC (AUC) resultantes de 10 execuções do *10-fold cross-validation*. Para este exemplo foi utilizada uma ELM com 100 neurônios escondidos e o aprendizado ativo inicia utilizando-se apenas um padrão escolhido aleatoriamente. Como pode ser verificado, os resultados para os padrões apresentados aleatoriamente são melhores que aqueles obtidos aplicando-se a heurística supracitada, sendo que os dois modelos perdem capacidade de generalização quando o número de padrões fica em torno do número de neurônios da camada escondida. O mesmo comportamento foi verificado para todas as bases apresentadas na tabela I. Isso é indesejável, já que para se obter uma boa separabilidade dos dados é necessário utilizar uma camada escondida com um número muito elevado de neurônios [6], o que implica na necessidade de um número muito elevado de padrões rotulados.

O princípio utilizado por Huang et al. [5] é válido para

qualquer separador linear na saída da rede neural, como demonstrado por Frenay et al. [6]. Dessa forma, buscamos neste trabalho utilizar um separador linear do tipo perceptron na saída da rede, a fim de realizar o aprendizado ativo sem ajuste de parâmetros livres. Um método apropriado para esta tarefa é o modelo proposto recentemente por Fernandez-Delgado et al. [8], que é capaz de ajustar os pesos de um perceptron analiticamente.

B. Perceptron com ajuste analítico dos pesos

Recentemente Fernandez-Delgado et al. [8] propuseram 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 com multiplicadores de Lagrange iguais a 1. Os pesos são ajustados da seguinte forma: $\mathbf{w}_0 = \frac{\sum_{k=1}^{N} d_k \mathbf{x}_k}{||\sum_{k=1}^{N} d_k \mathbf{x}_k||}$. Para a dedução e maiores detalhes veja [8]. Nessa equação d_k é a saída desejada para cada padrão, que deve ser 1 ou -1, e os vetores $\mathbf{x}$ e $\mathbf{w}$ são vetores aumentados, ou seja, $\mathbf{x} = [x_1, x_2, ..., x_m, 1]^T$ e $\mathbf{w} = [w_1, w_2, ..., w_m, bias]^T$, onde $bias$ é o limiar de ativação do perceptron. Essa notação será utilizada ao longo de todo o artigo. Para se obter a saída do perceptron para um ponto desejado basta fazer $y(\mathbf{x}) = sign(\mathbf{w}_0^T \mathbf{x})$.

Essa abordagem, além de dispensar o ajuste do parâmetro C das SVMs, permite também o treinamento *on-line*, já que basta somar os novos padrões, multiplicados por sua saída desejada, ao vetor de pesos e normalizar novamente o vetor resultante dividindo-o por sua norma, a fim de se obter $||\mathbf{w}_0|| = 1$.

Como pode ser observado, o perceptron proposto por Fernandez-Delgado et al. [8] utiliza apenas os padrões de treinamento para ajustar os pesos do perceptron, multiplicados pelos seus rótulos. Dessa forma é esperado que o aprendizado ativo possa melhorar a capacidade de generalização deste modelo, escolhendo para treinamento apenas os padrões mais relevantes.

Uma forma de realizar esta escolha é utilizar a heurística de Tong et al. [10]. A figura 2 mostra a comparação entre a ELM da figura 1 e o modelo gerado utilizando-se a mesma camada escondida, porém com o perceptron de Fernandez-Delgado et al. na saída, tanto para o treinamento utilizando-se padrões escolhidos aleatoriamente quanto para a aplicação da heurística citada. Como pode ser observado o modelo proposto obtém melhores resultados que a ELM, utilizando menos de 100 padrões. A questão que temos que responder é: como encontrar o número ótimo de rótulos necessários para se obter máxima capacidade de generalização para esse modelo? Acreditamos que um bom caminho é estender o bem conhecido teorema de convergência do perceptron [12] para o modelo de Fernandez-Delgado et al. e utilizar o resultado como critério de parada para o algoritmo.

C. Critério de parada baseado no teorema de convergência do perceptron

O teorema de convergência do perceptron define que o algoritmo clássico irá convergir com um número de iterações

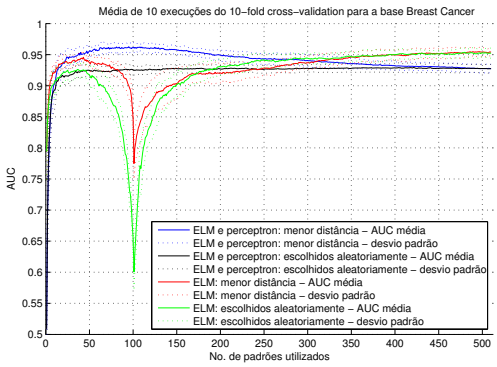

Figura 2. Comparação entre o modelo ELM clássico e o modelo ELM utilizando o perceptron de Fernandez-Delgado et al. na saída, tanto para escolha aleatória dos padrões quanto utilizando-se a heurística de Tong et al. [10].

menor ou igual a um valor máximo se o problema for linearmente separável [12]. Nesse algoritmo um padrão só é aprendido pelo perceptron se a sua classificação tiver sido incorreta. A dedução desse teorema, proposta por [9] e reapresentada por [12], pode ser estendida para o perceptron de Fernandez-Delgado et al. Nesse perceptron todos os padrões disponíveis são usados para o treinamento, independente se sua classificação foi correta ou incorreta. Dessa forma ao estender a dedução desse teorema para esse modelo iremos obter o número máximo de rótulos necessários para garantir que o algoritmo convergiu. A seguir apresentamos a adaptação da dedução apresentada em [12] para o modelo proposto. Ao final demonstraremos como o resultado dessa dedução pode ser utilizado como um novo critério de parada para um algoritmo de aprendizado ativo.

Suponha que se deseje classificar duas classes linearmente separáveis ζ_1 e ζ_2, pertencentes ao conjunto de dados ζ, e que $\mathbf{w}(0) = \mathbf{0}$. Para o perceptron utilizado neste trabalho, todos os padrões selecionados são usados para o treinamento. Assim para padrões $\mathbf{x}(1), \mathbf{x}(2), ..., \mathbf{x}(n)$ que possuem os rótulos $d(1), d(2), ..., d(n)$ (que podem ser 1 ou -1) temos que:

$$\mathbf{w}(n+1) = d(1)\mathbf{x}(1) + d(2)\mathbf{x}(2) + ... + d(n)\mathbf{x}(n) \quad (1)$$

Nesta equação não realizamos a normalização do vetor $\mathbf{w}(n+1)$ para facilitar a análise. Isso resulta em:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + d(n)\mathbf{x}(n) \quad (2)$$

Como as classes ζ_1 e ζ_2 são linearmente separáveis, existe uma solução $\mathbf{w}_0$ que classifica corretamente esses padrões. Assim, para uma solução fixa $\mathbf{w}_0$ pode-se definir um número positivo α através da equação 3.

$$\alpha = \min_{\mathbf{x}(n) \in \zeta} |d(n)\mathbf{w}_0^T \mathbf{x}(n)| \quad (3)$$

Multiplicando os dois lados da equação 1 pelo vetor $\mathbf{w}_0^T$ teremos: $\mathbf{w}_0^T \mathbf{w}(n+1) = d(1)\mathbf{w}_0^T \mathbf{x}(1) + d(2)\mathbf{w}_0^T \mathbf{x}(2) + ... +$

$d(n)\mathbf{w}_0^T\mathbf{x}(n)$. De acordo com a definição apresentada pela equação 3 teremos:

$$\mathbf{w}_0^T\mathbf{w}(n+1) \geq n\alpha \tag{4}$$

Utilizando a inequação de *Cauchy–Schwarz* teremos:

$$||\mathbf{w}_0||^2||\mathbf{w}(n+1)||^2 \geq [\mathbf{w}_0^T\mathbf{w}(n+1)]^2 \tag{5}$$

Observando a equação 4 podemos concluir que $[\mathbf{w}_0^T\mathbf{w}(n+1)]^2 \geq n^2\alpha^2$, ou de forma equivalente:

$$||\mathbf{w}(n+1)||^2 \geq \frac{n^2\alpha^2}{||\mathbf{w}_0||^2} \tag{6}$$

Em seguida reescrevemos a equação 2 como sendo $\mathbf{w}(k+1) = \mathbf{w}(k)+d(k)\mathbf{x}(k)$, onde $k = 1, ..., n$. Tomando o quadrado da norma euclidiana dos dois lados dessa equação obtemos:

$$||\mathbf{w}(k+1)||^2 = ||\mathbf{w}(k)||^2 + ||d(k)\mathbf{x}(k)||^2 + 2d(k)\mathbf{w}^T(k)\mathbf{x}(k) \tag{7}$$

A equação 7 pode ser reescrita da seguinte forma:

$$||\mathbf{w}(k+1)||^2 - ||\mathbf{w}(k)||^2 = ||d(k)\mathbf{x}(k)||^2 + 2d(k)\mathbf{w}^T(k)\mathbf{x}(k) \tag{8}$$

Somando essa equação para $k = 1, ..., n$ e utilizando a condição inicial $\mathbf{w}(0) = \mathbf{0}$ obtemos:

$$||\mathbf{w}(n+1)||^2 = \sum_{k=1}^{n} ||d(k)\mathbf{x}(k)||^2 + 2\sum_{k=1}^{n} d(k)\mathbf{w}^T(k)\mathbf{x}(k) \tag{9}$$

Como $||d(k)\mathbf{x}(k)||^2 = ||\mathbf{x}(k)||^2$, independente do valor de $d(k)$, podemos definir:

$$\beta = \max_{\mathbf{x}(k)\in\zeta} ||\mathbf{x}(k)||^2 \tag{10}$$

Além disso podemos definir:

$$\theta = \max_{\mathbf{x}(k)\in\zeta} |d(k)\mathbf{w}^T(k)\mathbf{x}(k)| \tag{11}$$

Utilizando as definições das equações 10 e 11, podemos garantir que a seguinte inequação é verdadeira:

$$||\mathbf{w}(n+1)||^2 \leq n\beta + 2n\theta \tag{12}$$

Pela equação 12 podemos concluir que o quadrado da norma do vetor de pesos cresce no máximo linearmente com o valor de n. Esse resultado é conflitante com aquele apresentado na equação 6 para valores suficientemente grandes de n. Assim podemos concluir que o valor de n não pode ser maior que um certo valor n_{max} onde as equações 6 e 12 são iguais:

$$\frac{n_{max}^2\alpha^2}{||\mathbf{w}_0||^2} = n_{max}(\beta + 2\theta) \tag{13}$$

Dessa forma $n_{max} = \frac{(\beta+2\theta)||\mathbf{w}_0||^2}{\alpha^2}$.

Como o perceptron de Fernandez-Delgado et al [8] possui $||\mathbf{w}_0|| = 1$ a equação anterior se resume na equação 14.

$$n_{max} = \frac{\beta + 2\theta}{\alpha^2} \tag{14}$$

Essa equação prova que o perceptron de Fernandez-Delgado et al. converge utilizando no máximo $\frac{\beta+2\theta}{\alpha^2}$ rótulos. Dessa forma a equação 14 é um bom indicador da convergência do algoritmo, sendo, portanto, um candidato para o critério de parada do algoritmo de aprendizado ativo, conforme apresentaremos a seguir.

D. Máquinas de Aprendizado Ativo e Extremo

Realizaremos o aprendizado ativo construindo um modelo composto por uma camada escondida ELM e uma camada de saída baseada no perceptron de Fernandez-Delgado et al. [8], de forma que a primeira camada irá projetar os dados em um espaço de características ELM e a segunda camada realizará a separação linear dos dados nesse novo espaço. A equação 15 apresenta a nova forma de ajuste dos pesos na camada escondida, sendo $\phi(\mathbf{x}_k)$ a projeção do padrão de treinamento $\mathbf{x}_k$ no espaço de características definido pela camada escondida da rede, que possui função de ativação tangente hiperbólica. A saída da rede será dada por $y(\mathbf{x}) = sign(\mathbf{w}_0^T\phi(\mathbf{x}))$.

$$\mathbf{w}_{ho} = \frac{\sum_{k=1}^{N} d_k\phi(\mathbf{x}_k)}{||\sum_{k=1}^{N} d_k\phi(\mathbf{x}_k)||} \tag{15}$$

Inicialmente todos os padrões serão propagados pela camada escondida formando o conjunto de dados C_{ELM}. Um padrão (ou mais) desse conjunto é selecionado aleatoriamente e rotulado sendo utilizado para atualizar os pesos do perceptron através da equação 15 para construir o primeiro hiperplano separador. Todos os padrões C_{ELM} terão suas margens calculadas em relação ao hiperplano corrente. O padrão de menor margem será selecionado e este valor será atribuído à variável α da equação 14. A variável θ receberá o valor máximo entre a margem do vetor atual e o maior dos αs anteriores (valor que deve ser armazenado a cada iteração do algoritmo). A variável β consistirá no valor máximo entre a norma do vetor atual e a maior norma entre os vetores utilizados no treinamento anterior (valor que também deve ser armazenado a cada iteração do algoritmo). De posse desses valores é calculado o valor de n_{max}, sendo que, se o valor for maior que o número de padrões utilizados para o treinamento significa que o algoritmo ainda não convergiu do ponto de vista do padrão atual e o mesmo deverá ter o rótulo solicitado e ser removido do conjunto C_{ELM}, sendo que os pesos do perceptron deverão ser ajustados da seguinte forma: $\mathbf{w}_{t+1} = \frac{\mathbf{w}_t+d\phi(\mathbf{x})}{||\mathbf{w}_t+d\phi(\mathbf{x})||}$. Caso n_{max} seja menor que o número de padrões utilizados para o treinamento, significa que o algoritmo já convergiu, não sendo

Entrada: Conjunto de padrões C, número inicial de padrões utilizados para treinamento m
Saída : Vetor de pesos **w**, número de padrões necessário para convergir n
Método :

Propague todos os padrões de C pela camada escondida formando o conjunto C_{ELM};
Retire aleatoriamente m padrões de C_{ELM} e solicite os rótulos;
Ajuste **w** com a equação 15 utilizando os padrões selecionados;
Faça n = Inf;
enquanto *($n > m$) e ($C_{ELM} \neq \emptyset$)* **faça**
    Calcule $\alpha = |\mathbf{w}^T \phi(\mathbf{x})|$ para todos os padrões de C_{ELM} e escolha o menor α;
    Calcule β usando a equação 10 e θ usando a equação 11;
    Calcule n usando a equação 14 ;
    se $n > m$ **então**
        Retire $\phi(\mathbf{x})$ de C_{ELM};
        Solicite o rótulo d de $\phi(\mathbf{x})$;
        $\mathbf{w} = \frac{\mathbf{w}+d\phi(\mathbf{x})}{||\mathbf{w}+d\phi(\mathbf{x})||}$;
        $m = m + 1$;
    fim
fim

Algoritmo 1: Máquina de aprendizado ativo e extremo

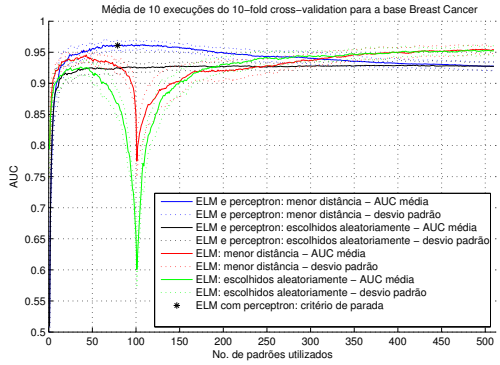

Figura 3. Resultado da aplicação do algoritmo proposto no exemplo da figura 2. A solução do algoritmo é bem próxima do valor ótimo.

necessário solicitar o rótulo deste padrão. Nessa situação o algoritmo termina. O algoritmo 1 apresenta o método proposto em detalhes.

A figura 3 mostra o resultado médio da execução do algoritmo proposto no exemplo da figura 2. Como pode ser verificado, a solução do algoritmo é bem próxima da solução ótima. O mesmo comportamento foi verificado para as bases apresentadas na tabela I. Para a construção das figuras 1, 2 e 3 foram utilizadas em cada execução a mesma camada escondida para todos os métodos e a mesma distribuição dos *folds* da validação cruzada.

IV. Experimentos e Resultados

Os experimentos realizados neste trabalho têm o objetivo de comparar o método proposto, aqui chamado EALM (*Extreme Active Learning Machines*), com os perceptrons de Dasgupta et al. [2], denominado *PDKCM*, de Cesa-Bianchi et al. [1], denominado *PCBGZ*, com a SVM de Tong et al. [10], denominado *SVMTK*, e com uma SVM treinada com todos os padrões de treinamento disponíveis, denominada *SVMALL*. O objetivo é utilizá-los como saída linear para uma ELM. Em todos os casos a camada escondida é a mesma, sendo composta por 1000 neurônios e com os pesos escolhidos aleatoriamente na faixa [-3, 3], conforme proposto por [6]. A tabela I apresenta as características das bases utilizadas, que foram obtidas no repositório UCI Machine Learning [11]. Todos os padrões com dados faltantes foram removidos. Os modelos *PDKCM* e *PCBGZ* foram adaptados para utilizarem a heurística de Tong et al. [10]. Todos os dados foram normalizados de forma que os padrões de entrada tenham média 0 e desvio padrão 1.

Tabela I. Características das bases utilizadas

Nome	Sigla	No. Padrões	No. Entradas
Heart Disease	HRT	297	13
Wisc. Breast Cancer Original	WBCO	699	9
Wisc. Breast Cancer Diagnostic	WBCD	569	31
Pima Diabetes	PIMA	768	8
Sonar	SNR	208	60
Ionosphere	ION	351	34
Australian Credit	AUST	690	14
Liver Disorder	LIV	345	6
German Credit	GER	1000	24
Spam	SPAM	4601	58

Em todos os casos, 30% das bases foram separados para realizar o ajuste de parâmetros dos modelos *PDKCM*, *PCBGZ* e as SVMs, lembrando que para o modelo EALM não é necessário nenhum ajuste de parâmetros. O ajuste foi realizado utilizando a técnica *10-fold cross-validation*, similarmente ao trabalho de Monteleoni et al. [3]. Este ajuste foi realizado a fim de se obter a melhor AUC possível com um número reduzido de rótulos. Para o caso do *PCBGZ* além do parâmetro ajustado também foram feitos testes para o parâmetro ótimo $b = (max_{x \in C}||x||^2)/2$ descrito em [1], sendo o modelo denominado *PCBGZ-OPT*. Para as SVMs o parâmetro de regularização C foi ajustado usando os valores da faixa $\{2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 1, 2, 2^2, ..., 2^{14}\}$ conforme proposto por [8]. Para verificar a capacidade de generalização de cada um dos modelos foi realizado o *10-fold cross-validation* obtendo-se a acurácia média e a AUC média bem como o número médio de rótulos utilizados. A validação cruzada foi executada 10 vezes. Os modelos EALM, PDKCM e PCBGZ foram inicializados com um padrão escolhido aleatoriamente. O modelo SVMTK utilizou um conjunto inicial composto por 2 padrões, sendo um de cada classe e escolhidos aleatoriamente, conforme proposto por [10].

A tabela II apresenta os resultados. Foram calculados tanto o número de rótulos obtidos pelo aprendizado ativo (Rótulos AL), quanto o número de rótulos efetivamente utilizados (Rótulos Efetivos), que é composto pelos rótulos de ajuste de parâmetros somados aos Rótulos AL. Como pode ser observado, os melhores resultados foram obtidos para o modelo EALM e para o modelo SVMTK. Os resultados do modelo EALM são próximos ao do modelo SVMALL e do modelo SVMTK, sendo que este último seleciona um número menor de rótulos durante o aprendizado ativo, porém com custo computacional muito mais elevado e com a necessidade de se realizar o retreinamento sobre todo o conjunto de treinamento a

Tabela II. RESULTADOS MÉDIOS DE 10 EXECUÇÕES DO 10-FOLD CROSS-VALIDATION

	EALM				PDKCM			
Key	Rótulos AL	Rótulos Efetivos	Ac	AUC	Rótulos AL	Rótulos Efetivos	Ac	AUC
HRT	79.97 ± 3.80	79.97	0.84 ± 0.01	0.83 ± 0.01	21.76 ± 1.88	102.76	0.65 ± 0.05	0.65 ± 0.05
WBCO	33.23 ± 1.64	33.23	0.98 ± 0.00	0.98 ± 0.00	14.63 ± 9.89	219.63	0.91 ± 0.05	0.91 ± 0.04
WBCD	71.96 ± 5.66	71.96	0.98 ± 0.00	0.97 ± 0.01	19.16 ± 2.96	190.16	0.82 ± 0.03	0.82 ± 0.03
PIMA	203.55 ± 3.65	203.55	0.77 ± 0.01	0.73 ± 0.01	76.36 ± 16.86	306.36	0.56 ± 0.04	0.55 ± 0.04
SNR	104.19 ± 2.55	104.19	0.71 ± 0.02	0.72 ± 0.02	16.26 ± 1.35	78.26	0.51 ± 0.05	0.52 ± 0.04
ION	99.69 ± 4.40	99.69	0.89 ± 0.01	0.86 ± 0.01	15.85 ± 3.20	120.85	0.56 ± 0.08	0.57 ± 0.07
AUST	122.02 ± 4.72	122.02	0.86 ± 0.01	0.86 ± 0.01	83.14 ± 8.08	290.14	0.60 ± 0.07	0.60 ± 0.07
LIV	170.35 ± 6.78	170.35	0.60 ± 0.02	0.61 ± 0.02	23.80 ± 2.70	127.80	0.51 ± 0.03	0.51 ± 0.04
GER	308.08 ± 7.13	308.08	0.75 ± 0.01	0.66 ± 0.01	157.48 ± 7.80	457.48	0.52 ± 0.04	0.52 ± 0.03
SPAM	520.10 ± 9.37	520.10	0.92 ± 0.00	0.91 ± 0.00	447.00 ± 40.64	1827.00	0.59 ± 0.06	0.59 ± 0.05
	PCBGZ				PCBGZ-OPT			
Key	Rótulos AL	Rótulos Efetivos	Ac	AUC	Rótulos AL	Rótulos Efetivo	Ac	AUC
HRT	57.75 ± 1.35	138.75	0.74 ± 0.02	0.72 ± 0.03	161.50 ± 0.86	242.50	0.81 ± 0.00	0.80 ± 0.01
WBCO	135.67 ± 3.51	340.67	0.96 ± 0.00	0.95 ± 0.01	300.65 ± 2.14	505.65	0.97 ± 0.00	0.96 ± 0.00
WBCD	124.29 ± 2.17	295.29	0.94 ± 0.00	0.93 ± 0.01	292.31 ± 1.16	463.31	0.94 ± 0.00	0.94 ± 0.01
PIMA	89.41 ± 2.89	319.41	0.65 ± 0.00	0.51 ± 0.01	393.16 ± 1.63	623.16	0.68 ± 0.00	0.56 ± 0.01
SNR	88.35 ± 1.87	150.35	0.66 ± 0.02	0.68 ± 0.02	128.81 ± 0.39	190.81	0.64 ± 0.02	0.66 ± 0.02
ION	136.38 ± 4.78	241.38	0.68 ± 0.04	0.73 ± 0.02	215.92 ± 0.50	320.92	0.69 ± 0.01	0.72 ± 0.01
AUST	133.04 ± 4.49	340.04	0.70 ± 0.02	0.67 ± 0.02	386.64 ± 1.67	593.64	0.80 ± 0.00	0.78 ± 0.01
LIV	78.68 ± 2.42	182.68	0.50 ± 0.01	0.51 ± 0.01	193.55 ± 0.72	297.55	0.50 ± 0.01	0.51 ± 0.01
GER	102.06 ± 1.63	402.06	0.71 ± 0.01	0.51 ± 0.01	561.83 ± 2.56	861.83	0.72 ± 0.00	0.55 ± 0.00
SPAM	751.68 ± 12.81	2131.68	0.78 ± 0.00	0.72 ± 0.01	2229.68 ± 2.14	3609.68	0.81 ± 0.00	0.76 ± 0.00
	SVMTK				SVMALL			
Key	Rótulos AL	Rótulos Efetivos	Ac	AUC	Rótulos AL	Rótulos Efetivos	Ac	AUC
HRT	45.70 ± 4.24	126.70	0.82 ± 0.01	0.82 ± 0.01	170.00	251.00	0.81 ± 0.01	0.83 ± 0.08
WBCO	18.70 ± 1.83	223.70	0.97 ± 0.00	0.97 ± 0.00	430.00	635.00	0.97 ± 0.00	0.97 ± 0.02
WBCD	40.00 ± 2.79	211.00	0.98 ± 0.00	0.97 ± 0.00	358.00	529.00	0.98 ± 0.00	0.99 ± 0.02
PIMA	138.60 ± 26.06	368.60	0.75 ± 0.01	0.72 ± 0.01	485.00	715.00	0.74 ± 0.01	0.71 ± 0.08
SNR	59.60 ± 9.71	121.60	0.75 ± 0.02	0.75 ± 0.02	131.00	193.00	0.81 ± 0.01	0.82 ± 0.10
ION	55.80 ± 5.77	160.80	0.90 ± 0.01	0.87 ± 0.01	221.00	326.00	0.91 ± 0.01	0.90 ± 0.07
AUST	68.90 ± 13.90	275.90	0.85 ± 0.01	0.85 ± 0.01	434.00	641.00	0.84 ± 0.01	0.84 ± 0.04
LIV	96.40 ± 26.01	200.40	0.64 ± 0.02	0.64 ± 0.02	217.00	321.00	0.67 ± 0.02	0.62 ± 0.11
GER	207.40 ± 17.33	507.40	0.74 ± 0.01	0.66 ± 0.01	630.00	930.00	0.75 ± 0.01	0.64 ± 0.04
SPAM	340.00 ± 25.26	1720.00	0.92 ± 0.00	0.92 ± 0.00	2898.00	4278.00	0.93 ± 0.00	0.93 ± 0.02

cada passo, não sendo, portanto, tão prático quanto o modelo EALM. Além disso, o número efetivo de rótulos utilizados pelo modelo EALM é menor que todos os outros modelos.

V. CONCLUSÃO

Neste trabalho apresentamos um modelo de aprendizado ativo capaz de classificar problemas não-linearmente separáveis minimizando o número de rótulos necessários e sendo o método livre de ajuste de parâmetros. Demonstramos que o teorema de convergência do perceptron clássico pode ser adaptado para o perceptron de Fernandez-Delgado et al. e utilizado como critério de parada para o aprendizado ativo. Foi demonstrado que utilizar a forma padrão de treinamento das ELMs para realizar o aprendizado ativo não é adequado, uma vez que o número de padrões deve ser muito maior que o número de neurônios utilizados. O modelo proposto foi comparado com outros algoritmos presentes na literatura. Foi verificado que os resultados obtidos em termos da acurácia e da AUC são muito próximos dos obtidos utilizando-se SVMs com aprendizado ativo e comum. Foi verificado que o número de rótulos necessários para a construção do modelo EALM é menor que outros métodos, uma vez que devemos considerar tanto os rótulos separados para o ajuste de parâmetros quanto os rótulos obtidos pelo aprendizado ativo. Por fim, o método se mostrou extremamente prático e ágil obtendo bons resultados.

REFERÊNCIAS

[1] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Worst-Case Analysis of Selective Sampling for Linear Classification," *Journal of Machine Learning Research*, vol. 7, pp. 1205–1230, 2006.

[2] S. Dasgupta, A. T. Kalai, and C. Monteleoni, "Analysis of Perceptron-Based Active Learning," *Journal of Machine Learning Research*, vol. 10, pp. 281–299, 2009.

[3] C. Monteleoni and M. Kääriäinen, "Practical Online Active Learning for Classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 328, no. 7449, 2007.

[4] A. Guillory, E. Chastain, and J. Bilmes, "Active Learning as Non-Convex Optimization," in *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 5, Clearwater Beach, Florida, 2009.

[5] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, Dec. 2006.

[6] B. Frénay and M. Verleysen, "Using SVMs with randomised feature spaces: an extreme learning approach," in *European Symposium on Artificial Neural Networks*, no. April, 2010, pp. 315–320.

[7] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *Ieee Transactions On Electronic Computers*, vol. EC-14, no. 3, pp. 326–334, 1965.

[8] M. Fernandez-Delgado, J. Ribeiro, E. Cernadas, and S. B. Ameneiro, "Direct parallel perceptrons (DPPs): fast analytical calculation of the parallel perceptrons weights with margin control for classification tasks." *IEEE transactions on neural networks*, vol. 22, no. 11, pp. 1837–48, Nov. 2011.

[9] N. Nilsson, *Learning Machines*. New York: McGraw-Hill, 1965.

[10] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *Journal of Machine Learning Research*, pp. 45–66, 2001.

[11] A. Frank and A. Asuncion, "Uci machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[12] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.