

Predição do Desempenho do Aluno usando Sistemas de Recomendação e Acoplamento de Classificadores

Reginaldo Gotardo¹, Paulo Roberto Massa Cereda, Estevam Rafael Hruschka Junior¹

¹Departamento de Computação - Universidade Federal de São Carlos – São Carlos, Brasil

{reginaldo, estevam}@dc.ufscar.br, paulocereda@gmail.com

***Abstract.** In this paper we present an approach predict student performance using Recommendation System techniques in Environment Education Web. Our approach is based on the concept of Coupled-Learning. Based on an initial set of data we apply traditional machine learning algorithms to cooperate with each other, forming various views on its outputs and allowing the data set to be classified incrementally. In our approach, we use the incremental learning based on pairs in order to increase the initial training set in order to allow the generation of more recommendations which is the student performance.*

***Resumo.** Este trabalho apresenta uma abordagem para prever o desempenho de alunos usando técnicas de Sistemas de Recomendação num ambiente educacional. A abordagem é baseada no conceito de Acoplamento, usando um conjunto inicial de dados são aplicadas técnicas tradicionais de aprendizado de máquina que cooperam entre si, formando visões diferentes sobre o resultado. Estes resultados são avaliados e usados de maneira incremental para melhorar o desempenho do classificador.*

1. Introdução

A maneira como se tem aprendido (e ensinado) passou e passa por transformações ao longo dos últimos séculos através das novas formas de comunicação e pela revolução tecnológica. Dois produtos destas transformações são: o Ensino a Distância e o Ensino Eletrônico. O ensino a distância, diferente do ensino eletrônico, não foi inspirado apenas pelo surgimento de novas tecnologias, mas sim, pelo enfrentamento das dificuldades de locomoção ou disponibilização de tempo das pessoas para a educação. O ensino eletrônico, cujas características fundamentais são a existência de um projeto instrucional bem definido e a dependência de algum meio tecnológico de interação com o usuário, surgiu como ferramenta de auxílio à aprendizagem mediada por computador, os sistemas CAI (Instrução Assistida por Computador). O ensino eletrônico abrange não apenas os STI (Sistemas Tutores Inteligentes) - que garantem a adaptação de conteúdos e práticas pedagógicas -, mas também toda forma de comunicação que visa a aprendizagem, valendo-se de recursos hipermídia e de interatividade.

Atualmente, estas abordagens são combinadas na forma de sistemas que utilizam recursos e ferramentas existentes na Web. Tais sistemas são denominados Sistemas Educacionais Baseados na Web. A educação baseada na Web (*Web-based Education - WbE*) tem sido usada para auxiliar processos de educação presencial e à distância. Cada

vez mais o uso de sistemas inteligentes de apoio tem ganhado papel importante neste processo. A Mineração de Dados em Sistemas Educacionais permite fornecer aos estudantes recursos como: recomendação de atividades, personalização do aprendizado, sugestão de itens interessantes, caminhos diferentes ou atalhos de acordo com o perfil deste estudante. Já aos educadores, a Mineração de Dados pode prover feedback das tarefas, análise do comportamento dos estudantes, predição de performance, criação de grupos de alunos com características afins, descoberta de padrões irregulares ou indesejáveis e determinação de atividades mais efetivas nos cursos [da Silva, Mendes Neto e Jácome Jr 2011] [Ferro, et al. 2011].

A avaliação e manutenção dos cursos, a organização dos recursos institucionais (materiais e humanos) e a melhoria do processo de tomada de decisão também são benefícios que podem ser obtidos pela Mineração de Dados em Sistemas Educacionais. Sistemas de Recomendação possuem um papel bem definido: recomendar algo a alguém. Através de técnicas de Inteligência Artificial como Mineração de Dados e Aprendizado de Máquina é possível gerar sistemas de recomendações para tratar grandes conjuntos de dados para muitos usuários.

Este trabalho apresenta uma abordagem que usa algoritmos de aprendizado acoplados para integrar diferentes técnicas de aprendizado e mineração de dados e explorar um conjunto de dados educacionais oferecendo recomendação sobre o desempenho do aluno. Os experimentos realizados demonstraram que a abordagem proposta auxilia a obtenção de recomendações sobre o desempenho num ambiente com poucos dados sobre os alunos através da geração de modelos de classificação. Além disto, a análise e visualização destes dados, posteriormente numa ferramenta específica para o professor ou tutor, poderá melhorar a tomada de decisão sobre os métodos educacionais e as necessidades para o melhor desempenho da turma.

2. Sistemas de Recomendação, Tecnologias Educacionais e Mineração de Dados

Um sistema de recomendação é uma nova abordagem para ajudar usuários a encontrar informações relevantes. Basicamente, todo sistema de recomendação faz o mesmo: tenta identificar os itens mais importantes para os usuários e então os apresenta a eles [Shardanand e Maes 1995]. Basicamente, um sistema de recomendação visa responder a uma função de relevância/utilidade para o usuário alvo. Mais formalmente de acordo com Adomavicius e Tuzhilin (2005), o problema de recomendação pode ser formulado como sendo A o conjunto de todos os usuários do sistema e I o conjunto de itens que podem ser recomendados (por exemplo livros, filmes, produtos eletrônicos, objetos de aprendizagem, pessoas, etc). Tanto o conjunto A quanto o I podem ser muito grandes (com milhares ou milhões de elementos). Considere u uma função de *utilidade* que mede quão útil o item i é para o usuário a :

$u: A \times I \rightarrow R$, onde R é um conjunto limitado e conhecido (por exemplo, valores de 0 a 5). O que o sistema precisa é encontrar e escolher o item $i' \in I$ com maior utilidade para o usuário a :

$$\forall a \in A, i'_a = \arg \max_{i \in I} u(a, i)$$

Em geral, nos sistemas de recomendação, esta função de utilidade é um *rating* que indica como um usuário em particular gostou de um item. Por exemplo, João gostou do

filme Superman, pois deu nota 5 (variando de 0 a 5). Este *rating* pode ser um valor explícito dado pelo usuário ou calculado pela aplicação com base em métricas (por exemplo, pode ser o resultado de uma avaliação, pode representar aprovado ou reprovado numa disciplina, etc). Os *ratings* referem-se às avaliações anteriormente feitas pelos usuários e, por isto, é necessário calcular a estimativa de um rating para um item que o usuário não viu para que as recomendações possam ser feitas. Veja o exemplo na tabela 1 onde “-“ representa itens que o usuário não avaliou. As formas de se estimar (ou prever) os valores de rating são, basicamente:

1. Recomendações Baseadas em Conteúdo
2. Recomendações Baseadas em Colaboração
3. Abordagem Híbrida

Tabela 1. Exemplo de Avaliações Feitas por Usuários a Itens

Usuários	Itens							
	Item1	Item2	Item3	Item4	Item5	Item6	Item7	Item8
A1	5	-	-	4	5	1	-	-
A2	5	-	4	4	-	1	-	1
A3	-	4	-	1	1	4	-	4
A4	5	-	-	3	4	1	4	-
A5	-	2	1	-	4	1	-	-
A6	3	5	2	-	-	-	4	4

Com os avanços na área as técnicas de recomendação se dividiram e incorporaram novos conceitos distribuindo-se da seguinte forma [Ricci, Rokach e Shapira 2010]:

1. Baseado em Conteúdo: visa recomendar itens similares a outros que o usuário gostou no passado
2. Filtragem Colaborativa: visa recomendar itens de acordo com preferências de usuários similares ou de itens similares
3. Demográfico: baseado em informações demográficas do perfil do usuário
4. Baseado em Conhecimento: tipicamente baseados em casos anteriores
5. Baseado em Comunidade: visa utilizar informações sobre relações entre os usuários
6. Híbridos: reúne mais de uma das técnicas apresentadas

Os usuários e itens possuem conteúdo associados a eles. Cada elemento do conjunto *A* pode ser definido por um *perfil* ou *modelo de usuário* que inclui várias características do usuário. O mesmo vale para as informações dos itens no conjunto *I*. Um modelo de usuário (MU) representa explicitamente propriedades de um determinado usuário e suas informações num sistema que podem ser usadas para realizar a recomendação [Brusilovsky e Millan 2007].

Neste trabalho é apresentada uma variação da Filtragem Colaborativa usando dados demográficos. Os dados do usuário serão seu id e um conjunto de descritores sobre seu comportamento no sistema no acesso a um curso, enquanto o valor de rating é determinado pela aprovação ou não do usuário no curso. A filtragem colaborativa mede a similaridade entre usuários ou itens para então gerar a recomendação com base na

predição dos melhores resultados. Aqui a predição será calculada com base num classificador, pois usamos a classificação para gerar recomendações. A coleta de informações foi feita com base no log de acessos do sistema e com base em informações dos usuários e cursos.

A Educação Baseada na Web (EBW do termo inglês *Web-based Education*) popularizou-se no final do ano de 1996. Diversos Sistemas Educacionais Baseados na Web (definidos às vezes como Educação Mediada pela Web ou Educação baseada na Web) como *Sakai*, *Moodle*, *Blackboard*, dentre muitos outros, possuem recursos e características similares que visam, basicamente, a manutenção de um processo de ensino-aprendizagem através da Web. Este processo pode ser presencial, semipresencial ou à distância, e é apoiado por ferramentas como chats, fóruns, comunicadores instantâneos, agendas, *dropbox*, páginas pessoais, conteúdos multimídia e hiperlinks, criação de comunidades virtuais, entre outros. Estas ferramentas tratam da comunicação, da cooperação e da troca de conteúdos entre os participantes do curso e os tutores (Minaei-Bidgoli 2004). A interação com estas ferramentas pode ser usada para medir características relacionadas ao desempenho do aluno e predizer se o mesmo terá condições de ser aprovado ou não com base nestas características.

Nos últimos anos uma nova área de aplicação, chamada de Mineração de Dados na Educação (ou *Educational Data Mining – EDM*) surgiu como uma nova área de pesquisa visando dois objetivos principais: (a) entender melhor o estudante com base em sua interação com os sistemas usados e (b) as configurações e características que levam ao aprendizado [Romero, Ventura e Pechenizkiy, et al. 2010] [Koedinger, et al. 2008] [Romero e Ventura, 2006]. A ideia do EDM consiste em converter dados obtidos de sistemas educacionais num conjunto de informações que permitam a representação do modelo educacional e a reflexão sobre este modelo. Com isto é possível obter novos conhecimentos baseando-se no monitoramento das atividades do usuário e (a) o sistema educacional pode ser avaliado para checar se atende ao método proposto e (b) ou o processo de aprendizagem pode ser corrigido ou melhorado caso anomalias sejam detectadas (por exemplo descobrir que as ferramentas não são usadas e isto pode influenciar no mau desempenho do aluno) [Romero e Ventura, 2006] [Merceron e Yacef, 2005]. A ideia discutida em EDM pode ser estendida para o conceito de *Technology Enhanced Learning* com a exploração de diversos tipos de dados e processos de recomendação através do uso de logs, perfis de alunos e materiais, além do estudo do comportamento do usuário no ambiente [Manouselis, et al. 2010].

3. Acoplamento de Classificadores para Predição do Resultado do Aluno

Na literatura o processo de aprendizado de máquina apresenta algumas abordagens abordagens sobre como decidir sobre novos dados adquiridos e seus relacionamento. O aprendizado supervisionado usa dados já rotulados previamente para predizer os rótulos de novos dados. Esta abordagem também depende de um especialista que verifica os resultados e se eles estão de acordo com resultados esperados/desejados. A maior vantagem do aprendizado supervisionado é a consistência estabelecida pelas relações descobertas. Em contrapartida, é necessária uma base de dados considerável para que estas relações tenham um desempenho satisfatório.

Para suprir a falta desta quantidade de dados, neste trabalho é apresentada uma abordagem de aprendizado sem-fim usando múltiplos algoritmos e visões para garantir

que os dados obtidos aproximem-se ao máximo do conjunto real e suas relações existentes. A abordagem aqui descrita é parte do trabalho de doutorado de um dos autores. A abordagem completa do trabalho é descrita na figura 1. Neste trabalho nós apresentamos o funcionamento do componente *Supervision-based Model Generator* e resultados obtidos com o teste do mesmo.

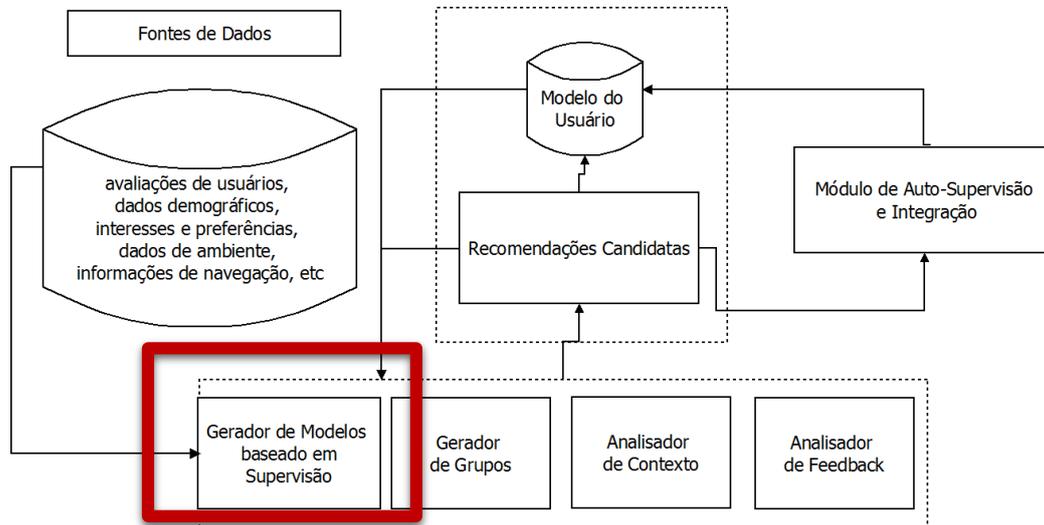


Figura 1: Abordagem Geral

É importante citar que a autossupervisão é feita internamente através da própria integração de algoritmos diferentes, colaborando entre si, e externamente ao componente através de regras que associam os resultados dos algoritmos. Neste trabalho são discutidas as regras internas do componente. Para aumentar o tamanho da base de dados disponível foi utilizada a abordagem de *Bootstrapping* na qual o conjunto de treinamento utilizado vai sendo incrementado escolhendo-se as melhores instâncias classificadas do conjunto de teste. Nos testes foi usada a probabilidade de acerto do classificador nos testes com o NaiveBayes. O problema é que o uso de apenas um classificador para a tarefa não apresentou bons resultados. A partir de um determinado tamanho de conjunto de treinamento o classificador passava a diminuir continuamente sua taxa de acerto. O mecanismo interno de controle desenvolvido usa mais de uma visão proporcionada por mais de um classificador atuando em conjunto com outro e com dados de treinamento independentes. A solução apresentada na figura 2 usa a abordagem de aumentar o tamanho do conjunto de treinamento a cada execução do algoritmo. De maneira resumida o algoritmo é mapeado da seguinte forma:

1. O conjunto de treinamento (dados já rotulados) é dividido entre os classificadores;
2. Cada um deles gera um modelo e faz a classificação das instâncias não rotuladas (conjunto de testes);
3. Os dados já rotulados são ordenados de acordo com a probabilidade de acerto dada pelo seu classificador (bayesiano);
4. As instâncias melhores ranqueadas são escolhidas para serem inseridas no conjunto de treinamento do outro classificador (ou seja, um elege as melhores instâncias e as insere do outro e vice-versa). O intuito disto é fazer com que os classificadores cooperem entre si enquanto “decidem” as classificações dadas.
5. O processo é repetido e a base de treinamento vai aumentando seu tamanho.

Esta técnica é conhecida como *bootstrapping* e o problema da mesma é que muitos erros podem ser propagados nos dados rotulados. Para diminuir este efeito foi usado mais de um algoritmo trabalhando em conjunto e rotulando os dados de outro algoritmo, de forma cruzada. Este acoplamento permite um modelo com melhor desempenho e é baseado no conceito de *Co-Training* (Blum e Mitchell 1998).

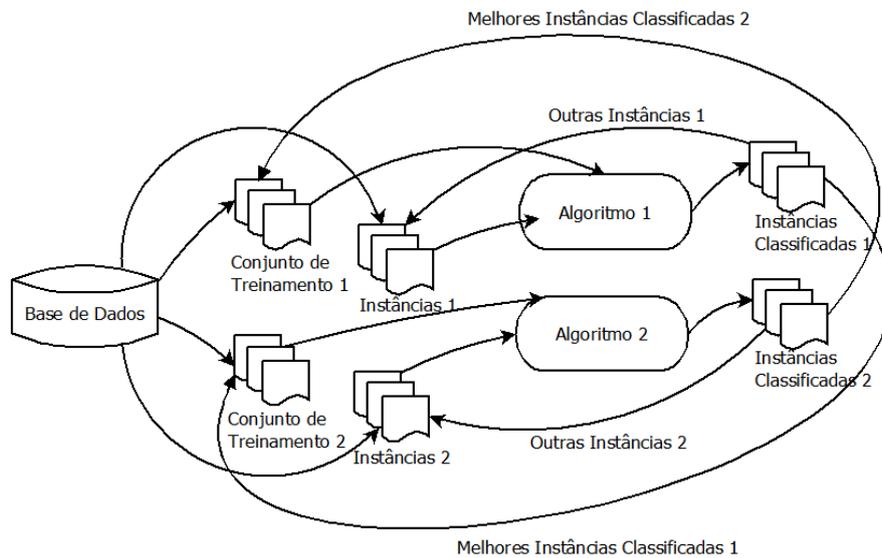


Figura 2: Aprendizado por Acoplamento e o Modelo usado para garantir novas instâncias.

Como é possível observar na figura 2 os dois algoritmos possuem seu próprio conjunto de instâncias que é dividido entre treinamento e testes. Cada algoritmo será executado e gerará instâncias classificadas com base no seu modelo gerado. Estas instâncias serão ordenadas de acordo com o critério de probabilidade de acerto. As primeiras N instâncias serão inseridas no conjunto de treinamento do outro algoritmo. Esta abordagem permite que o algoritmo possa melhorar seu conjunto de treinamento a partir da visão e resultados de outro algoritmo. O *bootstrapping* sozinho não é suficiente para obter os mesmos resultados. O algoritmo desenvolvido está descrito abaixo:

```

Input: A - Conjunto de Dados é extraído da base
W <- parse(A) // o conjunto de dados é estruturado em W
R = getInstances(W, 0.05) // conjunto de treinamento
V = getInstances(W, 0.95) // conjunto de testes
// Veja que W = R U V
C <- set of classifiers
for Ci in C do // faça iteração para todos os classificadores
    Ti <- R // conjunto de treinamento
    Si <- V // conjunto de testes
repeat
    for Ci in C do // para cada classificador
        trainClassifier(Ci, Ti) //treine-o com o conjto. de treinamento.
        Xi <- evaluateModel(Ci, Si) //avalié-o com o conjto. de testes
        Yi <- {} //crie o conjunto de instâncias classificadas
        for x in Xi do
    
```

```
        Yi <- Yi + x // adicione instâncias ao conjunto
    end
end
for Ci in C do //para cada classificador
    for Cj in C, Cj ≠ Ci do // adicione a melhor instância de um classificador
no outro
        Tj <- Tj + best(Yi)
        Sj <- Sj - best(Yi)
    end
end
end
until size(S) <= 0.10 // repete o algoritmo até um tamanho de base indicado.
```

No começo da execução o conjunto de dados é extraído de toda a base disponível e organizado da forma mais estruturada possível. Alguns elementos podem ser discretizados de acordo com as regras do modelo semântico do domínio para melhor atender os objetivos da extração e classificação. A próxima fase consiste em dividir o conjunto em dois subconjuntos: um deles representando o conjunto de treinamento e o outro representando o conjunto de testes. Geralmente a divisão é feita com uma destruição de 5/95 ou 10/90 (em porcentagem de instâncias). O próximo passo consiste em usar um *pool* de classificadores bem conhecidos. Neste trabalho são apresentados os resultados da utilização de dois deles: C4.5 (Quinlan 1993) (implementado pelo algoritmo J48) e NaiveBayes. Como já comentado, cada classificador trabalha com seu conjunto de treinamento e testes, inserindo no outro seus melhores resultados. O algoritmo repete o processo indefinidamente.

4. Experimentos e Resultados

Para testar a abordagem proposta foram realizados experimentos com uma base de dados que representa a interação de um conjunto de alunos, seus cursos e pólos, no ambiente *Moodle* de uma Universidade à Distância. A base de teste contempla um curso com 5 pólos e 252 alunos (não foram excluídos alunos que desistiram ou tiveram matrícula cancelada, pois investigava-se o comportamento dos alunos até então).

Tabela 2: Descrição da Base de dados do Experimento

Tipo de Dado	Descrição	Tipo de Dado	Descrição
Assignment	Submissão de Tarefas	Resource	Arquivos
Choice	Perguntas e Respostas	Upload	Upload de Arquivos
Course	Lista de Cursos	User	Informações do Usuário
Forums	Acesso ao Fórum	Days	Qtd. de dias de Acesso
Journal	Jornal	Avg_days	Média de Acessos por Dia
Label	Postagens e Atividades para Casa	Result	Informação sobre Aprovação/Reprovação
Quiz	Perguntas		

Na tabela 2 são descritas as ferramentas monitoradas no *log* e usadas para compor a informação de perfil do usuário. Os dados consistem em um sumário do acesso do usuário considerando as ferramentas (como fóruns, chats, etc) e sua frequência de acesso.

Como resultado final no perfil do usuário tem-se a informação que indica se foi ou não aprovado no curso. Cada elemento na tabela 2 representa a ferramenta disponível no sistema e utilizada pelo usuário. A tabela 3 mostra alguns exemplos de perfis (*profiles/user model*) dos usuários/alunos. No experimento foi comparada a porcentagem de acerto de um classificador usando os dados reais nas porcentagens indicadas (10%, 20%, 30% e 40% do conjunto de todos os dados) como conjunto de treinamento e a abordagem aqui proposta partindo-se de apenas 10% da base de dados e fazendo a reinserção de acordo com os treinamentos dos algoritmos. Nos gráficos apresentados nas figuras 3 e 4 pode-se verificar que as situações são bem similares e isto indica que é possível incrementar a base de dados com algum grau de confiança usando esta abordagem.

Tabela 3: Exemplo do Conjunto de Dados.

Userid	Cours eid	Assign ment	Foru m	...	Days	Avg_d ays	Result
115	5	Yes	No	...	30	5	Yes
113	5	No	Yes	...	40	4	No
98	1	No	Yes	...	10	2	Yes
...

Na figura 3 é apresentada a comparação entre o algoritmo J48 e a abordagem deste trabalho usando o J48 acoplado. Neste experimento os resultados se mantiveram próximos até o equivalente a 60% da base dados.

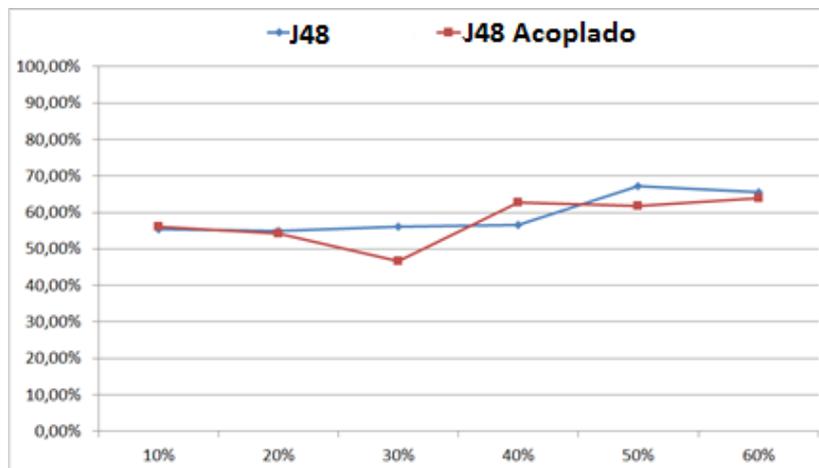


Figura 3. Comparação do Desempenho (eixo Y) entre um algoritmo que usa a base de dados original (J48) e o algoritmo acoplado (J48 + NaiveBayes). O eixo X representa o tamanho do conjunto de treinamento em relação ao tamanho total da base original.

Também foi comparado o algoritmo NaiveBayes com 10%, 20%, 30% e 40% da base de dados como conjunto de treinamento e a abordagem acoplada usando 2 algoritmos NaiveBayes começando com 10% no conjunto de treinamento e crescendo gradualmente. Como é possível observar na figura 3 o eixo X mostra o tamanho do conjunto de treinamento e o eixo Y mostra o percentual de acerto, considerando o restante dos dados. Os resultados obtidos mostram que o comportamento do algoritmo na abordagem proposta é similar ao uso dos dados originais da amostra. Isto quer dizer que, partindo de

apenas 10% da base de dados original o algoritmo foi capaz de incrementá-la a até o equivalente a 40%, usando o aprendizado acoplado e a reinserção das instâncias.

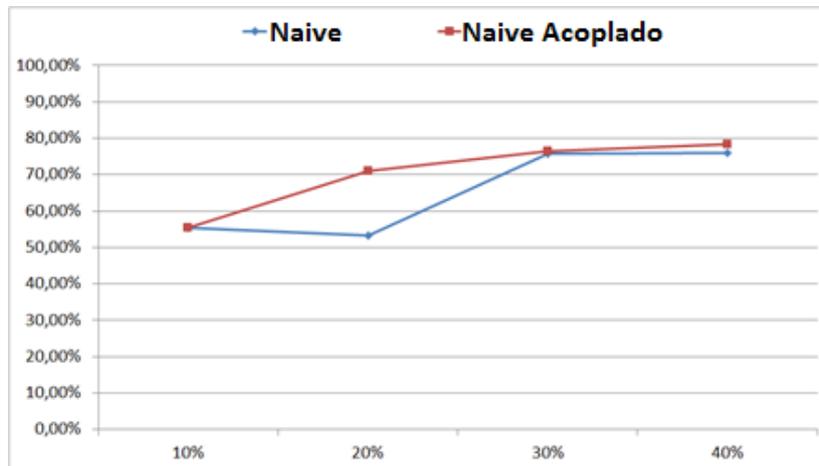


Figura 4. Comparação do Desempenho (eixo Y, % de acerto) entre um algoritmo que usa a base de dados original (NaiveBayes) e o algoritmo acoplado (NaiveBayes + NaiveBayes).

Através destes resultados nós criamos recomendações que possam indicar preferências do aluno que obtém melhores resultados, uma vez que é possível melhorar o acerto sobre o desempenho do aluno com base em seu comportamento. Os trabalhos futuros incluem a integração dos resultados obtidos com os demais módulos e a oferta de outros tipos de recomendação, além do desempenho do aluno em si.

5. Conclusão

Pela análise dos resultados observados nos experimentos é possível concluir que o uso do acoplamento para gerar recomendações sobre o desempenho do aluno incrementa a base inicial de dados (conjunto de treinamento) e permite a oferta de recomendações com dados mais próximos de uma base de dados real. O *bootstrapping* sozinho não conseguia obter os mesmos resultados e por isto a abordagem de algoritmos acoplados foi utilizada. Futuramente será possível comparar a abordagem com outras técnicas de recomendação e integrá-la ao restante do projeto, sendo uma das geradores de recomendações candidatas no sistema que ainda contará com um módulo para decidir a oferta das recomendações.

6. Agradecimentos

Nós gostaríamos de agradecer a CAPES por ter apoiado este projeto.

Referências

- Adomavicius, Gediminas, e Alexander Tuzhilin. (2005) "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." Knowledge and Data Engineering, IEEE Transactions on, Junho.
- Blum, A, e T Mitchell. (1998) "Combining labeled and unlabeled data with co-training." COLT'98 Proceedings of the eleventh annual conference on Computational learning. New York: ACM.

- Brusilovsky, P., e E. Millan. (2007) "User Models for Adaptive Hypermedia and Adaptive Educational Systems." In: *The Adaptive Web: Methods and Strategies*. Berlin, Heidelberg, New York: Springer-Verlag.
- da Silva, Luiz C N, Francisco M Mendes Neto, e Luiz Jácome Jr. (2011) "MobiLE: Um ambiente Multiagente de Aprendizagem Móvel para Apoiar a Recomendação Sensível ao Contexto de Objetos de Aprendizagem." *Simpósio Brasileiro de Informática na Educação - SBIE*.
- Ferro, Márcio Robério, Hélio Martins Nascimento Jr, Fábio Costa Paraguaçu, Evandro Barros Costa, e Larissa Artemis Luna Monteiro. (2011) "Um Modelo de Sistema de Recomendação de Materiais Didáticos para Ambientes Virtuais de Aprendizagem." *Simpósio Brasileiro de Informática na Educação*.
- Koedinger, K, K Cunningham, A Skogsholm, e B Leber. (2008) "An open repository and analysis tools for fine-grained, longitudinal learner data." In: *Proceedings 1st Int. Conf. Educ. Data Mining*. Montreal, Canadá.
- Manouselis, Nikos, Hendrik Drachsler, Riina Vuorikari, Hans Hummel, e Rob Kuper. (2010) "Recommender Systems in Technology Enhanced Learning." In: *Recommender Systems Handbook*, por Francesco Ricci, Lior Rokach, Bracha Shapira e Paul B Kantor, 387-418. Springer.
- Merceron, A, e K Yacef. (2005) "Educational data mining: A case study." In: *Proc. Int. Conf. Artif. Intell. Educ.* Amsterdam, The Netherlands.
- Minaei-Bidgoli, B. (2004) "Data Mining for a Web-Based Educational System." Doctor of Philosophy. Department of Computer Science and Engineering Michigan State University.
- Quinlan, J R. (1993) *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc..
- Ricci, Francesco, Lior Rokach, e Bracha Shapira. (2010) "Introduction to Recommender Systems." In: *Recommender Systems Handbook*, por Francesco Ricci, Lior Rokach, Bracha Shapira e Paul B. Kantor, 1-38. Springer.
- Romero, C, e S Ventura. (2006) *Data Mining in E-Learning*. Ashurst: Wit Press.
- Romero, C, S Ventura, M Pechenizkiy, e R Baker. (2010) *Handbook of Educational Data Mining*. New York: Taylor & Francis.
- Shardanand, Upendra, e Pattie Maes. (1995) "Social Information Filtering: Algorithms for Automating "Word of Mouth"." *CHI '95 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM Press/Addison-Wesley Publishing Co..