**Capitulo 3 – O Método**

**(Descrição do capítulo)**

**3.1. Sistema de seleção dinâmica**

O problema analisado neste trabalho baseia-se no procedimento clássico de seleção dinâmica de classificador (DCS), que é divido em três níveis [17]: (1) A geração do ensemble que define como os classificadores bases são gerados; (2) Região de competência que define a região na qual serão realizadas a busca pelos melhores classificadores; (3) Seleção dinâmica que define a regra que seleciona o classificador (*DCS*) ou o conjunto de classificadores (*DES*), gerados no primeiro nível baseado na informação extraída das regiões definidas no segundo nível.

O classificador ou o conjunto de classificadores selecionado no terceiro nível é usado para classificar o padrão de consulta. A figura a seguir mostra uma visão geral do sistema de seleção dinâmica de classificador modificado para o uso da seleção de conjunto de classificadores (*DES*), no terceiro nível.

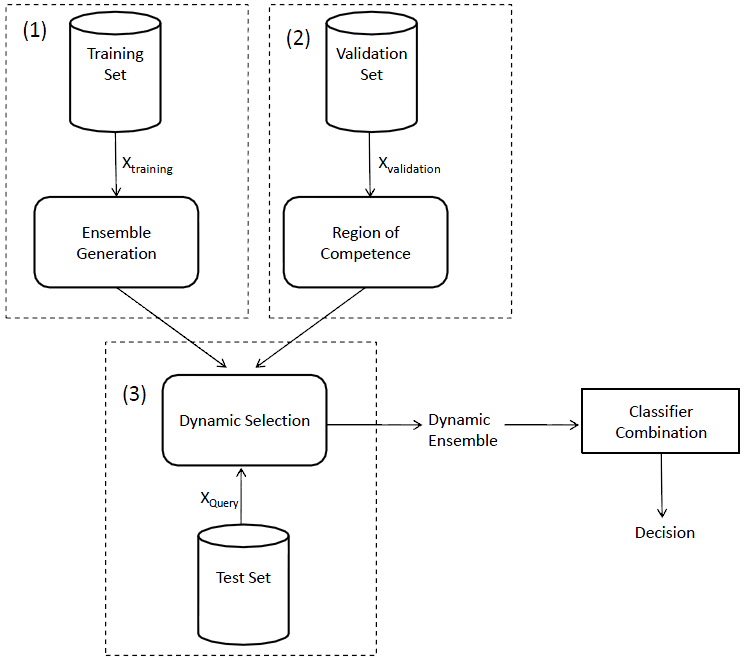


Figura 1. Visão geral do sistema de seleção dinâmica de classificador modificado para o uso de seleção dinâmica de ensemble.

Muitos estudos têm sido realizados no primeiro e no terceiro nível. No primeiro nível, os algoritmos mais utilizados são *bagging* [18], *boosting* [19] e *random subspace* [20]. No terceiro nível, Woods em [13] propôs o DCS-LA. Nesta técnica, a estimativa da precisão de cada classificador na vizinhança do padrão de teste é computada e o classificador com o melhor resultado é selecionado. No entanto, dado o fato que a seleção de um único classificador é muito propenso a erros, alguns trabalhos preferem a seleção de um subconjunto de classificadores. O KNORA proposto por Ko em [14] é uma das técnicas mais utilizadas para a seleção dinâmica de ensemble, porém falta uma investigação maior de outras abordagens para a seleção dinâmica de ensemble.

No segundo nível, pouca atenção tem sido dada a região de competência e como a qualidade da região influencia o resultado. A regra definida para a seleção de classificadores, no terceiro nível, depende da qualidade da informação na região de competência. A seleção dinâmica provavelmente deve falhar se existir muitos padrões de ruído na região de competência.

Portanto o objetivo deste trabalho é utilizar a abordagem DES-FA descrita por (Rafael), para o segundo nível e utilizar diferentes abordagens além do KNORA-E para o terceiro nível. Primeiro, será mostrado o método proposto por (Rafael) e como o desempenho de seleção dinâmica fica limitado pela qualidade da região de competência. Um exemplo prático é usado para ilustrar os casos quando o sistema de seleção dinâmica falha por causa de ruídos na região de competência. Depois, será mostrada uma abordagem DES utilizando uma adaptação dos métodos DCS-LA(OLA e LCA) baseado no método de seleção de características IWSS.

**3.2. Melhorando a qualidade da região de competência**

A abordagem proposta por (Rafael) sugere uma nova técnica de seleção dinâmica de ensemble que permite alcançar resultados mais precisos melhorando a qualidade das regiões de competência. Isto é feito usando duas estratégias: uma é um filtro que remove as amostras que são consideradas ruído, criando uma fronteira de decisão suave. A outra é uma versão adaptativa do algoritmo k-NN que usa pesos para indicar se um padrão está perto de padrões de classes diferentes ou não. O objetivo é eliminar os padrões de ruídos antes da execução da seleção dinâmica (terceiro nível), melhorando assim o desempenho geral do sistema.

Com intenção de mostrar a eficiência dessa abordagem será visto que o desempenho da técnica de seleção fica limitado pelo desempenho do algoritmo que cria a região de competência. Depois disso, será mostrado que essa abordagem não só aumenta a taxa de reconhecimento, como também diminui o tempo computacional, uma vez que torna mais fácil para o sistema selecionar os classificadores.

**3.2.1 Análise da influência da região de competência**

Para analisar a influência da região de competência no sistema de seleção dinâmica, foi utilizado o algoritmo KNORA-Eliminate visto que ele tem um desempenho ligeiramente melhor que outros algoritmos de seleção dinâmica [14]. Em seguida será visto um exemplo prático da influência da qualidade da região de competência.

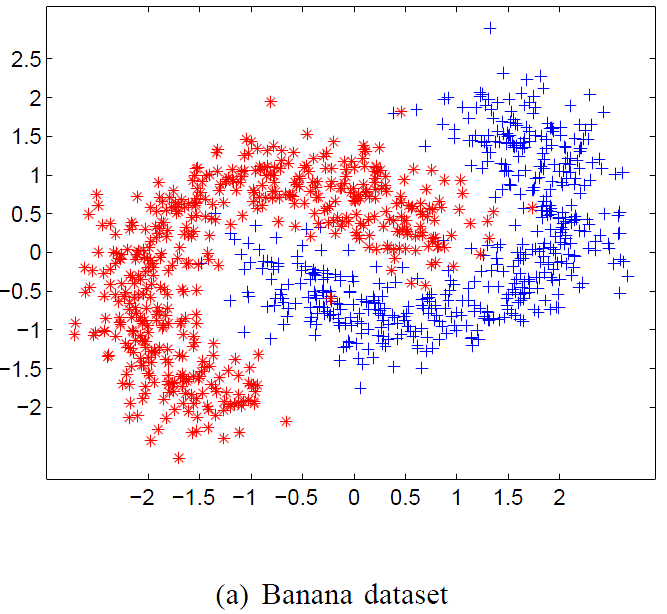
**3.2.1.1 KNORA-Eliminate**

Esta abordagem explora o conceito Oracle selecionando dinamicamente os classificadores. Dado que , , são os vizinhos mais próximos do padrão de teste e um ensemble de classificadores , , o ensemble dinâmico é composto pelos classificadores que classificam corretamente cada . Os classificadores que erram na classificação de qualquer um dos vizinhos são eliminados. Se nenhum classificador pode classificar corretamente todos os vizinhos, o valor de é diminuído e o método continua a procurar até pelo menos um classificador classificar corretamente todos os vizinhos. Se no final o algoritmo não encontrar nenhum classificador, todos os classificadores de são usados para dar a resposta final.

Uma vantagem deste método é que o número de vizinhos não é fixo, embora pode somente diminuir. No entanto, o custo da redução da vizinhança e o método para recalcular são computacionalmente caros. Como as outras técnicas de seleção dinâmica, este método é dependente da qualidade dos padrões vizinhos.

**3.2.1.2 Análise**

Para demonstrar o problema que as técnicas de seleção dinâmica têm com a qualidade da região de competência, será mostrado abaixo um experimento usando um ensemble de 10 *perceptrons* gerados usando o algoritmo *bagging*. Foi utilizado o valor da vizinhança . A figura 2 mostra os erros de classificação obtidos pelo KNORA-E para a base Banana. A figura 2 (a) mostra a forma dos dados da base Banana. A figura 2 (b) mostra os erros obtidos na base (em vermelho) e o conjunto de validação (em azul). O conjunto de validação é usado para computar a região de competência. A figura 2 (c) mostra alguns padrões da classe (em vermelho) que embora eles estejam mais próximos da média da classe, eles foram classificados erroneamente porque existe um padrão da outra classe entre eles. Este padrão está mais perto da média da média da outra classe do que a média da própria classe . Portanto, ele pode ser considerado um ruído.



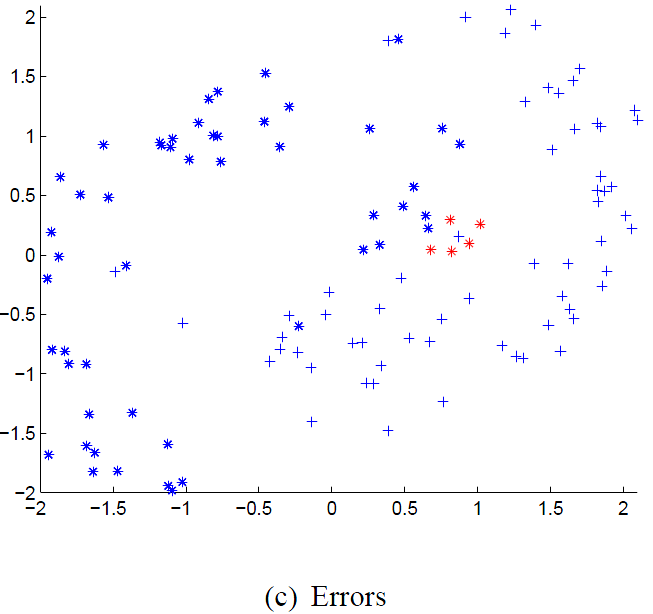
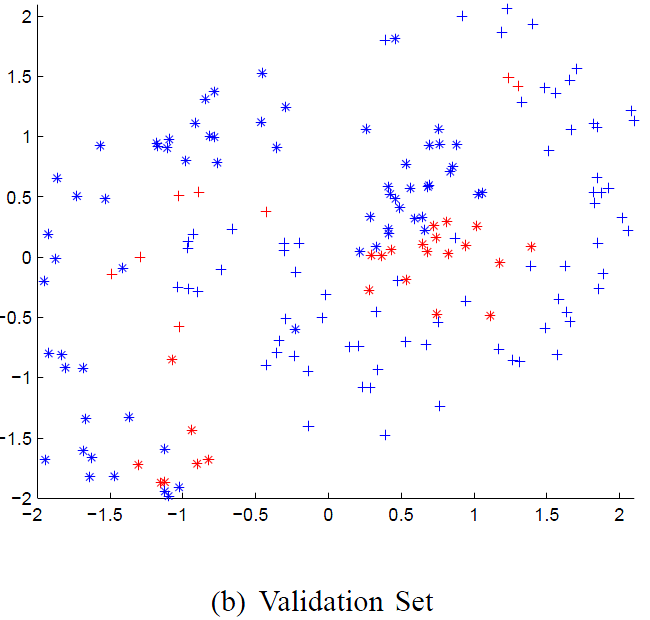


Figura 2. Problemas com a informação da vizinhança

Os sistemas de seleção dinâmica falham quando situações como esta acontecem. Quando há padrões ruidosos perto do padrão de consulta, os sistemas acabam selecionando os classificadores errados porque o classificador reconhece aqueles padrões de ruídos e, portanto alcançam uma maior precisão na vizinhança com probabilidade de ter *overfitting* na região. Isso explica porque os métodos de seleção são limitados ao desempenho do algoritmo que define a região de competência. Assim, se melhorar a qualidade dos padrões vizinhos, o desempenho do método de seleção dinâmica de classificador/ensemble também irá melhorar. Este é um ponto importante para melhorar a taxa de reconhecimento do sistema, que não recebeu muita atenção.

**3.2.2 A abordagem DES-FA**

Nesta seção serão descritas as ideias para melhorar a qualidade dos padrões vizinhos e consequentemente a seleção dinâmica. Duas técnicas foram usadas. Primeiro, é aplicado um filtro para a redução de ruído ao conjunto de validação (dados onde às regiões de competências são computadas) para remover os ruídos. Este passo é feito durante a etapa de treinamento. Depois disso, uma variação do algoritmo k-NN é usada a fim de melhorar a qualidade dos vizinhos computados. A figura 3 mostra uma visão geral do sistema proposto. é o conjunto de treinamento, o conjunto de validação e os dados de teste (generalização). Durante o estágio de treinamento, o ensemble é gerado usando conjunto de dados . O filtro *ENN* (*Edited Nearest Neighbor*) é aplicado aos dados de validação gerando o conjunto de dados , . O filtro *ENN* trabalha eliminando o ruído nas fronteiras de decisão. Portanto, o algoritmo produz uma fronteira de decisão suave.

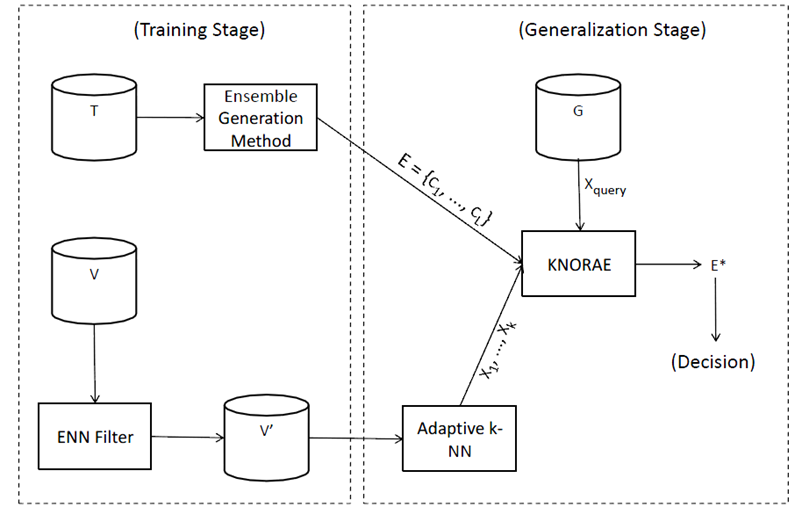


Figura 3. Visão geral do sistema DES-FA.

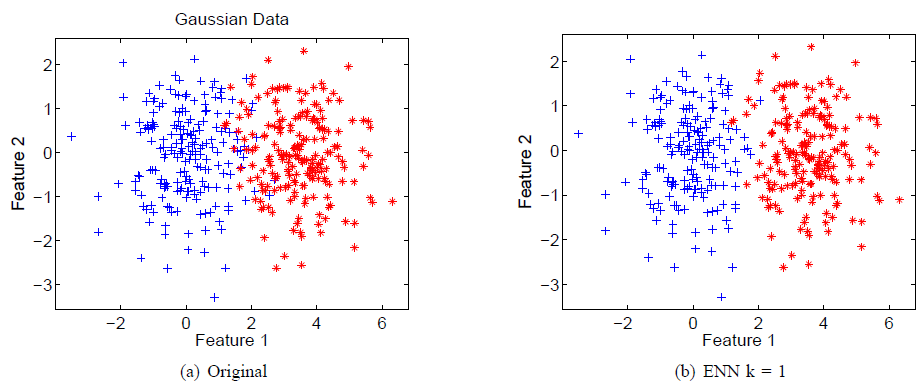
Na fase de teste, a região local é computada com o algoritmo do k-NN adaptativo [27] usando os padrões do conjunto de dados filtrado . O k-NN adaptativo é uma variação do k-NN tradicional que usa pesos para indicar quão próximo um padrão de treinamento está dos padrões de classes diferentes. O peso é usado para ter uma maior probabilidade de seleção dos padrões que estão distantes da fronteira. Assim, os padrões com maior probabilidade de ser ruído são menos prováveis de serem escolhidos. No terceiro nível é utilizado o método KNORA-Eliminate para a seleção dinâmica do ensemble usando a região de competência definido pelo algoritmo k-NN adaptativo. Esse método é chamado de DES-FA (Dynamic Ensemble Selection by Filter + Adaptive distance). O filtro ENN e o k-NN adaptativo são descritos a seguir.

**3.2.2.1 ENN**

O método ENN (*Edited Nearest Neighbor Filter*) funciona como um filtro de redução de ruído com intenção de criar os limites da classe mais suave. Os pontos centrais das classes são preservados. O algoritmo 1 mostram os passos do algoritmo ENN. O algoritmo funciona da seguinte forma: Seja o conjunto de treinamento, e o conjunto filtrado, o algoritmo realiza a classificação do vizinho mais próximo para cada usando como referência. Se é classificado errado usando o algoritmo k-NN, este é considerado um ruído e é removido do conjunto final .

|  |
| --- |
| **Algoritmo 5** ENN |
| **Entrada:** conjunto de treinamento .  Saída: conjunto filtrado .   1. **para** cada  **faça** 2. se então |

A figura 5 mostra um exemplo da aplicação do ENN. Os dados foram construídos usando duas distribuições Gaussianas geradas com , e . A figura 5 (a) mostra a distribuição original. A figura 5 (b), (c) e (d) apresentam o resultado depois da execução do algoritmo ENN com respectivamente.



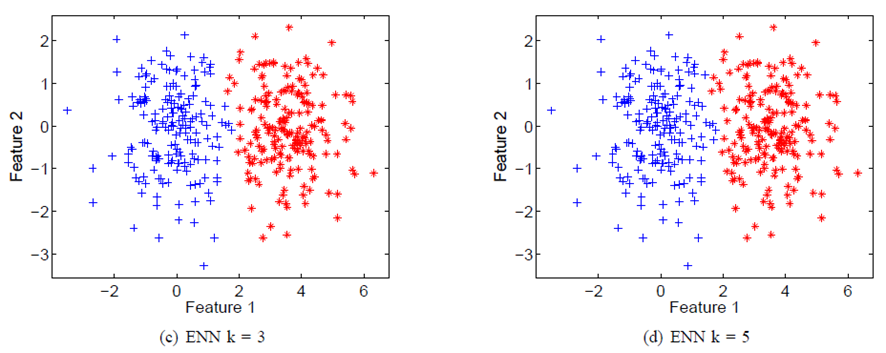


Figura 4. Resultado do algoritmo ENN para duas distribuições gaussianas.

**3.2.2.2 K-NN com distância adaptativa**

A distância adaptativa [27] calcula, para cada amostra de treinamento , a maior esfera centrada em , , que exclui cada padrão de treinamento de classes diferentes , . Esta é realizada pelo cálculo da distância mínima (raio da esfera) entre o padrão de treinamento e as amostras de treinamento das classes diferentes (Eq. 1). Com o raio , a distância adaptativa entre o padrão de teste e é definida pela (Eq. 2). A distância pode ser qualquer distância, tal como, a distância Euclidiana ou a Manhattan.

Usando este método, amostras próximas da média de sua classe tem o maior raio do que as amostras que estão perto das fronteiras da classe. Assim, amostras que estão mais próximas dos limites da classe tornam-se mais distantes para o padrão de consulta enquanto as próximas a média da classe torna-se mais perto. Portanto, a probabilidade de selecionar um ruído como vizinho é menor.

A ideia por trás de usar as técnicas do filtro ENN e o k-NN adaptativo vem do fato que eles reduzem o número de padrões indesejados na região de competência. No entanto, não é garantido que o ENN irá eliminar todos os padrões indesejáveis. O k-NN adaptativo funciona da forma que o padrão mais próximo dos limites de decisão e, portanto, mais provável de ser ruído tem menos chance de ser selecionado. Portanto, mesmo se um padrão indesejado não foi eliminado usando o ENN, a probabilidade de selecionar este padrão usando o k-NN adaptativo é menor. Assim, é interessante o uso de ambas as técnicas em que uma pode superar a limitação da outra.

**3.3. Abordagem DES utilizando os métodos DCS-LA**

Como visto anteriormente a abordagem DES-FA sugere modificações no nível 2 do sistema de seleção dinâmica a fim de melhorar as regiões de competência. Na descrição do DES-FA, foi utilizado o KNORA-Eliminate para a seleção dinâmica de ensemble no nível 3, visto que este tem um desempenho melhor que outros métodos de seleção dinâmica. Com o objetivo de explorar outros métodos para a seleção dinâmica de ensemble além do KNORA, nesta seção serão apresentadas novas abordagens para a seleção dinâmica de ensemble a partir das técnicas DCS-LA.

**3.3.1 O DCS-LA**

O método DCS-LA utiliza o conceito da estimativa da precisão local “*Local Accuracy Estimates*” para fazer a seleção dinâmica de classificador (DCS). A ideia é estimar a precisão de classificação de cada classificador em regiões locais definidas pelo k-NN no conjunto de treinamento, para a descrição original do método, ou para o conjunto de validação, de acordo com o sistema de seleção dinâmica descrito anteriormente. A estimativa da precisão local pode ser realizada de duas maneiras: usando a precisão local total (OLA) ou a precisão local da classe (LCA). O OLA computa percentual de acerto dos vizinhos da amostra de teste para cada classificador. O LCA computa o percentual de classificações corretas realizados pelo classificador com relação à classe de saída da amostra de teste dada por esse classificador. Da mesma forma. Em ambos os métodos o classificador que tiver a maior precisão local é selecionado.

Como visto esse é um método DCS, ou seja, seleciona apenas um classificador. No entanto, a seleção de um único classificador pode tornar a classificação mais propensa a erros. Sendo assim, a seguir será descrita uma proposta de utilização dos métodos DCS-LA (OLA e LCA) para a seleção dinâmica de ensemble (DES).

Para entender a ideia da abordagem do método DES proposto a seguir, primeiro é preciso saber que os métodos OLA e LCA descritos acima usam uma abordagem de ranking de classificadores. O ranking é construído ordenando os classificadores pela estimativa da precisão local, em que o classificador com a maior precisão local estará no topo do rank. Portanto o método original DCS-LA seleciona apenas o primeiro classificador do ranking. Sendo assim, uma adaptação simplificada dos métodos DCS-LA para um DES poderia ser simplesmente selecionar os primeiros classificadores do ranking, com . No entanto, essa abordagem não é robusta o suficiente para a seleção dinâmica de ensemble, podendo não dar bons resultados. Para entender melhor porque essa ideia pode falhar, na próxima seção será feita uma analogia da seleção de classificadores com a seleção de características ou *FSS (Feature Subset Selection)*. Em seguida será explicado o método *IWSS* para a seleção de atributos para então apresentar o método proposto.

**3.3.2 Analogia DES com *FSS***

A seleção subconjunto de características (*FSS*) apresenta algumas semelhanças com a seleção de classificadores como, por exemplo, melhorar a precisão de classificação selecionando os membros de um conjunto original em um subconjunto. Na seleção de características, os atributos são os membros do conjunto, e a importância da seleção é reduzir a quantidade de atributos, consequentemente reduzindo a base, mantendo apenas os atributos que em conjunto apresentem um bom desempenho. Assim, a seleção de características visa eliminar os atributos redundantes ou indesejáveis sem prejudicar o desempenho de classificação.

Não há uma garantia que selecionar apenas os atributos que são considerados bons ou discriminantes individualmente seja a melhor solução de subconjunto. Pode acontecer que um subconjunto de características que tenha um atributo que é fraco ou pouco discriminante individualmente tenha um desempenho melhor que um subconjunto apenas com os melhores atributos individuais. Isso ocorre devido ao atributo fraco tornar-se relevante dado outro atributo. Portanto, não se pode afirmar que um atributo que seja ruim ou pouco discriminante não seja selecionado para o subconjunto de características.

Da mesma forma, para a seleção de classificadores o melhor subconjunto de classificadores para classificar uma amostra de teste , não necessariamente terá que conter apenas os melhores classificadores que classificam os seus vizinhos. Pois, um classificador que é fraco, também pode se tornar relevante para o subconjunto de classificadores. Dessa forma, a busca pelo subconjunto de classificadores deve tomar outra abordagem do que apenas apostar nos melhores classificadores computados pelos métodos OLA e LCA.

**3.3.3 IWSS**

Em problemas de FSS, devido à alta cardinalidade () do espaço de busca, a busca exaustiva é intratável mesmo para valores moderados de atributos, então diferentes estratégias de busca são empregadas para esse problema. Os algoritmos de *FSS* resultam da combinação de (1) um método de busca e (2) uma estratégia de avaliação para pontuar a importância dos atributos candidatos.

A avaliação das características pode ser realizada na forma de filtro ou *wrapper*. Na abordagem do filtro, a importância de um atributo ou conjunto de atributos é estimada pelas propriedades intrínsecas dos dados, enquanto que na abordagem *wrapper* a importância de um determinado subconjunto é obtida através da aprendizagem e avaliação de um classificador usando apenas as variáveis incluídas no subconjunto proposto.

Uma vez que os subconjuntos candidatos são pontuados (usando um filtro ou *wrapper*), um grande número de estratégias de busca pode ser usado para procurar um subconjunto (quase) ideal, entre elas a está a estratégia incremental. Esta estratégia utiliza um rank, medida de filtro, e avalições dos subconjuntos candidatos na forma de *wrapper*.

O IWSS (Incremental Wrapper Subset Selection) (Ruiz et al., 2006; Flores et al., 2008) é um método que utiliza a estratégia incremental para a seleção de subconjunto de características. Este método funciona em duas etapas:

* *Filtro:* é avaliado cada variável independentemente com relação à classe, a fim de criar um ranking. A incerteza simétrica é geralmente a medida de pontuação utilizada para o ranqueamento do atributo de acordo com sua importância.
* *Wrapper:* o subconjunto selecionado é inicializado com o primeiro atributo do ranking, então o algoritmo tenta iterativamente incluir em o próximo atributo do ranking pela avaliação do desempenho do subconjunto aumentado . A avaliação dos subconjuntos candidatos é feito na forma de wrapper, e se uma diferença positiva é obtida, então é adicionado ao subconjunto , caso contrário é descartado.

Para exemplificar o algoritmo *IWSS*, assuma um problema com oito atributos . Supondo que os seguintes valores são computados no passo do *filtro*: , , , , , , e . Portanto o ranking do filtro é . Então a execução da fase do wrapper é:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Passo | Atributo testado | Subconjunto testado | Precisão de classificação | Decisão | Subconjunto resultante |
| 1 |  |  | 0.6 | Aceito |  |
| 2 |  |  | 0.7 | Aceito |  |
| 3 |  |  | 0.68 | Rejeitado |  |
| 4 |  |  | 0.71 | Aceito |  |
| 5 |  |  | 0.71 | Rejeitado |  |
| 6 |  |  | 0.65 | Rejeitado |  |
| 7 |  |  | 0.7 | Rejeitado |  |
| 8 |  |  | 0.75 | Aceito |  |

Tabela 1. Iterações wrapper *IWSS*

E portanto, é o subconjunto selecionado.

**3.3.4 O método DES proposto**

Nesta seção será descrito a abordagem utilizada para adaptação dos métodos DCS-LA (OLA e LCA) para a seleção dinâmica de ensemble (DES) baseado no método de seleção de subconjunto de características IWSS.

Como visto o OLA e LCA são métodos baseado na abordagem de ranking de classificadores. Onde o classificador que está em primeiro no ranking tem a maior estimativa da precisão local. Analisando *IWSS* pode perceber claramente que é facilmente adaptável para o problema de seleção dinâmica de classificadores.

O IWSS é executado em duas fases: (1) filtro, construção do ranking, e (2) wrapper, avaliação dos subconjuntos. Para adaptar os métodos DCS-LA, o método proposto utiliza o OLA ou o LCA para a fase 1 do IWSS. Assim o método DCS-LA irá construir o ranking dos classificadores ordenando de acordo com a maior estimativa da precisão local para uma amostra de teste . Para a fase 2, wrapper, o subconjunto é inicializado com o classificador do ranking com a maior precisão local, primeiro do ranking. Então iterativamente o algoritmo adiciona o próximo classificador ao subconjunto e realiza a avaliação do subconjunto de classificadores para estimar a precisão local do novo subconjunto. Caso a inclusão do novo classificador ao subconjunto aumente a precisão local com relação à , então ele é adicionado ao subconjunto final , caso contrário ele é descartado e o próximo classificador do ranking será avaliado na próxima iteração do wrapper. No final da execução do método, o algoritmo terá como resultado um subconjunto com classificadores, onde é o total de classificadores do ensemble inicial.

Para avaliar o método proposto, também foram investigadas duas outras abordagens para o método LCA, que foram chamadas de LCA2 e LCA3, pois o LCA só analisa o percentual de acerto que o classificador classificou como sendo da classe de saída do padrão e não investiga o percentual do acerto da classe em si. Essa diferença pode ser esclarecida pela figura a seguir:

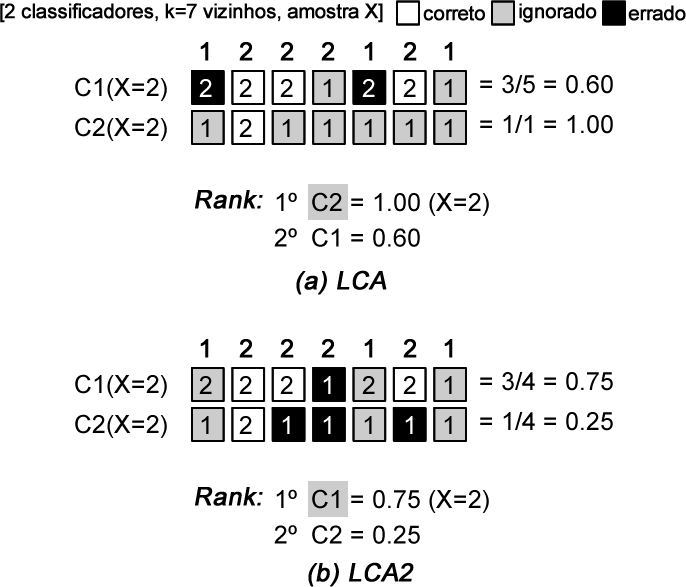


Figura 5. Exemplo da diferença do LCA (a) e o LCA2 (b) proposto. O LCA3 é apenas uma média das estimativas do LCA e LCA2.

Pode ser visto na figura que apesar do classificador C1 acertar mais exemplos da classe 2 que o classificador C2, a precisão local de C2 é maior que a de C1, pois como o classificador C2 classificou apenas uma amostra como sendo da classe 2, e o classificou corretamente. Assim, C2 obteve uma precisão local de 100%. Enquanto que o classificador C1, que classificou 5 amostras como sendo da classe 2, porém acertou 3, teve uma precisão local de 60%. Dessa forma segundo o LCA, o classificador C2 é mais preciso localmente que o classificador C1.

Sendo assim, o LCA2 sugere ver o outro lado da estimativa da classe, ou seja, do total de vizinhos que são da mesma classe do padrão de teste, quantos deles o classificador classificou corretamente. E por fim, o LCA3 utiliza ambas as abordagens fazendo uma média das duas precisões locais, do LCA e LCA2.