2
Uma Proposta para Seleção de Dados em Modelos LVQ

Abordagens diferentes têm sido encontradas na literatura para seleção de dados. Entretanto, todas as técnicas buscam de fato o mesmo objetivo, seleção inteligente de dados, ou seja, manipular o conjunto de treinamento de forma a alcançar determinados interesses futuros, seja ele o de melhor generalização dado um modelo prévio, economia de custo computacional ou encontrar um melhor modelo entre vários. Na verdade, a procura é feita sempre de modo que o aprendizado seja maximizado, ou seja, consiga-se encontrar o melhor resultado para um determinado problema. No apêndice A, a questão de seleção de dados é discutida com mais detalhes. A principal contribuição desta dissertação é propor um algoritmo para seleção de dados em aprendizado por quantização vetorial.

Aprendizado por quantização vetorial é um algoritmo proposto por Kohonen [8], usado em problemas supervisionados. Pela eficiência do método e facilidade de implementação, ele se tornou um algoritmo extremamente conhecido em problemas de classificação e reconhecimento de padrões. Além disso, variações dessa técnica surgiram acompanhadas do estudo de outros detalhes como a análise das margens [12]. No apêndice B podem ser encontrados mais detalhes sobre esse algoritmo.

2.1.
Seleção de Dados em LVQ

Em geral, o treinamento dos protótipos dos modelos LVQ é iniciado como uma medida de posição de cada uma das classes, por exemplo, a média. A figura 1 mostra um exemplo para duas classes em dimensão 2, onde cada classe possui 1000 padrões e os protótipos são as cruzes vermelha e azul das classes 1 e 2, respectivamente.

No apêndice B são apresentados diversos procedimentos para atualização de protótipos dentro da filosofia do LVQ. Ao final do treinamento, o protótipo converge para uma posição do espaço de acordo com um determinado critério e com os dados disponíveis. A figura 2 mostra a posição dos protótipos ao final do treinamento, utilizando LVQ 1.

Entretanto, em problemas reais, nem sempre a posição final garantirá uma boa classificação para o modelo. No exemplo apresentado nas figuras 1 e 2, houve uma mudança pouco significativa dos protótipos em relação às posições originais. A figura 3 mostra que o ponto marcado em preto, que pertence à classe azul se encontra mais próximo ao protótipo da outra classe, ou seja, seria classificado de forma errada, apesar de todos os padrões terem sido utilizados para treinar o modelo. Se observarmos atentamente a figura 3, podemos ver que não só o ponto preto, mas vários pontos pertencentes à vizinhança dele teriam problemas de classificação. Esses pontos pertencem à zona de risco [6],[7], pois podem ser facilmente confundidos com pontos da outra distribuição. A definição 3.1 ilustra o conceito de zona de risco aplicado ao contexto de LVQ 1.

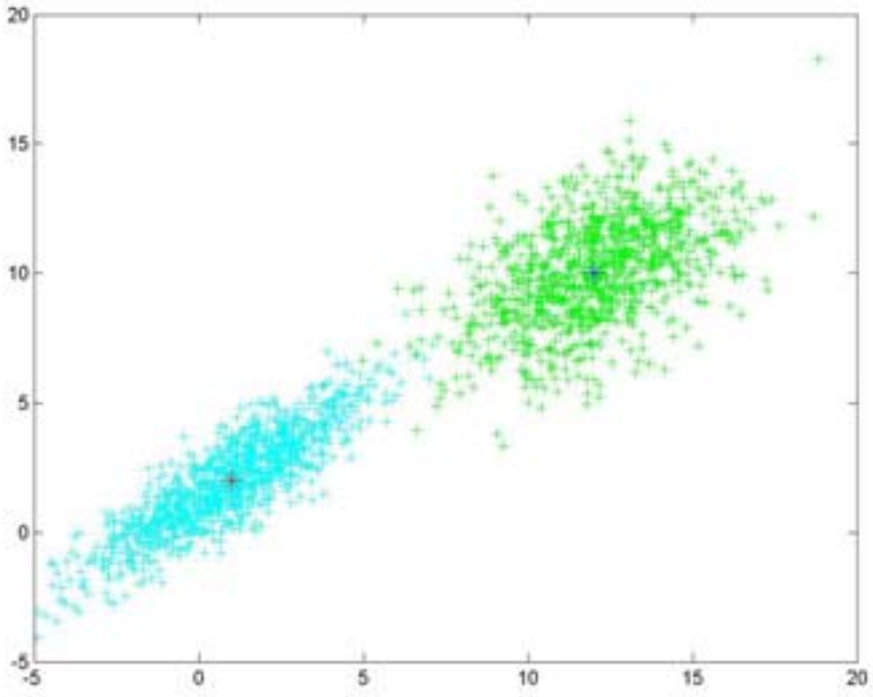

Figura 1 - Distribuições 1 e 2 com protótipos nos centróides

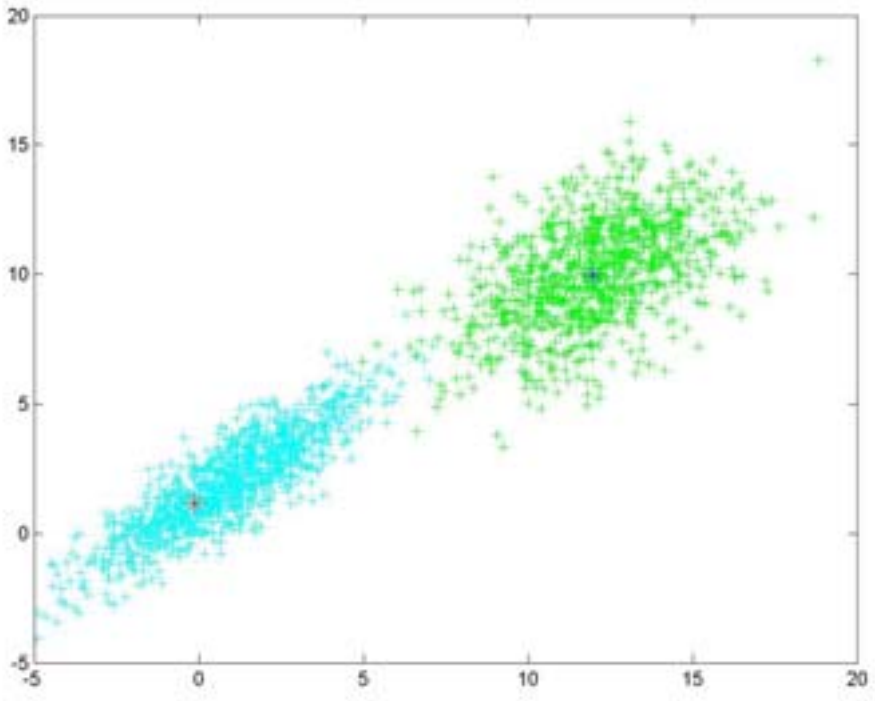

Figura 2 - Distribuições 1 e 2 com protótipos treinados

Definição. 2.1 Seja D um conjunto de treinamento para o modelo LVQ 1 com n classes diferentes. Os subconjuntos $D_1,...,D_n$ de D, pertencem a **zona de risco** das classes 1,...,n respectivamente, se cada $x_j \in D_i$, i = 1,...,n está mais próximo a um protótipo m_k, k = 1,...,n, $k \neq i$.

Em outras palavras, um ponto pertence à zona de risco de uma certa classe se ele está mais próximo a um protótipo de outra classe.

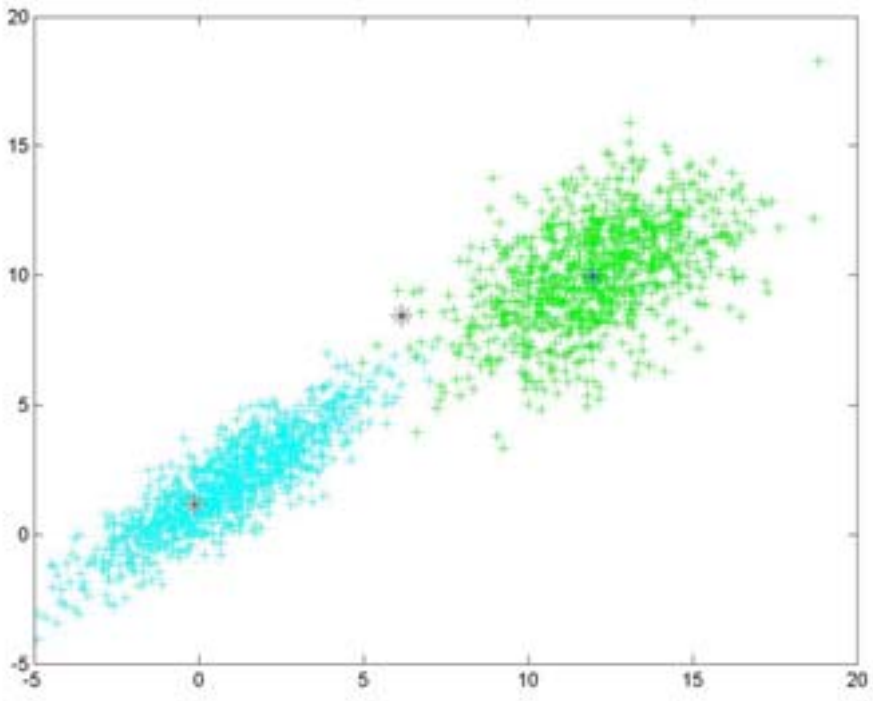

Figura 3 - Distribuições 1 e 2 com ponto classificado errado

Assim é necessário realizar algum procedimento para que, na generalização, pontos que pertençam à zona de risco, ou, então, estejam nas fronteiras de decisão não sejam prejudicados e possam ser classificados de forma correta.

No exemplo apresentado pelas figuras 1, 2 e 3, temos um problema relativamente fácil. Podemos manter os protótipos das classes localizados na média e a classificação ainda assim se mantém muito boa, nesse caso, ainda ligeiramente melhor em comparação aos protótipos atualizados. Mas é uma boa ilustração de como isso é uma questão pertinente ao se trabalhar com LVQ. A tabela 1 mostra a classificação antes e depois do treinamento dos protótipos.

Classificação antes e depois da Atualização

	Antes	Depois
Acerto	99,5%	99,4%
Erro	0,5%	0,6%

Tabela 1

As distribuições das classes 1 e 2 também possuem outra característica que faz com que a classificação seja boa mesmo com os protótipos nas médias. Elas não são distribuições ruidosas. A presença de ruídos nas distribuições pode prejudicar a atualização de protótipos.

Definição. 2.2 Seja D um conjunto de treinamento para modelo LVQ 1 com n classes diferentes. Os subconjuntos $R_1,...,R_n$ de D, são ditos **ruídos** das classes 1,...,n respectivamente, se cada $x_j \in R_i$, i = 1,...,n possui a saída de uma classe k, k = 1,...,n, $k \neq i$.

Assim os ruídos são aqueles pontos que possuem a saída de uma classe diferente da qual pertencem. Dessa forma, durante o processo de atualização, se eles estiverem mais próximos ao protótipo de sua classe verdadeira, eles afastarão esse protótipo ao invés de aproximarem.

Assim, em problemas ruidosos, algum tipo de seleção de dados já pode trazer uma eficiência maior, caso consiga eliminar ao menos parte do ruído presente nas distribuições.

Em [6], [7], a idéia de privilegiar pontos pertencentes à zona de risco mistura o conceito de seleção de dados com análise de margem. Apenas um conjunto especial de pontos é utilizado em [6] e [7] para atualizar os protótipos, e, esses pontos, são aqueles que estavam nas zonas de risco, podendo ser classificados erroneamente. Os protótipos se aproximavam das chamadas zonas de risco, procurando assim uma alternativa para que os pontos pertencentes a sua distribuição permanecessem classificados acertadamente. A utilização de múltiplos protótipos nesse exemplo pode trazer uma performance ainda melhor.

Distribuições ruidosas podem conduzir a sérias dificuldades durante o treinamento de um modelo LVQ. Isso porque um ponto que teoricamente deveria pertencer a uma distribuição pela sua localização no espaço, mas que durante o aprendizado se verifica que ele pertence à outra distribuição, pode fazer com que protótipos sejam afastados da região que verdadeiramente deveriam ocupar,

fazendo com que eles convirjam para regiões do espaço distante da posição ótima prejudicando fortemente a generalização.

Dessa maneira, em se tratando de modelos LVQ, manipular os dados de forma a evitar ruído e privilegiar pontos pertencentes à fronteira de decisão é a chave para se obter um treinamento eficiente que garanta uma boa generalização.

2.2. Uma Proposta de Seleção de Dados: Algoritmo de Alexander – Abu-Mostafa

Nesta seção será apresentada a técnica que é a principal motivação desta dissertação, apresentada originalmente em [15].

2.2.1. Aprendizado Exaustivo

O objetivo em qualquer modelo de aprendizado é descobrir uma função f: X → Y, onde X e Y são espaços conhecidos, ou seja, encontrar o mapeamento do conjunto de entrada no conjunto de saída. f é chamada de função alvo.

Definição. 2.3 Dados X e Y espaços de entrada e saída. **Modelo de aprendizado** é o conjunto G de todas as funções candidatas a exercerem o papel de função alvo.

Um algoritmo geral de validação de dados foi apresentado em [15], baseado em aprendizado exaustivo, isto é, seleção aleatória de funções alvo pertencentes ao modelo de aprendizado. O processo é chamado exaustivo porque qualquer hipótese, isto é, qualquer função pertencente ao modelo de aprendizado pode ser selecionada.

2.2.2. Generalização sob Aprendizado Exaustivo

A idéia de generalizar utilizando aprendizado exaustivo está em verificar, dada uma função alvo, seu erro no treinamento e o erro na fase de testes dessa

função. A generalização ideal, segundo [15], ocorre quando os dois erros, de treinamento e teste, são iguais.

A maneira utilizada para medir o sucesso da generalização, ainda em [15], se dá pela esperança do erro quadrático entre erro de treinamento e erro de teste, ou seja,

$$E[(erro\_treinamento - erro\_teste)^2]$$

É importante citar que uma boa generalização não está associada a um critério que estabeleça um patamar de erro no teste, mesmo porque isso varia de acordo com o problema, mas, sim, a observação de que, para uma determinada função, o erro durante a fase de treinamento provê uma boa estimativa do que será o erro no teste.

Entretanto, sob algoritmos supervisionados, é importante verificar se o conjunto de treinamento conduz o modelo a uma boa generalização. No apêndice A é mostrado que, podem-se selecionar subconjuntos do conjunto de treinamento que possibilitem resultados melhores em termos de generalização.

2.2.3. Correlações entre Erro de Treinamento e Erro de Teste

Sob aprendizado exaustivo [15], overfitting surge quando os erros dos pontos do conjunto de treinamento não são bons indicadores da taxa geral de erro, ou seja, esses erros são pouco correlatados com o erro esperado.

Dessa forma, para cada elemento pertencente ao conjunto de treinamento, pode-se estabelecer o erro de treinamento como sendo o erro médio do conjunto com exceção do ponto em questão, e o erro de teste como sendo o erro desse ponto. Ao medir a correlação entre esses dois erros, estamos medindo o quanto o erro de um ponto está correlatado com o erro geral. Essa é uma contribuição importante de [15].

Assim, para cada $x_i \in D$, calcula-se o erro médio de D – $\{x_i\}$ como $\frac{1}{n-1}\sum_{j=1}^{n-1} erro(x_j)$, para $j \neq i$, denotado por erro_dentro, onde x_i é um elemento hipotético de D cuja correlação com o erro geral deseja-se calcular, e calcula-se também o erro de x_i, denotado por erro_fora. Temos:

$$\rho(D) = corr[erro\_dentro, erro\_fora]$$

Ou seja,

$$\rho(D) = \frac{E[(erro\_dentro)(erro\_fora)] - E[erro\_dentro]E[erro\_fora]}{\sqrt{Var[erro\_dentro]}\sqrt{Var[erro\_fora]}}$$

$\rho(D)$ representa o cálculo da correlação para todo o conjunto, ponto a ponto. Entretanto, o interesse está no resultado para um ponto específico x_i. Se $\rho(x_i) = 0$, então o erro de x_i não diz nada sobre o erro geral. Se $\rho(x_i) < 0$, então a hipótese que classifica x_i corretamente tende a ter um erro de teste maior do que as hipóteses que cometem um erro na classificação de x_i. Temos, então, que ρ reflete o quanto os exemplos são úteis para o aprendizado.

2.2.4. Análise de ρ

Associar ao conjunto de treinamento um vetor com os respectivos valores do cálculo de ρ, pode trazer benefícios para seleção de dados. Em muitos problemas [15], descartar todos os elementos que possuam $\rho < 0$ já pode ser suficiente para efetuar uma seleção de dados eficiente.

2.2.4.1. Estimação de Ruído

A análise dos valores de ρ para um certo número de hipóteses pode trazer informações importantes sobre a quantidade de ruído presente nas distribuições

examinadas, possibilitando a rejeição desses pontos, trazendo assim boas conseqüências para o aprendizado [15]. Possibilitar a rejeição de, ao menos uma certa quantidade de ruído, passa a ser uma característica de extrema importância dessa técnica.

Intuitivamente é possível compreender essa sensibilidade ao ruído. Ela ocorre porque ao sortear aleatoriamente funções alvo para o conjunto de dados, na maioria das vezes em que essas funções estiverem ajustadas para os pontos da distribuição, os ruídos terão classificação errada e vice-versa. Assim, em alguns casos, parte dos ruídos acaba descorrelatada do restante de sua distribuição e é possível rejeitá-la.

2.2.4.2. Fronteira de Decisão e Zona de Risco

Um problema dessa técnica é que pontos cuja dificuldade de classificação seja grande, podem acabar sendo confundidos com ruídos por estarem mais expostos ao erro de classificação. Assim, pontos relevantes pertencentes a fronteira de decisão ou a zona de risco podem acabar sendo eliminados do processo de treinamento dependendo dos valores apresentados durante o processo de cálculo de ρ. Independente do modelo escolhido, os pontos da fronteira de decisão são muito importantes porque eles que vão delinear a fronteira. Dentro do contexto de LVQ, os pontos pertencentes à zona de risco também devem ser defendidos para que, a partir do final do treinamento e início da generalização, pontos que caiam nessa região estejam protegidos por protótipos de sua distribuição. Torna-se interessante, então, a utilização de algum recurso que procure distinguir entre pontos que sejam efetivamente ruídos e os relevantes.

2.3. Uma Proposta de Seleção de Dados para Modelos LVQ

A principal contribuição desta dissertação é propor uma adaptação do método proposto em [15] para LVQ, procurando selecionar dados representativos do conjunto de entrada para treinamento do modelo e, quando possível, eliminar parte do ruído das distribuições.

Aprendizado exaustivo requer sorteio aleatório de funções alvo, isto é, modelos. No caso de LVQ, sortear modelos significa protótipos aleatórios para aplicar ao conjunto de treinamento e realizar o cálculo de ρ.

A idéia inicial é, dado um conjunto de treinamento escolher funções alvo aleatórias pertencentes ao modelo de aprendizado que, a partir de um certo momento possam oferecer informações suficientes para o cálculo de ρ. Aplicando essa etapa no contexto de LVQ, temos que a partir de um conjunto de treinamento, protótipos serão sorteados para testar o erro de cada ponto e o erro do restante da distribuição. Nesse momento, a classificação para cada ponto é similar a vizinhos mais próximos, usando como única referência o protótipo sorteado. Dessa forma, é possível calcular a correlação entre os dois erros para cada ponto e, após examinar todo o conjunto, teremos cada elemento associado a um valor de correlação.

O primeiro problema a ser resolvido ao utilizar LVQ é determinar as regiões do espaço onde os protótipos serão sorteados, pois essa região faz parte do modelo. É claro que, dentro dessas regiões o sorteio será aleatório, mas é preciso especificá-las, caso contrário os modelos não dirão nada a respeito das distribuições que representam, prejudicando a tentativa de selecionar dados representativos do espaço de entrada.

Uma vez estabelecidas às regiões para sorteio de protótipos e calculada a correlação entre o erro de cada ponto e o erro do restante da distribuição, é necessário estabelecer uma heurística que determine o ponto de corte nesse vetor de correlações, ou seja, apresentar a partir de que valor de correlação os pontos devem ser mantidos e os restantes eliminados do treinamento.

2.3.1. Região de Sorteio de Protótipos

Determinar então a região para sorteio de protótipos passa a ser um dos problemas chave na aplicação dessa técnica para LVQ. Note que, como os

protótipos de cada distribuição serão sorteados dentro das regiões, é imprescindível que todos, ou pelo menos a maioria, dos pontos pertencentes a essa área façam sentido ao serem tomados como protótipos. Se a região estipulada para uma determinada distribuição A for mais distante dela do que a região de outra distribuição B, todos os pontos de A estarão sempre mais próximos aos protótipos sorteados para B. Dessa forma, a análise de ρ fica impossibilitada.

A forma utilizada nessa dissertação para calcular regiões de sorteio de protótipos é calcular a média de cada atributo das distribuições, somando e diminuindo a essa média três desvios padrões.

2.3.2 Heurística para Seleção de Dados

Estabelecer o corte ideal no vetor com os valores das correlações entre o erro de cada ponto e o erro geral da distribuição passa a ser o objetivo. Conforme colocado na seção 2.2.4, eliminar os pontos que possuam o valor de ρ menor que zero já pode ser suficiente para uma seleção eficiente de dados. Entretanto, pontos de difícil classificação como aqueles pertencentes à zona de risco, ou outros relevantes para a atualização dos protótipos podem ser erroneamente eliminados dependendo do valor de corte do vetor. Assim, uma heurística é apresentada, visando obter em alguns casos uma separação mais justa entre ruídos e pontos relevantes.

Descrição da Heurística

Primeiramente, calcula-se o histograma de ρ, que terá valores pertencentes ao intervalo [-1, 1] e, utilizando Janela de Parzen [13], [23], estima-se a densidade de probabilidade de ρ. Em geral, distribuições ruidosas apresentam uma densidade que parece ser bimodal. Na verdade, não se trata disso, mas sim, de duas distribuições diferentes, a distribuição dos pontos relevantes e a distribuição dos ruídos. O que queremos de fato é encontrar o ponto de encontro entre essas duas distribuições, que seria o ótimo bayesiano [13]. Encontrar o mínimo na

região central dessa função pode, então, retornar um valor muito próximo desse ótimo bayesiano.

Retira-se, então, 5% dos pontos em cada extremo para evitar que o mínimo apareça nessas regiões, e se estipula que o mínimo estará entre os cinco maiores pontos da distribuição a esquerda e os cinco maiores da distribuição a direita. O próximo passo é calcular o mínimo da função.

Uma regra é estabelecida: Se o mínimo da função for menor que zero esse é o ponto de corte. Caso o mínimo seja maior ou igual a zero, três análises podem ser feitas: ou o ponto de corte é o próprio mínimo, ou passa a ser zero ou então -0.2. No caso do ponto ser -0.2, o corte é mais conservador que em zero, possibilitando a eliminação de um número menor de ruído, mas em contrapartida, poupando mais pontos relevantes para o treinamento.

A decisão de estabelecer um limite superior para o mínimo da função ocorre por duas razões:

1) Proteger dados de distribuições não ruidosas: como a técnica é justamente para selecionar dados relevantes e, quando possível, eliminar ruído das distribuições, caso a procura por um mínimo da função se encaminhe para a parte de correlação positiva, o algoritmo pode sugerir a eliminação de dados pertinentes ao conjunto de treinamento apenas porque a correlação do erro desses pontos com os dados não era muito alta.

2) Quando uma distribuição possui ruídos de outras distribuições e ela mesma não possui ruído algum: Nesse caso, a região de sorteio de protótipos coincidirá com a região da própria distribuição. Assim, os ruídos de outras distribuições que ela contém terão um índice de erro altíssimo, fazendo com que a correlação se aproxime de zero. A função estimada após a geração do histograma deixará de ter a aparência bimodal e os ruídos se acumularão em torno do zero. Procurar o mínimo nessa região pode eliminar novamente uma série de pontos pertinentes. O exemplo 2 do capítulo 3 mostra esse problema na prática.

2.3.3. O Algoritmo Principal

A fase inicial do algoritmo é composta por dois passos e, nesse momento, não há nenhuma diferença da proposta de [15], a não ser pelo fato de que, trabalhando com LVQ, serão sorteados protótipos para o aprendizado exaustivo, portanto, uma região deve ser estabelecida. A partir do passo 3 a etapa de seleção começa, sendo gerado o histograma de ρ e verifica-se a separação entre ruídos e relevantes. Seja D = $\{x_i, y_i\}_{i=1}^{N}$ conjunto de treinamento disponível de tamanho N e z o número de sorteio de protótipos.

Passo 1: Aplicar heurística apresentada na seção 2.3.1 para gerar região de sorteio de protótipos.

Passo 2: Para cada $\{x_i, y_i\}_{i=1}^{N}$, faça

Enquanto número do sorteio $\leq$ z

Calcular erro x_i e erro do restante da distribuição

Calcular $\rho(x_i)$

Passo 3: Calcular o histograma de ρ no intervalo [-1, 1].

Passo 4: Estimar a densidade de ρ usando Janela de Parzen.

Passo 5: Aplicar heurística apresentada na seção 2.3.2 para escolha do mínimo da função ρ.

Passo 6: Se o mínimo for menor que zero,

ponto de corte = mínimo da função

Senão

ponto de corte = mínimo da função ou zero ou -0.2

O processo termina na determinação do ponto de corte, realizando, assim, a seleção de dados para o treinamento.