Universidade Federal de Pernambuco
Centro de Informática

Graduação em Engenharia da Computação

Classificação Otimizada em Bases Desbalanceadas Utilizando Algoritmos Evolucionários

Guilherme Ramalho Magalhães

Trabalho de Graduação

Recife
13 de dezembro de 2011
Universidade Federal de Pernambuco
Centro de Informática

Classificação Otimizada em Bases Desbalanceadas Utilizando Algoritmos Evolucionários

Guilherme Ramalho Magalhães
grm@cin.ufpe.br

Trabalho de Graduação apresentado no Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE) como requisito parcial para obtenção do grau de Engenheiro da Computação.

Orientador: George Darmiton da Cunha Cavalcanti

Recife
13 de dezembro de 2011

Dedico este trabalho à minha família, fonte de força e paz na minha vida.

“Whatever you do will be insignificant, but it is very important that you do it”.

Mahatma Gandhi

Agradecimentos

Uma série de agredecimentos faz-se necessária diante da conclusão desse trabalho. Agradeço primeiramente a Deus pelas oportunidades colocadas em minha vida; à minha família pela compreensão de minhas ausências e pela força nas horas mais difíceis; aos amigos que fiz durante a graduação, por compartilharmos das mesmas angústias, pela motivação e pelos momentos de descontração; aos professores e especialmente ao meu orientador por em nenhum momento duvidar das minhas capacidades e mostrar que sempre podemos fazer melhor. E finalmente agradeço a todas as pessoas que de alguma forma me trouxeram inspiração: "Se você realmente quiser alguma coisa, você conseguirá". São palavras como essas que tentarei levar comigo pelo resto da vida.

Resumo

A classificação em bases desbalanceadas é um dos desafios atuais da pesquisa em mineração de dados. Através de técnicas de seleção de instâncias (*IS – Instance Selection*) pode-se reduzir o tamanho do conjunto inicial de treinamento aumentando a eficiência dessa fase, reduzindo dados redundantes e diminuindo o erro de classificação através da remoção de dados ruidosos.

Entre as técnicas mais destacadas na seleção de instância encontra-se a abordagem com algoritmos evolucionários, que são métodos adaptativos baseados na evolução natural e podem ser utilizados para busca e otimização. A idéia básica é manter uma população de cromossomos, os quais representam soluções plausíveis para o problema e que evoluem com o tempo através de um processo de competição e variação controlada.

Nesse trabalho, damos uma visão geral de métodos de seleção de instâncias baseados em algoritmos evolucionários, e aprofundamos o estudo sobre uma das técnica do estado-da-arte, usando bases de dados desbalanceadas para a análise.

Palavras-chave: Seleção de Protótipos, Algoritmos Evolucionários, bases desbalanceadas.

Abstract

The classification in unbalanced data sets is a current research challenge in data mining. Through the use of techniques of instance selection (IS), we can reduce the size of the initial training set, increasing the efficiency of this phase, reducing redundant data and the classification error by removing noisy data.

Among the techniques of instance selection, one of the most prominent is the approach with evolutionary algorithms, which are adaptive methods based on natural evolution that can be used for search and optimization. The basic idea is to maintain a population of chromosomes, which represent plausible solutions to the problem and to evolve them over time through a process of competition and controlled variation.

In this paper we give an overview of methods for instance selection based on evolutionary algorithms, and deepen the study over a recent and prominent state of the art technique, using unbalanced data set for the analysis.

Keywords: Prototype Selection, Evolutionary Algorithms, Artificial Neural Networks.

Assinaturas

_____

Aluno: Guilherme Ramalho Magalhães

_____

Orientador: George Darmiton da Cunha Cavalcanti

Recife, 13 de dezembro de 2011.

Sumário

Lista de Figuras

Lista de Tabelas

Tabela de Siglas

Aqui fornecemos uma tabela com todos os acrônimos presentes no texto, com seu significado e a página na qual é definido.

Sigla	Significado	Página
CHC	Crossgenerational elitist selection, Heterogeneous recombination, and Cataclysmic mutation	07
EBUS	Evolutionary Balancing Under-Sampling	13
EGIS-CHC	Evolutionary Generalized Instance Selection by CHC	17
EIS-PS	Evolutionary Instance Selection – Prototype Selection	04
EUS	Evolutionary Under Sampling	13
EUSGCM	Under-Sampling guided by Classification Measures	13
GGA	Generational Genetic Algorithm	05
GOCBR	Global Optimization of feature weighting and instance election using genetic algorithms for Case Based Reasoning	15
HGA	Hybrid Genetic Algorithm	15
IGA	Intelligent Genetic Algorithm	09
IS	Instance Selection	05
OSS	One Sided Selection	25
PBIL	Population Based Incremental Learning	07
PS	Prototype Selection	02
SGP	Self Generating Prototypes	25
SMOTE	Synthetic Minority Over-sampling Technique	21
SSGA	Steady State Genetic Algorithm	06
SSMA	Steady State Memetic Algorithm	09

CAPÍTULO 1

Introdução

1.1 Contexto

Em classificação supervisionada, encontra-se muitos problemas do mundo real nos quais os dados não possuem uma distribuição equilibrada entre as diferentes classes do problema abordado. Exemplos de domínios com essa característica incluem telecomunicações, finanças, a web, biologia, medicina, etc, os quais são referenciados como bases de dados desbalanceadas (GARCÍA, DERRAC, *et al.*, 2012). Essas bases são caracterizadas quando o número de instâncias de uma classe é muito menor que o número de instâncias de outra(s) classe(s). As técnicas mais utilizadas para lidar com esse problema consistem em pré-processar os dados anteriormente ao processo de aprendizado, chamada de abordagem externa, ou optar por criar ou modificar algoritmos de aprendizagem existentes para levar em conta o problema de desbalanceamento, chamada de abordagem interna (HE e GARCIA, 2009). Uma mistura das duas abordagens também é comum muitas vezes obtendo resultados ainda melhores.

A classificação em bases desbalanceadas é um dos desafios atuais da pesquisa em mineração de dados (YANG e WU, 2006). Uma vez que algoritmos de aprendizagem tradicionais são desenvolvidos para minimizar a medida de erro global, a qual é independente da distribuição das classes, isso causa um *bias* sobre a classe majoritária no treinamento dos classificadores e resulta em uma sensibilidade mais baixa em detectar exemplos da classe minoritária.

Um dos processos principais em mineração de dados é conhecido como *data reduction*. O objetivo é reduzir o tamanho do conjunto de treinamento principalmente para aumentar a eficiência dessa fase, reduzindo dados redundantes e diminuindo o erro de classificação, removendo dados ruidosos. A seleção de instâncias (*IS – Instance Selection*) é uma das técnicas de redução mais empregadas em mineração de dados (LIU e MOTODA, 2001) e historicamente tem sido usada principalmente para melhorar a eficiência de classificadores do tipo *Nearest Neighbor* (CANO, HERRERA e LOZANO, 2003).

Quando o conjunto de instâncias selecionados é usado como referência para classificação estamos nos referindo à uma sub-área de IS chamada de seleção de protótipos (*PS - Prototype Selection*). Para selecionar esses protótipos podemos destacar os chamados de algoritmos de seleção evolucionária de instâncias que consideram o problema de seleção como sendo um problema de busca.

Algoritmos evolucionários são métodos adaptativos baseados na evolução natural que podem ser utilizados para busca e otimização. A idéia básica é manter uma população de cromossomos, os quais representam soluções plausíveis para o problema e que evoluem com o tempo através de um processo de competição e variação controlada. Muito do sucesso da aplicação desses algoritmos é devido a sua habilidade de explorar a informação acumulada sobre um espaço de busca complexo e inicialmente desconhecido, oferecendo uma abordagem válida para problemas que requerem técnicas eficientes de busca.

1.2 Objetivos

Muitas propostas da literatura confirmaram a eficácia das técnicas de seleção evolucionária, algumas focando em melhoramento da regra do vizinho mais próximo e outras no aumento da performance dos modelos gerados por algoritmos conhecidos de *data mining* (CANO, HERRERA e LOZANO, 2003).

A proposta desse trabalho é analisar os principais métodos de seleção de protótipos baseados em algoritmos evolucionários, e reproduzir uma das técnicas mais recentes presente em (GARCÍA, DERRAC, *et al.*, 2012) que utiliza classificação baseada em instâncias generalizadas, selecionadas por um modelo CHC em bases desbalanceadas, pré-processadas pela técnica SMOTE. Uma série de experimentos foi realizada sobre bases de dados públicas e dessa forma espera-se conhecer profundamente as possíveis falhas e vantagens da abordagem..

1.3 Estrutura do trabalho

Este documento divide-se em cinco capítulos: o Capítulo 2 faz uma revisão da literatura sobre o uso de algoritmos evolucionários em seleção de protótipos. A técnica EGIS-

CHC é analisada no Capítulo3. Os experimentos e seus resultados estão detalhados no Capítulo 4. Por último, o Capítulo 5 expõe as considerações finais e as propostas de trabalhos futuros.

CAPÍTULO 2

Revisão de Literatura

Este capítulo apresenta o estado-da-arte do uso de algoritmos evolucionário para seleção de protótipos. Uma breve descrição das técnicas mais importantes também é fornecida.

2.1 Visão Geral

A primeira notícia da aplicação de um algoritmo evolucionário em problemas de seleção de protótipos pode ser encontrado em (KUNCHEVA, 1995). Kuncheva aplicou algoritmos genéticos para selecionar um conjunto de referência (protótipos) para um classificador K-NN. Seu algoritmo genético mapeia o conjunto de treinamento em uma estrutura de cromossomo composta por genes, cada um com dois possíveis estados (representação binária). A função de aptidão (*fitness function*) mede a taxa de erro aplicando a regra dos K-vizinhos mais próximos. Posteriormente o algoritmo proposto sofreu modificações em (KUNCHEVA e BEZDEK, 1998; ISHIBUCHI e NAKASHIMA, 1999).

Até então, todos os algoritmos de EIS-PS (seleção evolucionária de protótipos) considerados adaptavam um modelo de algoritmo genético clássico para o problema de seleção de protótipos. Mais tarde, surgiram algoritmos específicos voltados para a seleção. O primeiro exemplo dessa nova classe pode ser encontrado em (SIERRA, LAZKANO, *et al.*, 2001), que utiliza um algoritmo de estimação de distribuição. Outro exemplo pode ser encontrado em (HO, LIU e LIU, 2002), onde um algoritmo genético é proposto para obter um classificador K-NN ótimo baseado em *arrays* ortogonais.

O termo técnico EIS-PS foi adotado por (CANO, HERRERA e LOZANO, 2003), no qual é analisado o comportamento de diferentes algoritmos evolucionários. Algoritmos Genéticos Geracionais (GGAs - *Generational Genetic Algorithms*), Algoritmos Genéticos de

Estado Fixo (SSGAs - *Steady-State Genetic algorithms*), o modelo CHC (ESHELMAN, 1990), Aprendizado incremental baseado em população (PBIL - *Population Based Incremental Learning*) (BALUJA, 1994), o qual pode ser considerado um dos algoritmos de estimação de distribuição básicos. A função de aptidão usada nesses modelos combina dois valores: Taxa de classificação (clasRat) utilizando um classificador 1-NN e a percentagem de redução de protótipos de *S* (Base reduzida) em relação a *TR* (Conjunto de Dados de Treinament) *percRed:*

$$Fitness(S) = \alpha \cdot clasRat + (1 - \alpha) \cdot percRed$$

Onde α é uma fator de peso geralmente configurado em 0.5, e *percRed* é definido como:

$$percRed = 100 \cdot (|TR| - |S| / |TR|)$$

Uma das abordagens mais recentes em EIS-PS emprega algoritmos meméticos (*memetic algorithms*) (ONG, KRASNOGOR e ISHIBUCHI, 2007) que são buscas heurísticas em problemas de otimização que combinam um algoritmo baseado em população com uma busca local. O algoritmo memético empregado em (GARCÍA, CANO e HERRERA, 2008) incorpora uma busca local *ad hoc* específicamente desenvolvida para otimizar a busca em problemas de seleção de protótipos com o objetivo de considerar a escalabilidade. Outra proposta recente (GIL-PITA e YAO, 2008), concentra-se no melhoramento da função de aptidão e dos operadores de mutação e *crossover* quando aplicado a problemas de seleção de protótipos.

Pesquisas posteriores foram mais além, focando seus interesses em outros tópicos que não melhorar o design de algoritmos de EIS-PS. Muitos esforços foram orientados em desenvolver métodos que serão capazes de enfrentar novos desafios em mineração de dados.

2.2 Técnicas de Seleção Evolucionárias de Protótipos

Nessa seção, descreveremos de forma detalhada os métodos mais representativos de seleção evolucionária de protótipos.

2.2.1 Generational Genetic Algorithm

Sua idéia básica é manter uma população de cromossomos, os quais representam soluções possíveis para o problema apresentado, evoluindo-a sobre uma sucessão de iterações (formando novas gerações) por um processo de competição e variação controlada. Cada cromossomo na população tem um valor de aptidão associado para determinar no processo de competição quais cromossomos serão utilizados para formar novos cromossomos. Tais cromossomos são criados utililizando operadores genéticos como crossover (recombinação) e mutação.

Em GGAs, o mecanismo de seleção produz uma nova população $P(t)$ com cópias dos cromossomos da antiga população $P(t-1)$. O número de cópias recebidas para cada cromossomo depende da sua aptidão – cromossos com aptidão maior têm uma chance maior de contribuir com cópias para $P(t)$.

O GGA foi o primeiro esquema desenvolvido para seleção de protótipos evolucionária e embora seus resultados em sua maioria sejam superados por propostas subsequentes, ainda representa um marco importante para a área de seleção de protótipos.

2.2.2 Steady-State Genetic Algorithm

O SSGA foi empregado como uma técnica de seleção evolucionária de protótipos pela primeira vez em (CANO, HERRERA e LOZANO, 2003). No SSGA geralmente uma ou duas populações filhas (*offspring*) são produzidas a cada geração. Os pais são selecionados para produzir os filhos e então é necessário decidir quais indivíduos na população serão removidos para abrir espaço para os novos filhos.

Na construção do SSGA, é possível escolher a estratégia de subistituição (subistituição dos piores, dos mais velhos ou de um indivíduo escolhido randômicamente) e a condição de substituição (e.g., susbtituição em caso do indivíduo ser melhor, ou subistituição incondicional). Uma combinação largamente usada é substituir o pior indivíduo apenas se o novo indivíduo for melhor (i.e., possui maior aptidão). Além disso, em (GOLDBERG e DEB, 1991), é sugerido que a remoção dos piores indivíduos pode induzir a uma alta pressão

seletiva, mesmo quando os pais são escolhidos randômicamente. Esta alta pressão seletiva pode ajudar o SSGA a melhorar a performance do GGA no processo de seleção de protótipos. Assim como aconteceu com o GGA, o desempenho do SSGA foi superado pela maioria propostas evolucionárias dos últimos anos.

2.2.3 Population-Based Incremental Learning

O PBIL (BALUJA, 1994) é um algoritmo de estimação de distribuição especificamente desenvolvido para espaços de busca binários que tenta manter estatísticas do espaço de busca para decidir onde amostrar em seguida. O objetivo do algoritmo é criar um vetor de Probabilidade Vp com valores reais, que quando amostrado, revela vetores solução de alta qualidade e com alta probabilidade. Inicialmente, os valores de Vp são fixados em 0.5. A amostragem a partir deste vetor gera uma vetor com uma solução randômica porque a probabilidade de gerar um 1 ou 0 para cada gene é igual. Com o progresso da busca, os valores de Vp gradualmente são deslocados para representar melhores vetores de soluções. PBIL é uma das propostas evolucionárias mais representativas para a seleção de protótipos, pois é um dos únicos métodos evolucionários no qual o processo de busca não é baseado em um algoritmo genético. Em geral o PBIL obtém bons resultados quando comparado com algoritmos genéticos, como no estudo presente em (CANO, HERRERA e LOZANO, 2003).

2.2.4 CHC

O algoritmo CHC (ESHELMAN, 1990) é um algoritmo genético codificado de forma binária que involve a combinação de uma estratégia de seleção com uma alta pressão seletiva, e alguns componentes que estimulam uma forte diversidade. É uma algoritmo evolucionário robusto, que quase sempre oferece bons resultados em muitos problemas de busca. Os 4 principais componentes do algoritmo são:

- **Uma seleção elitista**: Para forma uma nova geração, os melhores indivíduos entre pais e filhos são selecionados. No caso em que um pai e um filho tenham o mesmo valor de fitness, a preferência é dada ao filho.

- **Cruzamento (*crossover*) altamente disruptivo:** o HUX (*Half Uniform Crossover*), que cruza exatamente metade dos genes não emparelhados, onde os bits a serem trocados são escolhidos randômicamente sem subistituição.

- **Um mecanismo para prevenção de incesto**: o qual, apenas permite o cruzamento de pares de indivíduos que tiverem uma distância de hamming (número de genes diferentes entre os cromossomos) maior que um limiar préviamente estabelecido. O limiar diminui com o tempo para ajudar a população a convergir.

- **Um processo de reinicialização (Cataclysmic mutation)**: Substituindo a mutação comum de Algoritmos Genéticos, é aplicado quando a população convergiu (quando o limiar atinge o valor zero). Gera uma nova população alterando randômicamente uma percentagem, geralmente 35%, dos bits do indivíduo de maior aptidão da população antiga. A Figura 1 apresenta o pseudo código do CHC:

```
Pseudocódigo CHC
Crie população inicial: P
L := comprimento do cromossomo
N := Tamanho da população

threshold := L/4
Evolua até condição de parade ser satisfeita:
  CPop := {}
  Para i := 1 à N/2 faça:
    Escolha dois pais randômicamente: P1 e P2
    Se (bits diferentes entre P1 e P2) / 2 > threshold então:
      Criar filhos C1 e C2 usando HUX
      Adicionar C1 e C2 à CPop

  Se não houver filhos em Cpop então:
    threshold := threshold - 1
  Caso contrário:
    P := Melhores N indivíduos de P e CPop

  Se threshold < 0 então:
    Restart_Cataclysmic(P)
    threshold := L/4
```

Figura 1 - Pseudocódigo CHC

No estudo realizado em (CANO, HERRERA e LOZANO, 2003), o algoritmo CHC foi selecionado como a melhor estratégia, conseguindo sobrepujar os resultados dos outros métodos de seleção de protótipos (tanto dos evolucionário quando dos não-evolucionários).

Uma conclusão interessante derivada do estudo foi que a característica principal desse algoritmo é sua habilidade para selecionar as instâncias mais representativas independente de sua posição no espaço de busca, satisfazendo os objetivos de aumento da precisão e taxa de redução.

2.2.5 Algoritmos Genéticos Inteligentes

Ho, Liu e Liu (2002) propuseram o algoritmo genético inteligente (*Intelligent Genetic Algorithm - IGA*) baseado em um design ortogonal experimental, utilizado para seleção de protótipos e seleção de características. Apesar da definição inicial, também pode ser aplicado apenas para a seleção de protótipos, sem mudar os objetivos iniciais de aumento da precisão da classificação e das taxas de redução do conjunto de treinamento.

Uma algoritmo genético inteligente é um *GGA* que incorpora um operador de cruzamento (crossover) inteligente. O operador constrói um array ortogonal a partir de uma par de cromossomos pais e busca dentro do array pelos dois indivíduois mais adaptados de acordo com a função de aptidão. Leva cerca de $2^{log_z(y-1)}$ avaliações de aptidão para realizar uma operação de crossover inteligente onde y é o número de bits que diferem entre ambos os pais. Nota-se que a aplicação de um operador de crossover inteligente para cromossomos extensos (resultantes de um banco de dados grande) pode consumir um grande número de avaliações. O autor conclui que o emprego do operador de cruzamento inteligente permite o IGA ser superior aos algoritmos genéticos convencionais quando aplicado a problemas onde o espaço de soluções não é tão grande e complexo (i.e., padrões de baixa dimensionalidade e sem sobreposição) e bancos de dados de tamanho médio ou pequeno.

2.2.6 Steady-State Memetic Algorithm

O algoritmo memético de estado fixo (*Steady-State Memetic Algorithm* - SSMA) foi proposto em (GARCÍA, CANO e HERRERA, 2008) para cobrir uma desvantagem dos método convecionais de seleção evolucionária de protótipos até então conhecidos: a falta de convergência quando confrontavam problemas grandes e complexos. O SSMA faz uso de uma busca local ou *meme* especificamente desenvolvida para esse problema de seleção de protótipos. O entrelaçamento das fases de busca local e global permite que a duas influenciem uma a outra; i.e. o SSGA escolhe bons pontos iniciais, e a busca local provê uma representação precisa daquela região do domínio. Este esquema de busca local associa um valor de probabilidade para cada cromossomo por crossover e mutação, C_{novo} :

$$P_{LS} = \begin{cases} 1, & se\ Fitness(C_{novo})\ \text{é}\ melhor\ que\ Fitness(C_{pior}) \\ 0.625, & caso\ contrário \end{cases}$$

O mecanismo PLS é um médoto adaptativo baseado na aptidão. Uma descrição dele e o meme específicamente desenvolvido para a tarefa de seleção de protótipos pode ser encontrada em (GARCÍA, CANO e HERRERA, 2008). O mecanismo de otimização meme é uma busca local específicamente desenvolvida para o problema de seleção de protótipos, tentando melhorar o cromossomo inicial gerando seus vizinhos descartando um dos protótipos atualmente selecionado. Esses vizinhos são avaliados por uma função de *fitness* especial a qual pode consumir apenas avaliações parciais, economizando recursos computacionais para todo o processo evolucionário. O objetivo do mecanismo de otimização meme é ajustado dinamicamente na execução do SSMA. Cada vez que um certo número de avaliações é realizada, a precisão e as taxas de redução alcançadas pelo melhor cromossomo da população são registradas. Se a precisão da classificação não aumentou, então a otimização meme inicia a fase de melhoramento de precisão, onde apenas os melhores resultados em precisão são aceitos pela busca local. Em contrapartida, se a taxa de redução não aumentou, então a otimização meme inicia uma etapa para evitar a convergência prematura, na qual a busca local aceita soluções com baixa aptidão para aumentar a diversidade da população.

Na conclusão dos autores, o SSMA apresenta uma boa taxa de redução e tempo computacional. De fato, ele é capaz de sobrepujar os resultados dos algoritmos clássicos de seleção de protótipos, quando a precisão e taxa de redução são considerados, sendo bastante útil quando o tamanho da base de dados aumenta.

2.2.7 Algoritmo Genético baseado em erro médio quadrático, cruzamento clusterizado e mutação inteligente e rápida

Um novo algoritmo genético (baseado no modelo GGA) foi proposto em (GIL-PITA e YAO, 2008) como um modelo evolucionário para a seleção de protótipos. Este algoritmo inclui a definição de uma função de aptidão baseada no erro médio quadrático, uma técnica de crossover clusterizado e um esquema de mutação rápida e inteligente. A função de apitidão empregada é apresentada a seguir:

$$F = \frac{1}{NC}\sum_{n=1}^{N}\sum_{i=1}^{C}(\frac{K_n[i]}{K} - d_n[i])^2$$

Onde N é o número de padrões de treinamento, C é o número de classes, Ké o número de vizinhos mais próximos empregados, K_n é o número dos K vizinhos mais próximos pertencentes a classe i, e d_n é 1 quando a saída desejada para a instância n é a classe I, e 0 quando não. Como constatado por seus autores, a superfície de erro definida por essa função é mais suave do que aquela obtida utilizando funções baseadas em estimativa de contagem, tornando mais fácil encontrar o mínimo local. O operador de crossover formador de *clusters* primeiramente realiza uma clusterização dos cromossomos utilizando o k-means, extraindo o centróide de todos os clusters encontrados (o número de clusters é estabelecido de forma randômica). Dessa forma, os centróides são empregados com um operador de crossover clássico de ponto randômico produzindo os indivíduos da nova geração. O procedimento de mutação rápida e inteligente computa o efeito da mudança de um bit do cromossomo sobre sua função de aptidão, testando todas as possibilidades. A mudança que produz o melhor valor de aptidão é aceita como resultado do operador de mutação e é aplicado para cada indivíduo da população. Ao fim de sua aplicação, o valor de aptidão dos indivíduos é atualizado, iniciando então uma nova geração do processo evolucionário. O resultados obtidos pelos autores no estudo experimental realizado sugere que o uso conjunto dos três métodos propostos podem ser interessantes no caso de conjunto de treinamento não muito grandes.

2.3 Estratificação da seleção evolucionária para lidar com o problema da escalabilidade

Cano, Herrera e Lozano (2005), propuseram um método para lidar como o problema da escalabilidade em seleção evolucionária. O método apresentado consiste em uma estratégia estratificada que divide o conjunto de dados inicial em camadas dijuntas com distribuição de classes idêntica. O número de camadas escolhidas determina o tamanho de cada partição, a partir do tamanho do conjunto de dados. Utilizando um número de camadas apropriado esse método é capaz de reduzir significantemente o conjunto de treinamento, driblando os efeitos negativos do problema da escalabilidade. Seguindo a estratégia estratificada, o conjunto inicial D é dividido em t conjuntos dijuntos de mesmo tamanho D_j $(D_1, D_2, \dots, D_t)$, mantendo a distribuição das classes dentro de cada subconjunto. Logo, algoritmos de seleção de protótipos serão aplicados para cada D_j obtendo um subconjunto selecionado DS_j. Dessa forma, os subconjuntos TR e TS serão obtidos segundo:

$$TR = \bigcup_{j \in J} D_j \, , J \subset \{1,2, \dots, t\}$$

$$TS = D - TR$$

E o subconjunto estratificado de protótipos selecionados (SPSS) é definido como:

$$SPSS = \bigcup_{j \in J} DS_j \, , J \subset \{1,2, \dots, t\}$$

O classificador 1-NN é então avaliado utilizando como conjunto de treinamento SPSS, e TS como conjunto de teste. Logo o processo de classificação pode ser realizado em conjuntos de tamanho maior, evitando as desvantagens comuns e ainda alcançando resultados aceitáveis. A conclusão do estudo foi que uma escolha apropriada do número de camadas torna possível diminuir significantemente o tempo de execução e consumo de recursos, mantendo o comportamento do algoritmo de seleção evolucionária de protótipos quanto a precisão e percentual de redução. Também no estudo foi verificado que o CHC foi o melhor

algoritmo quando empregado com o processo de seleção evolucionária estratificada de protótipos.

2.4 Algoritmos de Seleção evolucionária em dados desbalanceados

Em (GARCÍA e HERRERA, 2009), um estudo completo de algortimos de EUS (*Evolutionary Under-Sampling*) é realizado. Um conjunto de métodos EUS é proposto, levando em consideração a natureza do problema e utilizando diferentes funções de aptidão, para obter uma boa troca entre equilíbrio das classes e performance. Oito algoritmos diferentes compoem o conjunto de métodos propostos no estudo. Além disso cada método compartilha da mesma estrutura básica, a qual é desenvolvida utilizando o algoritmo CHC como modelo evolucionário. Há três características ques os diferenciam:

- O Objetivo que tentam alcançar
    - Buscar por um equilíbrio ótimo dos dados sem perder efitividade na precisão de classificação. Modelos EUS que seguem essa tendência são chamados EBUS (Evolutionary Balancing Under-Sampling)
    - Buscar por um poder de classificação ótimo sem levar em conta o equilíbrio dos dados, considerando o último como um sub-objetivo que pode ser um processo implícito. Modelos EUS que seguem essa tendência são chamados EUSGCM (*Under-Sampling guided by Classification Measures*).
- A maneira que eles executam a seleção de instâncias
    - Se o esquema de seleção age sobre qualquer tipo de instância, então é chamado de seleção global. Isto é, o cromossomo contém o estado de todas as instâncias pertencentes ao conjunto de treinamento e remoções das instâncias da classe minoritária são permitidas.

- Se o esquema de seleção age apenas sobre as instâncias da classe majoritária então é chamado de seleção majoritária. Nesse caso, o cromossomo salva o estado das instâncias que pertencem à classe majoritária e a remoção de uma instância da classe minoritária não é permitida.

- A medida de precisão utilizada pela sua função de aptidão

  - São chamado de métodos de média geométrica se utilizam a média geométrica como medida de precisão. A média geométrica foi primeiramente empregada em (Barandela et al., 2003) e é definida como: $GM = \sqrt{a^+.a^-}$, onde a^+ denota a precisão em exemplos pertencentes a classe minoritária e a a^- em exemplos pertencentes a classe majoritária.

Também existem métodos que utilizam a área sob acurva ROC ao invés da média geométrica como medida de precisão. A AUROC (*Area under ROC Curve*) pode medir a eficácia de vários classificadores simultaneamente, utilizando as taxas de positivos reais (*true positives* - *tp*) e falsos positivos (*false positives* - *fp*) de um processo de classificação. Os oito métodos foram testados com várias bases de dados desbalanceadas, e os resultados obtidos foram contrastados utilizando procedimentos estatísticos não-paramétricos. As principais conclusões do estudo foram:

1. Algoritmos de seleção de protótipos não devem ser utilizados para tratar problemas desbalanceados. Eles são moldados para obter performance global eliminando exemplos pertencentes a classe minoritária considerados como ruído.

2. Durante o processo de EUS, o uso do mecanismo de seleção majoritária ajuda a obter subconjuntos de instâncias mais precisos dos que os obtidos pelo mecanismo de seleção global. No entanto, a seleção global é necessária para obter maiores taxas de redução.

3. Conjuntos de dados com uma taxa de desbalanceamento baixa devem ser resolvidos com modelos EUSGCM, e devem usar em particular o modelo com um mecanismo de seleção global e avaliação com a medida da média geométrica.

4. Conjuntos de dados com uma alta taxa de desbalanceamento foram melhor tratados com modelos EBUS, e devem usar em particular esse modelo com um mecanismo de seleção majoritária e avaliação por meio da medida da média geométrica.

2. 5 Abordagem mista de seleção evolucionária de protótipos

Nos últimos anos, algumas propostas apareceram realizando não apenas o processo de seleção de protótipos com algoritmos evolucionários, mas também realizando outro processo de preparação de dados. Nessa subseção revisaremos algumas dessas abordagens.

Ros, Guillaume, *et al.*, (2008) propuseram um algoritmo genético híbrido (*Hybrid Genetic Algorithm* - HGA), o qual realiza seleção de instâncias e seleção de características simultaneamente. Os objetivos são aumentar a precisão dos k-vizinhos mais próximos sobre o conjunto de referência, minimizar o número de características selecionadas (reduzindo os dados de referência) e maximizar o número de instâncias selecionadas (para reter o máximo de informação possível sem prejudicar a precisão de classificação). O HGA é dividido em 3 fases:

- Um algoritmo genético é aplicado na primeira fase. Inclui uma esquema de seleção sofisticado e alguns mecanismos para gerenciar a diversidade e o elitismo (incluindo um arquivo da população e uma análise dinâmica da diversidade dela).

- Empregando um histograma da frequência com a qual cada característica foi selecionada, um processo de seleção de características é conduzido de forma a simplificar o problema.

- O algoritmo genético é aplicado novamente sobre a população. Além disso, alguns filhos produzidos por cada geração são ajustados utilizando procedimento de busca local.

Apesar dos objetivos contraditórios no número de instâncias e características selecionadas, o HGA consegue ralizar um processo duplo de seleção de instância e de características com sucesso, sendo um método evolucionário apropriado para relizar a tarefa de redução de dados.

Uma outra abordagem mista, GOCBR (*Global Optimization of feature weighting and instance selection using genetic algorithms for Case Based Reasoning*) foi proposta em (AHN e KIM, 2006; AHN e KIM, 2009). Esta abordagem realiza uma seleção de instâncias e um processo de ponderação de características em um framework de um sistema de ressonância baseado em casos. O processo de busca do GOCBR consiste da aplicação de um algoritmo genético com os operadores comuns (seleção, crossover e mutação) onde seu indivíduos são representados de forma binária, codificando os pesos das características utilizando 14 bits para cada uma, e a informação de seleção de instâncias na segunda parte do cromossomo utilizando o esquema binário usual. A função de aptidão mede apenas a precisão obtida empregando apenas o conjunto de referência definido pelo cromossomo para classificar os dados de treinamento em um classificador KNN. Um sistema GOCBR foi aplicado com sucesso pelos autores em vários problemas, como classificação de perfis de consumo e modelos de predição de falência. Além disso, é memorável que seja o único método evolucionário conhecido até então que realize simultaneamente a seleção de instâncias e o processo de ponderação de características.

CAPÍTULO 3

EGIS-CHC

O método proposto em (GARCÍA, DERRAC, *et al.*, 2012) chamado EGIS-CHC (*Evolutionary Generalized Instance Selection by CHC*) é um melhoramento da técnica desenvolvida em (GARCÍA, DERRAC, *et al.*, 2011) aplicada a bases de dados desbalanceadas e pertence a família do *Nested Generalized Exemplar* (NGE) cujo aprendizado é alcançado armazenando hiperretângulos no espaço euclidiano n-dimensional. A classificação dos padrões de teste é realizada calculando sua distância para os "exemplares generalizados". O método é otimizado pela seleção dos exemplos generalizados mais apropriados através de algoritmos evolucionários (utilizando o modelo CHC). Uma análise experimental foi realizada sobre uma grande variedade de bancos de dados com uma alta taxa de desbalanceamento.

3.1 Aprendizado baseado em NGE

A teoria de NGE foi introduzida em (SALZBERG, 1991) e sua principal caracterísitca é o armazenamento de exemplos na memória permitindo que eles possam ser generalizados. Eles são fortemente relacionados ao classificador KNN e foram propostos como forma de extendê-lo. A generalização toma a forma de hiperretângulos no espaço euclidiano n-dimensional. No que diz respeito a cassificação baseada em instâncias, o uso de de generalizações aumenta a compreensão dos dados armazenados para realizar a classificação dos novos dados e acarreta uma compressão substancial dos mesmos, reduzindo os requisitos de armazenagem.

Exemplos generalizados de classes podem ser representado como hiperretângulos ou como instâncias simples. Em NGE, um conjunto inicial de pontos dado no espaço euclidiano n-dimensional é generalizado em um conjunto menor de hiperretângulos em relação ao número de elementos contidos. A escolha de como e qual hiperretângulo é generalizado a partir de um conjunto de pontos, depende da implementação do algoritmo própriamente dito.

A regra de classificação utilizado por esse tipo de método permite um certo grau de customização, se desejado. Falando de forma geral, o processo calcula a distância entre um exemplo dado e um exemplar generalizado guardado em memória. Chamaremos a instância a ser classificada de E e o exemplo generalizado de G, independetemente se G é um simples ponto ou um hiperretângulo. O modelo computa um *match score* entre E e G medindo a distância euclidiana entre dois objetos. A distância Euclidiana é bem conhecida quando G é um ponto simples. Quando G é um hiperretângulo, a distância é computada da seguinte forma:

$$D_{EG} = \sqrt{\sum_{i=1}^{M} \left(\frac{dif_i}{max_i - min_i}\right)^2},$$

$$dif_i = \begin{cases} E_{fi} - G_{superior}\ quando\ E_{fi} > G_{superior} \\ G_{lower} - E_{fi}\ quando\ E_{fi} < G_{inferior} \\ 0\ caso\ contrário, \end{cases}$$

Onde M é o número de atributos do dado, E_{fi} é o valor do i-ésimo atributo do exemplo, $G_{superior}$ e $G_{inferior}$ são os valores superior e inferior de G para um atributo específico (as arestas do retângulo) e max_i e min_i são o máximo e mínimo valor para o i-ésimo atributo nos dados de treinamento, respectivamente.

Nota-se que os pontos internos de um hiperretângulo têm distância 0 para aquele retângulo. Para o caso de retângulos sobrepostos, muitas estratégias podem ser seguidas, mas normamalmente é aceito que o ponto em uma região de sobreposição pertença ao retângulo menor (o tamanho é medido em termos do volume). O volume é calculado como:

$$Volume(G) = \prod_{i=1}^{M} (G_{superior} - G_{inferior})$$

3.2 Seleção Evolucionária com o modelo CHC

O problema de gerar um número ótimo de exemplos generalizados para classificar um conjunto de pontos é NP-difícil. Os exemplos generalizados produzidos podem ser

irrelevantes para classificação, por isso é necessário que o exemplos generalizados mais influentes sejam destinguidos. A idéia então é utilizar um algoritmo evolucionário como forma de selecionar os exemplos generalizados mais apropriados em domínios de classificação desbalanceados, aumentando a precisão de classificação através da seleção. Então é feita uma seleção evolucionária utlizando o modelo CHC, detalhado na seção anterior. Durante cada geração o CHC executa os seguintes passos:

- Utiliza a população pai de tamanho N para gerar uma população intermediária de N indivíduos, que são randômicamente emparelhados para gerar potencialmente N indivíduos filhos (o termo "potencialmente", é devido ao mecanismo de prevenção de incesto citado na seção anterior).

- Realiza uma competição por sobrevivência onde os melhores N cromossomos são selecionados para formar a proxima geração.

Assume-se um conjunto TR com P instâncias, cada uma delas com M atributos, um conjunto de instâncias generalizadas GS contendo N instâncias generalizadas com M condições (valores numéricos no intervalo [0,1]). Seja S um subconjunto de instâncias selecionadas a partir de GS, resultante da seleção do algoritmo. A seleção de instâncias generalizadas é considerada um problema de busca que é aplicado ao CHC. O espaço de busca associado é formado por todos os subconjuntos possíveis de instâncias generalizadas (Para um conjunto com n exemplos generalizados temos 2^n possíveis subconjuntos). Isso é conseguido utilizando-se uma representação binária dos cromossomos. Um cromossomo é composto por N genes (Um para cada amostra em GS) com dois possíveis estados: 0 e 1. Se o gene for 1, o exemplar generalizado associado a ele é incluído no subconjunto de GS que ele representa, caso contrário ele não é incluído. Uma vez que um subconjunto de GS pode ser codificado por um cromossomo, é definida uma função de aptidão baseada na área sob a curva ROC avaliada sobre TR:

$$Fitness(S) = \propto AUC + (1 - \propto) red\_rate$$

Onde AUC denota o valor computado utilizando TR como conjunto de teste e S como conjunto de treinamento. O valor de red_rate denota a razão dos exemplos generalizados selecionados e é calculada da seguinte forma:

$$red\_rate = 100.\frac{(|GS| - |S|)}{|GS|}$$

O objetivo do CHC então é maximizar a função de aptidão definida. Para $\propto$ é mantido o valor utilizado em (CANO, HERRERA e LOZANO, 2003) de 0,5. Para a classificação das instâncias de teste o mecanismo utilizado é o mesmo apresentado em (SALZBERG, 1991):

- A classe do Hiperretângulo mais próximo define a predição da classe do exemplo de teste.

- Se o padrão se apresenta em uma região de sobreposição de hiperretângulos, o de menor volume é escolhido para predizer a classe do padrão.

O conjunto inicial de hiperretângulos é gerado generalizando cada instância do conjunto de treinamento, encontrando para cada um deles os K – 1 vizinhos mais próximos sendo o K-ésimo vizinho um padrão de uma classe diferente. Em seguida, cada exemplar é expandido considerando os K-1 vizinhos mais próximos utilizando o maior e o menor valor como limites do intervalo. Uma vez que todos os exemplos generalizados são obtidos, os duplicados são removidos, fazendo |GS| ≤ |TR|. A partir desse conjunto de hiperretângulos é aplicado o CHC. Essa é uma heurística simples e rápida que obtém bons resultados.

3.3 Métrica de Avaliação de Performance

Nos experimentos realizados pelos autores comparou-se a abordagem do EGIS-CHC como os modelos de aprendizado baseado em exemplos generalizados mais representativos: BNGE (WETTSCHERECK e DIETTERICH, 1995), RISE (DOMINGOS, 1996) e INNER (LUACES e BAHAMONDE, 2003), e dois métodos de aprendizado por regra de indução: RIPPER (COHEN, 1995) e PART (FRANK e WITTEN, 1998), e uma grande quantidade de bases de dados desbalanceadas do repositótio KEEL (ALCALÁ-FDEZ, FERNANDEZ, *et al.*, 2011) foi utilizada para a análise.

Para medir a performance a métrica utilizada foi a curva ROC (*Receiver Operating Characteristic*) que é bastante útil no trato de domínios desbalanceados e que possuam custo de classificação diferentes por classe (BRADLEY, 1997), permitindo a vizualização do equilíbrio entre benfícios e custo. A área sob a curva ROC (AUC) corresponde a probabilidade de identificar corretamente quais de dois estímulos é ruído e qual é sinal mais ruído, provê um valor simples para medir a performance de algoritmos de aprendizagem. A área sob a curva em (GARCÍA, DERRAC, *et al.*, 2012) é obtida como:

$$AUC = \frac{1 + True\_Positive\_Rate - False\_Positive\_Rate}{2}$$

3.4 Pré-processamento dos dados

Os autores do EGIS-CHC ainda incluíram um estudo sobre o uso da técnica de pré-processamento SMOTE ("Synthetic Minority Over-sampling Technique") (CHAWLA, BOWYER, *et al.*, 2002) que equilibra a distribuição de exemplos de treinamento em ambas as classes como forma de lidar com o problema de desbalanceamento. Nessa abordagem a classe positiva (minoritária) sofre um *over-sampling*, tomando-se cada amostra e introduzindo exemplos sintéticos ao longo dos segmentos de reta que ligam qualquer um dos k vizinhos mais próximos da classe minoritária. O processo é mostrado na Figura 2.

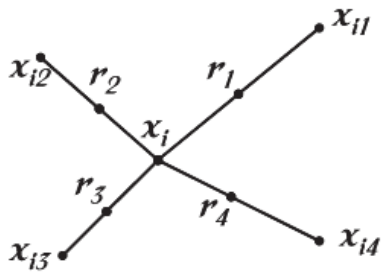

Figura 2 - Ilustração da criação de dados sintéticos com SMOTE. Retirada de (GARCÍA, DERRAC, *et al.*, 2012).

Onde x_i é um ponto qualquer, x_{i1} à x_{i4} são alguns vizinhos mais próximos selecionados e r_1 à r_4 são os pontos criados por um procedimento de interpolação randômico. Exemplos sintéticos são gerados tomando-se a diferença entre a amostra considerada e seu

vizinho mais próximo de mesma classe e adicionando à amostra essa diferença multiplicada por um número randômico entre 0 e 1. Para exemplificar:

$$r_2 = x_i + [0,1].(x_{i2} - x_i)$$

A justificativa para o uso da técnica é que ela força a região de decisão da classe minoritária a tornar-se mais geral. Uma visão geral do experimento é mostrada na Figura 3.

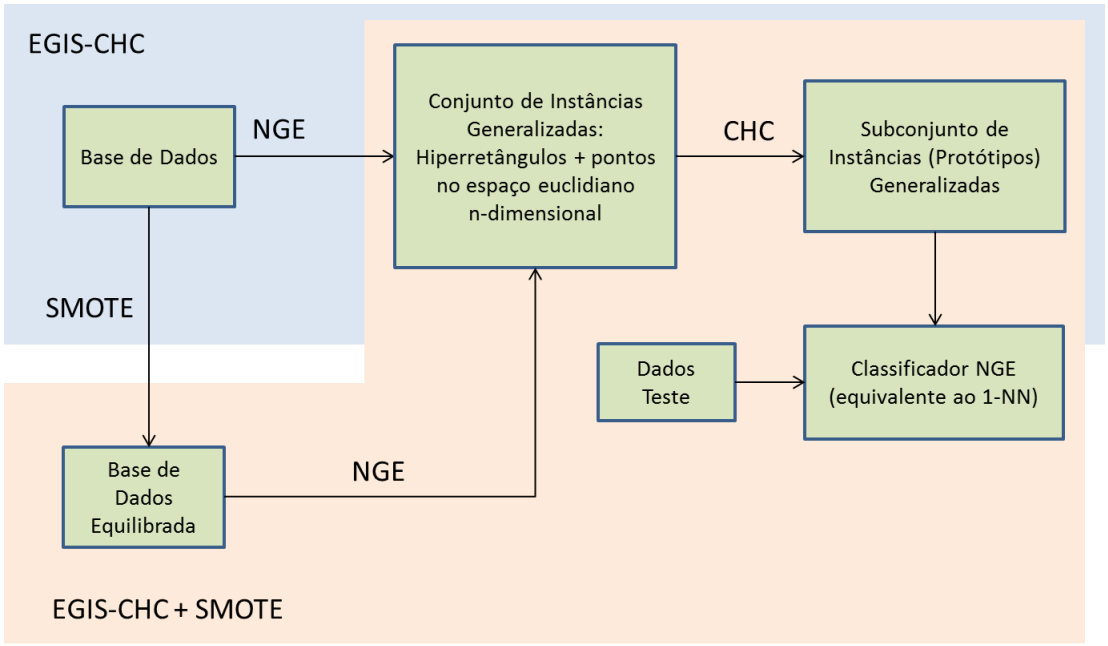

Figura 3 - Visão Geral de Experimento sobre o EGIS-CHC utilizando SMOTE.

CAPÍTULO 4

Análise Experimental

Nessa seção descrevemos a metodologia e os resultados dos experimetos realizados comparando a performance do EGIS-CHC com diferentes tipos de técnicas de *IS*.

4.1 Base de dados

Nos nossos experimentos o EGIS-CHC foi aplicado em cinco bases do repositório KEEL (ALCALÁ-FDEZ, FERNANDEZ, *et al.*, 2011). A Tabela 1 dá uma visão geral das dimensões e distribuição dos dados das bases e mostra a taxa de desbalanceamento (*IR* – *Imbalance Ratio*) obtida dividindo-se a quantidade de instâncias na classe majoritária pela quantidade de instâncias na classe minoritária.

Tabela 1 - Descrição das bases de dados desbalanceadas

Data sets	#Instâncias	#Atributos	Classe(min., maj.)	%Classe(min., maj.)	IR
glass0	214	9	(build-win-float-proc, remainder)	(32.71,67.29)	2.06
vehicle3	846	18	(opel, remainder)	(28.37,71.63)	2.52
newthyroid1	215	5	(hyper, remainder)	(16.28,83.72)	5.14
vowel0	988	13	(hid, remainder)	(9.01,90.99)	10.10
ecoli0137vs26	281	7	(positive, negative)	(2.49,97.51)	39.15

4.2 Configuração dos Experimentos

Os conjuntos de dados considerados foram normalizados e particionados utilizando o procedimento *5-fold cross validation*. Os parâmetros utilizados foram os mesmos presentes em (GARCÍA, DERRAC, *et al.*, 2012): Tamanho da população de 50 indivíduos, 10000 iterações e $\alpha = 0.5$. Os métodos estocásticos também foram executados três vezes com diferentes *seeds* para evitar a repetição de números pseudo-randômicos.

Como mostrado pelos autores o uso da técnica SMOTE como forma de pré-processamento para o EGIS-CHC afeta negativamente a sua performance possívemente por dois motivos:

- O aumento no número de instâncias na base de treinamento, produzindo consequentemente um aumento no número de genes codificados pelos cromossomos do CHC influencia a performance e o *trade-off* entre *exploration* (varrer o espaço de busca) e *exploitation* (refinar um resultado procurando um máximo ou mínimo).
- A inserção de instâncias ruidosas pelo mecanismo de interpolação do SMOTE, cria dados artificiais nas fronteiras de decisão, o que pode acarretar a formação de exemplos generalizados irrelevantes no processo de inicialização do EGIS-CHC.

Por essa razão não reproduzimos os experimentos com pré processamento dos dados.

4.3 Resultados

Tabela 2 - Resultado da execução da abordagem com 100 iterações sobre as bases de dados consideradas

Data sets	AUC médio	Taxa de Redução
glass0	0.6252	0.79592
vehicle3	0.52683	0.6284
newthyroid1	0.92857	0.80159
vowel0	0.75	0.61846
ecoli0137vs26	0.5	0.7561

Tabela 3 - Resultado da execução da abordagem com 10000 iterações sobre as bases de dados consideradas

Data sets	AUC médio	Taxa de Redução
glass0	0.7404	0.9374
vehicle3	0.7298	0.9755
newthyroid1	0.9944	0.9844
vowel0	0.9489	0.9781
ecoli0137vs26	0.8445	0.99

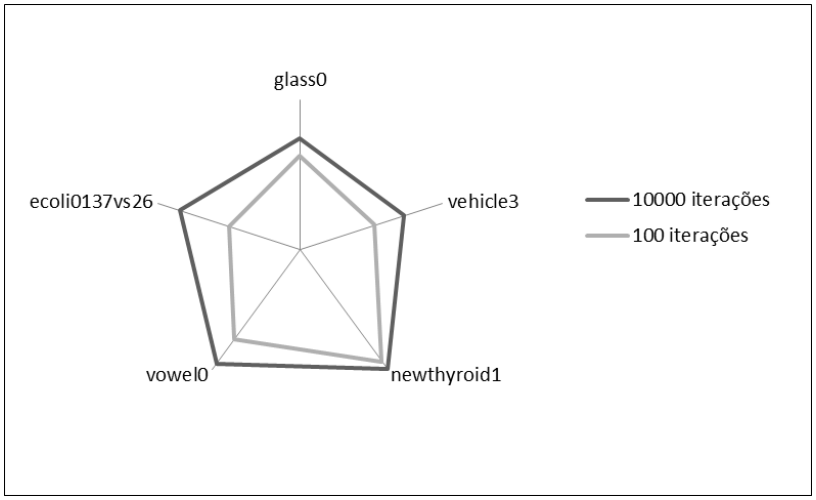

Figura 4 - Gráfico estrela comparando resultados entre a mesma abordagem com diferente número de iterações

Os resultados parciais com 100 e 10000 iterações do algoritmos EGIS-CHC, nas Tabelas 2 e 3 respectivamente, foram colocados para comentar a necessidade do parâmetro elevado para que o processo de busca do algoritmo não fique preso em mínimos locais.

Para avaliar o desempenho do EGIS-CHC comparamos os resultados de técnicas conhecidas de seleção de protótipos como *One Sided Selection* (*OSS*) (KUBAT e MATWIN, 1997) e *Self Generating Prototypes* (FAYED, HASHEM e ATIYA, 2007) como pré-processamento de um classificador 1-NN. Além disso, avaliamos ainda o desempenho do algoritmo CHC selecionando as instâncias mais representativas (ao invés de hiperretângulos) e o algoritmo de apendizado baseado em exemplos generalizados BNGE, como forma de avaliação das etapas do EGIS-CHC de forma isolada. Uma visão geral dos resultados é mostrada na tabela abaixo.

Data sets	EGIS-CHC		BNGE		OSS		SGP		CHC	
	AUC	SD	AUC	SD	AUC	SD	AUC	SD	AUC	SD
glass0	0.7404	0.0547	0.7333	0.0650	**0.8149**	0.0198	0.5165	0.0410	0.5241	0.0482
vehicle3	**0.7298**	0.0387	0.5778	0.0356	0.7244	0.0148	0.5530	0.0576	0.5	0.0
newthyroid1	**0.9944**	0.0076	0.9515	0.0539	0.9662	0.0343	0.8376	0.0480	0.5285	0.0571
vowel0	0.9489	0.0659	0.8377	0.0742	**0.9994**	0.0011	0.5916	0.0692	0.7126	0.0859
ecoli0137vs26	**0.8445**	0.2216	0.7500	0.0099	0.7466	0.1781	0.8135	0.1846	0.4981	0.0037

Tabela 4 - Resultados Comparataivos dos AUCs médios e o desvio padrão

Avaliamos também a redução de instâncias colocando o número final médio dos 5 *folds* para os exemplos generalizados no caso do EGIS-CHC, BNGE e instâncias pontuais nos métodos restantes. O resultado é mostrado na tabela abaixo.

Data sets	EGIS-CHC	BNGE	OSS	SGP	CHC
glass0	**9.2**	61.6	83.8	62.8	12.6
vehicle3	**16.2**	290.6	290.6	243.7	24.6
newthyroid1	**2.4**	8.8	37.84	17.2	4
vowel0	**14.2**	57.2	134.2	36.4	20.2
ecoli0137vs26	**2**	14.4	11.2	28	5.8

Tabela 5 - Média de exemplos generalizados ou instâncias selecionados

4.4 Análise Global dos Resultados

O EGIS-CHC obteve bons resultado e o menor número de instância generalizadas em todas as bases, o que implica em menor complexidade do modelo. Seu tempo de execução foi o maior entre os métodos testados e os resultados obtidos confirmam o desempenho do algoritmo mostrado em (GARCÍA, DERRAC, *et al.*, 2012).

O desempenho do CHC de forma isolada mostra que ele não é indicado para bases desbalanceadas, a menos que haja um pré-processamento dos dados ou uso de instâncias generalizadas como no EGIS-CHC. Como a função de aptidão para o CHC isolado utiliza a taxa de acerto e não a área sob a curva roc, era de se esperar que o resultado obtido ignorasse instâncias da classe minoritária e buscasse maximizar a taxa de acerto.

O OSS é uma técnica desenvolvida para lidar com bases desbalanceadas e também apresentou bons resultados de AUC médio, no entanto as matrizes de confusão geradas no experimento mostraram que esse algoritmo forma um *bias* forte sobre a classe majoritária aumentando a taxa de falsos positivos.

O SGP sem fator de generalização foi utilizado. Embora apresente boa taxa de acerto, mostrou péssimos resultados na matriz de confusão, classificando não menos que a metada dos padrões da classe minoritária erroneamente. É uma falha conhecida do SGP a sua falta de robustez diante de bases de dados desbalanceadas e o propósito de sua inclusão nesse estudo foi de fornecer mais algumas referências para o desempenho desse algoritmo na literatura.

4.4 Gráficos Comparativos

Incluímos representações dos resultados comparados em gráficos do tipo estrela. Estes gráficos representam a performance a partir da distância das pontas para o centro; logo, uma área maior representa um melhor AUC médio, permitindo comparar os algoritmos para cada conjunto de dados e de forma geral.

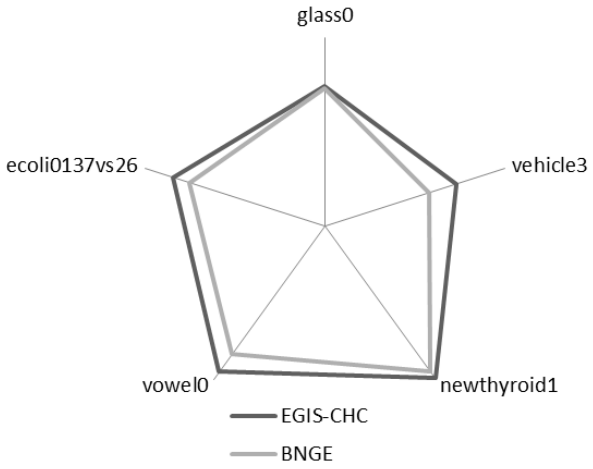

Figura 5- EGIS-CHC vs BNGE

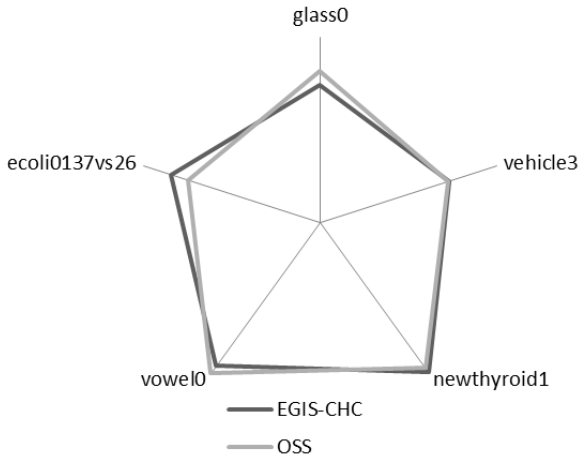

Figura 6 - EGIS-CHC vs OSS

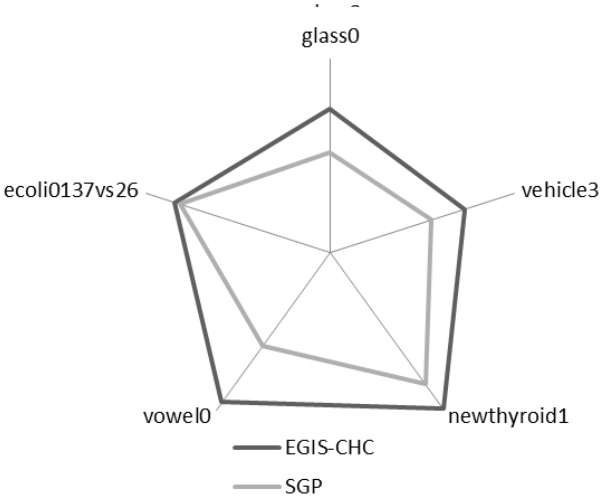

Figura 7 - EGIS-CHC vs SGP

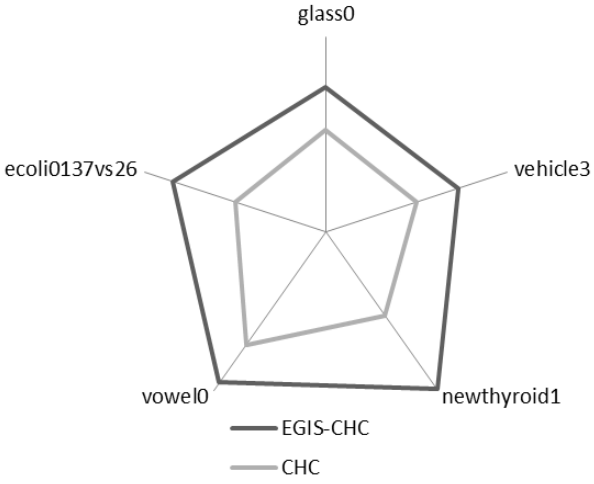

Figura 8 - EGIS-CHC vs CHC

CAPÍTULO 5

Conclusão

Os algortimos evolucionários mostram-se importantes ferramentas para superar problemas desafiadores como a escalabilidade dos dados e o desbalanceamento das classes em problemas de diferentes domínios como detecção de fraude, detecção de intruso, identificação biológica e médica.

Nesse trabalho discutimos brevemente uma série de algoritmos e propostas do estado-da-arte, mostrando que a pesquisa na área de seleção evolucionária de protótipos produziu diversos avanços nos últimos anos, melhorando a qualidade dessas técnicas.

O EGIS-CHC, o algoritmo analisado, mostrou-se uma técnica robusta frente ao problema do desbalanceamento e reduzindo a complexidade dos modelos formados.

Trabalhos Futuros

Para nosso experimentos utilizamos bases de dados de tamanho relativamente pequeno. As diretivas de pesquisas posteriores deverão focar também no estudo de escalabilidade de EIS-PS em bancos de dados extensos e complexos, seja utilizando estratégias de estratificação das bases ou desenvolvendo novas técnicas híbridas, mesclando algoritmos de *IS* clássicos e algoritmos evolucionários.

Também é interessante notar, na abordagem do EGIS-CHC, que uma leve modificação pode ser proposta para considerar problemas com mais de duas classe avaliando a área sob a curva roc gerada com subgrupos de classes dois-a-dois. Além disso, pode se concentrar esforços no melhoramento da seleção do conjunto de hiperretângulos iniciais do aprendizado NGE, utilizando possívelmente técnicas de *clustering*.

A verificação de técnicas de under-sampling da classe majoritária também é uma possível abordagem ainda não explorada constituindo dessa forma também uma nova diretiva de pesquisa.

Referências

AHN, H.; KIM, K. Hybrid genetic algorithms and case-based reasoning systems for customer classification. **Expert Systems: International Journal of Knowledge Engineering and Neural Networks**, v. 23, n. 3, p. 127–144, 2006.

AHN, H.; KIM, K. Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach. **Applied Soft Computing**, v. 9, n. 2, p. 599-607, 2009.

ALCALÁ-FDEZ, J. et al. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. **Journal of Multiple-Valued Logic and Soft Computing**, v. 17, n. 2–3, p. 255–287, 2011.

BALUJA, S. **Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning**. Carnegie Mellon University. Pittsburgh, PA. 1994.

BARANDELA, R. et al. Strategies for learning in class imbalance problems. **Pattern Recognition**, v. 36, n. 3, p. 849–851, 2003.

BRADLEY, A. P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. **Pattern Recognition**, v. 30, n. 7, p. 1145-1159, 1997.

CANO, J.; HERRERA, F.; LOZANO, M. Stratification for scaling up evolutionary prototype selection. **Pattern Recognition Letters**, 2005. 953–963.

CANO, J. ; HERRERA, F.; LOZANO, M. Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study. **IEEE Transactions on Evolutionary Computation**, v. 7, n. 6, p. 561-575, 2003.

CHAWLA, V. et al. SMOTE: Synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research**, n. 16, p. 321–357, 2002.

COHEN, W. **Fast effective rule induction**. Proceedings of the Twelfth International Conference on Machine Learning. [S.l.]: [s.n.]. 1995. p. 115–123.

DOMINGOS, P. Unifying instance-based and rule-based induction. **Machine Learning**, v. 24, p. 141–168, 1996.

EIBEN, A.; SMITH, J. **Introduction to Evolutionary Computing**. Berlin: Springer, 2003.

ESHELMAN, L. **The CHC adaptive search algorithm:** How to have safe search when engaging in nontraditional genetic recombination. Foundations of Genetic Algorithms. San Mateo, CA: Morgan Kaufmann. 1990. p. 265-283.

FAYED, H. A.; HASHEM, S. R.; ATIYA, A. F. Self-generating prototypes for pattern classification. **Pattern Recognition**, v. 40, n. 5, p. 1498-1509, 2007.

FRANK, E.; WITTEN, I. **Generating accurate rule sets without global optimization**. Proceedings of the Fifteenth International Conference on Machine Learning (ICML). [S.l.]: [s.n.]. 1998. p. 144–151.

GARCÍA, S. et al. Evolutionary selection of hyperrectangles in nested generalized exemplar learning. **Applied Soft Computing**, v. 11, n. 3, p. 3032-3045, 2011.

GARCÍA, S. et al. Evolutionary-based selection of generalized instances for imbalanced classification. **Knowledge-Based Systems**, v. 25, n. 1, p. 3-12, 2012.

GARCÍA, S.; CANO, J.; HERRERA, F. A memetic algorithm for evolutionary prototype selection: A scaling up approach. **Pattern Recognition**, v. 41, n. 8, p. 2693–2709, 2008.

GARCÍA, S.; HERRERA, F. Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. **Evolutionary computation**, v. 17, n. 3, p. 275-306, 2009.

GIL-PITA, R.; YAO, X. Evolving edited k-nearest neighbor classifiers. **International Journal of Neural Systems**, v. 18, n. 6, p. 1–9, 2008.

GOLDBERG, D.; DEB, K. A comparative analysis of selection schemes used in genetic algorithms. In: RAWLINS, G. **Foundations of genetic algorithms and classifier systems**. San Mateo, CA: Morgan Kaufmann, 1991. p. 69-93.

HE, H.; GARCIA, E. A. Learning from Imbalanced Data. **IEEE Transactions on Knowledge and Data Engineering**, v. 21, n. 9, p. 1263-1284, 2009.

HO, S. -Y.; LIU, C.; LIU, S. Design of an optimal nearest neighbor classifier using an intelligent
genetic algorithm. **Pattern Recognition Letters**, v. 23, n. 13, p. 1495–1503, 2002.

ISHIBUCHI, H; NAKASHIMA, T. **Evolution of reference sets in nearest neighbor classification**. Selected papers from the Second Asia-Pacific Conference on Simulated Evolution and Learning. [S.l.]: LNCS 1585. 1999. p. pp. 82-89.

KUBAT, M.; MATWIN, S. **Addressing the curse of imbalanced training sets:** one-sided selection. Proceedings of the 14th international conference on machine learning. [S.l.]: Morgan Kaufmann. 1997. p. 179–186.

KUNCHEVA, L. Editing for the k-nearest neighbors rule by a genetic algorithm. **Pattern Recognition Letters**, v. 16, n. 8, p. 809-814, 1995.

KUNCHEVA, L.; BEZDEK, J. Nearest prototype classification: Clustering, genetic algorithms, or random search? **IEEE Transactions on Systems, Man, and Cybernetics**, v. 28, n. 1, p. 160–164, 1998.

LIU, ; MOTODA, H. **Instance selection and construction for data mining**. [S.l.]: Springer, 2001.

LUACES, O.; BAHAMONDE, A. Inflating examples to obtain rules. **International Journal of Intelligent Systems**, v. 18, n. 11, p. 1113–1143, 2003.

ONG, Y.; KRASNOGOR, N.; ISHIBUCHI, H. Special issue on memetic algorithms. **IEEE Transactions on Systems, Man and Cybernetics**, v. 37, n. 1, p. 2-5, 2007.

ROS, F. et al. Hybrid genetic algorithm for dual selection. **Pattern Analysis & Applications**, v. 11, p. 179–198, 2008.

SALZBERG, S. A nearest hyperrectangle learning method. **Machine Learning**, v. 6, n. 3, p. 251-276, 1991.

SIERRA, B. et al. **Prototype selection and feature subset selection by estimation of distribution algorithms. A case study in the survival of cirrhotic patients treated with tips**. 8th Conference on Artificial Intelligence in Medicine in Europe. Cascais, Portugal: Springer. 2001. p. 20-29.

WETTSCHERECK, D.; DIETTERICH, G. An experimental comparison of the nearest neighbor and nearest-hyperrectangle algorithms. **Machine Learning**, v. 19, n. 1, p. 5-27, 1995.

YANG, Q.; WU, X. 10 challenging problems in data mining research. **Journal of Information Technology**, v. 5, n. 4, p. 597-604, 2006.

Apêndice

Introdução aos Algoritmos Evolucionários

Nessa seção apresentaremos um esquema geral que formam a base comum de todas as variações de algoritmos evolucionários. O conteúdo aqui foi adpatado do segundo capítulo de (EIBEN e SMITH, 2003).

Algoritmos Evolucionários

Existem diferentes variações de algoritmos evolucionários. A idéia pincipal por trás dessas técnicas é a mesma: dada uma população de indivíduos, a ação do ambiente causa uma seleção natural (sobrevivência do mais adaptado) o acarreta um aumento na aptidão da população. Dada uma função de qualidade a ser maximizada, pode-se criar randomicamente um conjunto de soluções candidatas, i.e., elementos do domínio da função, e aplicar a função de qualidade como uma medida abstrata de aptidão – quanto maior seu valor mais bem adaptado é o indivíduo. Baseado nessa função de aptidão, alguns dos melhores candidatos são escolhidos para dar origem a próxima geração, aplicando recombinação e/ou mutação neles.

Recombinação é um operador aplicado a dois ou mais candidatos selecionados (os chamados pais) e resultam em um ou mais novos candidatos (os filhos). Mutação é aplicada em um candidato e é aplicado e resulta em um novo candidato. Executar recombinação e mutação leva a um novo conjunto de candidatos (a cria/ offspring) que competem – baseado em sua aptidão (e algumas vezes idade) – com os mais velhos por um lugar na próxima geração. Esse processo pode se repetir até que um candidato com qualidade suficiente (uma solução) seja encontrado ou um limite para a computação definido préviamente seja atingido. Nesse processo há duas forças fundamentais que formam a base dos sistemas evolucionários:

- Operadores de variação (recombinação e mutação) criam a diversidade necessária e consequentemente facilitam a renovação da população.

- A seleção natural atua como força de elevação da qualidade da população.

A aplicação combinada de variação e seleção geralmente leva a um aumento das taxas de aptidão em populações posteriores. É fácil (embora um pouco enganoso), ver tal processo como se a evolução otimizasse, ou pelo menos aproximasse, chegando cada vez mais próximo de valores ótimos durante a execução. Alternativamente, a evolução ás vezes é vista como um processo de adaptação. Dessa perspectiva, a aptidão não é vista como uma função objetivo a ser otimizada, mas sim uma expressão de requisitos do ambiente. Atender esses requisitos de forma mais próxima implicam em uma maior viabilidade, refletida em um maior número de filhos. O processo evolucionário faz com que a população se adapte ao ambiente cada vez melhor.

Muitos componentes de um processo evolucionários são estocásticos. Durante a seleção, indivíduos mais adaptados tem uma chance maior de serem selecionados, mas tipicamente até o menos adaptados têm uma chance de ser pais ou de sobreviver. Para recombinação de indivíduos a escolha de quais pedaçoes (genes) serão recombinados é randômica. De forma similar para a mutação, os pedaçoes que serão modificados em uma solução candidata, e os novos pedaçoes que os substituirão, são escolhidos randomicamente.

1- Pseudocódigo geral de uma algoritmo genético

```
BEGIN
    INICIALIZE população com soluções candidatas randômicas;
    AVALIE cada candidato;
    REPITA ATÉ (CONDIÇÃO DE PARADA ser satisfeita) {
        1 SELECIONE PAIS;
        2 RECOMBINE  pares de pais;
        3 EXECUTE MUTAÇÃO na população resultante
        4 AVALIE novos candidatos
       5 SELECIONE indivíduos para a próxima geração.
     }
END
```

Algoritmos genéticos por esse esquema, fazem parte de uma categoria de algoritmos de tentativa-e-erro. A avaliação da função de aptidão representa uma heurística de estimativa da qualidade da solução encontrada e o processo de busca é guiado pelos operadores de variação e seleção.

Componentes dos Algoritmos Evolucionários

Algoritmos evolucionários tem um número de componentes , procedimentos ou operadores que devem ser especificados para definir um algoritmo evolucionário em particular. Os principais componentes são:

1. Representação (definição de indivíduos)
2. Função de aptidão
3. População
4. Mecanismo de seleção de pais
5. Operadores de variação, recombinação e mutação
6. Mecanismo de seleção de sobreviventes (Substituição)

Além disso, para termos um algoritmo funcionando o processo de inicialização e condição de parada devem também ser definidos.

Representação

O primeiro passo definindo um EA é fazer o link entre o contexto original do problema e seu espaço de solução. Objetos formando soluções possíveis dentro do contexto original do problema são chamados de fenótipos, sua codificação, os indivídus no algoritmos evolucionário são chamados de genótipos. A representação é a primeira etapa de desenvolvimento.

Funções de Aptidão (*fitness*)

O papel da função de aptidão é representar os requisitos aos quais uma possível solução deve atender. Isso forma a base para a seleção, e consequentemente facilita as melhorias. Mais precisamente, define o que as melhorias significam. Da perspectiva de

solução do problema, representa a tarefa a ser resolvida no contexto evolucionário. Técnicamente, é uma função ou procedimento que designa uma medida de qualidade de soluções no espaço evolucionário.

População

O papel da população é de carregar soluções possíveis. Uma população é um multi-conjunto de genótipos. A população forma a unidade de evolução. Indivíduos são objetos estáticos que não mudam ou se adaptam, é a população que o faz. Dada uma representação, definir uma população pode ser tão simples quanto especificar quantos indivíduos estão presentes nela, que chamamos, definir o tamanho da população. Na maioria dos EAs o tamanho da população é constante, não mudando durante uma busca evolucionária. A diversidade de uma população é uma medida de diferentes soluções presentes.

Mecanismo de seleção de pais

O papel da seleção de pais ou seleção de emparelhados é destacar entre os indivíduos baseados em sua qualidade, em particular, permitindo que os melhores indivíduos sejam pais da próxima geração. Um indivíduo é um pai se ele for selecionado para sofrer variação para que gere novos indivíduos. Junto com o mecanismo de seleção de sobreviventes, a seleção de pais é responsável pelas melhoras na qualidade da população.

Operadores de Variação

Mutação

Um operador de variação unário (apenas uma entrada) é geralmente chamado de mutação. É aplicado em um genótipo e produz um mutante “levemente modificado”, o filho

ou cria dele. Um operador de mutação é sempre estocástico: Sua saída – o filho – depende do resultado de uma série de escolhas randômicas. Nota-se que um operador unário arbitrário não é necessariamente visto como mutação.

Recombinação

Um operador de variação binário é chamado de recombinação ou *crossover*. Como indica o nome, o operador mescla informações de dois genótipos pais em um ou dois genótipos filhos. De forma similar à mutação, recombinação é um operador estocástico: A escolha de quais partes de cada pai serão combinadas, e como essas partes são combinadas, dependem de decisões randômicas. Operadores de recombinação utilizando mais de dois pais são matematicamente possíveis e fáceis de implementar, e mesmo não possuindo nenhum equivalente biológico, alguns estudos indicam que eles têm efeitos positivos na evolução.

O princípio da recombinação é simples: Emparelhando dois indivíduos com características diferentes mas desejáveis, podemos produzir uma cria que combina ambas características. Algoritmos evolucionários aceitam que alguns filhos terão combinações de traços indesejáveis, a maioria pode não ser melhor ou até mesmo pior que seus pais, e esperam que alguns tenham características melhoradas.

Mecanismo de Seleção de Sobreviventes (Subsituição)

O papel da seleção de sobreviventes, ou seleção do ambiente, assemelha-se a seleção de pais, mas acontece em um estágio diferente do ciclo evolucionário. O mecanismo de seleção é chamado depois da criação dos filhos dos pais que foram selecionados. Como mencionado antes o tamanho da população é constante, então uma escolha deve ser feita para saber quais indivíduos terão permissão de passar para a próxima geração. Essa decisão é baseada quase sempre em seus valores de aptidão, favorecendo aquele de maior qualidade, no entanto o conceito de idade também é usado. Em oposto a seleção de pais a qual é tipicamente estocástica, seleção de sobreviventes é quase sempre determinística, por padrão rankeando o

conjunto total de pais e filhos e selecionando o segmento superior (enviesado por aptidão), ou selecionando apenas os filhos (enviesado por idade).

Inicialização

A primeira população é semeada por indivíduos gerados randomicamentes. Em princípio, heurísticas específicas do problema podem ser usadas nessa etapa visando uma população inicial com maior aptidão. Se isso aumenta o custo do esforço computacional ou não depende muito da aplicação em questão.

Condição de parada

Podemos distinguir dois casos de condições de paradas apropriadas. Se o problema tem um nível de aptidão ótimo, provavelmente vindo de um ponto ótimo conhecido de uma função objetivo, então alcançar esse nível (talvez apenas como uma precisão dada e>0) deveria ser usado como uma condição de parada. No entanto, como EAs são estocásticos e não garantem encontrar um resultado ótimo, essa condição pode nunca ser alcançada e o algoritmo pode não parar nunca. Isso requer que a condição seja extendida para uma que certamente faça o algoritmo parar. As opções mais usadas nesse caso:

- Um tempo máximo de CPU é decorrido.
- O número total de avaliações de aptidão alcança um limite dado.
- Para um período de tempo dado (i.e., para um número de gerações ou avaliações de adaptação), as melhorias no valor de aptidão permanecem abaixo de um limiar dado.
- A diversidade da população desce para abaixo de um limiar dado.
- O critério de parada nesses casos é uma disjunção: Valor ótimo atingido ou condição x satisfeita. Se o problema não tiver um ótimo conhecido, então precisamos da disjunção, simplesmente uma condição das listadas acima ou uma similar que garanta que a execução do algoritmo pare.