Universidade Federal de Pernambuco
Centro de Informática

Graduação em Engenharia da Computação

Método para Seleção Dinâmica de Conjunto de Classificadores

Henrique Alexandre de Menezes Sabino Almeida

Trabalho de Graduação

Recife
13 de dezembro de 2011

Universidade Federal de Pernambuco
Centro de Informática

Henrique Alexandre de Menezes Sabino Almeida

Método para Seleção Dinâmica de Conjunto de Classificadores

Trabalho apresentado ao Programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Orientador: *George Darmiton da Cunha Cavalcanti*

Recife
13 de dezembro de 2011

A Deus, meus pais e meus irmãos.

Agradecimentos

Agradeço a Deus pelas coisas boas da minha vida e por me dar força e fé nos momentos difíceis. Agradeço a Murilo e Zélia, meus pais, pelo amor incondicional, por me educar e pelo apoio ao longo da vida. Agradeço aos meus queridos irmãos, Carol e Murilinho, pelos bons momentos vividos juntos. Agradeço ao meu orientador, George, pelo aprendizado e a oportunidade de conhecer essa área. Agradeço aos meus amigos e colegas de curso que nesses cinco anos fizeram parte da minha vida, compartilhando momentos bons e difíceis da faculdade. Agradeço aos meus familiares, e por fim agradeço aos meus amigos da minha cidade natal Floresta-PE.

Tudo o que um sonho precisa para ser realizado
é alguém que acredite que ele possa ser realizado.
—ROBERTO SHINYASHIKI

Resumo

O uso de sistemas de múltiplos classificadores tem sido útil para alcançar altas taxas de reconhecimento. A seleção dinâmica de classificadores é uma abordagem de combinação de múltiplos classificadores que seleciona um subconjunto de classificadores mais adequados para classificar um padrão de consulta. Neste trabalho propomos adaptações dos métodos *DCS-LA* (*OLA* e *LCA*) para a seleção dinâmica de ensemble (*DES*) a fim de comparar com a abordagem *KNORA*. Para isso, fizemos um estudo da influência da região de competência nesses métodos, utilizando a abordagem *DES-FA*. Os resultados obtidos mostram que as adaptações dos métodos foram satisfatórias comparadas com o *KNORA-E* e que seu custo computacional é bastante atrativo.

Abstract

The use of multiple classifier systems has been useful to achieve high recognition rates. The dynamic ensemble selection is an approach of combining multiple classifiers that selects a subset of classifiers more appropriate to classify a query pattern. In this work we propose adaptations of DCS-LA methods (OLA and LCA) to dynamic ensemble selection (DES) in order to compare with the KNORA approach. For this, we studied the influence of the region of competence in these methods, using the DES-FA approach. The results show that the adaptations of the methods were satisfactory compared with the KNORA-E and that its computational cost is very attractive.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

1.1 Contextualização

Para que seja possível a criação de diversos sistemas computacionais inteligentes, como a verificação de assinatura *off-line*, reconhecimento de dígitos, reconhecimento de fala, entre outros sistemas, é necessário o uso de técnicas de aprendizagem de máquina/reconhecimento de padrões.

Segundo o teorema "*No Free Lunch*", não existe um único classificador que seja melhor que os outros classificadores para todos os problemas. Assim, quando não se tem conhecimento das distribuições dos dados do problema, não se pode afirmar que um classificador é em média melhor que outro. Isso torna difícil a tarefa de encontrar um único classificador que melhor resolve o problema. Muitos estudos têm mostrado que problemas de classificação são mais precisos quando é usado uma combinação de classificadores ao invés de um classificador individual. Por exemplo, diversos classificadores "fracos" ao serem combinados são capazes de superar um classificador individual específico para o problema.

Portanto, o uso de sistemas de múltiplos classificadores (*MCS - Multiple Classifier Systems*) ou conjunto de classificadores (*EoC – Ensemble of Classifiers*) ou ainda "*ensemble learning*" tem sido útil para aumentar a precisão de classificação (melhorar as taxas de reconhecimento). Isso é possível devido à combinação das vantagens individuais dos classificadores em uma solução final. Essa ideia é bastante intuitiva uma vez que imita a natureza humana em buscar opiniões de diversas fontes a fim de ter uma opinião/decisão melhor.

Existem duas principais abordagens para a combinação de múltiplos classificadores: fusão de classificador e seleção de classificador. Nas técnicas de fusão,

cada classificador é usado e suas saídas são agregadas através de uma função (votação majoritária, soma, produto, máximo, mínimo) ou até mesmo por meio de outro classificador final. Essa abordagem baseia-se no pressuposto que os erros de classificação são independentes para cada classificador. No entanto, não há garantia que um método particular de geração de classificadores (*bagging*, *boosting*, *random subspaces*, etc) vai conseguir independência de erro. Quando a condição de independência não se verifica, não se pode assegurar que a fusão dos classificadores irá melhorar o desempenho da classificação final.

A abordagem de seleção de classificador baseia-se no princípio de regiões de competência, na qual acredita-se que um classificador ou um conjunto de classificadores sejam os mais competentes para classificar uma região. Portanto, é conhecido como seleção de classificador (*CS - Classifier Selection*), a seleção feita de um único classificador para dar a resposta final. E seleção de ensemble (*ES - Ensemble Selection*), quando um conjunto de classificadores é selecionado e suas saídas são combinadas por meio de fusão para a resposta final.

Estes métodos de combinação de classificadores podem ser estáticos (mesma combinação para cada padrão de consulta) ou dinâmicos (a combinação depende do padrão de consulta). Porém como diferentes padrões de teste, em geral, são associados a diferentes dificuldades de classificações com bases nas características de cada padrão, muitos estudos têm mostrado que a seleção dinâmica obtém melhores resultados que a seleção estática. E como a seleção de um único classificador é muito propenso a erros, muitos pesquisadores têm focado em métodos de seleção dinâmica de ensemble (*DES – Dynamic Ensemble Selection*) ao invés de métodos de seleção dinâmica de classificador (*DCS – Dynamic Classifier Selection*). Porém, muitos dos métodos *DES* são bastante influenciados pelos métodos *DCS*, assim como o *KNORA* (*K-nearest-oracles*) método *DES* que usa os mesmos conceitos de métodos *DCS* como o *DCS-LA* (*Dynamic Classifier Selection by Local Accuracy*).

1.2 Objetivo

Este trabalho tem como objetivo desenvolver um método de seleção dinâmica de classificadores (*DES*) com o intuito de melhorar as taxas de classificação e analisar o seu desempenho com relação a outros métodos de seleção.

1.3 Estrutura

A sequência do trabalho é dividida da seguinte forma: o Capítulo 2 terá o estado da arte, com uma revisão da literatura e a descrição dos métodos de geração, seleção e combinação de classificadores; no Capítulo 3 é descrito o método de seleção proposto; o Capítulo 4 contém os dados dos resultados obtidos dos experimentos e das bases utilizadas, e apresenta a conclusão do trabalho.

CAPÍTULO 2

Estado da Arte

2.1 Revisão da Literatura

A combinação de opiniões especialistas é um tema estudado desde a metade do século XX. No início, os estudos eram voltados para aplicações como economia, democracia e decisões militares. Um dos primeiros modelos de aprendizagem de um sistema com múltiplos especialistas é a arquitetura *Pandemonium*, descrita por Oliver Selfridge em 1959. Posteriormente, alguns estudos importantes foram realizados sobre a combinação de especialistas usando diferentes termos e abordagens, mas a área de sistemas de múltiplos classificadores (*MCS – Multiple Classifier Systems*) ou *ensemble learning* tornou-se popular a partir da década de 1990, (PONTI-JR, 2011).

Hansen e Salamon (1990) mostraram a propriedade de redução da variância em um sistema de ensemble, e que o desempenho de generalização de uma rede neural pode ser melhorado usando um conjunto de redes neurais de configuração semelhante (HANSEN e SALAMON, 1990). Mas foi o trabalho de Schapire que colocou os sistemas de *ensemble* no centro das pesquisas de aprendizagem de máquina, quando ele provou que um classificador forte no sentido provavelmente aproximadamente correto (*PAC – Probably Approximately Correct*) pode ser gerado pela combinação de classificadores fracos através de um procedimento que ele o chamou de *boosting* (SCHAPIRE, 1990).

Esses trabalhos foram fundamentais para que as pesquisas em *ensemble learning* tenham expandido rapidamente e foram categorizados como sendo de seleção ou fusão, caso os classificadores sejam selecionados ou se suas saídas são combinadas, respectivamente. Lembrando que esses dois métodos, seleção e fusão, podem ser combinados.

Nos trabalhos de fusão a preocupação se restringe a duas etapas: i) procedimento de geração dos classificadores individuais ou ensemble. ii) estratégia empregada para combinação dos classificadores. Assim diversos métodos de geração de ensemble surgiram como: *bagging* (BREIMAN, 1996), *boosting* (SCHAPIRE, 1990), *AdaBoost* (FREUND e SCHAPIRE, 1996), *Random Subspace Method* (HO, 1998). Esses métodos de geração têm como objetivo construir um bom ensemble, onde os classificadores sejam precisos e diversos quanto possíveis. Para a segunda etapa, a combinação pode ser aplicada às classes de saída ou a um conjunto de valores contínuos para cada classe específica resultante da classificação de um especialista individual do ensemble (KITTLER, HATEF, *et al.*, 1998). Neste último caso, as saídas são frequentemente normalizadas no intervalo [0, 1], e esses valores são interpretados como um suporte dado pelo classificador para cada classe. Tal interpretação permite a aplicação de várias regras algébricas de combinação (votação majoritária, máximo, mínimo, soma, produto e outras combinações de probabilidades a posteriori) ((KITTLER, HATEF, *et al.*, 1998), (KUNCHEVA, 2002), (ROLI e GIACINTO, 2002)), e mais recentemente, modelos de decisão (*decision templates*) (KUNCHEVA, BEZDEK e DUIN, 2001).

Porém, vários estudos têm investigado bastante o uso da abordagem de seleção de classificador como uma opção ao invés da fusão, pois esta exige a independência de classificadores. Como resultado, vários métodos de seleção dinâmica (*DCS* e *DES*) têm sido propostos na literatura como visto a seguir.

Em 1997, Woods propôs em (WOODS, KEGELMEYER JR. e BOWYER, 1997) uma abordagem para seleção dinâmica de classificador, o *DCS-LA* (*Dynamic Classifier Selection by Local Accuracy*), baseado no conceito "*Local Accuracy Estimates*", em que a precisão local de cada classificador em relação a um padrão de teste é estimada e o classificador com a maior precisão local é selecionado. Woods ainda propôs dois métodos para estimar a precisão local (local accuracy): *OLA* (*Overall Local Accuracy*), e o *LCA* (*Local Class Accuracy*). Em que o *OLA* é uma simples precisão de classificação ao redor de uma amostra de teste para cada classificador, e o *LCA* é similar ao *OLA*, só que a precisão local é com respeito as classes de saída.

Já Giacinto propôs em (GIACINTO e ROLI, 1999) um *framework* para a seleção dinâmica de classificadores usando os métodos de seleção *A Priori* e *A Posteriori* e um

algoritmo baseado em seu *framework* proposto. Esses métodos de seleção são baseados na estimativa da probabilidade da classificação correta em uma região local do espaço de características em torno do padrão de teste desconhecido. Portanto, assim como os métodos de Woods baseados na estimativa da precisão local, o classificador que tiver a maior probabilidade de classificação correta sobre a amostra de teste é selecionado, em que o método *A Priori* não usa a informação da classe atribuída pelo classificador ao padrão de teste, e *A Posteriori* usa a informação da classe.

Uma maneira de obter o máximo de desempenho de um *EoC* é através do conceito *Oracle*, em que temos um seletor ótimo ou ideal para um conjunto de classificadores, ou seja, o *Oracle* classifica corretamente um dado padrão se existe algum classificador do *EoC* que o classifica corretamente, assim o *Oracle* é o limite superior do desempenho de um *EoC*.

A partir do conceito *Oracle*, Ko propôs em (KO, SABOURIN e JR., 2008) a abordagem *KNORA* (*K-nearest-oracles*). O *KNORA* é similar aos conceitos *OLA*, *LCA*, *A Priori* e *A Prosteriori* em considerar a vizinhança do padrão de teste, mas difere destes pelo uso direto de sua propriedade tendo as amostras de treinamento na região com a qual encontra o melhor ensemble para uma determinada amostra. Para uma amostra de teste, o *KNORA* encontra os seus *k* vizinhos mais próximos no conjunto de validação, descobre quais os classificadores que classificam corretamente esses vizinhos no conjunto de validação e usa-os como um ensemble para classificar o padrão de teste fornecido. Ko ainda propôs quatro diferentes métodos para a abordagem *KNORA*: *KNORA-E* (*KNORA-Eliminate*), *KNORA-U* (*KNORA-Union*), e suas versões ponderadas, *KNORA-E-W* e *KNORA-U-W*. A partir desta abordagem, algumas variantes *KNORA* surgiram posteriormente.

Outro trabalho notável foi o de Dos Santos, que em (DOS SANTOS, SABOURIN e MAUPIN, 2008) propôs uma estratégia dinâmica de *OCS* (*Overproduce-And-Choose Strategy*), que combina otimização e seleção dinâmica em uma fase de seleção de dois níveis. Em que no nível de otimização é realizado uma busca multi-objetiva através de algoritmos genéticos, e no nível de seleção dinâmica é baseada em medidas de confiança, ao invés da precisão do classificador nas regiões de competência como nos métodos de seleção tradicionais.

Por fim, existem outras abordagens que podem ser usadas para melhorar o desempenho de *EoC*, como as estratégias que visam melhorar a geração dos classificadores bases (DUIN, 2002) ou aprimorar as regiões de competência através da eliminação de ruídos por meio de filtro e distância adaptativa (CRUZ, CAVALCANTI e REN, 2011).

2.2 Geração de Ensemble

A intuição diz que em um bom ensemble os classificadores devem ser precisos o quanto possível e diversos o quanto possível. Sabe-se que quanto mais precisos são os classificadores menores são as taxas de erros que eles apresentam para novas entradas (HANSEN e SALAMON, 1990). No entanto, a diversidade não é um conceito preciso (BROWN, WYATT, *et al.*, 2005). Definir este conceito e como isto influencia o desempenho não é trivial.

Um ponto de consenso é que, quando os classificadores cometem erros estatisticamente independentes, a combinação tem o potencial de aumentar o desempenho do sistema. Se o conhecimento sobre a diversidade de um ensemble for boa, isso torna possível escolher melhor o método de fusão. Por exemplo, sabe-se que simples fusões podem ser usadas para classificadores com uma simples complementação de padrões, mas que fusões complexas são necessárias para classificadores com um modelo de dependência complexa (PONTI-JR, 2011).

A ligação como às medidas de diversidade estão relacionadas com a precisão do ensemble é um assunto de investigação ainda. No entanto, estudos experimentais observaram os resultados esperados: quanto maior diversidade maior o desempenho. Eles também mostraram que as propriedades de um ensemble que são desejáveis (alta precisão e diversidade) para obter uma combinação de sucesso não são comuns na prática (PONTI-JR, 2011).

Um exemplo didático é mostrado na **Figura 1** onde 3 classificadores (com 0.6 de precisão individual) tem que classificar 10 objetos desconhecidos. A figura mostra quatro casos: um caso estatisticamente independente, um caso com classificadores idênticos e dois casos de dependência de classificadores. Por este exemplo é claro que a independência contribui para o aumento do desempenho. Quando a dependência é

assumida, podemos ter classificação correta ou incorreta da dependência, dando resultados muito diferentes. Avaliar a diversidade de um classificador também não é uma tarefa trivial, já que o conceito em si não é claro, mas existem medidas *pairwise* que podem ser usadas.

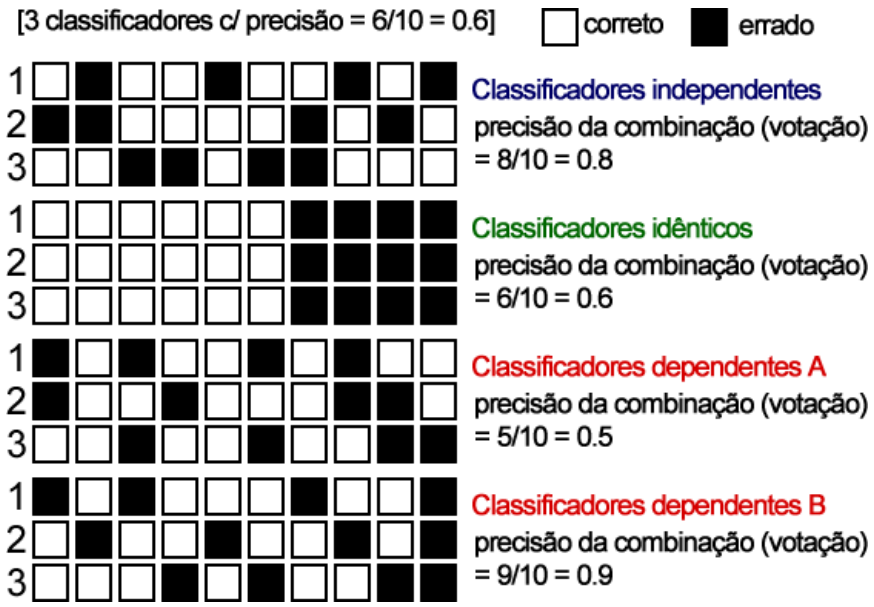


Figura 1 Exemplo de diversidade de ensemble. Quatro casos indicando o possível desempenho, mostrando que classificadores independentes aumenta potencialmente o desempenho individual, e dois casos de dependência de classificador com resultados diferentes.

Portanto, as diversas abordagens de geração de classificadores visam criar bons classificadores (precisos e diversos) para o ensemble a fim de terem um melhor desempenho em sua combinação. Alguns dos métodos mais populares são vistos a seguir.

2.2.1 Bagging

O *Bagging* (BREIMAN, 1996), um acrônimo para *Bootstrap AGGregatING*, é um dos primeiros, mais intuitivos, e talvez um dos mais simples algoritmos baseado em ensemble, com um bom desempenho. O *bagging* é baseado na ideia que amostras de *bootstrap* do conjunto de treinamento original irão apresentar uma pequena mudança com respeito ao conjunto de treinamento original, mas uma diferença suficiente para produzir classificadores diversos. Assim, para o *bootstrap* os diferentes subconjuntos de dados de treinamento são sorteados randomicamente - com reposição - do conjunto de

dados de treinamento inteiro. Depois cada subconjunto de dados de treinamento é usado para treinar um classificador base diferente. E por fim os classificadores individuais são então combinados pela média ou votação majoritária de suas decisões. O pseudocódigo do algoritmo pode ser visto abaixo:

Algoritmo 1 Bagging

Entrada: tamanho L do ensemble; conjunto de treinamento S de tamanho N.

Inicialização

1. $E = \emptyset$, ensemble

Fase de Treinamento

2. **para** i = 1 até L **faça**
3. $S_i \leftarrow$ Sorteio de amostras de S, com reposição. *bootstrap*
4. Constrói o classificador h_i usando S_i
5. Adiciona o classificador ao ensemble $E = E \cup h_i$

Fase de Classificação

6. **para** cada novo padrão **faça**
7. **se** se as saídas são contínuas **então**
8. Combine as saídas dos classificados de E pela média
9. **senão se** as saídas são os rótulos das classes **então**
10. Combine as saídas dos classificados de E pela votação majoritária

Apesar do *bagging* ser um bom algoritmo, como à diversidade dos classificadores é obtida usando várias "réplicas" *bootstrap* do conjunto de treinamento, o *bagging* só é eficaz quando uma pequena mudança no conjunto possa causar uma mudança significativa no modelo. Assim para fazer uso das variações do conjunto de treinamento, o classificador base deve ser instável ou não linear, ou seja, as pequenas alterações no conjunto de treinamento devem levar a grandes mudanças na saída do classificador. Caso contrário, o conjunto resultante será uma coleção de classificadores quase idênticos, portanto, pouco provável de melhorar o desempenho de um classificador único. Exemplos de classificadores instáveis são redes neurais e árvores de decisão, enquanto k-vizinho mais próximo é um exemplo de um classificador estável.

Como pode ser observado anteriormente o algoritmo do *bagging* é uma solução ensemble completa, ou seja, o algoritmo executa desde a geração do ensemble de classificadores até a classificação. Porém, o objetivo de relacionar o *bagging* ou outros algoritmos como sendo geração de ensemble é simplesmente pela preocupação apenas com a geração dos classificadores bases, no caso do *bagging* a fase de treinamento.

2.2.2 Boosting

Similar ao *bagging*, o *boosting* (SCHAPIRE, 1990) também cria um ensemble de classificadores por subconjuntos de amostras dos dados de treinamento e combina a saída por votação majoritária. No entanto, no *boosting*, a amostragem é estrategicamente orientada para fornecer os dados de treinamento mais informativos para cada classificador consecutivamente. Em essência, cada iteração do *boosting* cria três classificadores fracos: o primeiro classificador C1 é treinado com um subconjunto aleatório dos dados de treinamento. O subconjunto de dados de treinamento para o segundo classificador C2 é escolhido como o subconjunto mais informativo, dado C1. Mais especificamente, C2 é treinado com os dados de treinamento somente metade dos quais é corretamente classificados por C1, e a outra metade que é classificada erroneamente. O terceiro classificador C3 é treinado com as instâncias em que C1 e C2 discordarem. Os três classificadores são combinados através da votação majoritária de três vias. O detalhe do pseudocódigo é mostrado na figura abaixo a seguir:

Algoritmo 2 Boosting

Entrada: conjunto de treinamento S de tamanho N com as classes w_l, $\Omega = \{w_1, w_2\}$; algoritmo de aprendizagem fraco **WeakLearn**.

Fase de Treinamento

1. S_1 ← Sorteio de $N_1 < N$ amostras sem reposição de *S*.
2. C_1 ← Chama **WeakLearn** e treina usando S_1.
3. S_2 ← Cria o subconjunto S_2 com os mais informativos, dado C_1, tal que metade de S_2 é classificada corretamente por C_1 e a outra metade são erroneamente classificada.
    a. Jogue uma moeda. Se cara, selecione amostras de *S*, e apresente a C_1 até a primeira instância erroneamente classificada. Adicione essa instância à S_2.

b. Se coroa, selecione amostras de *S*, e apresente a C_1 até a primeira instância corretamente classificada. Adicione essa instância à S_2.
c. Continue lançando moedas até que nenhum padrão possa ser mais adicionado a S_2.

4. C_2 ← Treina o segundo classificador usando S_2.
5. S_3 ← Cria S_3 selecionando aquelas instâncias pras quais C_1 e C_2 discordam.
6. C_3 ← Treina o terceiro classificador usando S_3.

Fase de Classificação – dado um padrão de teste **x**

7. Classifique **x** por C_1 e C_2. Se eles concordarem na classe, esta classe é a classificação final.
8. Se eles discordarem, escolha a classe predita por C_3 como a classificação final.

Schapire mostrou que o erro desse algoritmo tem um limite superior: se o algoritmo A usado para criar os classificadores C_1, C_2, C_3 tem um erro de ϵ (calculado em S), então o erro do ensemble é limitado acima por $f(\epsilon) = 3\epsilon^2 - 2\epsilon^3$. Note que $f(\epsilon) < \epsilon$ para $\epsilon < 1/2$. Ou seja, enquanto o algoritmo original A pode fazer pelo menos melhor do que adivinhar aletoriamente, então o ensemble boosting que combina os três classificadores gerados por A de três distribuições de S descrito acima, sempre terá melhor desempenho que A. Portanto, um classificador forte é gerado a partir de três classificadores fracos. Um classificador forte no sentido provavelmente aproximadamente correto (*PAC*) pode ser criado por aplicações recursivas do *boosting*. Uma limitação específica do *boosting* é que ele se aplica somente a problemas de classificação binária. Esta limitação é superada com o algoritmo *AdaBoost* (*Adaptive Boosting*).

2.2.3 Random Subspace

Uma maneira de melhorar a diversidade em um *ensemble* é treinar os classificadores individuais com diferentes subconjuntos do espaço de características. A seleção de características visa uma computação mais eficiente e uma maior precisão do conjunto. O *RSM* (*Random Subspace Method*) (HO, 1998) é um método de seleção de subconjuntos de características aleatória para a construção de *ensemble*. É similar ao *bagging* mas ao invés de fazer a amostragem de instâncias, ele faz uma amostragem de características sem repetição, uma vez que seria inútil incluir uma característica mais de uma vez. O *RSM* seleciona randomicamente um número arbitrário de subespaços de

características do espaço original, e constrói cada classificador em cada subespaço. Essa randomização deve criar classificadores que são complementares e assim a combinação pode ser feita por simples regras de fusão.

Algoritmo 3 Random Subspace Method

Entrada: tamanho L do ensemble; conjunto de treinamento S de tamanho N, onde o número de características é D; escolher d_i número de características para treinar cada classificador individual, onde $d_i < D$, para $i = 1, \dots, L$.

Inicialização

1. $E = \emptyset$, ensemble

Fase de Treinamento

2. **para** i = 1 até L **faça**
3. $S_i \leftarrow$ Sorteio de d_i características de D, sem reposição.
4. Constrói o classificador h_i usando S_i
5. Adiciona o classificador ao ensemble $E = E \cup h_i$

Fase de Classificação

6. **para** cada novo padrão **faça**
7. **se** se as saídas são contínuas **então**
8. Combine as saídas dos classificados de E pela média
9. **senão se** as saídas são os rótulos das classes **então**
10. Combine as saídas dos classificados de E pela votação majoritária

Experimentalmente evidências mostram que *RSM* trabalha bem com espaços de características com grandes conjuntos de características e redundâncias de características. Isso evita a "maldição" da dimensionalidade. Os conceitos de *RSM* podem ser relacionados à teoria da discriminação estocástica de Kleinberg.

2.3 Seleção DCS e DES

Nesta seção serão descritos os principais métodos de seleção dinâmica *DCS* e *DES*. Estas estratégias de seleção dinâmica tem o propósito de selecionar o classificador (*DCS*) ou o conjunto de classificadores (*DES*) que melhor classifica a região de competência, dado uma amostra de teste. A seguir serão explicados primeiros os

métodos de *DCS* baseado no *DCS-LA*: *OLA* e *LCA*. Em seguida os métodos *DES* baseados no *KNORA*: *KNORA-Eliminate* e *KNORA-Union*.

2.3.1 DCS-LA

O *DCS-LA* (*Dynamic Classifier Selection by Accuracy*) proposto por (WOODS, KEGELMEYER JR. e BOWYER, 1997) é uma abordagem para seleção dinâmica de classificador, baseado no conceito da estimativa da precisão local "*Local Accuracy Estimates*". Portanto, a ideia é estimar a precisão de cada classificador em regiões locais do espaço de características ao redor da amostra de teste desconhecida, e então usar a decisão do classificador mais preciso localmente. Na implementação de Woods, as "regiões locais" são definidas em termos do K-vizinhos mais próximos (*K-NN*) no conjunto de dados de treinamento. Woods propôs dois métodos para estimar a precisão local (*Local Accuracy*) que são: precisão local total (*OLA – Overrall Local Accuracy*) e a precisão local da classe (*LCA – Local Class Accuracy*). O pseudocódigo da abordagem *DCS-LA* é descrito abaixo:

Algoritmo 4 DCS-LA

Entrada: ensemble *E* de tamanho *L*; parâmetro *K* de vizinhos mais próximos (recomendado K=10); conjunto de dados de treinamento *T*.

1. **para** cada novo padrão de teste x_i **faça**
2. R_i^* ← Defina a região local ao redor da amostra de teste (x_i) como sendo o conjunto dos K-vizinhos mais próximos (K-NN) a partir do conjunto de treinamento *T*.
3. **para** cada classificador individual D_i do ensemble *E* **faça**
4. LA_i ← Calcule a estimativa de precisão local de D_i para a região local R_i^*.
5. Utilize a decisão do classificador D_i para classe da amostra x_i que tiver a maior estimativa da precisão local LA_i.

Os dois métodos, *OLA* e *LCA*, para calcular a estimativa de precisão local serão descritos nas próximas seções a seguir.

2.3.1.1 OLA

O *OLA* (*Overrall local accuracy*) usa a abordagem de *rank* de classificador (*Classifier Rank*), que seleciona o classificador que classifica corretamente mais vizinhos da amostra de teste na região local. O classificador selecionado é dito ter o maior "*rank*". Um exemplo prático do *OLA*, pode ser visto na **Figura 2**:

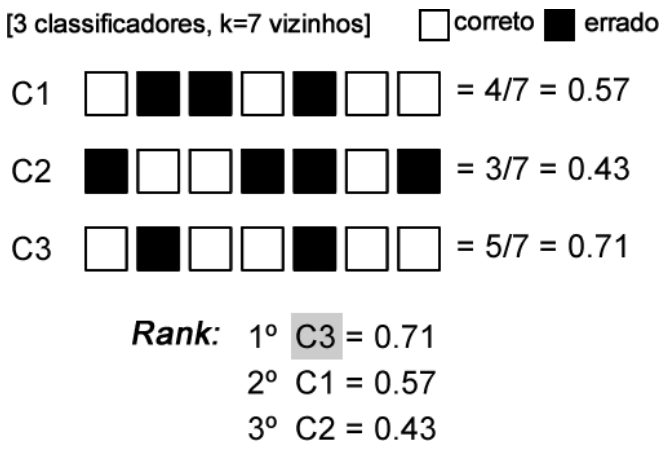


Figura 2 Exemplo da abordagem *OLA*, onde três classificadores (C1, C2 e C3) classificam os 7 vizinhos do padrão de teste. O *OLA* faz o *rank* dos classificadores com maior percentagem de acerto dos vizinhos, nesse exemplo o seleciona o classificador C3.

2.3.1.2 LCA

Este método é similar ao método *OLA*, a única diferença é que a precisão local é estimada em respeito a classe de saída. Considerando um classificador que atribui uma classe C_i a uma amostra de teste. Com isso é possível determinar a percentagem das amostras de treinamento atribuídas à classe C_i por este classificador que tem sido classificada corretamente. A seguir um exemplo do *LCA*:

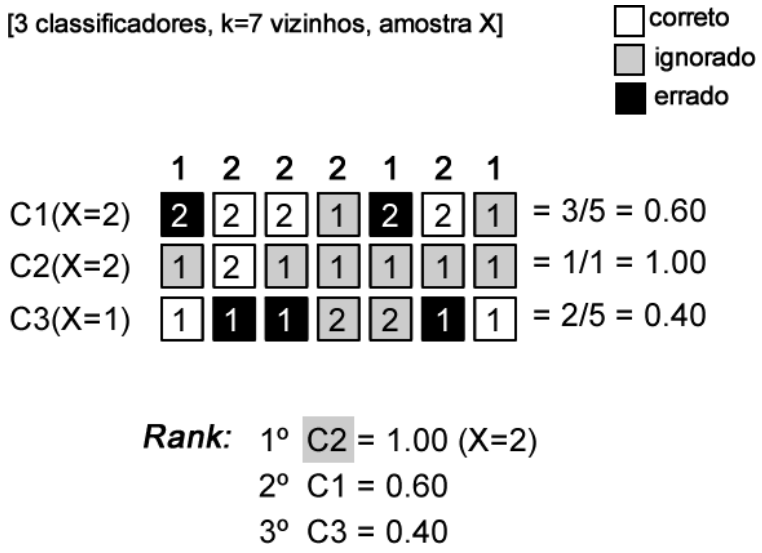


Figura 3 Exemplo da abordagem *LCA*. Os três classificadores C1, C2 e C3 classificam o padrão de teste X e suas precisões locais são calculadas baseadas na classe de saída atribuída a X. Conforme o exemplo, o classificador C3 tem a maior precisão local.

2.3.2 KNORA

Todos os métodos de seleção dinâmica descritos acima são projetados para encontrar o classificador com maior possibilidade de ser correto para uma amostra em uma vizinhança pré-definida. No entanto a abordagem do *KNORA* é outra: ao invés de encontrar o classificador mais adequado, é selecionado o conjunto mais adequado para cada amostra utilizando o conceito Oracle.

O conceito do *KNORA* (*K-nearest-oracles*) (KO, SABOURIN e JR., 2008) é similar aos conceitos de *OLA* e *LCA* na consideração da vizinhança de padrões de teste, mas pode ser distinguido dos outros pelo uso direto de sua propriedade de ter amostras de treinamento na região com o qual quer encontrar o melhor conjunto para uma determinada amostra. Para qualquer dado de teste, *KNORA* simplesmente encontra os *K* vizinhos mais próximos no conjunto de validação, descobre quais os classificadores classificam corretamente esses vizinhos no conjunto de validação e usa-os como ensemble para classificar o padrão fornecido naquele conjunto de teste.

Ko ainda propôs quatro diferentes métodos para a abordagem *KNORA*: *KNORA-E* (*KNORA-Eliminate*), *KNORA-U* (*KNORA-Union*), e suas versões ponderadas, *KNORA-E-W* e *KNORA-U-W*. As descrições detalhadas desses métodos serão vistas a seguir.

2.3.2.1 KNORA-Eliminate

O *KNORA-E* seleciona o subconjunto de classificadores por meio da eliminação de classificadores utilizando a abordagem *KNORA*. Portanto, dado *K* vizinhos x_i, $1 \leq j \leq K$ de um padrão de teste X, e supondo que o conjunto de classificadores $C(j)$, $1 \leq j \leq K$ classificam corretamente todos os seus *K*-vizinhos mais próximos, então cada classificador $c_i \in C(j)$ pertencente a este conjunto de classificadores corretos $C(j)$ deve apresentar um voto sobre a amostra X. No caso onde nenhum classificador pode classificar corretamente todos os *K*-vizinhos mais próximos do padrão de teste, então simplesmente é diminuído o valor de K até pelo menos um classificador classificar corretamente os seus vizinhos. Um exemplo prático pode ser visto na **Figura 4**:

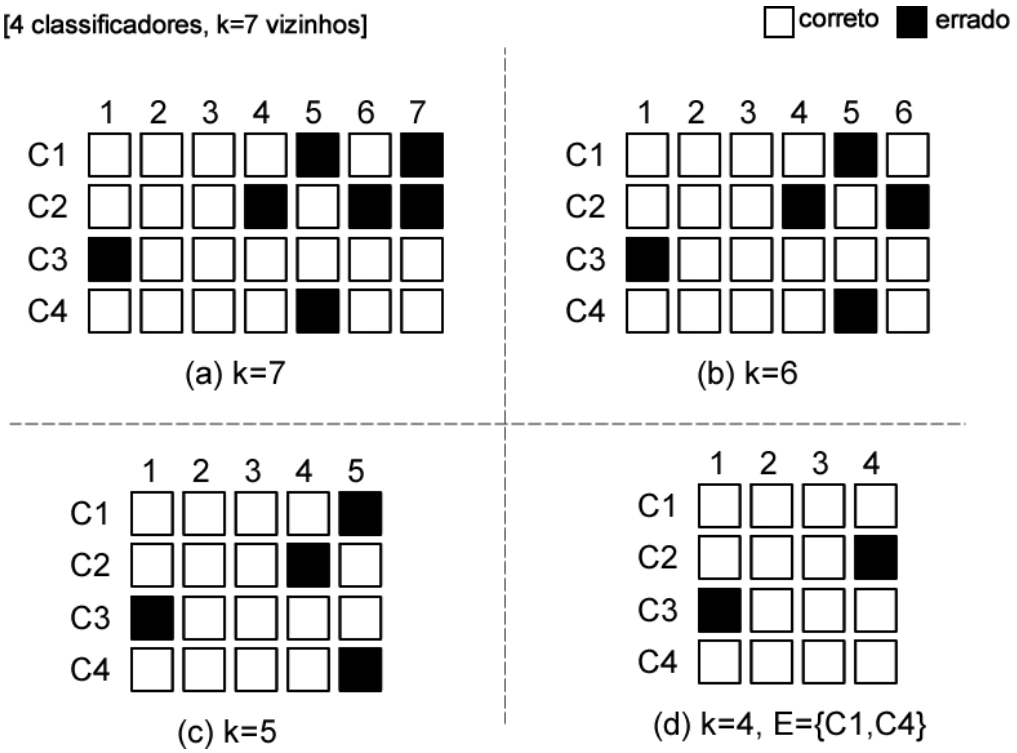


Figura 4 Exemplo da execução do KNORA-E para um ploblema com 4 e a região de competência com k=7 vizinhos. Depois de executar os passos mostrados em (a), (b), (c) e (d), o KNORA-E seleciona os classificadores C1 e C4.

2.3.2.2 KNORA-Union

O *KNORA-U* utiliza a abordagem *KNORA* selecionar os classificadores que classificam corretamente cada padrão vizinho. Dessa forma, dado *K* vizinhos x_i, $1 \leq j \leq K$ de um padrão de teste X, e supondo que o j-vizinho mais próximo tem sido corretamente classificado por um conjunto de classificadores $C(j)$, $1 \leq j \leq K$, então cada

classificador $c_i \in C(j)$ pertencentes ao conjunto de classificadores correto $C(j)$ devem apresentar um vote sobre a amostra X. Note que, uma vez que todos os K-vizinhos próximos são considerados, um classificador pode ter mais de um voto se este classificar corretamente mais de um vizinho. Quanto mais vizinhos o classificador classificar corretamente, mais votos este classificador terá para um padrão de teste. Um exemplo do *KNORA-U* pode ser visto na **Figura 5**:

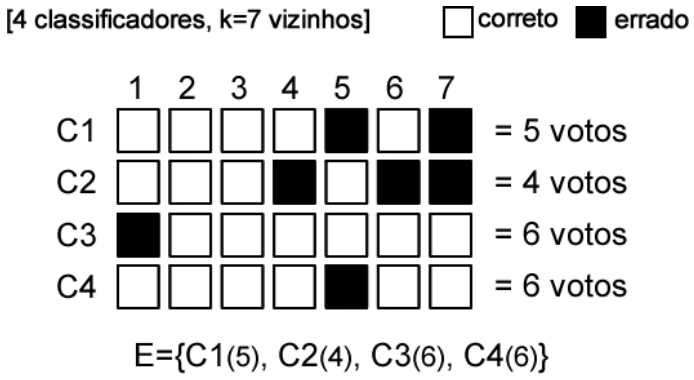


Figura 5 Exemplo do KNORA-Union com 4 classificadores e 7 vizinhos. Nesse exemplo, todos os classificadores foram selecionados com seus respectivos votos sobre a amostra X.

2.3.2.3 KNORA-Eliminate-W

Esse método é similar ao *KNORA-Eliminate*, a diferença é que cada voto do classificador é ponderado pela distância Euclidiana entre o padrão vizinho x_i e o padrão de teste X.

2.3.2.4 KNORA-Union-W

Esse método é similar ao *KNORA-Union*, a diferença é que cada voto é ponderado pela distância euclidiana entre o padrão vizinho x_i e o padrão de teste X.

2.4 Combinação de Decisões

Depois obter um ensemble, por um método de geração de ensemble ou por um método de seleção dinâmica de *ensemble* (*DES*). A decisão final pode ser obtida por várias regras de combinação (fusão). As possíveis maneiras de combinar as saídas dos *L* classificadores de um ensemble dependem de qual informação é obtida dos membros individuais. Alguma destas regras de combinação opera somente na classe de saída,

enquanto outros precisam das saídas contínuas que podem ser interpretadas como suporte dado pelo classificador para cada uma das classes. Segundo Xu em (XU, KRZYZAK e SUEN, 1992) os modelos de informação produzida pelos membros de um ensemble podem ser classificados em três tipos:

- **Abstrato** (*The Abstract Level*):
  - Cada classificador D_i produz um rótulo da classe de saída $s_i \in \Omega, i = 1, \dots, L$. Portanto, para qualquer amostra de teste $\boldsymbol{x}$ ser classificada, as saídas L classificadores definem um vetor $\boldsymbol{s} = [s_1, \dots, s_L]^T \in \Omega^L$. No nível abstrato, não há nenhuma informação que indica o grau certeza da classificação atribuída. Por definição, qualquer classificador pode ser capaz de produzir uma classe de saída para $\boldsymbol{x}$.
- **Rank** (*The Rank Level*):
  - A saída de cada classificador D_i é um subconjunto de Ω, com as alternativas em ordem de possibilidade de serem corretamente classificadas. Sendo assim é uma lista de classes ranqueadas para cada padrão de entrada.
- **Medição** (*The Measurement Level*):
  - A saída de cada classificador D_i produz um vetor c-dimensional $[d_{i,1}, \dots, d_{i,c}]^T$ de valores contínuos. O valor contínuo $d_{i,j}$ representa a estimativa da probabilidade a posteriori da classe ou o valor de confiança relacionado a classe que representa o suporte para a possível hipótese de classificação do padrão de teste $\boldsymbol{x}$. Sem perca de generalidade, pode ser assumido que o vetor de saída contem valores $d_{i,j} \in [0, 1]$.

2.4.1 Votação Majoritária

A votação majoritária é uma das estratégias mais antigas para fazer a decisão e trabalha no nível de abstração. Ela é utilizada por diversos métodos de criação de ensemble vistos anteriormente e é tido como um combinador ideal. Junto com as regras de média e produto, que serão vistos na próxima seção, a votação majoritária é um dos métodos mais usados.

Assuma que as saídas dos classificadores são dadas por um vetor binário c-dimensional $[d_{i,1}, \dots, d_{i,c}]^T \in \{0,1\}^c,\ i = 1, \dots, L$, onde $d_{i,j} = 1$ se o classificador D_i classificou $\boldsymbol{x}$ como sendo da classe ω_j, e $d_{i,j} = 1$ caso o contrário. Portanto, a decisão do ensemble é feita para a classe ω_k se este recebe a maior votação:

$$\sum_{i=1}^{L} d_{i,k} = max_{j=1}^{c} \sum_{i=1}^{L} d_{i,j}$$

Vários estudos tem mostrado que a votação majoritária é um dos métodos mais utilizados devido a sua propriedade característica: sob a condição de que as saídas dos classificadores são independentes, pode ser demonstrado que a combinação por votação majoritária vai sempre levar a uma melhoria de desempenho. Se há um total de L classificadores para um problema de duas classes, a decisão do ensemble estará correta se ao menos $\lfloor L/2 + 1 \rfloor$ classificadores escolher a classe correta.

Agora, assuma que cada classificador tem uma probabilidade p de tomar uma decisão correta. Então, a probabilidade do ensemble tomar uma decisão correta tem uma distribuição binomial, especificamente, a probabilidade de escolher $k > \lfloor L/2 + 1 \rfloor$ classificadores corretos de L é:

$$P_{ens} = \sum_{k=\lfloor L/2 \rfloor + 1}^{L} \binom{L}{k} p^k (1-p)^{L-k}$$

então,

$$P_{ens} \to 1,$$

quando $T \to \infty$ se $p > 0.5$ e

$$P_{ens} \to 0,$$

quando $T \to \infty$ se $p < 0.5$.

Note que a exigência de $p > 0.5$ é necessária e suficiente para um problema de duas classes, enquanto que é suficiente, mas não necessário para problemas multiclasse.

Uma variante da votação majoritária é a votação majoritária ponderada. Os votos são multiplicados por pesos que é frequentemente obtido pela estimativa da precisão dos classificadores no conjunto de validação. Uma possível estimativa é:

$$w_i = \log\left(\frac{p_i}{1 - p_i}\right)$$

Onde p_i é a precisão do i-ésimo classificador.

2.4.2 Combinadores Algébricos

Os graus de suporte dado por um padrão de entrada $\boldsymbol{x}$ pode ser interpretado em diferentes maneiras, às duas mais comuns são: o valor de confiança em uma sugestão de classificação e a estimativa das probabilidades a posteriori para as classes.

Dado que o padrão $\boldsymbol{x} \in \Re^n$ seja um vetor de características e $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$ seja o conjunto das classes do problema. Cada classificador D_i no ensemble $\mathcal{D} = \{D_1, \ldots, D_L\}$ têm c saídas de graus de suporte. Assumindo que todas as c saídas são valores no intervalo $[0, 1]$, então $D_i: \Re^n \rightarrow [0,1]^c$ trabalha no nível de medição. Portanto, denota-se $d_{i,j}(\boldsymbol{x})$ o suporte que o classificador D_i dá a hipótese que $\boldsymbol{x}$ é da classe ω_j. Quanto maior o suporte, maior a chance de ser da classe ω_j. As saídas dos L classificadores para um dado padrão $\boldsymbol{x}$ pode ser organizada em uma matriz de perfil de decisão $DP(x)$ (*decision profile*) mostrada na **Figura 6**:

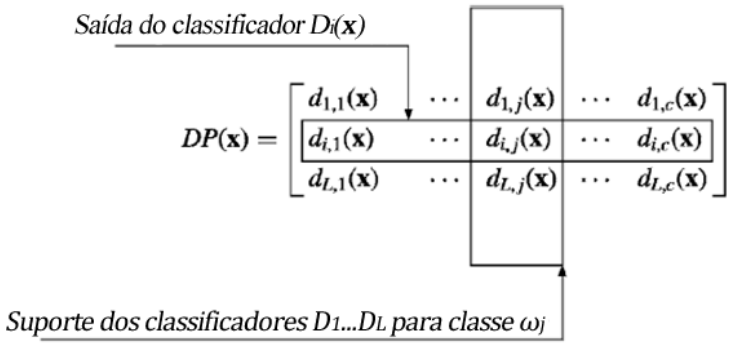


Figura 6 Matriz DP(x).

Os combinadores algébricos usa a matriz $DP(x)$ para encontrar o suporte geral de cada classe para entrada $\boldsymbol{x}$ com o maior valor de suporte. Isso é feito através de

$\mu_j(x) = \mathcal{F}[d_{1,j}(\boldsymbol{x}), \ldots, d_{L,j}(\boldsymbol{x})]$, o suporte de graus geral para ω_j, obtido após aplicar uma função de combinação $\mathcal{F}$ de expressões algébricas aos suportes individuais $d_{i,j}(\boldsymbol{x})$ da classe ω_j dado pelo ensemble. Sendo assim, a decisão final pode ser obtida por $h_{final}(x) = argmax_j\ \mu_j(x)$. As funções de combinação $\mathcal{F}$ podem computar o suporte geral da classe de diferentes maneiras como visto a seguir:

- **Média**: calcula a média dos suportes para cada classe, e a decisão final $h_{final}(x)$ é dada pela classe com maior média.

$$\mu_j(x) = \frac{1}{L}\sum_{i=1}^{L} d_{i,j}(\boldsymbol{x})$$

- **Soma**: realiza a soma dos suportes para cada classe, e a decisão final $h_{final}(x)$ é dada pela classe com maior soma. Essa regra é equivalente à média.

$$\mu_j(x) = \sum_{i=1}^{L} d_{i,j}(\boldsymbol{x})$$

- **Produto**: multiplica os suportes para cada classe, e atribui a decisão final $h_{final}(x)$ para a classe que tiver o maior produto.

$$\mu_j(x) = \prod_{i=1}^{L} d_{i,j}(\boldsymbol{x})$$

- **Máximo**: encontra o suporte máximo para cada classe, e atribui a decisão final $h_{final}(x)$ para a classe com o maior suporte máximo.

$$\mu_j(x) = \max_i\{d_{i,j}(\boldsymbol{x})\}$$

- **Mínimo**: encontra o suporte mínimo para cada classe, e atribui a decisão final $h_{final}(x)$ para a classe com o maior suporte mínimo.

$$\mu_j(x) = \min_i\{d_{i,j}(\boldsymbol{x})\}$$

- **Mediana**: encontra a mediana dos suportes para cada classe, e atribui a decisão final $h_{final}(x)$ para a classe com o maior mediana dos suportes.

$$\mu_j(x) = \underset{i}{\text{med}}\{d_{i,j}(\boldsymbol{x})\}$$

A **Figura 7** mostra um exemplo de combinação dos classificadores utilizando as funções algébricas através da matriz $DP(x)$.

	ω_1	ω_2	ω_3	ω_4
h_1	0.1	0.5	0.2	0.1
h_2	0.3	0.3	0.3	0.1
h_3	0.2	0.0	0.8	0.0

Regra					Resultado
Mínimo	0.1	0.0	0.2	0.0	ω_3
Máximo	0.3	0.5	0.8	0.1	ω_3
Produto	0.01	0.00	0.05	0.00	ω_3
Média	0.2	0.3	0.4	0.1	ω_3
Mediana	0.2	0.3	0.3	0.1	empate

Figura 7 Problema utilizando um ensemble de três classificadores e várias regras de combinação algébrica.

CAPÍTULO 3

O Método

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 (DOS SANTOS, SABOURIN e MAUPIN, 2008): (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.

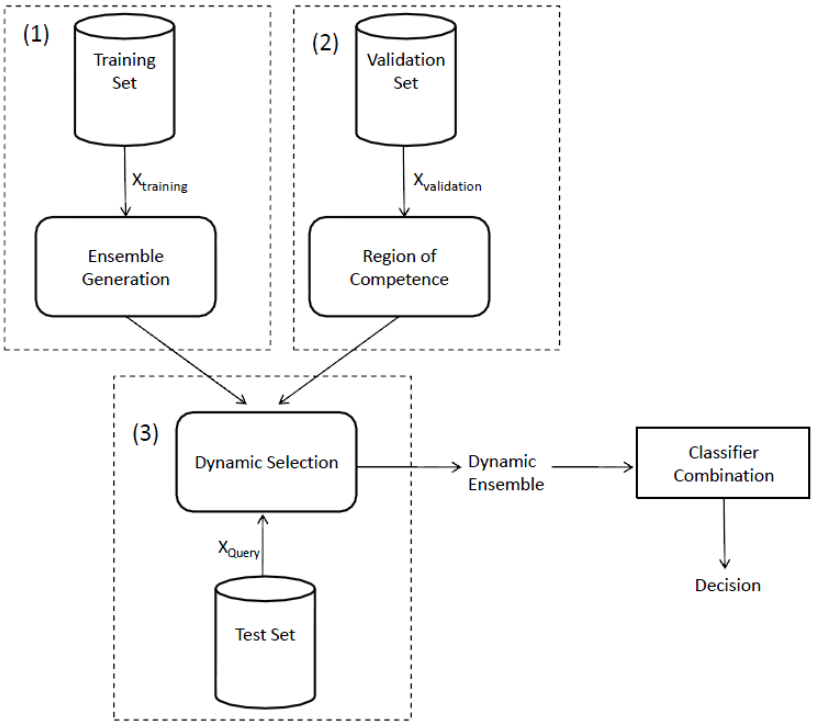


Figura 8 Visão geral do sistema de seleção dinâmica de classificador modificado para o uso de seleção dinâmica de ensemble.

Muitos estudos têm sido realizados no primeiro e no terceiro nível. No primeiro nível, os algoritmos mais utilizados são *bagging*, *boosting* e *random subspace*. No terceiro nível, Woods 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 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 competênci (CRUZ, CAVALCANTI e REN, 2011) a .

Portanto o objetivo deste trabalho é utilizar a abordagem *DES-FA* descrita em (CRUZ, CAVALCANTI e REN, 2011), para o segundo nível e utilizar diferentes

abordagens além do *KNORA-E* para o terceiro nível. Primeiro, será mostrado o método proposto por Cruz 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 Cruz 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 (KO, SABOURIN e JR., 2008). Em seguida será visto um exemplo prático da influência da qualidade da região de competência.

3.2.1.1 KNORA-Eliminate

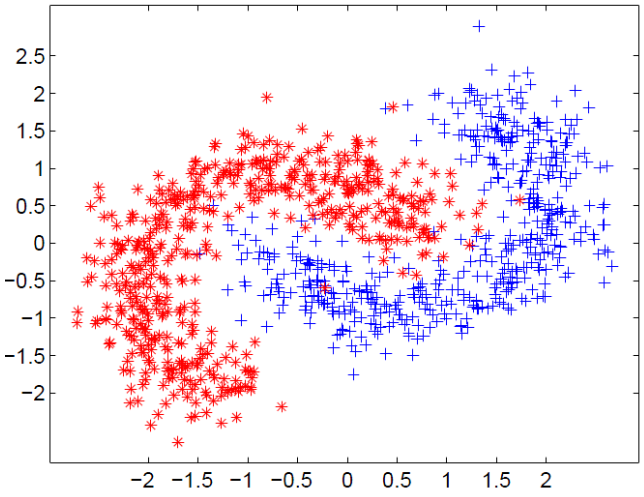
Esta abordagem explora o conceito *Oracle* selecionando dinamicamente os classificadores. Dado que X_i, $i = 1, \dots, k$, são os k vizinhos mais próximos do padrão de

teste X e um ensemble de L classificadores C_j, $j = 1, \dots, L$, o ensemble dinâmico L^* é composto pelos classificadores C_j que classificam corretamente cada X_i. Os classificadores que erram na classificação de qualquer um dos k vizinhos são eliminados. Se nenhum classificador pode classificar corretamente todos os vizinhos, o valor de k 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 L são usados para dar a resposta final.

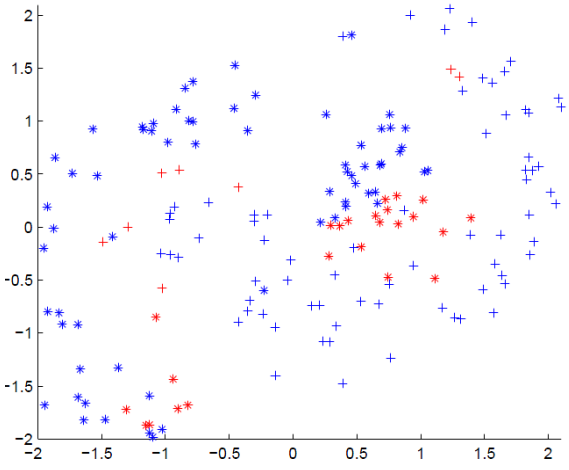
Uma vantagem deste método é que o número de vizinhos não é fixo, embora pode somente diminuir. No entanto, o custo da redução da vizinhança e o método para recalcular são computacionalmente caros. Como as outras técnicas de seleção dinâmica, este método é dependente da qualidade dos padrões vizinhos.

3.2.1.2 Análise

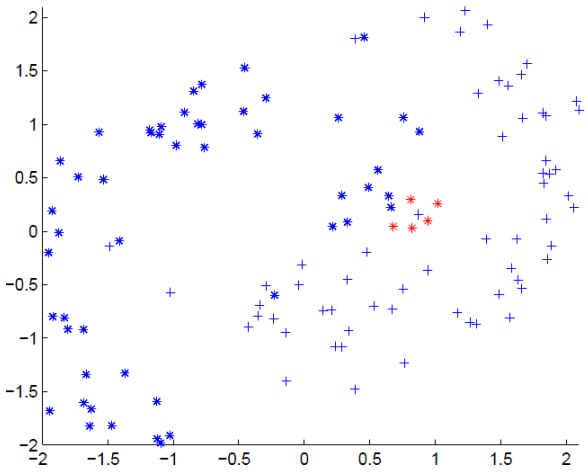
Para demonstrar o problema que as técnicas de seleção dinâmica têm com a qualidade da região de competência, será mostrado abaixo um experimento usando um ensemble de 10 perceptrons gerados usando o algoritmo bagging. Foi utilizado o valor da vizinhança $k = 7$. A **Figura 9** mostra os erros de classificação obtidos pelo KNORA-E para a base Banana. A **Figura 9** (a) mostra a forma dos dados da base Banana. A **Figura 9** (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 9** (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.



(a) Banana dataset



(b) Validation Set



(c) Errors

Figura 9 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 10** mostra uma visão geral do sistema proposto. T é o conjunto de treinamento, V o conjunto de validação e G os dados de teste (generalização). Durante o estágio de treinamento, o ensemble $E = \{C_1, \ldots, C_L\}$ é gerado usando conjunto de dados T. O filtro *ENN* (*Edited Nearest Neighbor*) é aplicado aos dados de validação V gerando o conjunto de dados V', $|V'| \leq |V|$. O filtro *ENN* trabalha eliminando o ruído nas fronteiras de decisão. Portanto, o algoritmo produz uma fronteira de decisão suave.

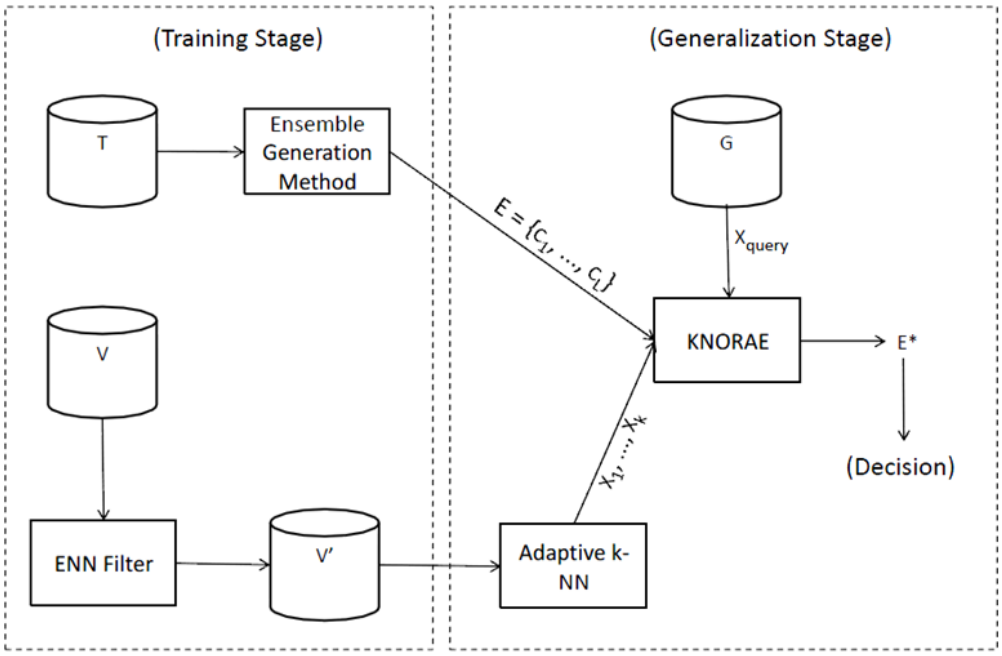


Figura 10 Visão geral do sistema DES-FA.

Na fase de teste, a região local é computada com o algoritmo do *k-NN* adaptativo (WANG, NESKOVIC e COOPER, 2007) usando os padrões do conjunto de dados

filtrado V'. 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 E^* 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*) (WILSON, 1972) 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 5 mostram os passos do algoritmo *ENN*. O algoritmo funciona da seguinte forma: Seja T o conjunto de treinamento, e S o conjunto filtrado, o algoritmo realiza a classificação do vizinho mais próximo para cada $X_i \in T$ usando T como referência. Se X_i é classificado errado usando o algoritmo *k-NN*, este é considerado um ruído e é removido do conjunto final S.

Algoritmo 5 ENN

Entrada: conjunto de treinamento T.

Saída: conjunto filtrado S.

1. $S = T$
2. **para** cada $X_i \in T$ **faça**
3. se $classe(X_i) \neq classe(kNN(X_i, T))$ então
4. $S = S - \{X_i\}$

A **Figura 11** mostra um exemplo da aplicação do *ENN*. Os dados foram construídos usando duas distribuições Gaussianas geradas com $\mu_1 = [0.0, 0.0]$, $\mu_2 = [3.5, 0.0]$ e $\sigma_1^2 = \sigma_2^2 = 1$. A **Figura 11** (a) mostra a distribuição original. A **Figura 11** (b), (c) e (d) apresentam o resultado depois da execução do algoritmo *ENN* com $k = 1, 3, 5$ respectivamente.

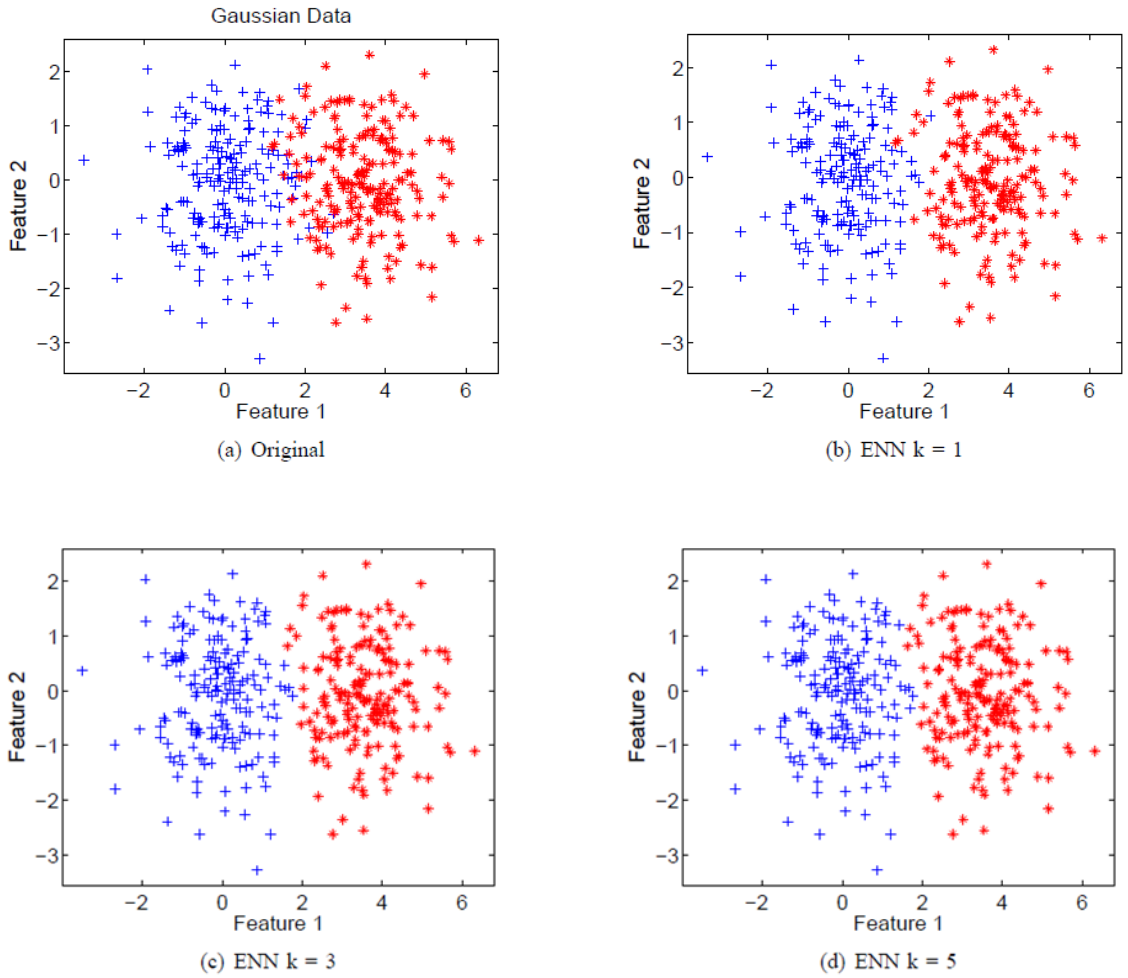


Figura 11 Resultado do algoritmo *ENN* para duas distribuições gaussianas.

3.2.2.2 K-NN com distância adaptativa

A distância adaptativa (WANG, NESKOVIC e COOPER, 2007) calcula, para cada amostra de treinamento X_i, a maior esfera centrada em X_i, $i = 1, \dots, N$, que exclui cada padrão de treinamento de classes diferentes X_j, $j = 1, \dots, N$. Esta é realizada pelo cálculo da distância mínima (raio da esfera) R_i entre o padrão de treinamento X_i e as amostras de treinamento das classes diferentes (Eq. 1). Com o raio R_i, a distância adaptativa entre o padrão de teste X_{test} e X_i é definida pela (Eq. 2). A distância $d(X_{test}, X_i)$ pode ser qualquer distância, tal como, a distância Euclidiana ou a Manhattan.

$$R_i = \min d(X_{test}, X_i) \ , c_i \neq c_j \quad (1)$$

$$D_{adap}(X_{test}, X_i) = \frac{d(X_{test}, X_i)}{R_i} \quad (2)$$

Usando este método, amostras próximas da média de sua classe tem o maior raio R_i 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 C_i 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 N primeiros classificadores do ranking, com $N \leq L$. 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* (*Incremental Wrapper Subset Selection*) para a seleção de atributos para então apresentar o método proposto.

3.3.2 Analogia DES com FSS

A seleção subconjunto de características (*FSS – Feature Subset Selection*) 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 X, não necessariamente terá que conter apenas os N 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 N melhores classificadores computados pelos métodos *OLA* e *LCA*.

3.3.3 IWSS

Em problemas de *FSS*, devido à alta cardinalidade (2^n) do espaço de busca, a busca exaustiva é intratável mesmo para valores moderados de n 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, RIQUELME e AGUILAR-RUIZ, 2006) 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 $SU(\,)$ é geralmente a medida de pontuação utilizada para o ranqueamento do atributo de acordo com sua importância.
- *Wrapper*: o subconjunto S selecionado é inicializado com o primeiro atributo do ranking, então o algoritmo tenta iterativamente incluir em S o próximo atributo X_i do *ranking* pela avaliação do desempenho do subconjunto aumentado $S_{aux} = S \cup \{X_i\}$. A avaliação dos subconjuntos candidatos é feito na forma de wrapper, e se uma diferença positiva é obtida, então X_i é adicionado ao subconjunto S, caso contrário é descartado.

Para exemplificar o algoritmo *IWSS*, assuma um problema com oito atributos $X_1, \dots, X_8$. Supondo que os seguintes valores são computados no passo do filtro: $SU(X_1, C) = 0.9$, $SU(X_2, C) = 0.8$, $SU(X_3, C) = 0.6$, $SU(X_4, C) = 0.5$, $SU(X_5, C) = 0.4$, $SU(X_6, C) = 0.2$, $SU(X_7, C) = 0.1$ e $SU(X_8, C) = 0.01$. Portanto o *ranking* do filtro é $r = X_1, X_2, \dots, X_8$. Então a execução da fase do *wrapper* é:

Tabela 1 Iterações *wrapper* do *IWSS*.

Passo	Atributo testado	Subconjunto testado	Precisão	Decisão	Subconjunto resultante
1	X_1	$\{X_1\}$	0.6	Aceito	$\|S\| = \{X_1\}$
2	X_2	$\{X_1, X_2\}$	0.7	Aceito	$\|S\| = \{X_1, X_2\}$
3	X_3	$\{X_1, X_2, X_3\}$	0.68	Rejeitado	$\|S\| = \{X_1, X_2\}$
4	X_4	$\{X_1, X_2, X_4\}$	0.71	Aceito	$\|S\| = \{X_1, X_2, X_4\}$
5	X_5	$\{X_1, X_2, X_4, X_5\}$	0.71	Rejeitado	$\|S\| = \{X_1, X_2, X_4\}$
6	X_6	$\{X_1, X_2, X_4, X_6\}$	0.65	Rejeitado	$\|S\| = \{X_1, X_2, X_4\}$
7	X_7	$\{X_1, X_2, X_4, X_7\}$	0.7	Rejeitado	$\|S\| = \{X_1, X_2, X_4\}$

8	X_8	$\{X_1, X_2, X_4, X_8\}$	0.75	Aceito	$\|S\| = \{X_1, X_2, X_4, X_8\}$

E portanto, $|S| = \{X_1, X_2, X_4, X_8\}$ 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 X. Para a fase 2, *wrapper*, o subconjunto S é inicializado com o classificador C_i do *ranking* com a maior precisão local, primeiro do *ranking*. Então iterativamente o algoritmo adiciona o próximo classificador C_j ao subconjunto $S_{aux} = S \cup \{C_j\}$ e realiza a avaliação do subconjunto de classificadores S_{aux} para estimar a precisão local do novo subconjunto. Caso a inclusão do novo classificador C_j ao subconjunto S_{aux} aumente a precisão local com relação à S, então ele é adicionado ao subconjunto final $S = S \cup \{C_j\}$, 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 S com $|S| \leq L$ classificadores, onde L 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:

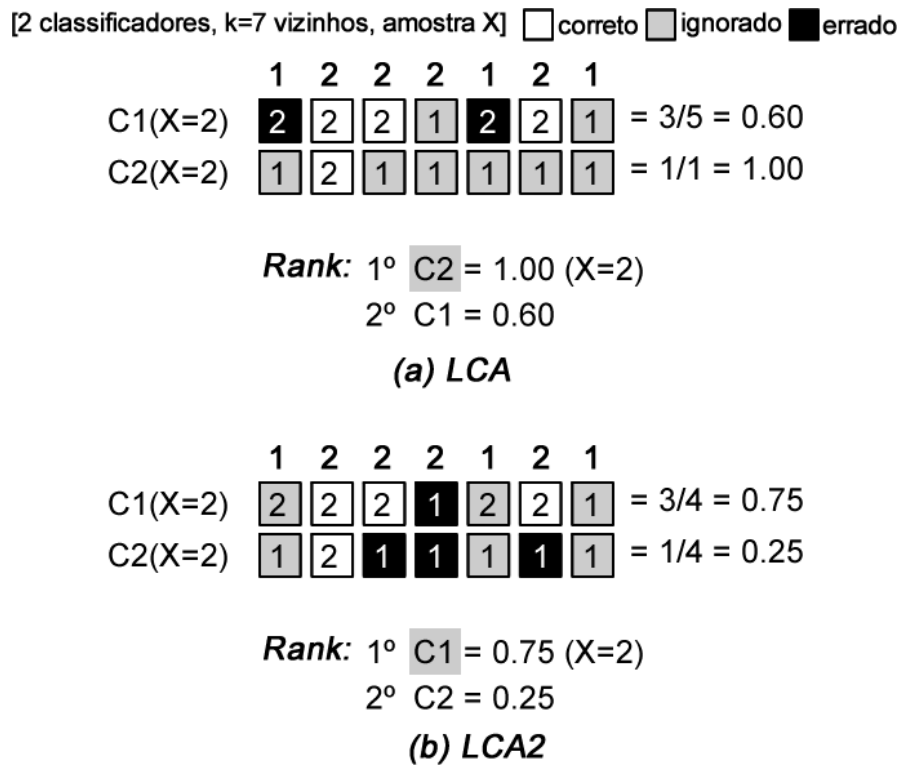


Figura 12 Exemplo da diferença do *LCA* (a) e o *LCA2* (b) proposto. O *LCA3* é apenas uma média das estimativas do *LCA* e *LCA2*.

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*.

Nas próximas seções as adaptações dos métodos *DCS-LA* (*OLA*, *LCA* e as variações do *LCA*) propostos para a seleção dinâmica de ensemble (*DES*) serão chamados de: *DES-OLA*, *DES-LCA*, *DES-LCA2* e *DES-LCA3*.

CAPÍTULO 4

Experimentos

4.1 Descrições das bases

Para avaliar o método proposto os experimentos foram realizados utilizando sete bases de dados, cinco repositórios da *UCI Machine Learning* e duas geradas artificialmente usando o *Matlab PRTOOLS toolbox*. As características das bases de dados são mostradas na **Tabela 2**.

Tabela 2 Características das bases de dados.

Base de dados	Número de instancias	Dimensão	Número de Classes
Pima	768	8	2
Liver Disorders	345	6	2
WDBC	568	30	2
Blood transfusion	748	4	2
Banana	600	2	2
Vehicle	846	18	4
Lithuanian	600	2	2

Os dados foram divididos em 50% pra o conjunto de treinamento e 50% pra o conjunto de teste. O conjunto de treinamento foi dividido em 75% para o treinamento e 25% para a validação. O conjunto de validação é usado para computar as regiões de competências. Recomenda-se utilizar um conjunto diferente do conjunto de treinamento para computar as regiões de competências porque os classificadores bases podem sofrer *overfitting* nos padrões de treinamento. Portanto, as informações da precisão podem ser enviesadas.

4.2 Metodologia

A geração do ensemble foi realizada usando a técnica *bagging*, que dá bons resultados quando utilizado um classificador base fraco e instável. O ensemble gerado é composto por 10 classificadores do modelo *Perceptron* e o número de vizinhos igual a 7 para computar a região de competência. O *Perceptron* foi escolhido por causa da sua instabilidade e por ser um modelo fraco. As regras de combinação testadas foram: votação majoritária, e as regras algébricas, média e produto.

Primeiro, foi feito a execução dos métodos (*KNORA-E*, *DES-OLA*, *DES-LCA*, *DES-LCA2* e *DES-LCA3*) de seleção, sem a utilização da abordagem *DES-FA* e utilizando a votação majoritária para combinação. Posteriormente foi utilizado o *DES-FA(1)*, com *k=1* para o *ENN* (melhor configuração *DES-FA* obtida pelo autor), e comparado com os resultados sem a utilização do *DES-FA*, a fim de mostrar como a qualidade da região de competência afeta o desempenho dos métodos de seleção. Em seguida foram testadas também as combinações de média e produto para os métodos, com o objetivo de investigar se a votação majoritária prejudicaria os métodos baseado no *DCS-LA*, nos casos de empate de votação nas iterações do *IWSS*. Por fim, foram avaliados os desempenhos em tempo de processamento dos métodos de seleção com e sem a abordagem *DES-FA*.

4.3 Resultados e Análise

Os resultados dos experimentos descritos na seção anterior são mostrados nas tabelas a seguir junto com a análise dos resultados, após cada tabela.

Tabela 3 Execução dos métodos DES sem a abordagem DES-FA e utilizando a regra da votação majoritária para combinação. O valor de cada célula é a média da precisão de classificação de três iterações, com exceção do Oracle* obtido pelo autor.

Base	KNORA-E	DES-OLA	DES-LCA	DES-LCA2	DES-LCA3	Oracle*
Vehicle	**81,20**	79,86	78,52	75,67	77,41	96,8
Pima	**74,22**	73,52	73,00	65,62	71,53	95,1
Liver Disorders	59,11	61,43	**63,57**	55,81	61,05	90,07
WDBC	**96,48**	95,66	96,24	92,84	94,25	99,13
Blood transfusion	72,19	76,11	**76,56**	61,23	66,22	94,2
Banana	91,20	93,40	**94,20**	52,33	56,27	94,75

Lithuanian	90,11	91,78	**92,00**	46,56	51,56	98,35

No experimento da **Tabela 3** Execução dos métodos DES sem a abordagem DES-FA e utilizando a regra da votação majoritária para combinação. O valor de cada célula é a média da precisão de classificação de três iterações, com exceção do Oracle* obtido pelo autor. (sem a abordagem *DES-FA*) pode ser visto que o algoritmo DES-LCA obteve o melhor resultado, sendo o melhor em quatro bases (*Liver Disorders*, *Blood transfusion*, *Banana* e *Lithuanian*) e superando o *KNORA-E* em média por 1.34 pontos percentuais. O *DES-OLA* também foi melhor que o *KNORA-E* em média 1.03 pontos percentuais, superando o *KNORA-E* nas mesmas quatro bases. Quanto às variações do *LCA*, *DES-LCA2* e *DES-LCA3*, os resultados não foram satisfatórios, tendo os resultados inferiores em quase todas as bases com relação ao *KNORA-E*, *DES-OLA* e *DES-LCA*.

Tabela 4 Execução dos métodos DES com a abordagem DES-FA(1) e utilizando a regra da votação majoritária para combinação. O valor de cada célula é a média da precisão de classificação de três iterações, com exceção do Oracle* obtido pelo autor.

Base	KNORA-E	DES-OLA	DES-LCA	DES-LCA2	DES-LCA3	Oracle*
Vehicle	**72,91**	71,09	71,09	67,38	68,56	96,8
Pima	72,57	**73,18**	71,79	66,93	66,58	95,1
Liver Disorders	62,21	63,37	**64,53**	57,95	59,69	90,07
WDBC	95,31	**96,36**	95,31	94,95	95,07	99,13
Blood transfusion	**77,90**	77,45	77,63	72,82	72,10	94,2
Banana	92,20	92,33	**92,60**	53,13	53,53	94,75
Lithuanian	**93,44**	92,78	90,33	48,33	51,00	98,35

No experimento da **Tabela 4** (com a abordagem *DES-FA*) pode ser visto que o *KNORA-E* obteve o melhor resultado no geral, sendo melhor em três bases (*Vehicle*, *Blood transfusion* e *Lithuanian*) das setes bases. Porém, comparando individualmente o *KNORA-E* com o *DES-OLA*, percebe-se que o *DES-OLA* obteve melhor resultado em quatro bases com relação ao *KNORA-E*. Enquanto que com relação ao *DES-LCA*, o *KNORA-E* foi superior. Ao analisar o *DES-LCA2* e *DES-LCA3* percebe-se que eles tiveram novamente os piores resultados, tendo os resultados de ambos quase equivalentes.

Segundo os resultados da **Tabela 3** (sem *DES-FA*) e **Tabela 4** (com *DES-FA*), pode ser visto que: para o método de seleção *KNORA-E* o resultado melhorou em 4 das 7 bases analisadas, com a abordagem *DES-FA*, piorando nas bases *Vehicle*, *Pima* e *WDBC*. Analisando o método *DES-OLA*, percebe-se que também melhorou em 4 das 7 bases, piorando nas bases *Vehicle*, *Pima* e *Banana*. Enquanto que o *DES-LCA* (com *DES-FA*) piorou em 5 das 7 bases, entre elas *Vehicle*, *Pima*, *WDBC*, *Banana* e *Lithuanian*. Já o método *DES-LCA2* (com *DES-FA*) conseguiu melhorar em 6 das 7 bases, piorando apenas na base *Vehicle*. Por fim, percebe-se que o *DES-LCA3* melhorou em apenas 2 das 7 bases, tendo resultados inferiores com as bases *Vehicle*, *Pima*, *Liver Disorders*, *Banana* e *Lithuanian*.

É importante esclarecer que isso pode ter acontecido devido ao filtro *ENN* provavelmente eliminar alguns padrões importantes nessas bases de dados. É interessante notar que a abordagem *DES-FA* chegou a piorar em torno de 8.3 pontos percentuais na base *Vehicle* para todos os métodos de seleção, o que fortalece a hipótese do filtro eliminar alguns padrões importantes principalmente nessa base. Portanto considerando apenas as 6 bases (sem o *Vehicle*), percebe-se que o *DES-LCA2* teve uma melhora em média de 3.3 pontos percentuais, enquanto que o *KNORA-E* melhorou em média 1.72 pontos percentuais, o *DES-OLA* melhorou em média 0.59 pontos percentuais, o *DES-LCA3* piorou em média 0.48 pontos percentuais e o *DES-LCA* piorou em média 0.56 pontos percentuais.

Tabela 5 Execução dos métodos DES com a abordagem DES-FA(1) e utilizando a regra do produto para combinação. O valor de cada célula é a média da precisão de classificação de três iterações, com exceção do Oracle* obtido pelo autor.

Base	KNORA-E	DES-OLA	DES-LCA	DES-LCA2	DES-LCA3	Oracle*
Vehicle	71,17	72,04	**72,43**	67,69	67,85	96,8
Pima	**74,39**	73,35	69,79	68,06	68,06	95,1
Liver Disorders	66,86	**68,60**	66,47	58,14	62,79	90,07
WDBC	**96,01**	95,77	95,19	94,84	95,19	99,13
Blood transfusion	**78,07**	77,18	77,63	72,10	72,19	94,2
Banana	89,13	**90,47**	90,00	55,53	56,40	94,75
Lithuanian	91,89	**92,78**	**92,78**	55,00	58,22	98,35

No experimento da **Tabela 5** (com abordagem *DES-FA* e produto como combinação). Pode ser visto que o *DES-OLA* foi o melhor em geral, pois apesar do *KNORA-E* e *DES-OLA* serem melhor em 3 das 7 bases cada e o *DES-LCA* ser melhor em 2 das 7 bases, o *DES-OLA* superou o *KNORA-E* em 4 das 7 bases com média de 0.38 pontos percentuais a mais. Enquanto que, o *KNORA-E* foi superior ao *DES-LCA* em média 0.46 pontos percentuais. Os métodos *DES-LCA2* e *DES-LCA3*, novamente tiveram resultados semelhantes e com os piores resultados dentre os métodos.

Comparando os resultados dos experimentos da **Tabela 4** e **Tabela 5**, pode ser visto que a combinação por produto foi melhor que a votação majoritária para todos os métodos com um aumento de pontos percentuais em média de: 0.14 para o *KNORA-E*, 0.51 para o *DES-OLA*, 0.14 para o *DES-LCA*, 1.41 para o *DES-LCA2* e 2.0 para o *DES-LCA3*.

Tabela 6 Execução dos métodos DES com a abordagem DES-FA(1) e utilizando a regra da média para combinação. O valor de cada célula é a média da precisão de classificação de três iterações, com exceção do Oracle* obtido pelo autor.

Base	KNORA-E	DES-OLA	DES-LCA	DES-LCA2	DES-LCA3	Oracle*
Vehicle	**77,57**	73,85	73,93	72,83	72,83	96,8
Pima	**73,18**	72,22	72,14	66,75	69,36	95,1
Liver Disorders	66,28	**67,64**	67,44	53,29	60,66	90,07
WDBC	**96,36**	95,89	94,48	94,25	94,25	99,13
Blood transfusion	**76,74**	76,02	73,44	68,18	67,29	94,2
Banana	**91,6**	91,33	91	47,13	47,93	94,75
Lithuanian	91,44	**93,11**	92,44	52,67	56,33	98,35

Para o experimento da **Tabela 6** (abordagem *DES-FA* e regra de produto para combinação), percebe-se que o *KNORA-E* foi o melhor entre todos os métodos, sendo superior em 5 das 7 bases. Porém o *DES-OLA* teve um resultado similar ao *KNORA-E* em todas as bases com exceção do *Vehicle*, que o *KNORA-E* teve em torno de 3.7 pontos percentuais a mais, e as bases *Liver Disorders* e *Lithuanian*, que o *DES-OLA* teve mais de 1.3 pontos percentuais a mais que o *KNORA-E*. Já o *DES-LCA* foi inferior ao *KNORA-E* em média 1.18 pontos percentuais. Os métodos *DES-LCA2* e *DES-LCA3* novamente tiveram os piores resultados.

Comparando os resultados dos experimentos das **Tabela 4** e **Tabela 6**, pode ser visto que a regra de combinação da média foi melhor que a votação majoritária, pois melhorou em média para todos os métodos com exceção ao *DES-LCA2*. Ao analisar os resultados da **Tabela 5** e **Tabela 6** com relação a **Tabela 4** nota-se que ambas as regras, produto e média, melhoraram consideravelmente seus resultados para a base *Liver Disorders* e pioraram para a base *Banana*.

Tabela 7 Tempo de execução dos métodos *DES* sem a abordagem *DES-FA*. O valor de cada célula é a média do tempo de processamento (em segundos) de três iterações.

Base	KNORA-E	DES-OLA	DES-LCA	DES-LCA2	DES-LCA3
Vehicle	125,58	76,75	40,77	**38,95**	50,48
Pima	184,58	98,55	72,51	**39,99**	92,36
Liver Disorders	137,3	64,47	48,68	**20,45**	58,08
WDBC	36,7	27,84	22,77	**22,41**	25,48
Blood transfusion	215,96	105,07	89,29	**40,58**	97,86
Banana	128,67	77,02	62,29	**49,78**	64,44
Lithuanian	94,15	54,39	42,04	**34,28**	45,02

Tabela 8 Tempo de execução dos métodos *DES* com a abordagem *DES-FA(1)*. O valor de cada célula é a média do tempo de processamento (em segundos) de três iterações.

Base	KNORA-E	DES-OLA	DES-LCA	DES-LCA2	DES-LCA3
Vehicle	119,31	72,38	41,22	**38,78**	43,83
Pima	90,61	56,85	42,23	**36,92**	48
Liver Disorders	73,6	43,19	30,61	**25,76**	39,88
WDBC	26,65	21,78	21,38	21,52	**21,11**
Blood transfusion	70,23	45,72	36,65	**31,56**	38,16
Banana	87,91	55,9	44,22	**43,32**	44,89
Lithuanian	61,41	35,99	28,57	**26,88**	29,4

Analisando o tempo de processamento dos métodos de seleção com e sem a abordagem *DES-FA*, na **Tabela 7** e **Tabela 8**, podemos perceber claramente que o *KNORA-E* é o método mais lento, enquanto que o mais rápido foi o *DES-LCA2*. O segundo método mais eficiente (tempo de processamento) foi o *DES-LCA*, em média 2.3 vezes mais rápido que o *KNORA-E*, depois foi o *DCE-LCA3* sendo em média 2.0 vezes mais rápido, e em seguida o *DES-OLA* com média 1.7 vezes mais rápido que o

KNORA-E. Percebe-se também que o uso da abordagem *DES-FA* reduziu bastante o tempo de processamento. Onde o *KNORA-E* reduziu em média cerca de 50% no tempo, o *DES-OLA* e *DES-LCA* cerca de 42%, o *DES-LCA2* cerca de 19% e o *DES-LCA3* cerca de 47% do tempo de execução.

4.4 Análise global

Tendo uma visão geral das análises da seção anterior vemos que o uso da abordagem *DES-FA*, filtro e distância adaptativa para a região de competência, influenciou diretamente nas precisões dos métodos de seleção dinâmica analisados. Pelos resultados, foi visto que em geral a abordagem *DES-FA* melhorou os resultados dos métodos *KNORA-E, DES-OLA e DES-LCA2*. Porém, o resultado em algumas bases, como o Vehicle, piorou com a abordagem *DES-FA* para alguns métodos. Isso pode ter ocorrido por causa da seleção do filtro *ENN* remover padrões importantes para a base.

Com relação ao desempenho dos métodos de seleção, foi visto que para:

- Seleção dinâmica sem uso da abordagem *DES-FA* (votação):
    - O *DES-LCA* foi o método que teve em geral o melhor desempenho, seguido pelo *DES-OLA*, e posteriormente o *KNORA-E*. Os métodos *DES-LCA2* e *DES-LCA3* obtiveram resultados insatisfatórios.
- Seleção dinâmica com uso da abordagem *DES-FA* (votação):
    - O *DES-OLA* teve o melhor desempenho comparando individualmente com os métodos, pois no geral o KNORA-E teve 3 dos melhores resultado enquanto que o *DES-OLA* teve 2. O *DES-LCA* teve o terceiro melhor desempenho com 2 melhores resultados no geral. Os métodos *DES-LCA2* e *DES-LCA3* obtiveram resultados insatisfatórios.
- Seleção dinâmica com uso da abordagem *DES-FA* (produto):
    - O *DES-OLA* teve o melhor desempenho comparando individualmente com cada método, em segundo foi o *KNORA-E* e depois o *DES-LCA*. Os métodos *DES-LCA2* e *DES-LCA3* obtiveram resultados insatisfatórios.
- Seleção dinâmica com uso da abordagem *DES-FA* (média):
    - O *KNORA-E* foi superior aos outros métodos em 5 das 7 bases, seguido por *DES-OLA* melhor em 2 das 7 e depois o *DES-LCA*.

Comparando o desempenho dos métodos de seleção de acordo com os métodos de combinação utilizados (votação majoritária, produto e média), foi visto que no geral que o desempenho dos métodos que utilizaram as regras de produto e média

melhoraram com relação a votação majoritária. Isso afetou principalmente a precisão de duas bases: *Liver Disorders*, melhorando significativamente, e a *Banana*, piorando.

Quanto ao tempo de processamento foi visto que é diretamente influenciado pela região de competência, pois o uso da abordagem *DES-FA* reduziu, em média, 40% do tempo de processamento. Também foi comparado o desempenho (tempo) dos métodos de seleção, em que o *DES-LCA2* mostrou-se o mais rápido, seguido por *DES-LCA* (em média 2.3 vezes mais rápido que o *KNORA-E*), depois o *DES-LCA3*, o *DES-OLA* (em média 1.7 vezes mais rápido que o *KNORA-E*) e por último o *KNORA-E*.

CAPÍTULO 5

Conclusão

Para melhorar a taxa de classificação em diversos sistemas o uso de sistemas de múltiplos classificadores tem sido bastante importante. Isso é possível devido à combinação das vantagens individuais dos classificadores. A abordagem de seleção dinâmica de classificadores tem mostrado ser um método de combinação bastante robusto quanto aos demais.

Baseado nessa abordagem, este trabalho propôs um estudo sobre o estado da arte, desde a geração do ensemble, os métodos de seleção, até as técnicas de combinação. Posteriormente, foi proposto um método de seleção dinâmica de ensemble (*DES*) baseado nos métodos de seleção dinâmica de classificador *DCS-LA* (*OLA*, *LCA* e duas variações propostas para o *LCA*), utilizando a abordagem *DES-FA* para melhorar as regiões de competências.

Para avaliar os resultados dos métodos de seleção propostos (*DES-OLA*, *DES-LCA, DES-LCA2* e *DES-LCA3*) várias experimentos foram realizados comparando com os métodos *KNORA-E* e *KNORA-E* com *DES-FA* utilizando várias combinações de ensemble (votação majoritária, produto e média).

Os resultados obtidos mostraram que os métodos *DES-LCA2* e *DES-LCA3* tiveram resultados de precisão de classificação insatisfatórios comparado com os outros métodos. Utilizando a abordagem *DES-FA*, o método *DES-OLA* mostrou resultados ligeiramente melhores para a votação majoritária e produto comparado ao método *KNORA-E*. Utilizando a média, o *KNORA-E* foi superior aos métodos propostos. Sem utilizar a abordagem *DES-FA*, o *DES-LCA* e o *DES-OLA* foram superiores ao *KNORA-E*. Também foi visto como a região de competência influência esses métodos pela abordagem *DES-FA*, mostrando melhorar o desempenho para três métodos, *KNORA-E*,

DES-OLA e *DES-LCA2*, e prejudicar para o *DES-LCA*, *DES-LCA3*. O trabalho mostrou também que a abordagem *DES-FA* reduziu em média 40% o tempo de processamento para os métodos. Quanto ao tempo de processamento, os métodos propostos todos foram melhores que o *KNORA-E*.

Para trabalhos futuros, é interessante investigar melhor a abordagem *DES-FA* para que a seleção não prejudique a região de competência, diminuindo a remoção de padrões importantes. Para os métodos *DES* propostos é importante avaliar outros métodos de busca de subconjuntos além do *IWSS*, e estudar outras métricas para avaliação do subconjunto de classificadores além da precisão local.

Referências

BREIMAN, L. Bagging Predictors. **Machine Learning**, v. 24, n. 2, p. 123–140, 1996.

BROWN, G. et al. Diversity creation methods: a survey and categorisation. **Journal of Information Fusion**, v. 6, n. 1, p. 5–20, 2005.

CRUZ, R.; CAVALCANTI, G.; REN, T. **A Method For Dynamic Ensemble Selection Based on a Filter and an Adaptive Distance to Improve the Quality of the Regions of Competence**. International Joint Conference on Neural Networks (IJCNN). San Jose: [s.n.]. 2011.

DOS SANTOS, E. M.; SABOURIN, R.; MAUPIN, P. A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. **Pattern Recognition**, v. 41, p. 2993–3009, 2008.

DUIN, R. P. W. **The combining classifier:** to train or not to train? Proceedings of the 16th International Conference on Pattern Recognition. [S.l.]: [s.n.]. 2002. p. 765–770.

FREUND, Y.; SCHAPIRE, R. E. **Experiments with a New Boosting Algorithm**. Proc. 13th International Conference on Machine Learning (ICML-96). [S.l.]: [s.n.]. 1996. p. 148–156.

GIACINTO, G.; ROLI, F. **Methods for dynamic classifier selection**. International Conference on Image Analysis and Processing (ICIAP 1999). [S.l.]: [s.n.]. 1999. p. 659–664.

HANSEN, L. K.; SALAMON, P. Neural network ensembles. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 12, n. 10, p. 993-1001, 1990.

HO, T. K. The random subspace method for constructing decision forests. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 20, n. 8, p. 832-844, 1998.

KITTLER, J. et al. On combining classifiers. **IEEE Trans. on Pattern Analysis and Machine Intelligence**, v. 20, n. 3, p. 226-239, 1998.

KO, A. H. R.; SABOURIN, R.; JR., A. S. B. From dynamic classifier selection to dynamic ensemble selection. **Pattern Recognition**, v. 41, p. 1735–1748, 2008.

KUNCHEVA, L. I. A theoretical study on six classifier fusion strategies. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 24, n. 2, p. 281-286, 2002.

KUNCHEVA, L. I.; BEZDEK, J. C.; DUIN, R. Decision templates for multiple classifier fu-sion: an experimental comparison. **Pattern Recognition**, v. 34, n. 2, p. 299-314, 2001.

PONTI-JR, M. P. **Combining Classifiers:** from the creation of ensembles to the decision fusion. SIBGRAPI. [S.l.]: [s.n.]. 2011. p. 1-10.

ROLI, F.; GIACINTO, G. Design of Multiple Classifier Systems. In: BUNKE, H.; KANDEL, A. **Hybrid Methods in Pattern Recognition**. [S.l.]: World Scientific Publishing, 2002. p. 199-226.

RUIZ, R.; RIQUELME, J. C.; AGUILAR-RUIZ, J. S. Incremental wrapper-based gene selection from microarray data for cancer classification. **Pattern Recognition**, v. 39, n. 12, p. 2383–2392, 2006.

SCHAPIRE, R. E. The Strength of Weak Learnability. **Machine Learning**, v. 5, n. 2, p. 197-227, 1990.

WANG, J.; NESKOVIC, P.; COOPER, L. N. Improving nearest neighbor rule with a simple adaptive distance measure. **Pattern Recognition Letters**, v. 28, p. 207–213, 2007.

WILSON, D. L. Asymptotic properties of nearest neighbor rules using edited data. **IEEE Transactions on Systems, Man and Cybernetics**, v. 2, n. 3, p. 408–421, 1972.

WOODS, K.; KEGELMEYER JR., W. P.; BOWYER, K. Combination of multiple classifiers using local accuracy estimates, v. 19, n. 4, p. 405-410, Abril 1997.

XU, L.; KRZYZAK, A.; SUEN, C. Y. Methods for combining multiple classifiers and their applications to handwriting recognition. **IEEE Transactions on Systems**, v. 22, n. 3, p. 418-435, 1992.