

Combinando Multi-Visões através de um Algoritmo de Nuvens de Partículas

Zilton Junior¹, Gisele L. Pappa¹, Filipe de L. Arcanjo¹,
Wagner Meira Jr.¹, Marcos A. Gonçalves¹

¹Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
CEP 31270-901 – Belo Horizonte – MG – Brasil

{zilton, glpappa, filipe, meira, mgoncalv}@dcc.ufmg.br

Abstract. *Classification problems can use several representations to express the same concept. Videos, for example, can be represented by their image or sound features. In recent years, researchers start to integrate these representations into models called multi-view, which can significantly improve the outcome of classification. This paper presents a study on two multi-view datasets, and introduces a new algorithm based on particle swarm optimization to weight views according to their relevance. As a result, we create MultiViL, an open-source tool for multi-view classification.*

Resumo. *Problemas de classificação podem utilizar diversas representações de um domínio do conhecimento. Por exemplo, um vídeo pode ser representado utilizando características de som ou imagem. Nos últimos anos, pesquisadores passaram a integrar essas representações em modelos denominados multi-visão, que integram atributos de visões diferentes para tornar a classificação mais robusta. Este artigo apresenta um estudo sobre o aprendizado multi-visão em duas bases de dados, e introduz um algoritmo baseado em nuvens de partículas para ponderar as visões dos classificadores de acordo com sua significância. Como resultado, geramos a MultiViL, uma ferramenta de código aberto para classificação multi-visão.*

1. Introdução

O ser humano é capaz de identificar e reconhecer objetos utilizando um conjunto de um ou mais de seus cinco sentidos básicos. Por exemplo, um incêndio pode ser percebido ao se ver as chamas, ao inalar a fumaça, através do alarme de incêndio, ou quando seu colega de trabalho lhe informa que estão abandonando o prédio e parece existir um princípio de incêndio. Todas as essas informações podem ser utilizadas independentemente para prever uma situação de risco, mas quando combinadas oferecem uma confiança maior na previsão do evento.

O mesmo acontece com aprendizado de máquina. Ao fornecer a um algoritmo diversas representações de um mesmo problema, este pode aprender diferentes modelos e depois combiná-los para obter um modelo de classificação/previsão único e mais robusto. Indiretamente, esse tipo de aprendizado já é comumente utilizado. Na área de processamento de imagens, por exemplo, imagens podem ser representadas utilizando diferentes

tipos de atributos, tais como aqueles baseados em cor, forma, textura, etc (também conhecido como aprendizado baseado em características multimodais [Wang et al. 1999]). Porém, esses atributos são comumente agrupados em uma única base de dados, que é fornecida como entrada para um algoritmo de aprendizado.

A ideia de aprender e combinar diferentes modelos a partir de um mesmo conjunto de dados já é antiga. Aprendizado com comitês (ou *ensembles*) [Kotsiantis and Pintelas 2004] mostraram que, ao aprender diversos modelos a partir de diferentes amostragens de uma única base de dados e depois combinar esses modelos pode levar a previsões mais confiáveis. Ao mesmo tempo, na literatura de aprendizado semi-supervisionado, métodos denominados multi-visão [Blum and Mitchell 1998] surgiram com o intuito de utilizar duas ou mais representações de um mesmo problema para aprender a rotular dados. Ao contrário do aprendizado supervisionado, onde rótulos para os conceitos a serem aprendidos são conhecidos, o aprendizado semi-supervisionado é caracterizado por um grande número de dados não rotulados e uma pequena quantidade de dados rotulados. A ideia do aprendizado multi-visão é utilizar conhecimento extraído dos dados rotulados para induzir os rótulos dos exemplos de rótulo desconhecido.

Em problemas de aprendizado multi-visão, um exemplo x é descrito por uma série de atributos distintos em cada visão. Considere, por exemplo, a tarefa de classificação, que será tratada nesse artigo. Em um domínio com k visões V_1, V_2, \dots, V_k , um exemplo rotulado pode ser visto como uma tupla $\langle x_1, x_2, \dots, x_k, l \rangle$, onde l é um rótulo e $[x_1], [x_2], \dots, [x_k]$ são conjuntos de dados em k visões. Definidas as visões, um modelo é criado de forma independente para cada visão, sendo necessária uma posterior combinação desses modelos para predição das classes de novos exemplos. Como mostrado em [Nigam et al. 2000], a utilização de k visões dos dados leva os classificadores a acertarem mais do que os métodos que utilizam apenas uma visão.

Apesar dessas conclusões partirem de problemas semi-supervisionados, elas podem ser facilmente extrapoladas para o contexto de aprendizado supervisionado. Ao utilizar várias visões, a dimensão dos dados com os quais o algoritmo trabalha é automaticamente reduzida, já que n atributos são distribuídos em um conjunto de k visões. Um outro ponto importante é que podemos utilizar classificadores diferentes para aprender em cada visão, de acordo com o tipo e representação de dados com o qual estamos trabalhando. Nesse caso, uma das dificuldades encontradas é como combinar previsões provenientes de classificadores diferentes. Por exemplo, enquanto o Naive Bayes [McCallum and Nigam 1998] retorna uma lista de probabilidades do exemplo pertencer a classe, o SVM [Vapnik 1995] retorna a distância a que o exemplo está da margem do classificador induzido. Combinar esse tipos de previsões não é trivial. Por último, em alguns domínios nem sempre dados para todas as visões estão disponíveis, e em alguns casos a informação de uma visão pode ser mais importante na classificação que outra. Por isso, métodos capazes de ponderar visões são também relevantes.

Este artigo apresenta um estudo de abordagens que chamaremos de multi-visão no contexto de aprendizado supervisionado. Note que, apesar desse termo ser comumente utilizado para se referir a métodos de aprendizado semi-supervisionado, ele será adotado neste artigo pois, no futuro, esse estudo também abrangerá abordagens semi-supervisionadas. São duas nossas contribuições: (i) introduzir um novo método para ponderação das visões utilizando um algoritmo de *Particle Swarm Optimization* (PSO)

ou Otimização por Nuvem de Partículas [Kennedy and Eberhart 2001], (ii) apresentar a ferramenta MultiViL (MultiView Learning), que permite a leitura, execução, combinação e visualização de classificadores em múltiplas visões dos dados.

Este trabalho está organizado da seguinte forma. A Seção 2 descreve os principais trabalhos no contexto de multi-visão semi-supervisionada. A Seção 3 introduz um algoritmo PSO para a ponderação de visões dos dados. A Seção 4 apresenta a ferramenta MultiViL, enquanto a Seção 5 mostra os resultados obtidos nos experimentos realizados. Finalmente, a Seção 6 traz conclusões e direções futuras.

2. Trabalhos Relacionados

O termo aprendizado multi-visão surgiu com [de Sa 1993], mas as pesquisas na área tornaram-se mais fortes a partir dos trabalhos realizados por [Blum and Mitchell 1998], com a criação do algoritmo *Co-training* para classificação de páginas *Web*. O *Co-training* utiliza aprendizado semi-supervisionado para construir dois classificadores: um baseado nos termos que aparecem na página, e outro sobre os termos que aparecem em *links* apontando para essa página. Os autores mostraram que a utilização de visões independentes de uma base de dados pode elevar a qualidade final da classificação.

A partir do artigo inicial de [Blum and Mitchell 1998], outros estudos tentaram caracterizar problemas em que a abordagem multi-visão seria eficiente, e propuseram diversas variações do *Co-training*, como o *Co-EM* [Nigam and Ghani 2000], o *Co-Testing* e o *Co-EMT* [Muslea et al. 2002]. Além disso, na parte de aprendizado semi-supervisionado, grande esforço foi colocado para definir os fundamentos teóricos por trás do aprendizado multi-visão.

Eles foram inicialmente baseados na suposição de que as visões são compatíveis e não-correlatas. Um problema tem visões compatíveis se todos os exemplos são rotulados de forma idêntica pelo conceito alvo em cada visão. Ao mesmo tempo, visões são ditas não-correlatas se, dado o rótulo de cada exemplo, suas descrições em cada visão são independentes. Logo, isso significa que, para as visões serem compatíveis e não-correlatas, em um exemplo $[x_1, x_2, l]$, x_1 e x_2 são independentes, dado o rótulo l [Muslea et al. 2002]. Porém, mais recentemente alguns trabalhos mostraram que mesmo visões não compatíveis podem melhorar o aprendizado [Christoudias et al. 2008].

3. Um Algoritmo de Nuvem de Partículas para Ponderação de Visões

Inspirados na simulação do comportamento social de um bando de pássaros em revoada, com movimento localmente aleatório mas globalmente determinado, Eberhart e Kennedy propuseram um método estocástico denominado *Particle Swarm Optimization* ou Otimização por Nuvem (ou Enxame) de Partículas (PSO) [Kennedy and Eberhart 2001]. O algoritmo trabalha com um conjunto de partículas que se movem no espaço através da cooperação e competição entre si por diversas iterações. A ideia básica do algoritmo é que aprendemos a partir da nossa própria experiência, mas também imitando o comportamento de pessoas próximas e mais bem sucedidas.

No algoritmo, cada partícula representa uma solução para o problema a ser resolvido, e é descrita por um posição no espaço $X_i = (X_{i_1}, X_{i_2}, \dots, X_{i_d})$, onde X_{i_d} representa uma dimensão. A cada uma dessas dimensões são atribuídas velocidades, que servem

para deslocar a partícula no espaço, e são representadas por $V_i = (V_{i_1}, V_{i_2}, \dots, V_{i_d})$. A partícula também possui memória, e conhece sua melhor posição ao longo de t iterações e também a posição da melhor partícula da nuvem até o momento.

A cada iteração, as partículas são avaliadas de acordo com uma função de aptidão pré-definida, que mede a qualidade da solução representada pela partícula para resolver o problema. Após essa avaliação, as partículas se movimentam no espaço através da alteração de sua velocidade e posição. Essa movimentação leva em conta dois fatores: (i) fator social, que determina a atração das partículas para a melhor posição descoberta por qualquer elemento da nuvem até o momento, (ii) fator cognitivo, que determina a atração da partícula por sua melhor posição até o momento.

A Eq. 1 descreve o cálculo da nova velocidade da partícula i de acordo com sua velocidade anterior, sua melhor posição até o momento ($pbest$), a posição da partícula da nuvem com melhor função de aptidão até o momento ($gbest$), e sua posição atual X_i . Note que essa é uma atualização da proposta original do PSO, modificada em [Kennedy and Eberhart 2001].

$$V_i = V_i + \varphi \times \left(\frac{\varphi_1 \times pbest + \varphi_2 \times gbest}{\varphi_1 + \varphi_2} - X_i \right) \quad (1)$$

Na Eq. 1, φ é um número aleatório $\in [0, 1]$, e φ_1 e φ_2 são duas constantes que têm seus valores atualizados a cada iteração de acordo com as Eqs. 2 e 3, e representam, respectivamente, os valores cognitivo e social das partículas, ou seja, o quanto elas confiam nelas mesmas ou nos vizinhos.

Como observado nas Eqs. 2 e 3, a cada iteração, o valor de φ_1 diminui e o de φ_2 aumenta, fazendo com que as partículas, ao longo do tempo, passem a confiar mais na nuvem do que em suas próprias soluções. Nessas equações, C_{MIN} e C_{MAX} são constantes, $NumMaxIter$ é o número máximo de iterações, e $iter$ é a iteração corrente. Atualizada a velocidade da partícula i , sua nova posição é definida por $X_i = X_i + V_i$.

$$\varphi_1 = C_{MAX} - \left(\frac{(C_{MAX} - C_{MIN}) \times iter}{NumMaxIter} \right) \quad (2)$$

$$\varphi_2 = C_{MIN} + \left(\frac{(C_{MAX} - C_{MIN}) \times iter}{NumMaxIter} \right) \quad (3)$$

Nesse trabalho, o PSO trabalha com k dimensões, onde k corresponde ao número de visões sendo consideradas no problema e que devem ser ponderadas. Cada dimensão pode assumir valores reais positivos. Cada partícula é avaliada utilizando a Micro-F1 obtida pelo classificador quando ponderando as confianças dos classificadores das k visões pelos k pesos obtidos pelo PSO.

A micro-média F1 pondera os F1 de cada classe com base na representatividade da classe na base de dados i.e., de acordo com o número de documentos em cada classe. A métrica F-1 representa a média harmônica entre a precisão (número de exemplos corretamente classificados) e a revocação (número de exemplos considerados como pertencentes a classe) obtidas pelo classificador. Assim, dado um problema com N classes representado por k visões, e a confiança c de um classificador prever a classe de um exemplo, o método de combinação do PSO decidiria a classe baseado em $\max(S_c)$, onde S_c é

a confiança do classificador e da visão em classificar um exemplo como pertencente a classe c , e é definido como $S_{c_j} = \sum_{i=1}^k c_{j_i} \times p_i$, onde p_i representa o peso encontrado pelo PSO para dimensão i . Alg. 1 mostra o código para o algoritmo PSO utilizado.

Algorithm 1: PSO

```

begin
  for (  $i = 0$  até número de partículas ) do
    for (  $j = 0$  até número de visões ) do
      inicializa velocidade  $V_j$  e posição  $P_j$  da partícula  $p_i$ 
    while (  $t <$  número de iterações ) do
      for (  $i = 0$  até número de partículas ) do
        Calcula Micro-F1 da partícula  $p_i$ 
        if ( Micro-F1 de  $p_i >$  Micro-F1 de  $p_{best_i}$  ) then
           $P_{p_{best_i}} = P_{p_i}$  /* Atualiza informação local da partícula */
        if ( Micro-F1 de  $p_i >$  Micro-F1 de  $g_{best}$  ) then
           $g_{best} = p_i$  /* Atualiza informação global da nuvem */
      for (  $i = 0$  até número de partícula ) do
        for (  $j = 0$  até número de visões ) do
          Calcula velocidade  $V_j$  da partícula  $p_i$ 
          Calcula posição  $P_j$  da partícula  $p_i$ 
    Retorna  $g_{best}$ 

```

4. MultiViL: Uma ferramenta para aprendizado Multi-visão

Esta seção apresenta a primeira versão da ferramenta MultiViL, criada para realizar aprendizado multi-visão, e disponível gratuitamente na Web¹. Até o momento, a MultiViL integra os algoritmos de classificação SVM-Perf [Vapnik 1995], *Naive Bayes* [McCallum and Nigam 1998], *Rocchio* [Rocchio 1971] e KNN (*K-Nearest Neighbor*). Ela também permite a combinação dos resultados das k visões, e inclui um esquema simples de visualização de cooperação entre as visões utilizando diagramas de Venn. A Fig. 1 mostra a tela inicial da MultiViL, e a tela de análise de cooperação das visões.

Além dos algoritmos citados acima, a MultiViL disponibiliza três métodos de combinação de resultados: o voto simples da maioria, o voto da maioria ponderado pela confiança e a ponderação das visões pelo PSO. O voto simples da maioria atribui uma classe C a um exemplo x_1 se esta é atribuída a X_1 por mais de $\frac{k}{2}$ visões. O voto da maioria com confiança pondera o voto das visões por usa confiança, e totaliza a confiança de cada classe de acordo com as k visões disponíveis. O método PSO é descrito na Seção 3.

5. Resultados Experimentais

Essa seção apresenta os resultados obtidos utilizando a abordagem multi-visão supervisionada em duas bases de dados: a ACM-DL (Biblioteca Digital da ACM)² e uma base de *spammers* do *YouTube* [Benevenuto et al. 2009]. A base da ACM-DL é uma sub-coleção de artigos científicos da biblioteca digital da *Association for Computing Machinery* (ACM), e contém 24.897 documentos divididos em 11 classes. Desta base foram extraídas três visões: termos presentes no *abstract* dos artigos, rede de autores dos artigos e rede de citações entre os artigos.

¹<http://multivil.sourceforge.net/>

²<http://portal.acm.org/dl.cfm>

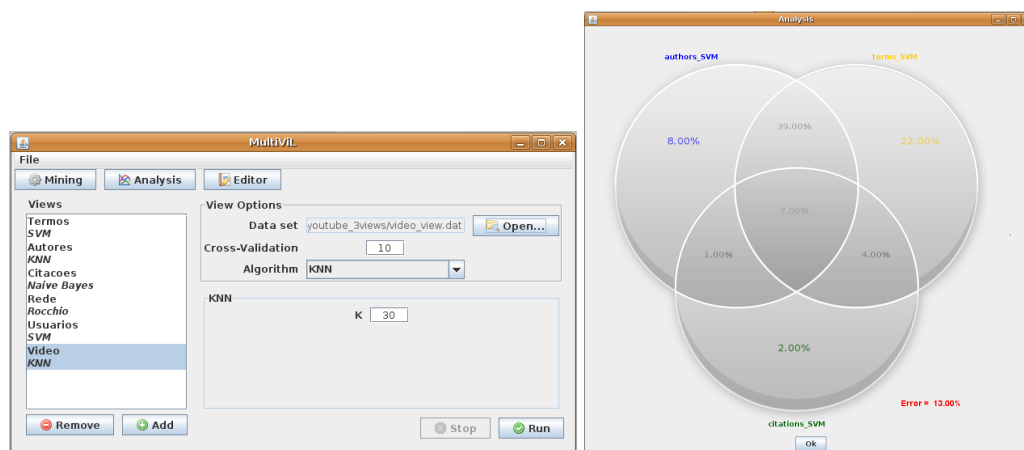


Figura 1. Tela Inicial e de Análise de Resultados da Ferramenta MultiViL

Algoritmos	Visões							Método de combinação			Pesos		
	A	T	C	A+T	A+C	T+C	A+T+C	Voto simples	Voto Confiança	PSO	A	T	C
SVM	57.44	73.70	29.21	75.20	54.86	73.48	74.98	74.45	74.57	73.35	0.05	0.22	21.64
Rocchio	46.72	68.21	13.67	70.20	48.05	68.70	70.62 ▲	70.64 ▲	70.10 ▲	72.67	8.28	3.55	3.89
KNN	58.59	73.18	31.65	77.23	58.91	68.81	75.44	75.42	74.88	75.91	12.85	7.54	3.61
NB	54.60	73.94	16.40	76.23	43.60	66.82	70.87 ▲	66.48 ▲	76.18	77.06	8.28	3.55	3.89

Tabela 1. Valores do Micro F1 para base ACM-DL nas visões de termos(T), autores(A) e citações(C).

A base de *spammers* do *YouTube* é formada por 829 exemplos, divididos em 3 classes. Desta base foram utilizadas três visões: informações dos usuários (*promoters*, *spammers* ou legítimos), atributos do vídeo e relação social estabelecida entre usuários de vídeos resposta.

Experimentos foram divididos em duas fases. Na primeira fase, os quatro classificadores disponíveis na ferramenta foram executados independentemente nas três visões das duas bases de dados, e também em conjuntos de treinamento que incluíam atributos das visões combinadas duas a duas, e com informações referentes às três visões. Os resultados foram combinados utilizando o voto de maioria simples e ponderado por confiança.

Numa segunda fase, o PSO foi utilizado para combinar as visões. Essa fase se iniciou com a configuração dos parâmetros do PSO através de experimentos preliminares. Pelo fato do problema apresentar um espaço de busca d dimensional relativamente pequeno, com $d = 3$, foi definida a utilização de 20 partículas e 20 iterações. Os valores das constantes utilizadas nas Eqs. 2 e 3 foram definidos como $C_{MIN} = 0.5$, $C_{MAX} = 4.0$. Conforme mostrado em [Kennedy and Eberhart 2001], esses valores fazem com que as partículas percorram o espaço de busca de forma sistemática.

As Tabelas 1 e 2 mostram os resultados de micro-F1 obtidos para cada uma das entradas de dados, e das combinações entre as visões. Todos os experimentos foram realizados utilizando *10-fold cross-validation* [Browne 2000]. A significância dos resultados foi avaliada utilizando um teste-t de *Student* com uma confiança de 99%. As células contém marcações em relação ao PSO: ▲ indica que o PSO é estatisticamente superior ao método e ▼ estatisticamente inferior. Nos outros casos não houve diferença estatística.

Fazendo uma análise das diferentes visões da ACM-DL, cujos resultados aparecem na Tabela 1, percebemos que a visão de termos é a mais robustas, e a de citações

Algoritmos	Visões				Método de combinação			Pesos		
	Rede	Usuário	Vídeo	Todas Visões	Voto Simples	Voto Confiança	PSO	Rede	Usuário	Vídeo
SVM	69.71	72.37	72.63	47.75 ▲	69.12 ▲	69.12 ▲	72.27	0.26	31.62	0.28
Rocchio	48.38	68.17	75.04	72.26 ▲	69.13 ▲	75.28	75.04	21.56	32.21	37.12
KNN	77.57	80.82	84.08	84.92 ▼	81.91	81.91	81.91	13.61	12.41	23.04
Naive Bayes	66.71	58.87	43.90	40.54 ▲	66.67 ▲	63.69 ▲	71.05	12.76	9.49	17.11

Tabela 2. Valores do Micro F1 para base YouTube [Benevenuto et al. 2009].

a menos robusta. Porém, temos que considerar que as visões de autores e citações são bastante esparsas, com muitos documentos apresentando informações de nenhum ou somente um autor ou citação. Observa-se também que utilizar apenas a visão de termos ou as três visões em uma única base de dados apresentam resultados estatisticamente equivalentes. Ao mesmo tempo, os resultados obtidos pelos três métodos de combinação considerados, incluindo o PSO aqui proposto, mostram dentro de uma mesma linha da tabela (um mesmo classificador), resultados equivalentes e, em alguns casos, melhores que os obtidos pela utilização das três visões em um único conjunto de treino.

Embora nossa intenção não seja comparar os classificadores entre si, podemos observar que os melhores resultados obtidos pelo PSO foram aqueles em que consideramos o uso do classificador Naive Bayes (77.06). Esse resultado é estatisticamente superior a qualquer outro apresentado na tabela, incluindo os obtidos pelo SVM. Esses resultados iniciais são promissores, e experimentos preliminares com classificadores diferentes em visões diferentes melhoram ainda mais esses resultados. Observando os pesos gerados pelo PSO, percebe-se que a visão de termos nem sempre recebe o maior valor. É importante ressaltar que esses pesos são normalizados antes de serem aplicados aos classificadores.

Em relação aos resultados da base do YouTube, podemos notar na Tabela 2 que as visões independentes são bastante equilibradas em seus valores de Micro F1. É interessante observar também que, nesse caso, quando todos os atributos são combinados em um único conjunto de treinamento, o Naive Bayes e SVM geram resultados muito inferiores a aqueles gerados pelas visões isoladas. Porém, nesse caso, nenhum dos métodos de combinação apresentados foi capaz de obter resultados melhores que aqueles gerados pelo KNN utilizando todas as visões. O PSO obteve resultados estatisticamente melhores que aqueles gerados pela votação simples e baseada em confianças quando utilizando o SVM e Naive Bayes. É interessante notar que os pesos atribuídos pelo PSO variam de classificador para classificador. Enquanto o SVM dá um peso maior a visão de usuários, os outros três classificadores dão mais importância a visão de vídeo. A investigação desse comportamento será feita nos próximos passos desse trabalho

A partir dos resultados aqui apresentados, podemos concluir que a ponderação das visões por meio do algoritmo PSO têm potencial. Além disso, a MultiViL pode tornar-se uma importante ferramenta no contexto de aprendizado multi-visão, pois fornece um novo ambiente para experimentação rápida e exaustiva de experimentos com bases de dados com mais de uma visão.

6. Conclusões e Trabalhos Futuros

Este artigo apresentou um algoritmo de nuvens de partículas (PSO) para ponderar visões de dados em aplicações de aprendizado multi-visão supervisionado. Além disso, ele introduziu uma ferramenta denominada MultiViL, disponível gratuitamente na Web e com

grande potencial para dar suporte a experimentação com bases de dados multi-visão. Os resultados obtidos pelo PSO na base da ACM foram estatisticamente superiores aos obtidos por diversas combinações de classificadores e métodos de combinação. Na base do *Youtube*, o PSO não conseguiu obter resultados melhores que a combinação de todas as visões utilizando o KNN. Porém, alguns parâmetros do PSO ainda podem ser otimizados, e as características da base do *Youtube* melhor compreendidas.

Como trabalhos futuros, pretendemos estender o presente estudo para o aprendizado semi-supervisionado, propondo novos métodos de combinação de visões, principalmente quando classificadores heterogêneos são utilizados em diferentes visões. Esses métodos também serão incluídos na MultiViL.

Agradecimentos

Esse projeto foi parcialmente financiado pela Fapemig, Santander, CNPq, e InWeb (Instituto Nacional de Ciência e Tecnologia para Web).

Referências

- Benevenuto, F., Rodrigues, T., Almeida, V., Almeida, J., and Gonçalves, M. (2009). Detecting spammers and content promoters in online video social networks. In *Proc. of Int'l ACM SIGIR*, Boston, MA, USA.
- Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proc. of the 11th Annual Conf. on Computational Learning Theory*, pages 92–100.
- Browne, M. W. (2000). Cross-validation methods. *Journal of Mathematical Psychology*, 44(1):108 – 132.
- Christoudias, C., Urtasun, R., and Darrell, T. (2008). Multi-view learning in the presence of view disagreement. In *Proc. of the 24th Annual Conf. on Uncertainty in Artificial Intelligence*.
- de Sa, V. R. (1993). Learning classification with unlabeled data. In Cowan, J. D., Tesauro, G., and Alspector, J., editors, *Proc. NIPS'93, Neural Information Processing Systems*, pages 112–119.
- Kennedy, J. and Eberhart, R. C. (2001). *Swarm intelligence*. Morgan Kaufmann Publishers Inc.
- Kotsiantis, S. B. and Pintelas, P. (2004). Combining bagging and boosting. In *IJCI '04: Proc. of the International Journal of Computational Intelligence*, pages 324–333.
- McCallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *Proc. of the 15th Nat. Conf. on Artificial Intelligence*, pages 41 – 48.
- Muslea, I., Minton, S., and Knoblock, C. A. (2002). Active + semi-supervised learning = robust multi-view learning. In *Proc. of the 19th Int. Conf. on Machine Learning*, pages 435–442.
- Nigam, K. and Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. In *CIKM '00: Proc. of the ninth international conference on Information and knowledge management*, pages 86–93, New York, NY, USA. ACM.
- Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Mach. Learn.*, 39(2-3):103–134.
- Rocchio, J. (1971). Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*, pages 313–323.
- Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag.
- Wang, H. L., Huang, J., Liu, Z., Wang, Y., Chen, Y., and Wong, E. K. (1999). Integration of multimodal features for video scene classification based on *hmm*. In *IEEE Workshop on Multimedia Signal Processing*, pages 53–58.