

Avaliação de uma Abordagem Híbrida para Aprender com Classes Desbalanceadas: Resultados Experimentais com o Indutor CN2

Claudia R. Milaré^{1,2}, Gustavo E. A. P. A. Batista¹, André C. P. de L. F. de Carvalho¹

¹Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (USP)
Caixa Postal: 668 – CEP: 13560-970 – São Carlos – SP, Brasil

²Centro Universitário das Faculdades Associadas de Ensino
CEP: 13870-377 – São João da Boa Vista – SP, Brasil

cmilare@gmail.com, {gbatista, andre}@icmc.usp.br

Abstract. *There is an increasing interest in application of evolutionary algorithms to induce classification rules. This hybrid approach can aid in areas that classical methods to rule induction have not been completely successful. One example is the induction of classification rules in imbalanced domains. Imbalanced data occur when some classes heavily outnumber other classes. Frequently, classical Machine Learning classifiers are not able to learn in the presence of imbalanced data sets, outputting classifiers that always predict the most numerous classes. In this work we extend the experimental evaluation of the hybrid approach proposed in [Milaré et al. 2009] to deal with the problem of inducing classification rules in imbalanced domains.*

Resumo. *Há um interesse crescente na aplicação de algoritmos evolucionários para induzir regras de classificação. Essa abordagem pode ajudar em áreas que métodos clássicos para indução de regras não têm obtido tanto sucesso. Um exemplo é a indução de regras de classificação em domínios desbalanceados. Dados desbalanceados ocorrem quando algumas classes possuem um número bem maior de exemplos se comparado a outras classes. Geralmente, em Aprendizado de Máquina tradicional os indutores não são capazes de aprender na presença de conjuntos de dados desbalanceados. Estes indutores geralmente classificam todos os exemplos como sendo da classe que possui o maior número de exemplos. Neste trabalho nós estendemos a avaliação experimental de uma abordagem híbrida proposta em [Milaré et al. 2009] para resolver o problema de indução de regras de classificação em domínios desbalanceados.*

1. Introdução

Há um interesse crescente na aplicação de algoritmos evolucionários para induzir regras de classificação. Essa abordagem pode ajudar em áreas que métodos clássicos para indução de regras não têm obtido tanto sucesso. Um exemplo é a indução de regras de classificação em domínios desbalanceados.

Dados desbalanceados ocorrem quando algumas classes possuem um número bem maior de exemplos se comparado a outras classes. Geralmente, em Aprendizado de Máquina tradicional os indutores não são capazes de aprender na presença de conjuntos de

dados desbalanceados. Estes indutores geralmente classificam todos os exemplos como sendo da classe que possui o maior número de exemplos. Entretanto, em muitos domínios, as classes minoritárias são as classes de maior interesse, as quais são atribuídos os custos mais altos. Por exemplo, na detecção de transações fraudulentas em cartões de crédito e na telefonia, no diagnóstico de doenças raras e na predição de eventos climáticos, classificar erroneamente exemplos da classe minoritária é mais caro do que classificar erroneamente um exemplo da classe majoritária. Um classificador que simplesmente classifica todos os exemplos com sendo da classe majoritária é inútil.

Em [Milaré et al. 2009] é proposta uma abordagem híbrida para resolver o problema de indução de regras de classificação em domínios desbalanceados. Nessa abordagem, o problema de aprender com classes desbalanceadas é visto como um problema de busca, e portanto, um algoritmo evolucionário é utilizado para melhor buscar o espaço de hipótese. A abordagem proposta cria vários conjuntos de dados balanceados com todos os exemplos da classe minoritária e uma amostra randômica de exemplos da classe majoritária. Esses conjuntos de dados balanceados são fornecidos a sistemas de Aprendizado de Máquina tradicionais, que produzem como saída conjuntos de regras. Os conjuntos de regras são combinados em um repositório de regras e um algoritmo evolucionário é utilizado para selecionar algumas regras desse repositório para construir um classificador.

O método proposto em [Milaré et al. 2009] utiliza a técnica *under-sampling* para criar vários conjuntos de dados balanceados. *Under-sampling* elimina os exemplos da classe majoritária para criar conjuntos de dados balanceados, e portanto, pode descartar dados úteis que poderiam ser importantes para o processo de indução. A abordagem híbrida proposta em [Milaré et al. 2009] não possui esta limitação pois muitas amostras de dados são criadas e, portanto, aumenta a probabilidade de todos os dados serem utilizados no aprendizado. Outra técnica utilizada para tratar dados desbalanceados é *over-sampling*. *Over-sampling* artificialmente aumenta o número de exemplos da classe minoritária. O maior problema desta técnica é que ela supostamente aumenta a probabilidade de ocorrência de *overfitting*, já que faz muitas cópias dos exemplos da classe minoritária. O método proposto em [Milaré et al. 2009] evita *overfitting* pois limita o número de regras de cada classificador.

Neste trabalho, os experimentos realizados em [Milaré et al. 2009] são estendidos para utilizar o sistema de Aprendizado de Máquina CN2 [Clark and Niblett 1989] para induzir classificadores de vários conjuntos de dados balanceados.

Este trabalho está organizado da seguinte forma: na Seção 2 são apresentados alguns trabalhos relacionados; na Seção 3 é descrita a abordagem proposta; na Seção 4 são descritos os experimentos realizados; e finalmente, na Seção 5 algumas conclusões e trabalhos futuros são apresentados.

2. Trabalhos Relacionados

Nesta seção são descritos alguns trabalhos relacionados a classes desbalanceadas e métodos para combinar classificadores.

2.1. Classes Desbalanceadas

A precisão de classificação de muitos algoritmos de Aprendizado de Máquina é altamente afetada pela distribuição dos exemplos entre as classes. Como muitos sistemas de apren-

dizado são projetados para trabalhar com conjuntos de dados balanceados, eles geralmente falham na indução de classificadores capazes de prever a classe minoritária.

A investigação de alternativas para trabalhar de forma eficiente com problemas que envolvem classes desbalanceadas é uma área importante de pesquisa, pois conjuntos de dados desbalanceados podem ser encontrados em diversos domínios. Por exemplo, na detecção de fraudes em chamadas telefônicas [Fawcett and Provost 1997] e em transações de cartão de crédito [Stolfo et al. 1997], o número de transações legítimas é geralmente muito maior do que o número de transações fraudulentas; em análise de risco em seguro [Pednault et al. 2000], poucos clientes requisitam o prêmio de seguro em um determinado intervalo de tempo; e em marketing direto [Ling and Li 1998], o retorno é geralmente muito pequeno (em torno de 1%) para a maioria das campanhas de marketing.

Muitos trabalhos têm analisado o problema de aprender sobre conjunto de dados desbalanceados (por exemplo, [Pazzani et al. 1994, Ling and Li 1998, Kubat and Matwin 1997, Fawcett and Provost 1997, Kubat et al. 1998a, Japkowicz and Stephen 2002, Batista et al. 2004]). Entre as estratégias utilizadas por estes trabalhos, três abordagens principais são geralmente usadas:

- Aplicar diferentes custos para classificações incorretas: os custos mais altos são aplicados às classes minoritárias;
- *Under-sampling*: balancear artificialmente os dados de treinamento eliminando exemplos da classe majoritária;
- *Over-sampling*: balancear artificialmente os dados de treinamento replicando exemplos da classe minoritária.

A abordagem híbrida proposta em [Milaré et al. 2009] utiliza *under-sampling* como um passo intermediário para criar muitos conjuntos de dados balanceados. Outros métodos utilizam uma abordagem semelhante para tratar classes desbalanceadas. Por exemplo, [Chan and Stolfo 1998] divide os exemplos da classe majoritária em muitos subconjuntos não sobrepostos com aproximadamente o mesmo número de exemplos da classe minoritária. Cada subconjunto é combinado com os exemplos da classe minoritária para formar conjuntos de dados balanceados que são fornecidos a um algoritmo de aprendizado. Os classificadores obtidos são integrados utilizando *stacking* [Wolpert 1992]. Uma abordagem similar é proposta em [Liu et al. 2009], em que Adaboost [Freund and Schapire 1997] integra a saída de muitos classificadores induzidos a partir de conjuntos de dados balanceados tratados com *under-sampling*.

A abordagem híbrida difere de trabalhos previamente publicados, pois o interesse é criar classificadores simbólicos, isto é, classificadores que podem ser facilmente interpretados por humanos. Embora *ensembles* possam ser construídos a partir de diversos classificadores simbólicos individuais, o classificador final não pode ser considerado um classificador simbólico, pois este classificador não pode ser facilmente interpretado.

2.2. Combinando Classificadores

Muitos trabalhos na literatura descrevem alternativas diferentes para combinar bases de conhecimento. Uma abordagem direta para combinar bases de conhecimento é utilizar *ensembles* [Opitz and Maclin 1999, Dietterich 1997b]. Um *ensemble* é composto por um conjunto de classificadores individuais cujas previsões são combinadas para determinar

a classe a que pertence um novo exemplo. Geralmente, um *ensemble* é mais preciso que seus classificadores individuais. Apesar do ganho em desempenho, que geralmente é obtido quando se utiliza *ensembles*, a combinação de classificadores simbólicos resulta em um classificador final não simbólico. Dois exemplos bem conhecidos de *ensemble* são Bagging e Booting.

Bagging [Breiman 1996] é a técnica mais antiga e simples para criar um *ensemble* de classificadores. Essa técnica utiliza voto majoritário para combinar previsões de classificadores individuais e aplica a classe mais frequentemente predita como a classificação final.

Diferente de Bagging, em Boosting [Schapire 1990], cada exemplo de treinamento é associado a um peso. Este peso está relacionado com a taxa de acerto da hipótese induzida para aquele exemplo particular. Uma hipótese é induzida por iteração e os pesos associados com cada exemplo deve ser modificado.

Uma segunda abordagem para combinar bases de conhecimento é integrar o conhecimento gerado por diferentes classificadores em uma única base de conhecimento e, então, utilizar um método de seleção de regra para criar um classificador. Em [Prati and Flach 2005], os autores propuseram um algoritmo denominado ROCCER para selecionar regras baseado no desempenho das regras sobre o espaço ROC (*Receiver Operating Characteristic*). Outra técnica que utiliza uma abordagem semelhante é o algoritmo GARSS [Batista et al. 2006]. GARSS utiliza um algoritmo evolucionário para selecionar regras que maximizam a AUC (*Area Under Curve*). Ambos ROCCER e GARSS são utilizados no contexto de classificação associativa. Outros trabalhos que utilizam algoritmo evolucionário para seleção de regras que combinam conhecimento de uma grande base de conhecimento podem ser encontrados em [Ghosh and Nath 2004, Bernardini et al. 2008].

A metodologia geral de induzir diversos classificadores de várias amostras de dados e integrar o conhecimento destes classificadores em um classificador final foi inicialmente proposta por Fayyad, Djorgovski, e Weir [Fayyad et al. 1996]. Esta metodologia foi implementada no sistema RULLER e utilizada no projeto SKICAT, cujo objetivo foi catalogar e analisar objetos de imagens digitalizadas do céu. De acordo com os autores, essa metodologia foi capaz de gerar um conjunto de regras robusto. Além disso, o classificador induzido foi mais preciso do que astrônomos para classificar objetos cósmicos de fotografias. A metodologia adotada no sistema RULLER foi estendida no sistema XRULLER (eXtended RULER) [Baranauskas and Monard 2003] para utilizar conhecimento induzido de diferentes algoritmos.

3. Abordagem Proposta

A abordagem proposta em [Milaré et al. 2009] utiliza um algoritmo evolucionário para selecionar regras com o objetivo de maximizar a AUC para problemas que envolvam classes desbalanceadas. Algoritmos evolucionários são algoritmos de busca baseados na seleção natural e genética [Goldberg 1989]. Seu funcionamento envolve um conjunto de soluções potenciais (população), geralmente codificadas como uma sequência de bits (cromossomos). A evolução é realizada pela aplicação de um conjunto de transformações (geralmente, os operadores genéticos *crossover* e *mutação*), e avaliação da qualidade (*fitness*) das soluções.

Como previamente descrito, a abordagem é baseada na técnica *under-sampling*. Entretanto, para reduzir a probabilidade de perda de informação quando alguns exemplos são descartados, a abordagem cria diversos conjuntos de treinamento. Dado um conjunto de treinamento $\mathcal{T} = \mathcal{T}^+ \cup \mathcal{T}^-$, no qual \mathcal{T}^+ é o conjunto de exemplos positivos (minoritário), e \mathcal{T}^- é o conjunto de exemplos negativos (majoritário), n amostras randômicas $\mathcal{T}_1^-, \dots, \mathcal{T}_n^-$ são criadas de \mathcal{T}^- . Cada amostra randômica possui o mesmo número de exemplos do conjunto de exemplos positivos, isto é, $|\mathcal{T}_i^-| = |\mathcal{T}^+|$, $1 \leq i \leq n$.

No total, n conjuntos de treinamento balanceados \mathcal{T}_i são criados pela junção de \mathcal{T}^+ com cada \mathcal{T}_i^- , isto é, $\mathcal{T}_i = \mathcal{T}^+ \cup \mathcal{T}_i^-$, $1 \leq i \leq n$. O valor do parâmetro n nos experimentos foi definido igual a 100. Os conjuntos de regras foram induzidos de cada conjunto de treinamento \mathcal{T}_i , as regras de todos os conjuntos de regras foram integradas em um único repositório de regras, e as regras repetidas foram descartadas. Nos experimentos deste trabalho, o algoritmo de Aprendizado de Máquina CN2 [Clark and Niblett 1989] foi utilizado para induzir os conjuntos de regras.

O repositório de regras é fornecido como entrada ao algoritmo evolucionário. Uma chave primária (um número natural) é associado com cada regra do repositório. Portanto, cada regra pode ser acessada independentemente por esta chave. O método utiliza a abordagem Pittsburgh para codificar classificadores como cromossomos. Então, um vetor de chaves é utilizado para representar um cromossomo, isto é, um conjunto de regras interpretado como um classificador. Na implementação utilizada, a população inicial é randomicamente composta por 40 cromossomos. A função de avaliação utilizada é a métrica AUC medida sobre os exemplos de treinamento. O método de seleção é a seleção proporcional, no qual o número de vezes que um cromossomo é esperado reproduzir é proporcional ao seu *fitness*. Um operador de *crossover* simples foi aplicado com probabilidade de 0.4. O operador de mutação altera o valor de elementos do cromossomo randomicamente selecionados, ou seja, uma regra randomicamente selecionada do cromossomo é trocada por outra regra também randomicamente selecionada da base de regras. O operador de mutação foi aplicado com probabilidade de 0.1. As taxas com que os operadores de mutação e *crossover* são aplicados foram escolhidas baseadas na experiência prévia com algoritmos evolucionários [Milaré et al. 2004]. O número de gerações é limitado em 20. Finalmente, a implementação utiliza um operador de elitismo para reposição da população. De acordo com este operador, o melhor cromossomo de cada população é preservado para a próxima geração.

Como mencionado anteriormente, cada cromossomo representa um classificador. O número de regras de cada cromossomo foi fixado para evitar *overfitting*. Como a função de *fitness* é a métrica AUC sobre o conjunto de treinamento e se fosse permitido que o cromossomo crescesse durante a execução do algoritmo evolucionário, os cromossomos poderiam possuir muitas regras e não generalizar bem. Nos experimentos, o número de regras de cada cromossomo difere para cada conjunto de dados. Este número é aproximadamente o número médio de regras obtidas pelos classificadores induzidos sobre cada \mathcal{T}_i . Quando este número é grande, optou-se por limitar o tamanho do cromossomo em no máximo 40 regras por questões de tempo de processamento.

4. Avaliação Experimental

Como descrito anteriormente, neste trabalho foram realizados novos experimentos, utilizando o sistema de Aprendizado de Máquina CN2 [Clark and Niblett 1989], para avaliar a abordagem proposta em [Milaré et al. 2009]. Os experimentos foram realizados sob nove conjuntos de dados de *benchmark* coletados do repositório UCI [Asuncion and Newman 2007], e três conjuntos de dados do “mundo real”: Mammography [Chawla et al. 2002]; Oil-spill [Kubat et al. 1998b]; and Hoarfrost [Bucene 2008]. Esses conjuntos de dados são relacionados a problemas de classificação de diferentes domínios de aplicação. Na Tabela 1 são resumidas as principais características dos conjuntos de dados. Essas características são: Identificador – identificador do conjunto de dados utilizado no texto; #Exemplos – número total de exemplos; #Atributos(quant., quali.) – número de atributos e número de atributos quantitativos e qualitativos; Classes (min., maj.) – rótulo das classes minoritária e majoritária; e Classes % (min., maj.) – porcentagem das classes minoritária e majoritária. Os conjuntos de dados na Tabela 1 estão listados em ordem crescente de grau de desbalanceamento. Conjuntos de dados com mais de duas classes foram transformados em problemas de classificação binário tornando uma das classes como classe minoritária (como indicado na coluna Classes) e concatenando as outras classes como classe majoritária.

Table 1. Descrição dos conjuntos de dados.

Identificador	#Exemplos	#Atributos (quant., quali.)	Classes (min., maj.)	Classes % (min., maj.)
CMC	1473	9 (2, 7)	(1, restante)	(42.73%, 57.27%)
Pima	768	8 (8, 0)	(1, 0)	(34.89%, 65.11%)
German	1000	20 (7, 13)	(Bad, Good)	(30.00%, 70.00%)
Blood	748	4 (4, 0)	(1, 0)	(24.00%, 76.00%)
Vehicle	946	18 (18, 0)	(van, restante)	(23.89%, 76.11%)
Flare	1066	10 (2, 8)	(C-class, restante)	(17.07%, 82.93%)
Page-blocks	5473	10 (10, 0)	(restante, text)	(10.22%, 89.78%)
Satimage	6435	36 (36, 0)	(4, restante)	(9.73%, 90.27%)
Hoarfrost	3044	236 (200, 36)	(positive, negative)	(6.11%, 93.89%)
Oil-Spill	937	48 (48, 0)	(2, 1)	(4.38%, 95.62%)
Abalone	4177	8 (7, 1)	(15, restante)	(2.47%, 97.53%)
Mammography	11183	6 (6, 0)	(2, 1)	(2.32%, 97.68%)

Cada um dos doze conjuntos de dados foi dividido em 10 pares de conjuntos de treinamento e teste utilizando amostragem randômica estratificada. Os exemplos de treinamento são 70% do conjunto de dados original e o conjunto de teste 30%. Dentro de cada partição, 100 conjuntos de dados balanceados foram criados com todos os exemplos da classe minoritária e uma amostra randômica dos exemplos da classe majoritária, como descrito previamente.

Esses conjuntos de dados balanceados foram fornecidos ao sistema de Aprendizado de Máquina CN2. Este algoritmo foi executado com seus parâmetros *default*. As regras de classificação geradas para cada conjunto de treinamento foram combinadas em um repositório de regras. Um algoritmo evolucionário foi utilizado para selecionar um subconjunto de regras e construir um classificador a partir do repositório de regras.

Como descrito na Seção 3, o tamanho do cromossomo, isto é, o número de regras de um classificador individual foi definido como o tamanho médio dos classificadores gerados pelo indutor CN2. Entretanto, quando o tamanho médio era muito grande, o tamanho dos indivíduos (cromossomos) foi limitado a no máximo 40 por questões de tempo de processamento. A Tabela 2 mostra o tamanho dos cromossomos utilizados pelo algoritmo evolucionário para cada conjunto de dados.

Table 2. Tamanho dos cromossomos.

Conjunto de Dados	CN2
CMC	20
Pima	20
German	20
Blood	20
Vehicle	12
Flare	20
Page-blocks	20
Satimage	40
Hoar-frost	24
Oil-spill	10
Abalone	40
Mammography	20

Na Tabela 3 são apresentados os resultados obtidos com o indutor CN2. Todos os resultados são valores médios de AUC calculados sobre os 10 pares de conjuntos de treinamento e teste. Os resultados estão divididos em três colunas. Na coluna CN2 são apresentados os resultados obtidos com o indutor CN2 executado sobre os dados desbalanceados; na coluna Under-sampling são apresentados os resultados obtidos pelo CN2 sobre o conjunto de dados balanceado obtido pela técnica *under-sampling*; e, na coluna EA-CN2 são apresentados os resultados obtidos pela abordagem híbrida. Os melhores resultados estão em negrito. Como pode ser observado, EA-CN2 e Under-sampling apresentam cada um valor médio de AUC mais alto para cinco dos conjuntos de dados, e CN2 para dois conjuntos de dados. No entanto, EA-CN2 apresenta melhores resultados para conjuntos de dados com mais alto grau de desbalanceamento.

Para verificar se a diferença dos resultados obtidos entre as abordagens (CN2, Under-sampling e EA-CN2) é significativa ou não, foi utilizado o teste de hipótese *k-fold cross-validated paired t* [Dietterich 1997a]. Under-sampling apresentou resultados significativos para os conjuntos de dados CMC e German quando comparado ao CN2 e EA-CN2.

5. Conclusão e Trabalhos Futuros

Neste trabalho foi estendida a avaliação da abordagem híbrida proposta em [Milaré et al. 2009] para resolver o problema de indução de regras de classificação em conjuntos de dados desbalanceados. Esta abordagem combina indutores simbólicos de Aprendizado de Máquina e algoritmos evolucionários. Nesta abordagem é utilizado um algoritmo evolucionário para realizar uma busca mais extensiva sob o espaço de hipótese.

Table 3. Valor da AUC com o indutor CN2.

Conjunto de Dados	CN2	Under-sampling	EA-CN2
CMC	64.29 (02.35)	65.33 (01.31)	58.33 (01.63)
Pima	78.81 (02.39)	78.77 (02.95)	77.64 (02.59)
German	72.93 (01.94)	73.85 (02.62)	66.13 (03.47)
Blood	63.37 (04.12)	66.05 (04.81)	65.90 (02.90)
Vehicle	95.97 (02.43)	96.56 (01.60)	96.65 (02.17)
Flare	61.16(04.19)	66.63 (03.67)	65.32 (02.81)
Page-blocks	96.12 (01.11)	95.42 (00.82)	97.08 (00.91)
Satimage	90.59 (00.83)	90.67 (00.67)	89.97 (01.53)
Hoar-frost	91.08 (03.70)	92.59 (02.24)	92.81 (03.20)
Oil-Spill	84.25 (07.07)	82.83 (05.76)	81.73 (09.22)
Abalone	65.95 (02.41)	61.56 (06.69)	69.94 (02.74)
Mammography	87.05 (08.67)	90.60 (03.19)	91.57 (01.77)

A avaliação experimental realizada neste trabalho utilizou o indutor CN2 para gerar classificadores de conjuntos de dados balanceados. Os resultados obtidos não foram tão promissores quanto os resultados obtidos em [Milaré et al. 2009] utilizando os indutores C4.5rules e Ripper para gerar os classificadores dos conjuntos de dados balanceados. Para alguns conjuntos de dados utilizados nos experimentos, os classificadores gerados pelo CN2 eram muito grandes, às vezes com mais de 100 regras. Nestes casos, optou-se por um tamanho menor dos cromossomos (classificadores) do algoritmo evolucionário. Portanto, os classificadores encontrados pela abordagem híbrida, para muitos conjuntos de dados, são bem menores do que os classificadores gerados pelo CN2 e pelo método Under-sampling, o que pode ter contribuído com os resultados não tão bons obtidos pela abordagem híbrida.

Como trabalhos futuros pretendemos investigar novas métricas para compor regras para gerar um classificador. Estas métricas indicam quais regras devem disparar nos casos em que múltiplas regras cobrem um exemplo. Estas métricas possuem uma influência direta sobre o desempenho dos classificadores e novas métricas, projetadas para classes desbalanceadas, podem potencialmente melhorar a classificação sobre conjuntos de dados desbalanceados.

Agradecimentos. Esta pesquisa foi financiada pela FAPESP e CNPq.

References

- Asuncion, A. and Newman, D. (2007). UCI machine learning repository.
- Baranauskas, J. A. and Monard, M. C. (2003). Combining symbolic classifiers from multiple inducers. *Knowledge Based Systems*, 16(3):129–136. Elsevier Science.
- Batista, G. E. A. P. A., Milaré, C. R., Prati, R. C., and Monard, M. C. (2006). A comparison of methods for rule subset selection applied to associative classification. *Inteligência Artificial*, (32):29–35.
- Batista, G. E. A. P. A., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations Newsletter*, 6(1):20–29. Special issue on Learning from Imbalanced Datasets.

- Bernardini, F. C., Prati, R. C., and Monard, M. C. (2008). Evolving sets of symbolic classifiers into a single symbolic classifier using genetic algorithms. In *International Conference on Hybrid Intelligent Systems*, pages 525–530, Washington, DC, USA. IEEE Computer Society.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.
- Bucene, L. C. (2008). Mineração de Dados Climáticos para Alertas de Geadas e Deficiência Hídrica. PhD Thesis, FEAGRI/UNICAMP.
- Chan, P. K. and Stolfo, S. J. (1998). Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In *International Conference on Knowledge Discovery and Data Mining*, pages 164–168.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.
- Clark, P. and Niblett, T. (1989). The CN2 Induction Algorithm. *Machine Learning*, 3:261–284.
- Dietterich, T. G. (1997a). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7):1895–1924.
- Dietterich, T. G. (1997b). Machine Learning Research: Four Current Directions. Technical report, Oregon State University.
- Fawcett, T. and Provost, F. J. (1997). Adaptive fraud detection. *Journal of Data Mining and Knowledge Discovery*, 1(3):291–316.
- Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34.
- Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.
- Ghosh, A. and Nath, B. (2004). Multi-objective rule mining using genetic algorithms. *Information Sciences*, 163(1-3):123–133.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.
- Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449.
- Kubat, M., Holte, R., and Matwin, S. (1998a). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30:195–215.
- Kubat, M., Holte, R. C., and Matwin, S. (1998b). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2-3):195–215.
- Kubat, M. and Matwin, S. (1997). Addressing the course of imbalanced training sets: One-sided selection. In *International Conference in Machine Learning*, pages 179–186. Morgan Kaufmann.

- Ling, C. X. and Li, C. (1998). Data mining for direct mining: Problems and solutions. In *International Conference on Knowledge Discovery and Data Mining*, pages 73–79.
- Liu, X. Y., Wu, J., and Zhou, Z. H. (2009). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2):539–550.
- Milaré, C. R., Batista, G. E. A. P. A., and Carvalho, A. C. P. L. F. (2009). A hybrid approach to learn with imbalanced classes using evolutionary algorithms. In *Proc. The 9th International Conference Computational and Mathematical Methods in Science and Engineering (CMMSE 2009)*.
- Milaré, C. R., Batista, G. E. A. P. A., Carvalho, A. C. P. L. F., and Monard, M. C. (2004). Applying genetic and symbolic learning algorithms to extract rules from artificial neural networks. In *Proc. Mexican International Conference on Artificial Intelligence*, volume 2972 of *LNAI*, pages 833–843. Springer-Verlag.
- Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198.
- Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., and Brunk, C. (1994). Reducing misclassification costs. In *International Conference in Machine Learning*, pages 217–225.
- Pednault, E. P. D., Rosen, B. K., and Apte, C. (2000). Handling imbalanced data sets in insurance risk modeling. Technical Report RC-21731, IBM Research Report.
- Prati, R. C. and Flach, P. A. (2005). ROCCER: An algorithm for rule learning based on ROC analysis. In *International Joint Conference on Artificial Intelligence (IJCAI'2005)*, pages 823–828.
- Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5(2):197–227.
- Stolfo, S. J., Fan, D. W., Lee, W., Prodromidis, A. L., and Chan, P. K. (1997). Credit card fraud detection using meta-learning: Issues and initial results. In *AAAI-97 Workshop on AI Methods in Fraud and Risk Management*, pages 83–90.
- Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5:241–259.