MARCELO CESAR CIRELO

UMA METODOLOGIA PARA O APRENDIZADO SEMI-SUPERVISIONADO DE CLASSIFICADORES BAYESIANOS

Dissertação apresentada à Escola
Politécnica da Universidade de
São Paulo para obtenção do
Título de Mestre em Engenharia

São Paulo

2005

MARCELO CESAR CIRELO

UMA METODOLOGIA PARA O APRENDIZADO SEMI-SUPERVISIONADO DE CLASSIFICADORES BAYESIANOS

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia.

Área de Concentração:
Engenharia Mecatrônica

Orientador:
Prof. Associado
Fabio Gagliardi Cozman

São Paulo

2005

AGRADECIMENTOS

Agradeço ao meu orientador, Fabio Cozman, por ter me aceitado no laboratório que dirige e pela ajuda e incentivo durante todo o desenvolvimento deste trabalho.

Agradeço ao pesquisador Ira Cohen que esteve no Laboratório de Tomada de Decisão (LTD) em fevereiro de 2002 pelos momentos de trabalho árduo e de diversão. Foi um privilégio ter participado de sua pesquisa que rendeu artigos tão relevantes.

Agradeço a minha família pelo apoio e aos meus amigos do LTD por tornarem os momentos em que lá estive muito agradáveis e enriquecedores.

Durante os meses em que estive no LTD, recebi uma generosa bolsa de mestrado do HP Labs. Agradeço a eles, ao Instituto de Pesquisas Eldorado (IPE) e ao Frederico D'Ávila do IPE que tornaram possível o recebimento dessa bolsa e a existência deste trabalho.

RESUMO

Neste trabalho são apresentados métodos para aprendizado de classificadores Bayesianos a partir de bases de dados contendo dados rotulados e não-rotulados (aprendizado semi-supervisionado). O trabalho apresenta dois novos algoritmos, SSS e CBL-EM, e compara estes algoritmos com versões de classificadores *Naive Bayes*, *Tree-Augmented Naive Bayes* e *Structural-EM*. As principais contribuições foram o desenvolvimento de um método para utilizar o algoritmo CBL1 em conjunto com o algoritmo EM (do inglês *Expectation-Maximization*) e a definição de uma metodologia para o aprendizado semi-supervisionado de classificadores Bayesianos. Os resultados empíricos mostram que os algoritmos propostos tem desempenho superior aos algoritmos existentes para aprendizado com dados rotulados e não-rotulados.

ABSTRACT

This work presents techniques for learning Bayesian classifiers from databases containing labeled and unlabeled data (semi-supervised learning). The work presents two new algorithms, SSS and CBL-EM, and compares their performance with versions of *Naive Bayes*, *Tree-Augmented Naive Bayes* and *Structural-EM* classifiers. The main contributions of this work are the development of a framework for using the CBL1 and EM algorithms together, and the development of a methodology for the semi-supervised learning of Bayesian classifiers. The empirical tests show that the proposed algorithms perform better than existing classifiers for labeled and unlabeled data.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

CBL	—	Algoritmo de Cheng, Bell e Liu
CL	—	Algoritmo de Chow e Liu
DAG	—	*Directed Acyclic Graph*
EM	—	*Expectation Maximization*
KL	—	Distância de Kullback-Leibler
MCMC	—	*Markov Chain Monte Carlo*
MDL	—	*Minimum Description Length*
MI	—	*Mutual Information*
CMI	—	*Conditional Mutual Information*
MWST	—	*Maximum Weight Spanning Tree*
NB	—	*Naive Bayes*
SEM	—	*Structural Expectation Maximization*
SSS	—	*Stochastic Structure Search*
TAN	—	*Tree Augmented Naive Bayes*

LISTA DE SÍMBOLOS

$X_1, X_2, X_3, \ldots$	Variáveis aleatórias.
$x_1, x_2, x_3, \ldots$	Valores observados. $X_1 = x_1,\ X_2 = x_2,\ X_3 = x_3, \ldots$
$\mathbf{X}$	Conjunto de variáveis aleatórias. $\mathbf{X} = \{X_1,\ X_2, \ldots,\ X_n\}$
D'	Base de dados com todos os registros rotulados.
D''	Base de dados com nenhum registro rotulado.
D^*	Base de dados composta por $D' \cup D''$ na qual os registros incompletos foram preenchidos com os valores esperados.
$\langle S,\ \theta \rangle$	Rede Bayesiana.
S	Grafo direcionado acíclico; estrutura da rede Bayesiana.
θ	Conjunto de parâmetros livres de uma rede Bayesiana.
i	Número de iterações máximo do EM.
j	Número de iterações máximo do SEM e CBL-EM.
y	Valor verdadeiro do rótulo de um registro.
m	Número de registros total de uma base de dados (registros rotulados mais não-rotulados).
n	Número de atributos de base de dados.
m'	Número de registros de uma base de validação.

1 INTRODUÇÃO

Com o contínuo avanço da tecnologia de aquisição e transferência de dados, em muitas áreas não é conveniente classificar dados de forma manual. Por exemplo, uma ferramenta de classificação automática pode ser usada para classificar mensagens eletrônicas como sendo mensagens indesejadas ou como mensagens legítimas. Para tarefas como essa, algoritmos são utilizados para construir classificadores a partir de bases de dados existentes.

Este trabalho aborda o aprendizado de classificadores Bayesianos a partir de uma base de dados composta por registros rotulados e não-rotulados. Na literatura, esses classificadores estão no domínio do *aprendizado semi-supervisionado*.

Nos últimos anos, vários autores utilizaram redes Bayesianas para aprendizado semi-supervisionado, com resultados promissores (NIGAM et al., 2000; BALUJA, 1999; GHANI, 2001; SHAHSHAHANI; LANDGREBE, 1994; SEEGER, 2000). Nesses trabalhos, dados não-rotulados aumentam o desempenho do classificador e levam a conclusões otimistas com relação ao valor dos dados não-rotulados. A utilização de redes Bayesianas é motivada pelo seguinte raciocínio. Quando um classificador é construído a partir de uma base de dados, em geral um "modelo" é assumido — ou seja, as probabilidades e decisões que constituem o classificador são parametrizadas de alguma forma específica. Por exemplo, classificadores como o *Naive Bayes* (NB) ou o *Tree-Augmented Naive Bayes* (TAN) assumem várias relações de independência entre atributos. Sabe-se que o conhecimento do modelo probabilístico correto garante a construção de um classificador ótimo (um classificador que minimiza o erro de classificação). Por isso, faz sentido adotar uma família de modelos que possam parametrizar modelos arbitrários, longe das restrições de classificadores como NB e TAN. Redes Bayesianas representam uma família de modelos com essas características. Neste trabalho adotamos redes Bayesianas como a família básica de classificadores. Os algoritmos

SSS e CBL-EM, propostos neste trabalho, são algoritmos que procuram utilizar todo o potencial e a flexibilidade de redes Bayesianas no contexto de aprendizado semi-supervisionado.

Neste trabalho foram utilizados os classificadores da literatura NB, TAN e uma adaptação do método *Structural-EM* (SEM) como base para estudos em aprendizado semi-supervisionado. Esses classificadores foram comparados a classificadores gerados por dois novos algoritmos, SSS e CBL-EM. O primeiro algoritmo de aprendizado, denominado SSS, foi desenvolvido juntamente com o pesquisador Ira Cohen (o trabalho principal de desenvolvimento foi realizado por ele). O algoritmo SSS realiza uma busca estocástica de redes Bayesianas guiada pelo erro estimado de classificação. O segundo algoritmo de aprendizado, denominado CBL-EM, é uma adaptação do algoritmo de aprendizado CBL1 (CHENG; BELL; LIU, 1997), que se baseia em testes de independência condicional para a escolha de uma rede.

1.1 Escopo

Neste trabalho foi escolhida uma abordagem estatística do problema da classificação, ou seja, funções de distribuição de probabilidade representam a situação de interesse e os parâmetros dessa distribuição são estimados utilizando-se o método da máxima verossimilhança.

Mostrou-se conveniente, também, trabalhar apenas com bases de dados discretas. Do ponto de vista prático, é sempre possível discretizar atributos contínuos presentes em bases de dados; portanto trabalhar com distribuições multinomiais discretas é pouco restritivo.

A literatura de aprendizado semi-supervisionado tem focado na utilização de poucos dados rotulados em comparação com a quantidade de dados não-rotulados. Os algoritmos de classificação estudados neste trabalho procuram utilizar uma quantidade maior de dados rotulados do que o encontrado na literatura, por considerar que essa é uma situação de maior relevância prática.

Durante a pesquisa feita para esta dissertação foi verificado que existem muitos caminhos possíveis que um analista pode tomar ao se deparar com o problema de construir um classificador Bayesiano a partir de uma base incompleta. Ele pode escolher um método sofisticado, como uma busca estocástica, ou mais simples como o NB-EM, ou qualquer um dos procedimentos entre esses dois extremos. O foco deste trabalho está em verificar em que condições a utilização dos dados não-rotulados é vantajosa e principalmente descrever uma metodologia para auxiliar o analista nessa tarefa.

1.2 Contribuições

Cinco contribuições distinguem este trabalho:

- Participação no desenvolvimento da primeira versão do algoritmo SSS, que representa o estado da arte em aprendizado semi-supervisionado (durante os primeiros meses do trabalho do mestrado, durante a visita do pesquisador Ira Cohen). Os resultados obtidos com o SSS representam como um teto de desempenho a ser obtido pelos outros classificadores.
- Desenvolvimento e implementação do algoritmo CBL-EM.
- Adaptação e implementação do algoritmo SEM (FRIEDMAN, 1997) para a tarefa de classificação.
- Comparação dos principais algoritmos para aprendizado semi-supervisionado de redes Bayesianas, com testes realizados em bases de dados reais (quatro bases provenientes do repositório da UCI (BLAKE; MERZ, 1998) e mais duas bases de reconhecimento de imagens).
- Descrição de uma metodologia que deve auxiliar os interessados em treinar classificadores utilizando dados rotulados e não-rotulados a experimentar os diversos algoritmos existentes e principalmente fazer o melhor uso dos algoritmos mais simples e de complexidade polinomial.

Como resultado desse trabalho, houve participação em artigo publicado na PAMI (*IEEE Transactions on Pattern Analysis and Machine Intelligence*) (COHEN et al., 2005), artigos publicados em congressos internacionais (COHEN et al., 2003; COZMAN; COHEN; CIRELO, 2003). Houve também publicação de resultados em congresso nacional (CIRELO; COZMAN, 2003).

2 EMBASAMENTO TEÓRICO

Neste capítulo são apresentados alguns conceitos necessários para a descrição dos algoritmos de classificação desenvolvidos no trabalho.

2.1 Classificação estatística

A tarefa de classificação consiste em associar um rótulo c a um vetor de atributos $\mathbf{x}$. Uma abordagem probabilística envolve ou a modelagem da distribuição conjunta dos atributos e rótulos, $P(C,\mathbf{X})$, ou a modelagem da distribuição *posterior* dos rótulos dados os atributos, $P(C|\mathbf{X})$. Conhecida a distribuição $P(C|\mathbf{X})$, a escolha do rótulo que maximiza a taxa de acerto esperada do classificador é feita utilizando-se a regra de decisão de Bayes (DUDA; HART; STORK, 2000), ou seja, escolhe-se o rótulo com maior probabilidade posterior. No entanto, $P(C|\mathbf{X})$ em geral não é conhecida e deve ser estimada a partir de uma base de dados D.

O problema está justamente na dificuldade de se estimar a distribuição conjunta ou a distribuição posterior. A solução mais comum está em assumir relações de independências entre os atributos, reduzindo o número de parâmetros livres da distribuição $P(C,\mathbf{X})$.

O mais simples classificador Bayesiano, o classificador *Naive Bayes*, parte do pressuposto de que todos os atributos são independentes se o rótulo for conhecido. Dessa forma temos uma classificador bastante compacto cujo número de parâmetros cresce linearmente com o número de atributos. A distribuição posterior representada por esse classificador é[1]:

[1] Na equação 2.1, assim como nas demais, as letras maiúsculas são utilizadas para representar variáveis aleatórias discretas, letras minúsculas indicam variáveis observadas, $|X|$ indica o número de valores possíveis que dada variável aleatória pode assumir, $|\mathbf{X}|$ representa o número de elementos do conjunto $\mathbf{X}$. Rótulos são representados por uma única variável C denominada variável de classe.

$$P(C|\mathbf{X}) = \frac{P(C)\ \prod_i^{|\mathbf{X}|} P(X_i)}{\sum_i^{|C|} P(C_i)\ \prod_i^{|\mathbf{X}|} P(X_i)}. \tag{2.1}$$

Nos problemas de interesse prático a hipótese de independência em geral não se sustenta e é comum tentar construir classificadores que permitam representar relações mais complexas entre atributos.

Redes Bayesianas, que constituem a base dos algoritmos apresentados neste texto, permitem a representação de distribuições conjuntas de probabilidades com relações arbitrárias de independência entre variáveis. Assim, redes Bayesianas se apresentam como modelos naturais para representar relações de dependência e independência entre atributos.

2.2 Redes Bayesianas e estimação de parâmetros

Redes Bayesianas são representadas pelo par $\langle S,\ \theta\rangle$, onde S é um grafo acíclico direcionado (DAG, na sigla em inglês) e θ é o conjunto de parâmetros que quantifica a distribuição. Nesses grafos as variáveis correspondem a nós enquanto relações de dependência direta são representadas por arcos (PEARL, 1988).

A estrutura de uma rede Bayesiana é o conjunto de relações de independência entre as variáveis de interesse: toda variável é independente das variáveis que não são suas descendentes, dados seus pais. Essas relações podem ser representadas por grafos acíclicos (SPIRTES; GLYMOUR; SCHEINES, 2000). Com efeito, parte da relevância das redes Bayesianas está em aproveitar as propriedades matemáticas e computacionais das operações com grafos (LAURITZEN, 1996).

Alguns conceitos são especialmente importantes e, por isso, são reproduzidos a seguir. Uma revisão desses conceitos pode ser encontrada em (CHICKERING, 2002; HECKERMAN, 1995).

- *Ordenação.* Para um DAG podemos extrair uma ou mais seqüências de variáveis, em que cada seqüência é ordenada segundo a seguinte regra: cada variável pode ser pai de uma ou mais das variáveis subseqüentes, mas não pode ser pai de nenhuma das variáveis anteriores. O caso inverso também é de interesse: pode ocorrer que a ordenação seja conhecida antes mesmo que uma DAG exista.

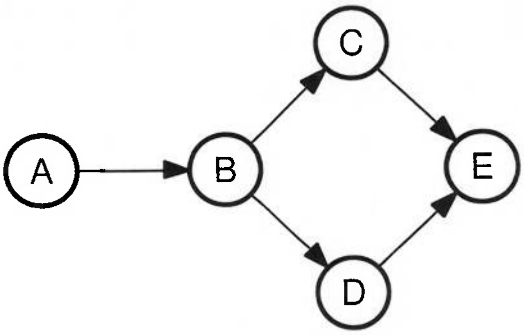


Figura 1: Grafo direcionado acíclico. Exemplo extraído de Spirtes, Glymour e Scheines (2000).

Nesse caso, existem algoritmos eficientes para o aprendizado quando a ordenação é conhecida (COOPER; HERSKOVITS, 1992; CHENG; BELL; LIU, 1997).

- *Estruturas-V.* Conjunto de três nós e dois arcos que apontam para o mesmo nó. Por exemplo, sub-estrutura $C \rightarrow E \leftarrow D$ da figura 1.

- *Equivalência.* Quando comparamos duas estruturas, dizemos que são equivalentes se possuírem o mesmo conjunto de variáveis e de arcos (ignorando suas direções) e as mesmas estruturas-V. A equivalência entre estruturas não implica equivalência entre as distribuições.

- *Estrutura Completa.* Estrutura que não contém nenhuma relação de independência entre as variáveis. Há arcos conectando cada par de variáveis e suas orientações são dadas pela ordenação das variáveis.

- *Estrutura Generativa.* Grafos nos quais o nó que corresponde à classe C é um nó que não tem pais. As distribuições cujas relações de independência estão restritas a uma estrutura generativa podem ser fatoradas na forma $P(C)\prod_{\mathbf{X}} P(X \mid Pa(X))$.

- *D-separação.* Apenas observando a configuração de um grafo acíclico é possível fazer afirmações sobre as relações de dependência entre as variáveis usando-se a propriedade da *d-separação*. Se duas variáveis X_i e X_j estão *d-separadas* por um conjunto $\mathbf{V}$, então elas serão independentes quando as variáveis que compõem $\mathbf{V}$ forem observadas. A definição de *d-separação* é apresentada na seção 3.2, onde é utilizada para justificar o algoritmo CBL1.

Os parâmetros de uma rede Bayesiana são os valores de probabilidade que definem uma única distribuição conjunta sobre todas as variáveis da rede. Para tanto, cada nó

X é associado a uma distribuição $P(X \mid Pa(X))$, onde $Pa(X)$ indica o conjunto de pais de X no grafo.

O valor de $P(C \mid \mathbf{X})$ é inferido usando-se a fórmula de Bayes:

$$P(C \mid \mathbf{X}) = \frac{P(C, \mathbf{X})}{P(\mathbf{X})} \tag{2.2}$$

Para obter os parâmetros da rede que permitem o cálculo destes valores de probabilidade, pode-se considerar a maximização da verossimilhança com relação a uma base de dados D. Note-se que neste trabalho sempre consideramos a estimação de parâmetros como a busca de parâmetros que maximizam verossimilhança, e sempre tomamos a verossimilhança como o logaritmo da probabilidade dos dados observados (DUDA; HART; STORK, 2000; GHAHRAMANI; JORDAN, 1994).

No caso de uma base de dados completa (ou seja, os valores de classe e atributos são conhecidos para todo registro), a verossimilhança pode ser maximizada de forma fechada. Com efeito, para dados rotulados a estimação de parâmetros reduz-se simplesmente a uma contagem feita sobre a base de dados (FRIEDMAN; GEIGER; GOLDSZMIDT, 1997).

$$\hat{P}(C = c) = \frac{n(C = c)}{m}, \quad \hat{P}(X = x \mid C = c) = \frac{n(X = x, C = c)}{n(C = c)}, \tag{2.3}$$

onde $n(.)$ representa o número observado de ocorrências encontradas na base de dados, e m é o número de exemplos existentes na base.

Para bases contendo dados não-rotulados, a maximização de verossimilhança é mais complexa. Considere inicialmente que a estrutura da rede Bayesiana é conhecida e desejamos estimar apenas os parâmetros θ (que são as distribuições associadas às variáveis da rede). O logaritmo da verossimilhança (LL) de uma base de dados D é:

$$\begin{aligned} LL(\theta \mid D) &= \log \prod_{i=1}^{|D'+D''|} P(\mathbf{x_i}, c_i \mid \theta) && (2.4) \\ &= \underbrace{\log \left(\prod_{i=1}^{|D'|} P(\mathbf{x_i}, C = y_i \mid \theta) \right)}_{\text{Dados Rotulados}} + \underbrace{\log \left(\prod_{i=|D'|+1}^{|D''|} \sum_{j=1}^{|C|} P(\mathbf{x}_i, c_{ij} \mid \theta) \right)}_{\text{Dados Não-rotulados}} && (2.5) \end{aligned}$$

$$= \underbrace{\sum_{i=1}^{|D'|} \log \left(P(C = y_i) \prod_{\mathbf{x}_i} P(x \mid Pa(x_i), \theta) \right)}_{\text{Dados Rotulados}} + \tag{2.6}$$

$$\underbrace{\sum_{i=|D'|+1}^{|D''|} \log \left(\sum_{j=1}^{|C|} P(c_{ij} \mid \theta) \prod_{\mathbf{x}_i} P(x \mid Pa(x_i), \theta) \right)}_{\text{Dados Não-rotulados}} , \tag{2.7}$$

onde y_i contém o valor verdadeiro do rótulo.

Nessa derivação, a verossimilhança dos parâmetros se decompõe sobre os registros rotulados e não-rotulados. Note que a equação (2.6) é a expressão de verossimilhança usada em aprendizado supervisionado convencional (ou seja, com todos os dados rotulados), e que leva a expressões fechadas para estimação. Já para a maximização do segundo membro da equação de verossimilhança (2.7) não existe solução fechada. Por isso, utilizamos o algoritmo iterativo EM (do inglês *Expectation-Maximization*) para maximização (GHAHRAMANI; JORDAN, 1994). Esse algoritmo e aspectos de implementação sua são discutidos a seguir.

O algoritmo EM é composto por duas etapas. No contexto de aprendizado semi-supervisionado, temos uma etapa de estimação de rótulos e uma etapa de estimação de parâmetros. Indicamos por θ^t os valores de parâmetros na t-ésima iteração do algoritmo — ou seja, estimativas de todas as probabilidades na rede. Temos:

- *Passo E:* Os valores esperados da variável C são calculados usando a estimativa atual dos parâmetros θ^t.
- *Passo M:* Calcula as estimativas de máxima verossimilhança como se a base de dados original tivesse sido complementada com os valores esperados pela classe; essas estimativas passam a ser θ^{t+1}.

No passo M, os estimadores de máxima verossimilhança utilizados são os mesmos que seriam utilizados caso todos os dados estivessem observados — essa é uma das características do EM.

A cada iteração calcula-se o valor atual de verossimilhança e compara-se esse valor com obtido na iteração anterior. Caso a diferença esteja abaixo de um limite especificado o processo é interrompido, pois o algoritmo convergiu para um máximo local da função de verossimilhança. O algoritmo também é interrompido caso um número máximo de iterações tenha sido alcançado.

No passo M do algoritmo EM a maximização dos parâmetros se reduz a uma simples contagem de "co-ocorrências probabilísticas" entre atributos na base de dados. No entanto, existe o problema da ocorrência de zeros nas tabelas de probabilidades condicionais (quando os valores de dois ou mais atributos nunca ocorrem simultaneamente na base de dados). Na verdade, dois são os problemas que os zeros acarretam. Em primeiro lugar, impedem o cálculo do logaritmo da verossimilhança (porém esse caso poderia ser facilmente contornado substituindo-se o zero por um número arbitrariamente pequeno). O segundo problema, mais sério, é decorrência do caráter iterativo do algoritmo EM. Zeros que aparecem em tabelas de probabilidade condicional impedem que a probabilidade posterior da classe seja atualizada nas iterações subseqüentes do algoritmo. Tendo em vista esses problemas, foi realizado amplo estudo sobre como contornar o aparecimento de zeros no algoritmo EM.

Após diversos testes realizados ficou claro que não seria recomendável simplesmente substituir o parâmetro com valor zero por uma constante, devido a sua influência no conjunto de parâmetros calculado, e conseqüentemente no desempenho do classificador. Apesar dessa questão ser levantada em muitos trabalhos no campo de aprendizado de máquina ((FRIEDMAN; GEIGER; GOLDSZMIDT, 1997; NIGAM et al., 2000; MCLACHLAN, 1992) para citar alguns), poucos trabalhos levantam uma comparação objetiva entre os principais métodos disponíveis. Um trabalho que apresenta essa comparação é Kohavi, Becker e Sommerfield (1997). Naquele trabalho, um método denominado *Laplace Smoothing* com fator de correção dependente do tamanho da base de dados é recomendado. Temos, então, os seguintes estimadores:

$$\hat{P}(C=c) = \frac{n(C=c)+1/m}{m+\mid C \mid /m}, \quad \hat{P}(X=x \mid C=c) = \frac{n(X=x, C=c)+1/m}{n(C=c)+\mid X \mid /m}. \quad (2.8)$$

Um problema da implementação do algoritmo EM é a sua inicialização (ou seja, a primeira estimativa dos parâmetros θ). Neste trabalho, adotamos como ponto de partida a porção de dados rotulados relevante a cada parâmetro. Com efeito, a quantidade de dados rotulados disponível tem importante papel no desempenho do classificador ao diminuir a importância de rotinas para eliminação de zeros.

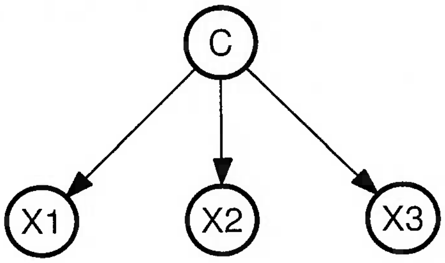


Figura 2: Rede Bayesiana tipo Naive Bayes.

2.2.1 *Naive Bayes* e *Tree-Augmented Naive Bayes*

Um classificador *Naive Bayes* (NB) pode ser representado como uma rede Bayesiana com estrutura fixa e conhecida (como exemplo veja a figura 2). Por essa razão, o aprendizado semi-supervisionado de classificadores NB pode ser feita por algoritmo EM — chamado aqui NB-EM.

Nigam et al. (2000) apresenta uma descrição detalhada da aplicação de NB-EM para classificação de textos da Usenet. É interessante notar que apesar das simplificações adotadas pelo *Naive Bayes*, Nigam et al. (2000) obtém resultados bastantes promissores em bases de dados com muitos atributos e com uma quantidade de dados rotulados muito reduzida.

A estimação supervisionada de redes Bayesianas restritas a árvores foi inicialmente proposta por Chow e Liu (1968). O algoritmo desenvolvido naquele artigo permite recuperar, em tempo polinomial, a árvore que melhor representa a base de dados (em termos de distância de Kullback-Leibler (KL)). O pseudo-código desse algoritmo, aqui chamado de CL, está no Algoritmo 1. Note que o passo 3 pode ser feito em tempo polinomial utilizando algum algoritmo para a determinação da árvore que maximiza a soma dos pesos dos nós (esses algoritmos são conhecidos por MWST sigla para *Maximum Weight Spanning Tree*).

No caso do algoritmo CL, o peso de cada arco $\langle X_i, X_j \rangle$ é dado pela Informação Mútua de X_i e X_j. O cálculo da Informação Mútua é feito pela seguinte equação:

$$\begin{aligned} MI(X_i, X_j) &= KL(\hat{P}(X_i, X_j), \hat{P}(X_i)\hat{P}(X_j)) && (2.9) \\ &= \sum_{X_i, X_j} \hat{P}(X_i, X_j) \log \frac{\hat{P}(X_i, X_j)}{\hat{P}(X_i)\hat{P}(X_j)}, && (2.10) \end{aligned}$$

onde a somatória percorre todos os valores possíveis para X_i e X_j, e o circunflexo indica

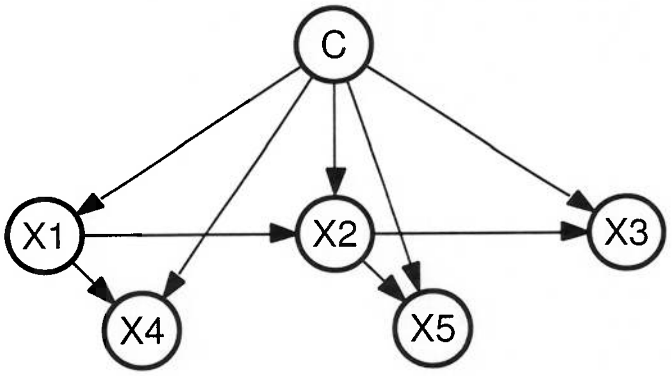


Figura 3: Rede Bayesiana tipo TAN.

que valores de probabilidade são estimados a partir da base de dados.

```
1. $L$ recebe uma lista com todos os arcos não direcionados possíveis.
2. **Para** cada arco $< X_i, X_j >$ em $L$.
    2.1. Associar ao arco $< X_i, X_j >$ o "peso" $MI(X_i,X_j)$.
3. $S$ recebe a árvore que maximiza a soma dos pesos dos arcos.
4. Escolhe uma variável qualquer e orienta todos os arcos partindo dessa
   variável
```

Algoritmo 1: Algoritmo CL para de redes Bayesianas restritas a árvores.

O trabalho de Friedman, Geiger e Goldszmidt (1997) adaptou esse algoritmo para classificação. Foi proposto um novo tipo de classificador, denominado *Tree Augmented Naive Bayes* (TAN), similar ao classificador NB, porém onde cada atributo pode ter um pai além da classe. Como resultado temos redes como aquela apresentada na figura 3 — se desconectarmos a classe dos atributos, temos uma árvore sobre os atributos. Para gerar um classificador TAN a partir de dados rotulados, Friedman, Geiger e Goldszmidt (1997) modificaram o algoritmo CL. A modificação básica é o uso de Informação Mútua Condicional à classe no lugar da expressão (2.10):

$$\begin{aligned} CMI(X_i,X_j \mid C) &= KL(\hat{P}(X_i,X_j \mid C),\hat{P}(X_i \mid C)\hat{P}(X_j \mid C)) && (2.11) \\ &= \sum_{X_i,X_j,C} \hat{P}(X_i,X_j,C) \log \frac{\hat{P}(X_i,X_j \mid C)}{\hat{P}(X_i \mid C)\hat{P}(X_j \mid C)}. && (2.12) \end{aligned}$$

É garantido que a estrutura obtida por esse algoritmo é a estrutura de maior verossimilhança entre as possíveis pelas restrições do modelo.

O algoritmo para TAN, por sua vez, foi posteriormente adaptado para aprendizado semi-supervisionado por Meila e Jordan (2000). O trabalho de Meila foi originalmente focado no aprendizado não-supervisionado de estruturas restritas a árvores, não necessariamente aplicadas para classificação. O resultado, aqui denominado TAN-EM, permite o aprendizado conjunto de estrutura e dos parâmetros, e é discutido na próxima seção.

2.3 Aprendizado da estrutura e dos parâmetros simultaneamente

Nesta seção apresentamos dois algoritmos disponíveis na literatura para aprendizado semi-supervisionado de estruturas para classificação: TAN-EM e SEM. O próximo capítulo propõe dois novos algoritmos para a seleção de redes Bayesianas arbitrariamente complexas a partir de busca ou de testes de independência, SSS e CBL-EM.

2.3.1 TAN-EM

As duas características mais importantes do algoritmo TAN são a garantia que irá recuperar a melhor estrutura (dentro do universo de estruturas restritas a árvores), e que o aprendizado é realizado em tempo polinomial. Assim, é natural que o algoritmo TAN seja um candidato ao aprendizado semi-supervisionado juntamente com o algoritmo EM. Meila-Predoviciu (1999) descreve o algoritmo TAN-EM, que adapta EM para aprendizado de TAN. Veja esquema no algoritmo 2.

```
1. $< S_0, \theta^0 > \leftarrow CL(D')$   /* Rede Inicial */
2. Faça até a convergência, $t = 0, 1, \ldots$
    2.1. $D^* \leftarrow$ calcula "rótulos probabilísticos" a partir de $< S_t, \theta^t >$ e
         das bases de dados $D'$ e $D''$.
    2.2. $< S_{t+1}, \theta^{t+1} > \leftarrow CL(D^*)$
```

Algoritmo 2: Algoritmo TAN-EM. O símbolo D^* é a lista que contém os não-rotulados complementados em uma dada iteração do EM. A saída produzida é uma rede Bayesiana $< S^t, \theta^t >$. Os "rótulos probabilísticos" correspondem ao valor esperado para o rótulo, ou seja $P(C \mid \mathbf{X})$

O algoritmo TAN-EM se destaca por sua simplicidade. Como todo o conjunto de parâmetros a ser estimado é conhecido, estes podem ser estimados de maneira eficiente em apenas uma passagem pelos dados. Além disso, a cada passo recupera-se a Informação Mútua entre os atributos dada a classe; Sahami (1998) utiliza essa informação para seleção de atributos.

2.3.2 Structural-EM

Originalmente o algoritmo SEM foi desenvolvido para estimar uma rede Bayesiana (grafo e parâmetros) de uma base de dados incompleta (FRIEDMAN, 1997). Este algoritmo realiza uma busca que maximiza a métrica MDL.

O MDL é uma métrica que busca um compromisso entre a aderência da rede Bayesiana ao conjunto de dados de treino (com relação à verossimilhança) e o número de bits necessários para armazenar a rede (o que é proporcional ao número de parâmetros da rede Bayesiana). O uso dessa métrica para o aprendizado de redes Bayesianas remonta ao trabalho de Lam e Bacchus (1994). A equação para calcular a métrica MDL de uma rede $\langle S, \theta \rangle$ é:

$$MDL = L(\theta \mid D) - \frac{\log m}{2} \mid S \mid \tag{2.13}$$

onde $\mid S \mid$ é o número de parâmetros da rede Bayesiana, m o número de registros disponíveis e D a base de dados disponíveis.

O algoritmo SEM é apresentado nos artigos de Friedman (1997, 1998). Os dois artigos diferem com relação ao critério para comparar as redes Bayesianas: o primeiro artigo, no qual a implementação apresentada neste trabalho foi baseada, utiliza o MDL, enquanto o segundo artigo utiliza uma métrica Bayesiana (COOPER; HERSKOVITS, 1992).

Usualmente o algoritmo EM está apenas envolvido no aprendizado de parâmetros, estando fixa a estrutura da rede. O algoritmo SEM é um extensão do EM de maneira que a cada passo-M do algoritmo, busca-se maximizar não apenas os parâmetros, mas também a estrutura utilizando o valor esperado da classe para os cálculos. Esse raciocínio é similar ao adotado por Meila-Predoviciu (1999) para justificar sua dedução do TAN-EM. No entanto, diferente do TAN-EM, não há como evitar a realização de uma busca para a escolha da estrutura da rede Bayesiana.

O algoritmo de busca implementado foi o *Hill Climbing*, o qual, a partir de uma estrutura inicial, lista todas as modificações possíveis (inserção de novo arco, remoção ou inversão dos já existentes). É nesta etapa que possíveis restrições na estrutura, ou algum conhecimento adicional, podem ser colocadas no modelo: arcos que possam causar ciclos orientados ou que violem a condição de estrutura generativa não são considerados como modificações válidas. Após cada modificação, o valor esperado do MDL é calculado. A modificação é revertida se o novo MDL for igual ou menor que o anterior. Fixada a rede, executa-se o EM convencional (para a estimação dos parâmetros). O algoritmo pára quando o incremento do MDL feito no último passo for menor que um determinado valor mínimo (definido em nossa implementação como 10^{-4}). Esquematicamente, o algoritmo SEM implementado é apresentado no Algoritmo 3.

```
1. $S^0 \leftarrow NaiveBayes$
2. $\theta^{0,0} \leftarrow$ Estima parâmetros $(S^0, D')$.
3. $D^* \leftarrow$ calcula "rótulos probabilísticos".
4. Repita até a convergência, $n = 1, 2, \ldots$
   4.1. Repita até a convergência, $l = 0, 1, \ldots$
        4.1.1. $\theta^{n,l+1} \leftarrow$ Estima parâmetro $(S^n, D^*)$.
   4.2. $D^* \leftarrow$ calcula "rótulos probabilísticos".
   4.3. $\mathbf{S} \leftarrow$ Estruturas candidatas $(S^n)$.
   4.4. Para cada estrutura $S^c$ em $\mathbf{S}$.
        4.4.1. $\theta^c \leftarrow$ Atualiza parâmetros $(S^c, D^*)$.
        4.4.2. $MDL^c \leftarrow$ Calcula MDL usando equação 2.13.
   4.5. $< S^{n+1}, \theta^{n+1,0} > \leftarrow$ Rede candidata com maior valor para o MDL
        calculado no passo 4.4.2.
```

Algoritmo 3: Algoritmo SEM.

2.4 Literatura relevante em aprendizado semi-supervisionado

O interesse em aprendizado semi-supervisionado é relativamente recente, embora o tema possa ser encontrado em publicações antigas (muitas vezes misturado com aprendizado não-supervisionado). A seguir são sumarizados três artigos recentes significativos e que motivaram o presente trabalho.

- O trabalho de Baluja (1999) aplica dados rotulados e não-rotulados para a classificação de orientação de faces (cinco posições variando do frontal ao perfil). São utilizados classificadores baseados no EM com estrutura fixa como o (NB) e estrutura variável (estruturas restritas a árvores).

- O artigo de Nigam et al. (2000) é geralmente citado como um dos pioneiros na utilização prática de classificadores Bayesianos para o aprendizado semi-supervisionado. O artigo detalha a implementação do algoritmo NB-EM especialmente destinado a classificação de textos. Nigam et al. (2000) realiza seus testes com bases de texto provenientes de mensagens escritas em listas de discussão na Internet. O objetivo proposto é classificar esses textos segundo seu tema. Como em outros trabalhos, a base de dados utilizada tem muitos atributos, é esparsa e a quantidade de dados rotulados é insignificante.

- No artigo de Blum e Mitchell (1998) é descrito o algoritmo *co-training*. Esse método necessita duas bases de dados redundantes, mas não completamente correlacionadas. O *co-training*, como descrito no artigo Blum e Mitchell (1998) tem a vantagem de se basear em classificadores convencionais para aprendizado semi-supervisionado. Com efeito, não há restrições com relação ao classificador a ser utilizado que poderia nem mesmo ser probabilístico. O sucesso do algoritmo co-training deu grande impulso à área de aprendizado semi-supervisionado nos últimos anos.

3 ALGORITMOS PARA APRENDIZADO SEMI-SUPERVISIONADO BASEADOS NA BUSCA PELO MELHOR MODELO

Neste capítulo são apresentados os dois novos algoritmos; o algoritmo SSS representa métodos de aprendizado baseados em busca, enquanto o algoritmo CBL se baseia em testes de independência. A idéia central é construir redes Bayesianas genéricas que representem a distribuição $P(C, \mathbf{X})$ em problemas de classificação. A partir dessa distribuição, uma decisão ótima sobre rótulos pode ser obtida. Dessa forma, o problema pode ser encarado como a estimação de redes Bayesianas genéricas a partir de dados rotulados e não-rotulados.

O algoritmo SSS foi desenvolvido primariamente pelo pesquisador Ira Cohen durante visita à Universidade de São Paulo, em 2002. Houve participação do autor durante as fases de idealização, implementação e testes. O segundo classificador é baseado no algoritmo de Cheng, Bell e Liu (1997) para aprendizado supervisionado, tendo sido adaptado para aprendizado semi-supervisionado.

3.1 Aprendizado de estruturas baseado em busca estocástica

O algoritmo SSS é baseado em busca estocástica; o algoritmo realiza uma busca no enorme espaço de estruturas de redes Bayesianas. Esta busca é guiada por uma métrica:

$$i(S') = \frac{1/p_{erro}(S')}{\sum_S 1/p_{erro}(S)}, \tag{3.1}$$

que, por sua vez, é baseada no erro de classificação estimado:

$$p_{erro} = \frac{\text{número de registros incorretamente rotulados}}{\text{número total de registros}}. \tag{3.2}$$

O algoritmo SSS procura gerar redes Bayesianas que tenham altos valores para essa métrica, varrendo uma porção do espaço de redes Bayesianas válidas. A métrica do SSS é normalizada para definir uma distribuição de probabilidades a partir do p_{erro}. O objetivo é dar ao SSS a capacidade de realizar uma busca mais bem informada do que algoritmos como o *Hill climbing* poderiam fazer. No entanto, a equação 3.1 não pode ser calculada diretamente pois a somatória do denominador percorre o vasto conjunto de redes Bayesianas válidas (varrer o espaço de estruturas somente é viável para problemas poucas variáveis). Por essa razão, foi utilizado o algoritmo Metropolis-Hastings (METROPOLIS et al., 1953) para realizar a busca. O algoritmo Metropolis-Hastings permite sortear elementos de distribuições que não podem ser diretamente caracterizadas.

O algoritmo Metropolis-Hastings é centrado na geração de um "estado candidato" e na probabilidade de aceitação desse estado candidato, dada pela equação (3.3). No presente contexto cada "estado" é uma rede Bayesiana. A probabilidade de aceitação depende do estado atual S^{atual}, o estado candidato S^{novo} e também de $q(S^a \mid S^b)$, que é a probabilidade de sair do estado S^b e ir para um estado S^a.

$$\min\left(1, \left(\frac{i(S^{novo})}{i(S^{atual})}\right)^{1/T} \frac{q(S^{atual} \mid S^{novo})}{q(S^{novo} \mid S^{atual})}\right) \tag{3.3}$$

Os estados percorridos durante a busca são determinados pela transições permitidas entre redes Bayesianas. O algoritmo SSS permite que dois estados subseqüentes possam diferir em apenas um arco (adicionado, removido ou invertido). Temos que para um dado estado S^{atual} podem haver Nv^{atual} possíveis novos estados. No SSS, definimos que a probabilidade de escolher o estado S^{novo} corresponde a $1/Nv^{atual}$, ou seja, distribuição uniforme.[1] De maneira análoga, Nv^{novo} corresponde ao número de

[1] Outras distribuições além da uniforme poderiam ser empregadas. Em especial poderíamos escolher alguma que privilegiasse estruturas mais simples (com menos parâmetros).

estados alcançáveis a partir de S^{novo}. Portanto, a probabilidade do novo estado ser aceito é:

$$\min\left(1, \left(\frac{p_{erro}(S^{atual})}{p_{erro}(S^{novo})}\right)^{1/T} \frac{Nv^{atual}}{Nv^{novo}}\right) \tag{3.4}$$

A descrição do algoritmo SSS está no Algoritmo 4.

```
1. $S^0 \leftarrow NaiveBayes$
2. $\theta^{0,0} \leftarrow$ Estima parâmetros $(S^0, D')$.
3. Estima $\hat{p}^0_{erro}$ a partir da base de validação.
4. $t = 0$.
5. **Repita** enquanto $t <$ Número máximo de iterações (MaxIter).
    5.1. Amostra nova estrutura $S^{novo}$, a partir da vizinhança de $S^t$, uniformemente, ou seja com probabilidade $\frac{1}{Nv^t}$.
    5.2. **Repita** até a convergência, $l = 0, 1, ....$
        5.2.1. $\theta^{novo} \leftarrow$ Estima parâmetro $(S^t, D', D'')$.
    5.3. Estima a probabilidade de erro da nova rede $\hat{p}^{novo}_{erro}$.
    5.4. $< S^{atual}, \theta^{atual} =< S^t, \theta^t >$
    5.5. Aceita $S^{novo}$ com probabilidade calculada por (3.4).
    5.6. **Se** $S^{novo}$ for aceita.
        5.6.1. $< S^{t+1}, \theta^{t+1} >=< S^{novo}, \theta^{novo} >$
        5.6.2. $\hat{p}^{t+1}_{erro} = \hat{p}^{novo}_{erro}$
    5.7. $t = t + 1$.
6. **Retorna** a estrutura $< S^j, \theta^j >$, cujo $j = \arg\min_{0 \leq j \leq MaxIter}(\hat{p}^j_{erro})$.
```

Algoritmo 4: Algoritmo SSS para aprendizado de classificadores utilizando busca estocástica.

Note que não existe uma métrica que relacione de forma simples a estrutura de redes Bayesianas com o desempenho de classificadores baseados em tais redes. Por essa razão o algoritmo SSS depende do erro de classificação estimado para guiar a sua busca.

3.2 Aprendizado de estruturas utilizando testes de independência

Nesta seção descrevemos a implementação de um classificador Bayesiano baseado no algoritmo CBL1. O objetivo aqui é investigar métodos de aprendizado que se baseiam em testes de independência, para concluir se esta estratégia de aprendizado pode levar a melhores resultados que a estratégia de busca estocástica.

Em algoritmos de aprendizado baseados em testes de independência, repetidos testes devem ser conduzidos para descobrir quais pares de atributos devem estar conectados por um arco ou não. O objetivo é gerar uma estrutura com conjunto mínimo de arcos necessários para representar as dependências entre variáveis (PEARL, 2000).

Um algoritmo que utiliza esta idéia é o PC (SPIRTES; GLYMOUR; SCHEINES, 2000). Esse algoritmo pode reconstruir uma rede Bayesiana utilizando apenas o resultado dos testes de independência, sem conhecimento *a priori* algum e, mesmo assim, comprovadamente, encontrar a rede correta. O custo dessa flexibilidade vem no número de testes de independência, que aumenta de maneira exponencial com relação ao número de atributos da base de dados. Por isso, o algoritmo PC é inviável para o aprendizado de redes com muitos atributos.

Dois exemplos podem esclarecer o tipo de conhecimento adicional em geral assumido por algoritmos de aprendizado, visando diminuir a complexidade computacional.

Primeiro, caso haja restrição para que toda variável tenha no máximo um pai, é possível aprender a rede em tempo determinístico (não há busca envolvida) utilizando o algoritmo CL (algoritmo de Chow e Liu para aprendizado de árvores) (CHOW; LIU, 1968).

Segundo, uma rede Bayesiana pode ser recuperada em tempo polinomial se os atributos da base de dados estejam previamente ordenados. A ordenação determina quais atributos podem ser pais dos demais. Por exemplo, se temos seqüência ordenada for X_1,X_3,X_2 temos que $X_1 \rightarrow X_2$, $X_3 \rightarrow X_2$ são arcos possíveis, mas $X_2 \rightarrow X_3$ não é válido. O algoritmo CBL1 proposto por Cheng (CHENG; BELL; LIU, 1997), discutido em mais detalhes adiante, é baseado na idéia de ordenação.[2]

[2] Cheng et al. também propõem um algoritmo, que como o algoritmo PC, não requer a prévia ordenação dos nós.

Resultados teóricos existentes sobre eficiência desses algoritmos partem do princípio que testes de independência sempre revelam a "verdade". Porém, quando utilizamos testes estatísticos, a eficiência do algoritmo está relacionada com o tamanho da base de dados e, no contexto do aprendizado semi-supervisionado, com o conjunto de parâmetros estimados. Devido à importância desse tópico no funcionamento do algoritmo CBL-EM, a próxima seção discute testes de independência.

3.2.1 Testes de independência de variáveis discretas

Um teste de hipóteses para independência procura decidir se duas variáveis, X_i e X_j, são independentes dado um conjunto de variáveis observadas, $\mathbf{S}$. A hipótese *nula* corresponde à situação em que a probabilidade conjunta $P(X_i, X_j \mid \mathbf{S})$ fatora enquanto que a hipótese alternativa nega essa possibilidade:

$$H_0 \ : \ \hat{P}(X_i \mid \mathbf{S})\hat{P}(X_j \mid \mathbf{S}) \tag{3.5}$$

$$H_1 \ : \ \hat{P}(X_i\, X_j \mid \mathbf{S}). \tag{3.6}$$

Foram utilizados dois testes para escolher entre as hipóteses: o teste de Informação Mútua condicional e o teste G^2.

3.2.1.1 Testes de independência usando Informação Mútua

A Informação Mútua (COVER; THOMAS, 1991), ou *cross entropy*, é um caso especial da medida da distância Kullback-Leibler de distribuições, avaliando a distância entre as distribuições $P(X_i)P(X_j)$ e $P(X_i, X_j)$. De maneira análoga, define-se a Informação Mútua Condicional, com relação às distribuições $P(X_i, X_j \mid \mathbf{S})$. As definições de Informação Mútua e Informação Mútua Condicional já foram apresentadas durante a descrição do algoritmo TAN (equações 2.10 e 2.12).

Caso a hipótese H_0 seja verdadeira (variáveis independentes), $MI(X_i, X_j) = 0$; caso H_1 seja verdadeira, essa quantidade é maior que zero. Como, na prática, as bases de dados reais são finitas e sujeitas a ruídos diversos, a Informação Mútua entre variáveis nunca é exatamente zero, mesmo que haja independência. Como não é simples descrever a distribuição da Informação Mútua (HUTTER, 2002), em geral utiliza-se uma pequena faixa de tolerância $[0, \delta]$ dentro da qual as variáveis são consideradas independentes. Na implementação usada para os experimentos neste trabalho $\delta = 0.01$.

Durante o aprendizado de uma rede Bayesiana na prática, podem ocorrer casos em que testes de independência condicionais envolvam um grande número de variáveis. Como é necessário estimar a probabilidade conjunta das variáveis envolvidas no teste, podemos nos deparar com situações em que o número de parâmetros é da mesma ordem de grandeza dos dados disponíveis, o que resulta em testes com baixo significado estatístico. Para evitar essas situações e manter a estrutura tão esparsa quanto possível, em nossa implementação interrompemos o teste e retornamos a H_0 como verdadeira quando temos uma relação número de parâmetros sobre número de registros superior a 0.1.

Observamos empiricamente que, caso a quantidade de dados disponíveis seja pequena, a estimativa de Informação Mútua não é confiável, e freqüentemente não há como saber se as variáveis testadas são efetivamente independentes. A escolha de δ passa a ser crucial. Nesse sentido, Meila-Predoviciu (1999) derivou um teto para o valor de δ baseado no tamanho da base de dados de treino e na dimensionalidade do teste (que, por sua vez depende do número de categorias das variáveis). Infelizmente, nos testes realizados neste trabalho, o δ fornecido por esse método era muito conservador para que fosse útil. O caminho buscado para contornar essa dificuldade foi a implementação do teste G^2, discutido a seguir.

3.2.1.2 Testes de independência usando G^2

A estatística G^2 é definida como (AGRESTI, 1990):

$$G^2 = 2 \sum_{X_i,\, X_j,\, \mathbf{S}} \hat{P}(X_i,\, X_j \mid \mathbf{S}) \ln\left(\frac{\hat{P}(X_i,\, X_j \mid \mathbf{S})}{\hat{P}(X_i \mid \mathbf{S})\hat{P}(X_j \mid \mathbf{S})}\right). \tag{3.7}$$

A estatística G^2 distribui-se assintoticamente segundo a distribuição χ^2 com grau de liberdade ν proporcional ao número de parâmetros que o teste envolve:

$$\nu = (\mid X_i \mid -1)(\mid X_j \mid -1)(\prod_{i=1}^{|\mathbf{S}|} \mid S_i \mid) \tag{3.8}$$

Com o resultado do cálculo do G^2 e do ν, podemos determinar o *p-value*, que corresponde à probabilidade de rejeitar H_0 mesmo ela sendo verdadeira. Em todos os testes realizados, utilizou-se por regra decidir que duas variáveis eram independentes se *p-value* fosse inferior a 0.03.

3.2.2 A *d-separação*

No centro do aprendizado baseado em testes de independência está a propriedade da *d-separação* (PEARL, 1988) que relaciona a estrutura da rede com as relações de dependência entre as variáveis. Se **V** *d-separa* X_i de X_j então $X_i \perp\!\!\!\perp X_j \mid \mathbf{V}$.

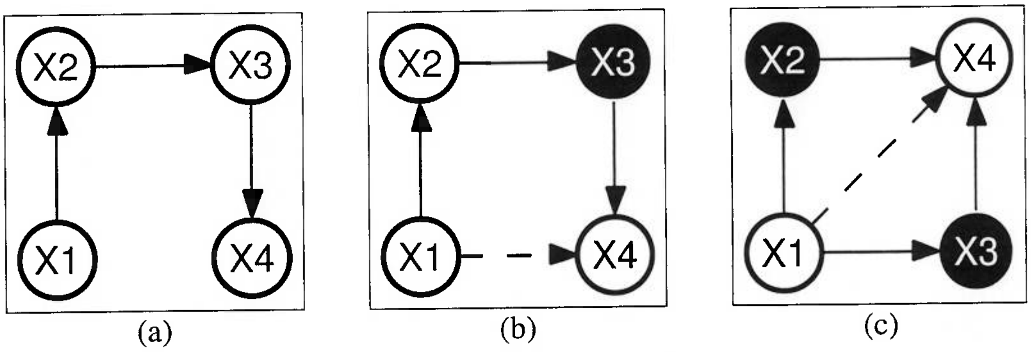
Definição: Diz-se que **V** *d-separa* X_i de X_j se não houver nenhum caminho partindo da variável X_i para a variável X_j em que:

1. todas as variáveis que possuem apenas arcos convergentes se encontram no conjunto **V**; e,

2. todos as demais variáveis não pertençam à **V**.

Para efeito de ilustração, a partir da figura 1, podemos inferir, usando a *d-separação* que: $C \perp\!\!\!\perp D \mid B$, porém $C \not\perp\!\!\!\perp D \mid B, E$.

No algoritmo CBL1, dadas duas variáveis é necessário determinar qual seria um grupo de variáveis que poderia separá-las, denominado *cutset*. Um teste de independência condicional é então realizado, por exemplo verificando se $MI(X_i, X_j \mid Cutset) = 0$. O ideal é encontrar o menor *cutset* possível, minimizando assim, o número de variáveis envolvidas no teste.



(a) (b) (c)

Figura 4: Determinação do *cutset*.

Nas redes em que a ordenação é previamente conhecida é muito mais simples determinar um *Cutset* válido. Com efeito, se numa dada ordenação X_i vier antes de X_j, um *Cutset* válido seria formando tomando-se os pais de X_j. Considere a seguinte situação: a partir do estado mostrado na figura 4(a), deve-se decidir pela inclusão ou não do arco $X_1 \longrightarrow X_4$ (figura 4(b)). Note que não há a necessidade de realizar um teste condicional ao grupo $\{X_2, X_3\}$; basta testar $X_1 \perp\!\!\!\perp X_4 \mid X_3$. O *cutset*, neste caso, é composto apenas por X_3. Na configuração de outro problema, o da figura 4(c), porém, o *cutset* é composto por $\{X_2, X_3\}$.

3.2.3 O classificador baseado no algoritmo CBL1

Um algoritmo baseado em testes de independência relatado na literatura é o algoritmo CBL1 (CHENG; BELL; LIU, 1997).[3] Esse algoritmo (veja algoritmo 5 requer uma ordenação pré-estabelecida para as variáveis.

O algoritmo é composto por três partes:

- Na primeira parte, obtemos um esboço da estrutura. Em nossa implementação esta fase produz a uma estrutura do tipo *Naive Bayes*. Esta etapa é portando realizada em apenas um passo.
- Na segunda parte (*loop* iniciado no passo 4), realizam-se testes de independência condicional para determinar quais arcos devem ser adicionados. Testam-se todos os pares de variáveis X_a, X_b com $a \neq b$ e X_a anterior à X_b na ordenação. Os testes são condicionais ao chamado *cutset*, que corresponde ao conjunto de variáveis capazes de *d-separar* X_a e X_b (SPIRTES; GLYMOUR; SCHEINES, 2000; PEARL, 2000). Sabemos que um *cutset* sempre válido é composto pelos pais de X_b. Em alguns casos esse conjunto pode não ser o mínimo possível, porém como em geral trabalhamos com redes bastante esparsas essa fato não chega a causar problemas. Note que como a rede vai se completando a cada arco adicionado, pode ocorrer que alguns testes não sejam realizados com o *cutset* completo, podendo ocasionar a adição desnecessária de arcos. Por essa razão existe mais uma etapa para reduzir a rede ao mínimo necessário.
- Na terceira parte (segundo *loop*, passo 5) é realizado um novo teste de independência condicional para cada arco para se decidir se esse arco deve permanecer na estrutura ou ser removido.

[3] Esse algoritmo aparece com outros nomes em artigos mais recentes, no entanto, foi mantida a sigla original, CBL, que aparece nas publicações mais antigas, por ser uma referência direta ao nome dos autores: Cheng, Bell e Liu (1997).

A descrição completa do algoritmo está no Algoritmo 5.

```
1. $LC \leftarrow$ Lista de arcos já conectados na estrutura inicial (Naive Bayes)
2. $LD \leftarrow$ lista de arcos candidatos, respeitando uma ordenação pré-estabelecida, excluindo aqueles contidos em $LC$
3. $S \leftarrow$ conecta arcos presentes em $LC$
4. **Para cada** arco candidato $X_i \rightarrow X_j$ presente em $LD$
    4.1. $Cutset \leftarrow pais(X_j)$ na estrutura $S$
    4.2. **Se** $X_i \perp\!\!\!\perp X_j \mid Cutset$
        4.2.1. Conecta arco em $S$
        4.2.2. Adiciona arco em $LC$
        4.2.3. Remove arco de $LD$
5. **Para cada** arco $X_i \rightarrow X_j$ presente em $LC$
    5.1. Desconecta arco de $S$
    5.2. $Cutset \leftarrow pais(X_j)$ na estrutura $S$
    5.3. **Se** $X_i \perp\!\!\!\perp X_j \mid Cutset$
        5.3.1. Conecta arco em $S$
```

Algoritmo 5: Algoritmo CBL1 — Algoritmo para o aprendizado de redes Bayesianas utilizando testes de independência. Na descrição deste algoritmo os seguintes símbolos foram utilizados: S, grafo direcionado acíclico; $X_i \rightarrow X_j$, arco que liga o nó X_i ao nó X_j; LC, lista de arcos já adicionados ao grafo S; LD, lista de arcos candidatos; $Cutset$, lista de nós candidatos a *d-separar* dois nós.

3.2.4 Descrição do algoritmo CBL-EM

Esta seção apresenta uma adaptação do algoritmo CBL1 para aprendizado semi-supervisionado. A adaptação do CBL1 para o aprendizado semi-supervisionado é análoga ao descrito em SEM. Isto é, a estrutura da rede Bayesiana é aprendida a cada passo M do EM utilizando os melhores estimadores disponíveis naquela iteração. O Algoritmo 6 traz uma descrição desse processo iterativo.

Ao contrário dos algoritmos TAN-EM e SEM, a aprendizagem de estruturas utilizando testes de independência não garante que para cada rede encontrada maximize a verossimilhança dos dados com relação aos parâmetros.

O algoritmo, denominado CBL-EM, é interrompido se a verossimilhança não aumentar entre duas iterações. O CBL1 não é orientado por uma métrica global, mas sim por testes de independência. Dada a natureza estatística desses testes, é esperado que ocorram erros e alguns arcos podem ser erroneamente adicionados (ou omitidos). Como resultado, a verossimilhança pode diminuir entre duas iterações consecutivas. Nessa situação, o processo é interrompido e a última rede válida é retornada. Assim como para o algoritmo EM, foi definido um número máximo de iterações. Os testes realizados sugerem que o processo de convergência é rápido: em raras situações o CBL-EM realizou mais de seis iterações.

Ao contrário do TAN-EM, o CBL-EM não exige que um número mínimo de arcos seja adicionado. Esse é um atrativo adicional do CBL-EM para os problemas mais simples, em que os atributos da base de dados são em sua maioria independentes entre si dada a classe.

```
1. $S^0 \leftarrow$ Aprende estrutura usando CBL1 a partir de $D'$.
2. $\theta^0 \leftarrow$ Estima parâmetros a partir de $D'$.
3. $D^* \leftarrow$ Atualiza "rótulos probabilísticos" calculando $P(C \mid \mathbf{X}, \theta^0)$.
4. $t \leftarrow 0$.
5. Repita
    5.1. $S^t \leftarrow$ Aprende estrutura usando CBL1 a partir de $D^*$.
    5.2. $\theta^t \leftarrow$ Estima parâmetros, usando EM, a partir de $D^*$.
    5.3. $D^* \leftarrow$ Atualiza "rótulos probabilísticos" calculando $P(C \mid \mathbf{X}, \theta^t)$.
    5.4. Se $t > 1$.
        5.4.1. $MDL^t \leftarrow$ Calcula $MDL$ usando $S^t, \theta^t, D^*$.
        5.4.2. Se $t > j$, retorna $< S^t, \theta^t >$.
        5.4.3. Se $MDL^t - MDL^{t-1} < 0$, retorna $< S^{t-1}, \theta^{t-1} >$.
        5.4.4. Se $MDL^t - MDL^{t-1} < \delta$, retorna $< S^t, \theta^t >$.
    5.5. $t = t + 1$
6. Retorna $< S^t, \theta^t >$.
```

Algoritmo 6: Algoritmo CBL-EM. A variável j indica o número máximo de iterações do CBL-EM.

3.2.5 CBL-EM para seleção de variáveis

A seleção automática de variáveis é um subproduto do CBL1 que foi "herdado" pelo CBL-EM. Como esse algoritmo decide quais arcos serão adicionados/ removidos, o grafo resultante pode ter uma configuração em que algumas variáveis simplesmente não são relevantes para classificação. Consideram-se relevantes as variáveis cujo estado influencia no valor $P(C \mid \mathbf{X})$.

As variáveis descartadas pelo CBL-EM são aquelas que não estão diretamente conectadas com a classe e que não são pais de variáveis filhas da classe. A figura 5 (a) ilustra essa situação.

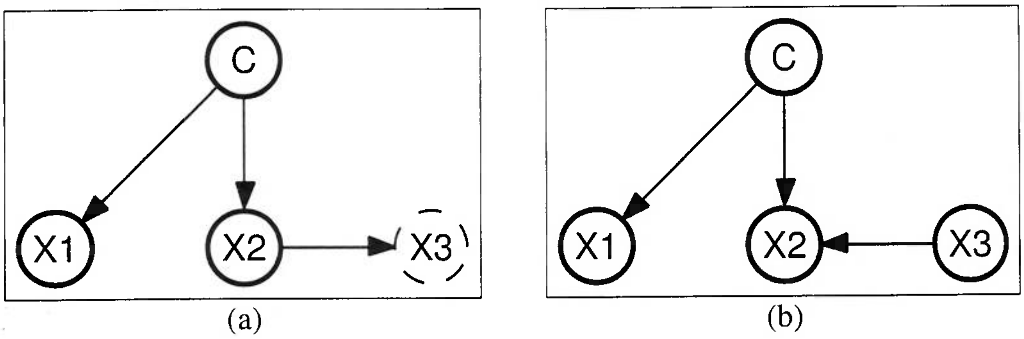


Figura 5: Figuras que exemplificam a seleção de atributos feita pelo CBL-EM. Na figura (a) a variável $X3$ não tem influência alguma para classificação; na figura (b), entretanto, a variável $X3$ ainda influencia indiretamente a distribuição *a posteriori* da classe já que é pai de uma variáveis filhas da classe.

4 TESTES E RESULTADOS

Este capítulo contém descrições de testes realizados durante o trabalho. O desempenho dos algoritmos aqui propostos (SSS e CBL-EM) é discutido e comparado ao desempenho dos algoritmos NB-EM, TAN-EM e SEM.

Foram utilizadas quatro bases de dados disponíveis no repositório da UCI. Essas bases foram escolhidas por terem uma grande quantidade de registros e terem menos de dez rótulos para a classe — a cardinalidade da classe tem influência direta no tempo de processamento do EM. Adicionalmente, foram utilizadas duas bases de dados fornecidas pelo pesquisador Ira Cohen, relacionadas a processamento de imagens.

Os testes nos permitem visualizar fenômenos interessantes do aprendizado semi-supervisionado, e mais importante, reconhecer onde nossas contribuições trouxeram progressos.

4.1 Descrição das bases de dados

Nesta seção são descritas as bases de dados utilizadas nos testes. Todas as bases são originadas de problemas reais sendo que quatro delas (Adult, Chess, Satimage e Shuttle) são provenientes do repositório da UCI (BLAKE; MERZ, 1998). As outras duas são bases de reconhecimento de expressões faciais (COHEN et al., 2003).

Na tabela 4.1 temos um resumo das características dessas bases de dados.

- *Adult*: Base que separa os americanos entrevistados em duas classes, os que recebem mais que $US\$50\,000$ ano e aqueles que recebem menos que esse valor. Os atributos contêm informações provenientes de um censo americano anterior a 1996 e trazem informações como idade, sexo, raça, nível educacional, etc. Esta base tem ao todo 13 atributos (originalmente eram 15 porém foram removidos os

Base de Dados	Treino		Teste
	# rotulado	# não-rotulado	
Satimage	600	3835	2000
Shuttle	100	43400	14500
Adult	6000	24163	15060
Chess	150	1980	1060
Cohn-Kanade	200	2980	1000
Chen-Huang	300	11982	3555

Tabela 1: Descrição das bases de dados.

atributos *país de origem* e *fnlwgt*, pois não eram relevantes para a classificação).

- *Chess:* Contém posições das peças no tabuleiro de xadrez e associa um rótulo *vence* ou *não vence* de acordo com a possibilidade das peças brancas vencerem o jogo. Além do rótulo existem 36 atributos, um para cada posição no tabuleiro.

- *Satimage:* Os registros representam pontos em imagens feitas de satélites de vários tipos de terreno. O objetivo é associar a imagem a um dos seis rótulos possíveis. Na base original há 36 atributos que podem ter um valor de intensidade de 0 até 255.

- *Shuttle:* Trata-se de um problema com nove atributos, todos numéricos, e sete valores possíveis para a classe. Uma das classes aparece em 80% dos registros, logo esse é a menor taxa de acerto possível de classificação. Quando todos os registros estão rotulados e a taxa de acerto é geralmente superior a 99%.

As bases de dados *Cohn-Kanade* e *Chen-Huang* estão relacionadas com o reconhecimento de expressões em segmentos de vídeo. Essas bases contêm as posições relativas de 12 marcações feitas nas imagens de rostos. Cada imagem na base de treino é rotulada com um seguintes valores: Neutralidade, Felicidade, Raiva, Desgosto, Medo ou Tristeza.

4.2 Preparação dos dados

Foram escolhidas as bases mais próximas do tipo de problema de interesse. Muitas das bases encontradas continham valores incompletos para seus atributos e/ou atributos contínuos. Registros que continham dados atributos com valor incompleto foram removidos. Em seguida, as bases completas (com os rótulos) foram discretizadas utilizando o pacote MLC++ (KOHAVI; SOMMERFIELD; DOUGHERTY, 1997) na sua configuração padrão.

Então a relação entre número de dados rotulados e não-rotulados foi definida; a partir desse número, sorteamos registros ao acaso e removemos o seu rótulo. Esse procedimento é necessário para garantir que o fato de determinado registro ter ou não um rótulo seja completamente independente dos valores dos atributos do registro. Para determinar a quantidade de dados rotulados foram levados em consideração o tamanho da base de dados disponível para treino e a complexidade do problema. A principal premissa era ter uma quantidade de dados rotulados suficientemente grande para que o ponto de partida do algoritmo EM não fosse completamente inválido. O tamanho da base de dados rotulados esteve entre 1% e 10%.

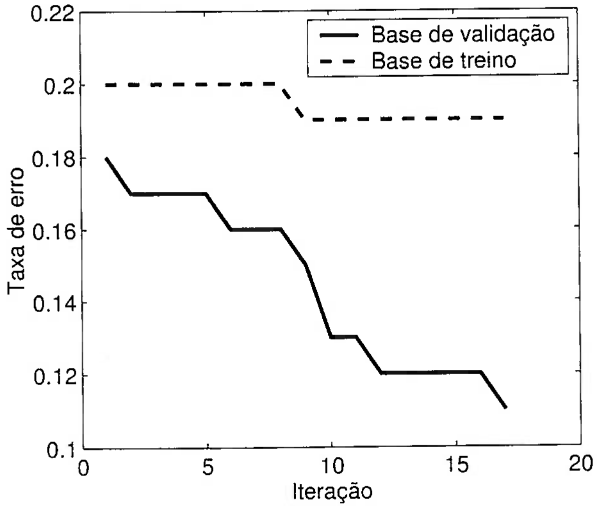


Figura 6: *Overfitting* em um caso real: este gráfico mostra a evolução da busca feita pelo SSS na base Satimage (200 registros rotulados; os demais registros eram não-rotulados). Após algumas iterações as redes encontradas já estavam "viciadas" na base de treino.

O SSS realiza uma busca guiada pelo valor estimado para $\hat{p}_{erro}$, devido a essa característica esse algoritmo está sujeito ao efeito do *overfitting*, ou seja especializar-se demais na base de treino e perder sua capacidade de generalização. Quando ocorre o *overfitting* os classificadores podem ter alto desempenho na base de validação e baixo nas bases de treino. A figure 6 mostra a ocorrência desse fenômeno durante as execuções preliminares do SSS. Note que apesar do erro de classificação estimado com relação à base de treino diminuir a cada iteração, o erro "real" (estimado sobre uma base independente destinada a testes) permanece praticamente inalterado.

Nos primeiros testes, havia a suposição que a grande massa de dados não-rotulados seria suficiente para evitar o *overfitting*, mas sua ocorrência foi observada de maneira consistente. Para contornar esse problema, com o mínimo de impacto no tempo de processamento, a técnica de *holdout* (DEVROYE; GYöRFI; LUGOSI, 1996) foi implementada. Essa técnica consiste em sortear um porção dos dados rotulados disponíveis, separá-los em uma base de validação e utilizá-la para estimar $\hat{p}_{erro}$. Foi utilizado um terço dos registros disponíveis rotulados na base de treino a cada teste. Com esse procedimento, bastante simples, o *overfitting* foi reduzido. O *holdout* para separar a base de validação da base de treino foi utilizado em todos os testes com o SSS. A figura 7 sintetiza a repartição dos dados.

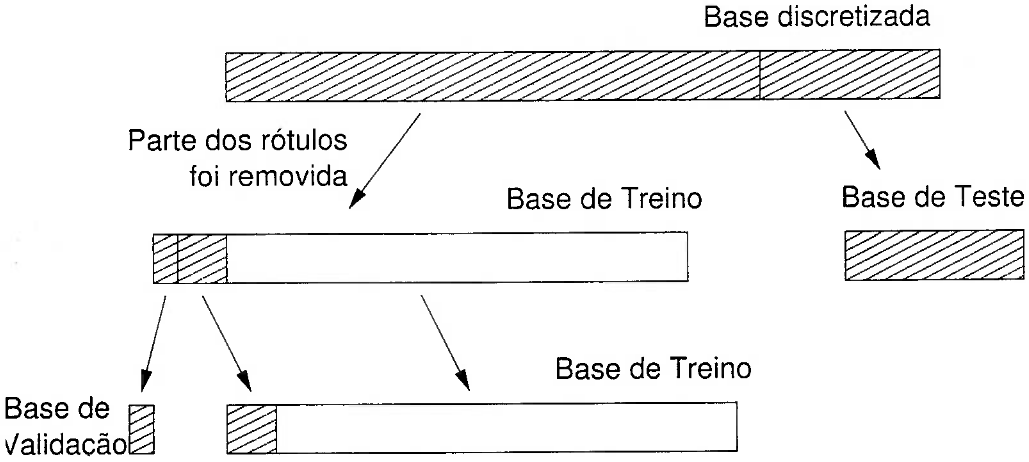


Figura 7: Repartição das bases de dados; as caixas hachuradas representam as bases com dados rotulados.

4.3 Parâmetros para os testes

Alguns parâmetros devem ser configurados antes da realização dos testes. Nesta subseção, são descritos como foi feita a escolha dos valores dos parâmetros.

- CBL — definição da ordenação

  O algoritmo CBL-EM necessita da ordenação completa dos atributos. Mesmo com uma ordenação incorreta (ou seja, inconsistente com a estrutura correta) é possível recuperar boa parte das relações de dependência entre os atributos, mas a rede resultante terá mais parâmetros. Esse excesso de parâmetros, como conseqüência do erro de ordenação, tem impacto negativo no desempenho do classificador. A solução escolhida foi a adoção do seguinte procedimento: sortear uma determinada quantidade de ordenações e escolher a rede com maior taxa de acerto estimada sobre uma base de validação, que no caso do CBL-EM, corresponde a própria base de dados rotulados de treino. Para cada base de dados, 200 ordenações foram testadas.

Apesar do número de ordenações máximo teórico crescer segundo $n!$, o número de ordenações reais é muito menor pois muitas delas são intercambiáveis. Por exemplo, para estruturas como o *Naive Bayes* todas as ordenações em que a classe aparece antes dos demais atributos é válida. A decisão de tomar um número fixo de ordenações para os testes mostrou-se adequado ao propósito deste trabalho, pois sorteando diversas ordenações foi possível descartar os piores casos com facilidade.

- SSS — número de iterações

  O número de iterações é relevante para que o algoritmo SSS alcance um regime permanente e as redes sejam geradas de acordo com a distribuição planejada. A definição do limiar a partir do qual o regime foi alcançado é um problema típico de algoritmos da família MCMC. Porém, quanto mais iterações forem realizadas, melhor. Foi definido que cada teste seria executado o maior número de iterações possível dentro da capacidade computacional disponível. O valor máximo foi 1000 iterações.

4.4 Resultados

Os resultados obtidos com dados rotulados e não-rotulados estão na tabela 3. Na tabela 2 estão os resultados com o NB e o TAN usando apenas dados rotulados e servem como base de comparação. Os resultados envolvendo os algoritmos desenvolvidos e o TAN-EM estão representados na figura 8.

Estes experimentos focam no erro de classificação. Como o primeiro objetivo do trabalho era fazer o melhor uso dos dados não-rotulados, tempo de processamento não foi um dado considerado central.

Base de dados	NB	TAN
Satimage	81,7±0,9	83,5±0,8
Shuttle	82,4±0,3	81,2±0,3
Adult	83,9±0,3	84,7±0,3
Chess	79,8±1,2	87,0±1,0
Cohn-Kanade	72,5±1,4	72,9±1,4
Chen-Huang	71,3±0,8	72,5±0,7

Tabela 2: Resultados obtidos (taxa de acerto; confiança de 95%) — Apenas dados rotulados

Comparando o CBL-EM com os algoritmos NB-EM e TAN-EM tivemos uma melhora do desempenho em quase todas as bases de dados testadas. Isso mostra o potencial de algoritmos baseados em testes de independência. No entanto, esse ganho só foi alcançado quando implementamos a busca no espaço de ordenações.

Como o algoritmo CBL-EM depende muito dos testes de independência há muito o que ganhar tornando esses testes mais confiáveis. Na implementação atual decidimos a independência dos atributos com base na comparação do resultado da Informação Mútua com um limite fixo. Verificamos que na realidade esse limite deve ser dependente da base de dados. Quanto maior a base de dados, melhores os estimadores e mais próximo de zero deve ser o limite (MEILA-PREDOVICIU, 1999).

Os resultados com o CBL-EM modificado para utilizar testes de independência usando G^2 (batizado de CBL-EM χ^2, pois essa é a distribuição da estatística G^2) mostram que a implementação de testes mais robustos resultou em um desempenho comparável ao algoritmo original, exceto para as bases Chess e Chen-Huang nas quais os resultados foram consideravelmente superiores.

Considerando as nuances do aprendizado semi-supervisionado, o algoritmo SSS tem as melhores chances que os demais algoritmos por ter menos restrições ao tipo de estrutura a ser gerada, essa característica se mostra nos resultados experimentais. O SSS teve o maior desempenho em praticamente todos os testes realizados.

Base de dados	NB-EM	TAN-EM	SEM	SSS	CBL-EM	CBL-EM χ^2
Satimage	77,5±0,9	81,0±0,9	77,8±0,9	83,4±0,8	83,5±0,8	82,8±0,8
Shuttle	76,1±0,4	90,5±0,2	73,0±0,3	96,3±0,2	91,8±0,2	87,2±0,2
Adult	73,1±0,4	80,0±0,3	81,3±0,3	85,0±0,3	82,7±0,3	83,2±0,3
Chess	62,1±1,5	71,2±1,4	62,9±1,4	76,0±1,3	81,0±1,2	90,5±0,9
Cohn-Kanade	69,1±1,4	69,3±1,4	67,9±1,4	74,8±1,4	66,2±1,5	69,2±1,4
Chen-Huang	58,5±0,8	62,9±0,8	63,7±0,8	75,0±0,7	65,9±0,8	67,8±0,7

Tabela 3: Resultados obtidos (taxa de acerto; confiança de 95%) — Dados rotulados e não-rotulados (para comparação)

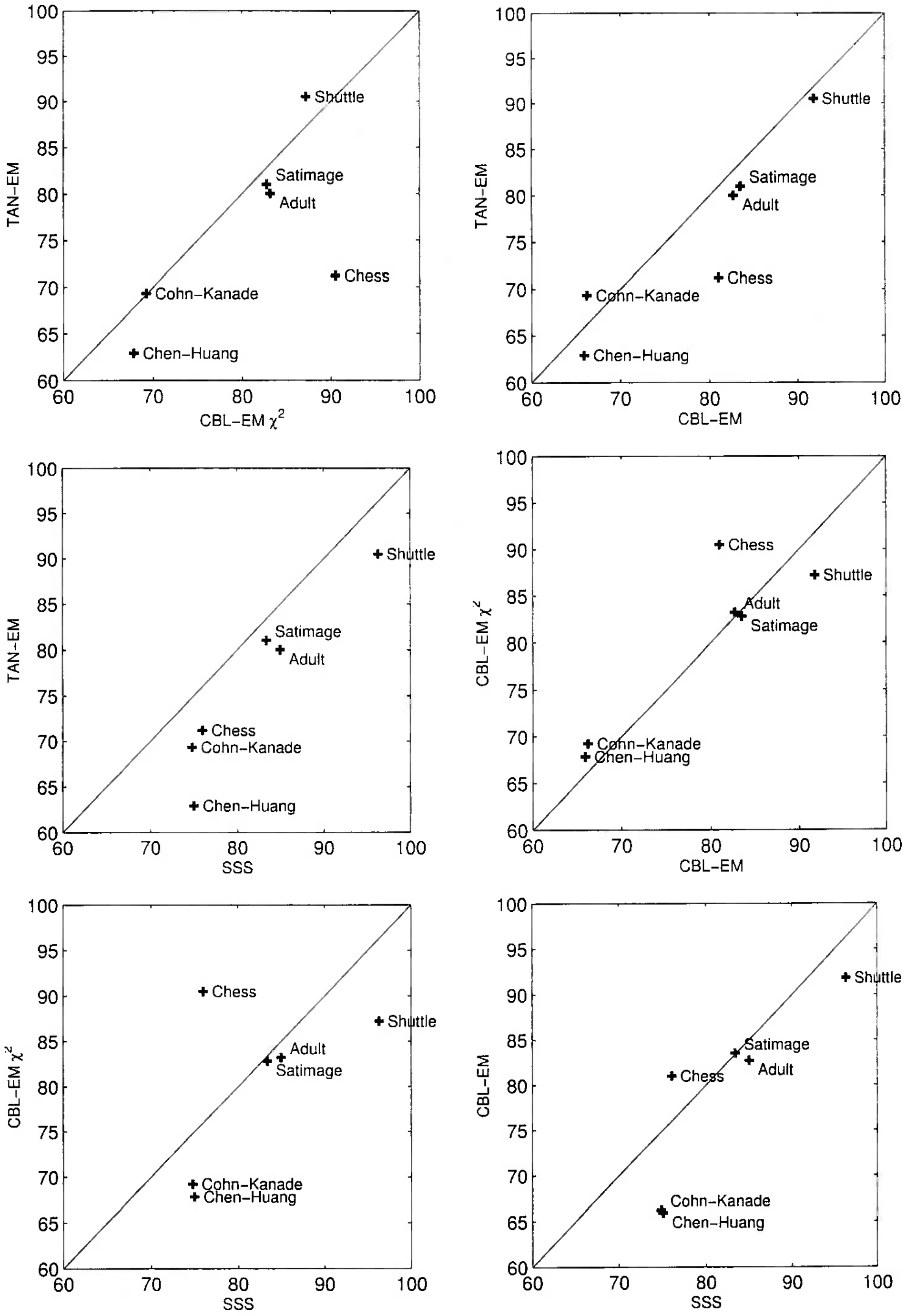


Figura 8: Comparação dos principais algoritmos: TAN-EM, CBL-EM, CBL-EM χ^2 e o SSS.

5 METODOLOGIA PARA O APRENDIZADO SEMI-SUPERVISIONADO DE CLASSIFICADORES

O trabalho apresentado até aqui teve como foco a busca de um classificador, sem consideração com o custo para isso (em termos de tempo de preparação dos testes e tempo de processamento). Não se pode, entretanto, fechar os olhos às restrições a que estão sujeitos os analistas responsáveis por colocar os métodos apresentados neste texto em prática. Neste capítulo é discutido um compromisso entre o ganho de desempenho e o custo computacional para tanto. Espera-se, assim, que a experiência acumulada durante o desenvolvimento desse trabalho, mesmo nos testes que eventualmente não apareceram no texto final, fique disponível a analistas da área.

De maneira resumida, as sugestões são as seguintes:

1. Usar informações *a priori* disponíveis sempre que possível (Seção 5.1).
2. Ter mais de um algoritmo à disposição (Seção 5.2).
3. Obter tantos dados rotulados quanto possível. (Seção 5.3).

5.1 Utilizando informações *a priori* para classificação

Em primeiro lugar, conhecendo-se o domínio do problema, é possível decidir se todos os atributos fornecidos na base de dados são relevantes. Uma verificação nesse sentido foi feita na base de dados Adult. Originalmente essa base possuía 15 atributos. Dois foram descartados. Um dos atributos, *fnlwgt*, foi descartado pelo discretizador por ter baixa correlação com a classe. O segundo atributo removido, *native-country*, embora seja composto por mais de 40 categorias, tem 90% dos registros pertencentes a apenas uma delas. Nos testes que realizamos com o NB e TAN, a remoção desses atributos não diminuiu o desempenho dos classificadores.

Além de pré-filtrar as bases de dados, podemos também restringir alguns relacionamentos entre os atributos. Diferentes algoritmos tem diferentes possibilidades para reduzir o espaço de busca de estruturas. No TAN, por exemplo, não há muito o que fazer senão aceitar a estrutura sugerida. No CBL-EM pode-se realizar uma busca num conjunto de ordenações em que certos casos sejam proibidos. Para evitar que um arco direcionado de $A \rightarrow B$, por exemplo, basta que nas ordenações fornecidas ao algoritmo o atributo A nunca apareça antes do atributo B. O algoritmo SSS também pode ser facilmente modificado para que estruturas que contenham os arcos proibidos não sejam incluídas na lista de estruturas sugeridas.

5.2 Escolha dos algoritmos

A classe de estruturas permitidas durante o aprendizado de redes Bayesianas está no centro da questão do aprendizado semi-supervisionado: quanto mais próximo da estrutura real, maiores serão as chances que uma diminuição no erro de estimação dos parâmetros também reduza o erro de classificação. Nos capítulos anteriores foram apresentados alguns algoritmos para explorar o espaço de estruturas (usando sempre estruturas generativas).

Quanto menos restrições foram colocadas para o aprendizado de estruturas, maior o tempo de processamento necessário, pois o espaço da busca é maior. Analisando os resultados apresentados no capítulo anterior, percebe-se que, em alguns casos ganha-se muito pouco em desempenho de classificação ao passo que o tempo de processamento aumenta demais. Por essa razão, apresenta-se a complexidade dos algoritmos (veja tabela 4 e detalhes no Apêndice A) implementados para permitir uma comparação.

De modo geral o NB-EM não é recomendado para o aprendizado semi-supervisionado pois a premissa de independência entre os atributos, que é a maior restrição que se pode impor à classe de estruturas generativas, raramente se sustenta nos problemas reais. Entretanto existem duas classes de problemas que, segundo a literatura se beneficiam do NB-EM. Há utilizações bem-sucedidas do NB-EM, como a classificação

Tabela 4: Complexidade dos algoritmos implementados. Nesta tabela, i corresponde ao número máximo de iterações do EM; j corresponde ao número máximo de iterações do SEM e do CBL-EM; m ao tamanho da base de dados; n ao número de variáveis da base de dados; $MaxIter$ o número de iterações do SSS.

Algoritmo	Complexidade
NB-EM	$O(imn)$
TAN-EM	$O(imn^2 + in^2k^3)$
SEM	$O(jmn^3)$
SSS	$O(MaxIter\ imn)$
CBL-EM	$O(jmn^3)$

de textos (NIGAM et al., 2000) que se caracteriza por bases de milhares de atributos e muito esparsas. Espera-se que base de outros domínios com as mesmas características também se beneficiem. Outro tipo de problema relevante consiste em bases de dados com muito poucos dados rotulados. Ou seja, situações em que os dados rotulados não seriam suficientes para construir classificadores com desempenho melhor que uma escolha aleatória. Existem publicações que reforçam essa afirmação (NIGAM et al., 2000; COZMAN; COHEN, 2002).

Dos métodos onde a construção de estruturas é guiada pela verossimilhança (SEM e TAN-EM), não há dúvidas que o TAN-EM é mais apropriado para classificação, ainda que o SEM tenha mais liberdade para encontrar estruturas arbitrárias. O TAN-EM é um algoritmo notável: primeiro, porque pode ser construído a partir de uma base de dados de maneira muito eficiente; segundo, porque como as estruturas que gera são restritas a um pai por nó, a quantidade de parâmetros é limitada. O TAN-EM combina a grande vantagem do NB-EM (o tempo de processamento) ao mesmo tempo que diminui a sua maior desvantagem (a premissa de independência entre variáveis).

Pelo que foi obtido como resultado experimental, o TAN-EM sempre esteve próximo do desempenho dos algoritmos mais complexos. Mais relevante, no entanto, é que o TAN-EM permite ao analista aprender um pouco sobre a estrutura do problema. Como exemplo, veja o caso do Shuttle, em que o desempenho do TAN-EM indica ser uma base com poucos relacionamentos, o que foi confirmado pelo SSS.

Um tipo de algoritmo onde a busca é guiada pelo erro estimado de classificação é representado pelo SSS. Cada iteração é processada rapidamente pois cada estrutura é derivada diretamente da estrutura definida na iteração anterior. Porém, por pertencer à classe dos algoritmos MCMC, o SSS exige um grande número de iterações até se chegar a estabilidade.

A vantagem clara do SSS é sua garantia de que a melhor estrutura será encontrada se a premissa que o modelo é generativo for correta. Com efeito, para a base Shuttle registra-se um desempenho de 96% e em testes preliminares chegou-se a construir um classificador com desempenho de 99%, o mesmo valor obtido quando todos os dados são rotulados. Para essa base e para ao menos duas outras bases representativas, Cohn-Kanade e Chen-Huang, o desempenho foi muito acima do obtido utilizando-se os outros classificadores.

Caso se tenha indícios que o modelo gerador da base de dados seja generativo, e o TAN-EM pode ajudar nesse sentido, e se disponha de algum tempo para o aprendizado, o SSS é a melhor opção disponível.

Finalmente temos o CBL-EM. Uma possível crítica ao SSS é que uma busca exclusivamente dirigida pelo erro de classificação estaria muito sujeita ao *overfitting*. O CBL-EM sofre menos desse problema pois a cada iteração, uma nova rede é gerada a partir dos teste de independência. Mesmo considerando que uma busca de ordenações foi implementada, o efeito do *overfitting* é menor, pois as ordenações são geradas de maneira aleatória e independente de p_{erro}.

O CBL-EM é indicado para problemas mais complicados, com muitos relacionamentos entre as variáveis e nos quais exista uma grande quantidade de dados para que seja possível realizar os testes de independência de maneira confiável, como ocorre com as bases Adult e Shuttle.

5.3 Valor práticos dos dados rotulados

Um diferencial deste trabalho com relação ao demais revistos no capítulo 2 é que os dados rotulados são uma parte importante do processo de aprendizagem. Essa abordagem é uma conseqüência da constatação que não há melhor estratégia senão realizar um busca no espaço de estruturas e decidir pela melhor usando o erro de classificação.

Sempre em que a busca for adotada, os dados rotulados são utilizados para treinar e validar as estruturas candidatas. Caso a quantidade de dados seja pequena, é quase certa a ocorrência de *overfitting*. Logo, uma outra razão para necessitar de dados rotulados é permitir que as bases de dados para treino e para a validação sejam diferentes utilizando-se *holdout* ou validação cruzada.

Essa abordagem, que depende de uma base de dados rotulados mediana, embora seja mais bem sucedida em problemas genéricos, traz consigo uma desvantagem: podem ocorrer casos em que a melhor estratégia seja simplesmente descartar os dados não-rotulados. Embora radical, essa alternativa deve ser considerada. Nessa situação a melhor alternativa, com base nos resultados experimentais obtidos, seria usar simplesmente o TAN.

6 CONCLUSÃO

Este trabalho comparou diversas abordagens para aprendizado semi-supervisionado de redes Bayesianas discretas. Até o momento trata-se da comparação empírica mais completa disponível na literatura. Houve o cuidado de realizar experimentos com bases de dados de domínios bastante diversos e representativos. Novamente essa característica é única ao nosso trabalho; por exemplo, Nigam et al. (2000) usa apenas bases provenientes de classificação de textos e Baluja (1999) utiliza apenas bases de reconhecimento de imagens.

Algumas conclusões interessantes podem ser tiradas deste trabalho. Primeiro, apesar do SSS e do CBL-EM terem apresentado resultados superiores aos demais, sempre devemos considerar o uso do TAN-EM — isso em razão da sua eficiência computacional (o algoritmo é polinomial e não há busca envolvida). Além disso, os resultados obtidos com TAN-EM estão bastante próximos dos obtidos com os algoritmos propostos.

Nos parece que enquanto não existirem métodos de estimação eficientes que relacionem a estrutura com o resultado de classificação, não há como evitar a busca para obter classificadores mais complexos que o TAN. A pesquisa em classificadores mais complexos que TAN e que ainda sejam computacionalmente viáveis é tópico importante a ser seguido.

Verificamos que em muitos casos é mais interessante descartar os dados não-rotulados e utilizar apenas os rotulados junto com classificadores convencionais como o NB e o TAN. Nesses casos a melhor alternativa ao uso de dados não-rotulados seria a aquisição de mais dados rotulados, se possível, realizando a escolha de dados a rotular de maneira otimizada. O estudo de métodos para escolha de dados a rotular ("aprendizado ativo") seria crucial para melhorar os resultados apresentados nesse trabalho.

Como indicado no Capítulo 1, o trabalho contribuiu com novos algoritmos (SSS e CBL-EM), com comparações entre vários algoritmos (NB, TAN, SEM, SSS, CBL-EM), e com uma metodologia para uso de dados não-rotulados. O trabalho certamente não esgotou o tema de aprendizado semi-supervisionado, deixando para o futuro um considerável número de oportunidades para pesquisa.

REFERÊNCIAS

AGRESTI, A. *Categorical Data Analysis.* New York: John Wiley & Sons, 1990.

BALUJA, S. Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. In: KEARNS, M. J.; SOLLA, S. A.; COHN, D. A. (Ed.). *Advances in Neural Information Processing Systems 11.* The MIT Press, 1999. p. 854–860.

BLAKE, C.; MERZ, C. *UCI Repository of machine learning databases.* 1998. Disponível em: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

BLUM, A.; MITCHELL, T. Combining labeled and unlabeled data with co-training. In: *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers.* 1998. p. 92–100.

CHENG, J.; BELL, D. A.; LIU, W. An algorithm for Bayesian network construction from data. In: *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics.* Ft. Launderdale, Florida, 1997. p. 83–90.

CHICKERING, D. M. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, v. 3, p. 507–554, 2002.

CHOW, C.; LIU, C. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, v. 14(3), p. 462–467, 1968.

CIRELO, M.; COZMAN, F. Aprendizado semi-supervisionado de classificadores Bayesianos utilizando testes de independência. In: *IV Encontro Nacional de Inteligência Artificial.* 2003. v. 1, p. 277. 1 CD-ROM.

COHEN, I.; COZMAN, F.; SEBE, N.; CIRELO, M.; HUANG, T. Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. *Pattern Analysis and Machine Intelligence*, v. 26, p. 1553–1567, 2005.

COHEN, I.; SEBE, N.; COZMAN, F. G.; CIRELO, M. C.; HUANG, T. S. Learning Bayesian network classifiers for facial expression recognition with both labeled and unlabeled data. In: *IEEE Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 2003. p. 595–604.

COOPER, G. F.; HERSKOVITS, E. A. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, v. 9, p. 309–347, 1992.

COVER, T.; THOMAS, J. *Elements of Information Theory.* New York: John Wiley & Sons, 1991.

COZMAN, F.; COHEN, I.; CIRELO, M. Semi-supervised learning of mixture models. In: *Proceedings of Twentieth International Conference on Machine Learning*. AAAI Press, 2003. p. 99–106.

COZMAN, F. G.; COHEN, I. Unlabeled data can degrade classification performance of generative classifiers. In: HALLER, S. M.; SIMMONS, G. (Ed.). *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*. Pensacola Beach, Flórida, 2002. p. 327–331.

DEVROYE, L.; GYöRFI, L.; LUGOSI, G. *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.

DUDA, R.; HART, P.; STORK, D. G. Pattern classification. New Yorque: Wiley-Interscience, 2000.

FRIEDMAN, N. Learning belief networks in the presence of missing values and hidden variables. In: *Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann, 1997. p. 125–133.

FRIEDMAN, N. The Bayesian structural EM algorithm. In: *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1998. p. 129–138.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. *Machine Learning*, v. 29, n. 2-3, p. 131–163, 1997.

GHAHRAMANI, Z.; JORDAN, M. I. Supervised learning from incomplete data via an EM approach. *Advances in Neural Information Processing Systems 7*, Morgan Kaufmann Publishers, Inc., v. 6, p. 120–127, 1994.

GHANI, R. Combining labeled and unlabeled data for text classification with a large number of categories. In: CERCONE, N.; LIN, T. Y.; WU, X. (Ed.). *Proceedings of the First IEEE International Conference on Data Mining*. San Jose, US: IEEE Computer Society, Los Alamitos, US, 2001. p. 597–598.

HECKERMAN, D. *A tutorial on learning with Bayesian networks*. Redmond, Washington: Microsoft Research, 1995. (Technical Report MSR-TR-95-06).

HUTTER, M. Distribution of mutual information. In: DIETTERICH, T. G.; BECKER, S.; GHAHRAMANI, Z. (Ed.). *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002. p. 399–406. Disponível em: <http://www.hutter1.de/ai/xentropy.htm>.

KOHAVI, R.; BECKER, B.; SOMMERFIELD, D. Improving simple Bayes. In: *Ninth European Conference on Machine Learning*. 1997. p. 78–87.

KOHAVI, R.; SOMMERFIELD, D.; DOUGHERTY, J. Data mining using MLC++: A machine learning library in C++. *International Journal on Artificial Intelligence Tools*, v. 6, n. 4, p. 537–566, 1997.

LAM, W.; BACCHUS, F. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, v. 10, p. 269–293, 1994.

LAURITZEN, S. L. *Graphical Models*. Oxford: Clarendon Press, 1996.

MCLACHLAN, G. J. Discriminant analysis and statistical pattern recognition. New Yorque: Wiley, 1992.

MEILA, M.; JORDAN, M. I. Learning with mixtures of trees. *Journal of Machine Learning Research*, v. 1, p. 1–48, 2000.

MEILA-PREDOVICIU, M. *Learning with Mixtures of Trees*. Tese (Doutorado) — Massachusetts Institute of Technology, 1999.

METROPOLIS, N.; ROSENBLUTH, A.; ROSENBLUTH, M.; TELLER, A.; TELLER, E. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, v. 21, p. 1087–1092, 1953.

NIGAM, K.; MCCALLUM, A. K.; THRUN, S.; MITCHELL, T. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, Kluwer Academic Publishers, Boston, v. 39, n. 2/3, p. 103–134, 2000.

PEARL, J. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo, California: Morgan Kaufmann, 1988.

PEARL, J. *Causality: Models, Reasoning, and Inference*. Cambridge: Cambridge University Press, 2000.

SAHAMI, M. *Using Machine Learning to Improve Information Access*. Tese (Doutorado) — Stanford University, 1998.

SEEGER, M. *Learning with labeled and unlabeled data*. Edinburgh: Institute for Adaptive and Neural Computation, University of Edinburgh, 2000.

SHAHSHAHANI, B.; LANDGREBE, D. Effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, v. 32, n. 5, p. 1087–1095, 1994.

SPIRTES, P.; GLYMOUR, C.; SCHEINES, R. *Causation, Prediction, and Search*. 2nd. ed. Cambridge: MIT Press, 2000.

APÊNDICE A – COMPLEXIDADE DOS ALGORITMOS

Neste apêndice, são apresentadas, mais detalhadamente, as relações de complexidade apresentadas na tabela 4.

- NB-EM. Exige apenas um passagem pela base de dados para a estimação dos parâmetros a cada iteração do *EM*. Como para cada registro analisado todos os contadores associados às variáveis devem ser atualizados, temos que a complexidade é dada por:

$$C_{NB-EM} = O(imn)$$

- TAN-EM. Algoritmo 2. Para cada iteração do *EM*, uma nova passagem pela base de dados é feita e todos os contadores são preenchidos. Para que se possa construir a *MWST* (*Maximum Weight Spanning Tree*, ou seja, árvore cuja soma dos pesos dos arcos é máxima) a Informação Mútua de cada par de variáveis deve ser calculada. A complexidade do cálculo da Informação Mútua é proporcional ao produto da cardinalidade das variáveis envolvidas. Assume-se que a cardinalidade das variáveis seja majorada por uma constante k suficientemente grande. No algoritmo TAN utiliza-se testes de Informação Mútua condicional à classe. Por isso, cada teste envolve as variáveis do arco e o nó. Existem $n(n-1)/2$ arcos possíveis em uma árvore (menor n^2). A constante i indica o número máximo de iterações do EM. A complexidade do TAN-EM é dada por:

$$C_{TAN-EM} = \underbrace{O(imn^2)}_{\text{Passo 2.1}} + \underbrace{O(in^2k^3)}_{\text{Passo 2.2}}$$

$$C_{TAN-EM} = O(imn^2 + in^2k^3)$$

- SEM. Algoritmo 3. Esse algoritmo iterativo é composto de uma série de procedimentos de complexidade polinomial. O maior número de estruturas ocorre quando a estrutura sendo testada está completamente conectada. Nesta situação existem $n(n-1)/2$ arcos que podem ser removidos e um mesmo número de arcos que podem ser invertidos. O número de candidatos é portanto menor que n^2. Como notação adicional temos a constante j que corresponde ao número máximo de iterações do SEM.

$$C_{SEM} = \underbrace{O(nm)}_{\text{Passo 2}} + \underbrace{j\left(\underbrace{O(imn)}_{\text{Passo 4.1}} + \underbrace{O(n^2(mn))}_{\text{Passo 4.3}}\right)}_{\text{Passo 4}}$$

$$C_{SEM} = j(O(imn) + O(mn^3))$$

$$C_{SEM} = O(jmn^3))$$

- SSS. Algoritmo 4. No SSS cada iteração é composta pela listagem das modificações possíveis ($O(n^2)$), pela seleção de uma dessas possibilidades ($O(1)$) e pelo teste da rede resultante ($O(imn)$). Do ponto de vista do desempenho computacional, o gargalo do SSS está no número de iterações, $MaxIter$ de ordem n^2. O símbolo m' se refere à base de validação que é muito menor que m. Portanto:

$$C_{SSS} = \underbrace{O(nm)}_{\text{Passo 2}} + \underbrace{O(nm')}_{\text{Passo 3}} + \underbrace{MaxIter\ O(imn)}_{\text{Passo 5}}$$

$$C_{SSS} = O(MaxIter\ imn)$$

- CBL-EM Algoritmo 6. No algoritmo original (CHENG; BELL; LIU, 1997) há uma avaliação indireta da complexidade desse algoritmo. São necessários $O(n^2)$ testes de Informação Mútua (o mesmo número de testes realizados pelo TAN). No entanto, a complexidade dos testes pode ser muito maior.

  Num caso extremo, poderia ser necessário realizar testes de independência condicional envolvendo todas as variáveis (tal situação envolveria em torno de k^n cálculos). Na prática dada a natureza dos problemas reais é raro que haja mais de seis variáveis sendo avaliadas. Não faz sentido realizar testes que envolvam mais variáveis pois certamente não haveria dados suficientes para que os testes fossem significativos estatisticamente. Na prática testes de independência que envolvam um número de parâmetros maior que o número de registros da base de dados são descartados. No entanto, para testes de independência genéricos (condicional a variáveis além da classe) é necessária uma nova passagem pela base de dados. Por essa razão esses testes são de ordem $O(mn)$. Portanto:

$$C_{CBL1} = n^2 \underbrace{O(mn)}_{\text{Passo 4.2}} + \underbrace{O(mn)}_{\text{Passo 5.3}}$$

$$C_{CBL1} = O(mn^3)$$

Como no algoritmo CBL-EM a parte predominate do processamento ocorre durante o aprendizado da rede usando o algoritmo CBL1, a complexidade do CBL-EM é a mesma do CBL1 vezes o número máximo de iterações j.

$$C_{CBL-EM} = O(jmn^3)$$

APÊNDICE B – PUBLICAÇÕES

Este trabalho fez parte de um projeto conduzido pelos pesquisadores Fabio G. Cozman (orientador) e Ira Cohen. Este último é pesquisador do HP Labs de Palo Alto, Estados Unidos, e iniciou sua participação neste projeto em 2002 durante visita ao Laboratório de Tomada de Decisão da Escola Politécnica. As contribuições individuais do presente trabalho se encaixam num projeto mais ambicioso que visa compreender melhor a teoria e prática de aprendizado semi-supervisionado.

Desse esforço conjunto, resultaram alguns artigos relevantes. O braço teórico do trabalho, encabeçado por Fabio G. Cozman, foi descrito no artigo *Semi-Supervised Learning of Mixture Models* (COZMAN; COHEN; CIRELO, 2003). O pesquisador Ira Cohen aplicou os algoritmos de aprendizado semi-supervisionado para o reconhecimento de expressões faciais; o resultado foi o artigo *Learning Bayesian Network Classifiers for Facial Expression Recognition with both Labeled and Unlabeled data* (COHEN et al., 2003). Finalmente, uma descrição mais abrangente de toda a pesquisa realizada foi publicada no *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, em artigo intitulado *Semisupervised learning of Classifiers: Theory, Algorithms, and Their Application to Human-Computer Interation* (COHEN et al., 2005)

Esses artigos estão reproduzidos neste Apêndice com o objetivo de colocar a participação do autor no contexto de todo o trabalho que foi produzido por essa equipe. Embora não seja o autor principal desses artigos, o autor teve participação seja desenvolvendo o algoritmo CBL-EM, conduzindo inúmeros testes com os algoritmos SSS, NB-EM e TAN-EM ou participando das discussões.

Semi-Supervised Learning of Mixture Models

Fabio Gagliardi Cozman FGCOZMAN@USP.BR
Escola Politécnica, University of São Paulo, Av. Prof. Mello Moraes, 2231 - Cidade Universitária 05508900, São Paulo, SP - Brazil

Ira Cohen IRACOHEN@IFP.UIUC.EDU
The Beckman Institute, 405 N. Mathews Ave., Urbana, IL 61801

Marcelo Cesar Cirelo MARCELO.CIRELO@POLI.USP.BR
Escola Politécnica, University of São Paulo

Abstract

This paper analyzes the performance of semi-supervised learning of mixture models. We show that unlabeled data can lead to an *increase* in classification error even in situations where additional labeled data would *decrease* classification error. We present a mathematical analysis of this "degradation" phenomenon and show that it is due to the fact that bias may be adversely affected by unlabeled data. We discuss the impact of these theoretical results to practical situations.

1. Introduction

Semi-supervised learning has received considerable attention in the machine learning literature due to its potential in reducing the need for expensive labeled data (Seeger, 2001). Applications such as text classification, genetic research and machine vision are examples where cheap unlabeled data can be added to a pool of labeled samples. The literature seems to hold a rather optimistic view, where "unclassified observations should certainly not be discarded" (O'Neill, 1978). Perhaps the most representative summary of recent literature comes from McCallum and Nigam (1998), who declare that "by augmenting this small set [of labeled samples] with a large set of unlabeled data and combining the two pools with EM, we can improve our parameter estimates."

Unfortunately, several experiments indicate that unlabeled data are quite often detrimental to the performance of classifiers (Section 3). That is, the more unlabeled data are added to a fixed set of labeled samples, the poorer is the performance of the resulting classifier. We make this statement cautiously, for some readers may find it obvious, while others may find it unbelievable — and some will dismiss it as incorrect. One might argue that numerical errors in EM or similar algorithms are the natural suspects for such performance degradation; thus we want to stress that our results concern performance degradation *even in the absence of numerical instabilities*. Some might object that unlabeled data are *provably* useful (Castelli & Cover, 1996), and so any "degradation" must come from incorrect analysis, while others might argue that unlabeled data could conceivably be deleterious in exceptional situations where modeling assumptions are clearly violated. Yet we note that unlabeled data can lead to performance degradation *even* in situations where labeled data can be useful to classification, so it must be the case that modeling assuptions have a rather different effect on these types of data. We have made extensive tests with semi-supervised learning, only to witness a complex interaction between modeling assumptions and classifier performance. Unlabeled data do require a delicate craftsmanship, and we suspect that most researchers are unaware of such complexities. With this paper we wish to contribute to a better understanding of semi-supervised learning by focusing on maximum-likelihood estimators and generative classifiers.

In Sections 2 and 3 we summarize relevant facts about semi-supervised learning. In Section 4 we show that performance degradation from unlabeled data depends on bias. Our main result is Theorem 1, where we characterize maximum-likelihood semi-supervised learning as a convex combination of supervised and unsupervised learning, and show how to understand performance degradation in semi-supervised learning. We indicate the reasons why we may observe labeled data to improve a classifier while unlabeled data may degrade the same classifier: in short, both labeled and unlabeled data contribute to a reduction of variance, but unlabeled data may lead to an increase in bias when modeling assumptions are incorrect. We present examples il-

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

lustrating such circumstances in semi-supervised learning. We finish by discussing the behavior of some practical classifiers learned with labeled and unlabeled data.

2. Semi-Supervised Learning

The goal is to classify an incoming vector of observables $\mathbf{X}$. Each instantiation $\mathbf{x}$ of $\mathbf{X}$ is a *sample*. There exists a *class variable* C; the values of C are the *classes*. To simplify the discussion, we assume that C is a binary variable with values $\{c', c''\}$. We want to build *classifiers* that receive a sample $\mathbf{x}$ and output either c' or c''. We assume 0-1 loss, thus our objective is to minimize the probability of classification errors. If we knew exactly the joint distribution $F(C,\mathbf{X})$, the optimal rule would be to choose class c' when the probability of $\{C = c'\}$ given $\mathbf{x}$ is larger than $1/2$, and to choose class c'' otherwise (Devroye et al., 1996). This classification rule attains the minimum possible classification error, called the *Bayes error*.

We take that the probabilities of $(C,\mathbf{X})$, or functions of these probabilities, are estimated from data and then "plugged" into the optimal classification rule. We assume that a parametric model $F(C,\mathbf{X}|\theta)$ is adopted. An estimate of θ is denoted by $\hat{\theta}$; we adopt the *maximum likelihood* method for estimation of parameters. If the distribution $F(C,\mathbf{X})$ belongs to the family $F(C,\mathbf{X}|\theta)$, we say the "model is correct"; otherwise we say the "model is incorrect." When the model is correct, the difference between the expected value $E_\theta[\hat{\theta}]$ and θ, $(E_\theta[\hat{\theta}] - \theta)$, is called *estimation bias*. If the estimation bias is zero, the estimator $\hat{\theta}$ is *unbiased*. When the model is incorrect, we use "bias" loosely to mean the difference between $F(C,\mathbf{X})$ and $F(C,\mathbf{X}|\hat{\theta})$. The classification error for θ is denoted by $\mathbf{e}(\theta)$; the difference between $E[\mathbf{e}(\hat{\theta})]$ and the Bayes error is the *classification bias*.

We assume throughout that probability models satisfy the conditions adopted by White (1982); essentially, parameters belong to compact subsets of Euclidean space, measures have measurable Radon-Nikodym densities and are defined on measurable spaces, all functions are twice differentiable and all functions and their derivatives are measurable and dominated by integrable functions. A formal list of assumptions can be found in (Cozman & Cohen, 2003).

In semi-supervised learning, classifiers are built from a combination of N_l labeled and N_u unlabeled samples. We assume that the samples are independent and ordered so that the first N_l samples are labeled. We consider the following scenario. A sample $(c,\mathbf{x})$ is generated from $p(C,\mathbf{X})$. The value c is then either revealed, and the sample is a *labeled* one; or the value c is hidden, and the sample is an *unlabeled* one. The probability that any sample is labeled, denoted by λ, is fixed, known, and independent of the samples. Thus the same underlying distribution $p(C,\mathbf{X})$ models both labeled and unlabeled data; we do not consider the possibility that labeled and unlabeled samples have different generating mechanisms.

The likelihood of a labeled sample $(c,\mathbf{x})$ is $\lambda p(c,\mathbf{x}|\theta)$; the likelihood of an unlabeled sample $\mathbf{x}$ is $(1-\lambda)p(\mathbf{x}|\theta)$. The density $p(\mathbf{X}|\theta)$ is a mixture model with *mixing factor* $p(c'|\theta)$ (denoted by η):

$$p(\mathbf{X}|\theta) = \eta p(\mathbf{X}|c',\theta) + (1-\eta)p(\mathbf{X}|c'',\theta). \qquad (1)$$

We assume throughout that mixtures (1) are identifiable: distinct values of θ determine distinct distributions (permutations of the mixture components are allowed).

The distribution $p(C,\mathbf{X}|\theta)$ can be decomposed either as $p(C|\mathbf{X},\theta)\,p(\mathbf{X}|\theta)$ or as $p(\mathbf{X}|C,\theta)\,p(C|\theta)$. A parametric model where both $p(\mathbf{X}|C,\theta)$ and $p(C|\theta)$ depend explicitly on θ is referred to as a *generative model*. A strategy that departs from the generative scheme is to focus only on $p(C|\mathbf{X},\theta)$ and to take the marginal $p(\mathbf{X})$ to be independent of θ. Such a strategy produces a *diagnostic model* (for example, logistic regression (Zhang & Oles, 2000)). In this narrow sense of diagnostic models, maximum likelihood cannot process unlabeled data for *any* given dataset (see Zhang and Oles (2000) for a discussion). In this paper we adopt maximum likelihood estimators and generative models; other strategies can be the object of future work.

3. Do Unlabeled Data Improve or Degrade Classification Performance?

It would perhaps be reasonable to expect an average improvement in classification performance for any increase in the number of samples (labeled or unlabeled): the more data are processed, the smaller the variance of estimates, and the smaller the classification error. Several reports in the literature seem to corroborate this informal reasoning. Investigations in the seventies are quite optimistic (Cooper & Freeman, 1970; Jr., 1973; O'Neill, 1978). More recently, there has been plenty of applied work with semi-supervised learning,[1] with some notable successes. There have also been workshops on semi-supervised learning at NIPS1998, NIPS1999, NIPS2000 and IJCAI2001. These publications and meetings have generally concluded that unlabeled data can be profitably used whenever available.

There have also been important positive theoretical results concerning unlabeled data. Castelli and Cover (1996) and Ratsaby and Venkatesh (1995) use unlabeled samples to es-

[1] Relevant references: (Baluja, 1998; Bruce, 2001; Collins & Singer, 2000; Comité et al., 1999; Goldman & Zhou, 2000; McCallum & Nigam, 1998; Miller & Uyar, 1996; Nigam et al., 2000; Shahshahani & Landgrebe, 1994b).

timate decision regions (by estimating $p(\mathbf{X})$), and labeled samples are used solely to determine the labels of each region (Ratsaby and Venkatesh refer to this procedure as "Algorithm M"). Castelli and Cover basically prove that Algorithm M is asymptotically optimal under various assumptions, and that, asymptotically, labeled data contribute exponentially faster than unlabeled data to the reduction of classification error. These authors make the critical assumption that $p(C,\mathbf{X})$ belongs to the family of models $p(C,\mathbf{X}|\theta)$ (the "model is correct").

However, a more detailed analysis of current empirical results does reveal some puzzling aspects of unlabeled data.[2] We have reviewed descriptions of performance degradation in the literature in (Cozman & Cohen, 2002); here we just mention the relevant references. Four results are particularly interesting: Shahshahani and Landgrebe (1994b) and Baluja (1998) describe degradation in image understanding, while Nigam et al. (2000) report on degradation in text classification and Bruce (2001) describe degradation in Bayesian network classifiers. Shahshahani and Landgrebe speculate that degradation may be due to deviations from modeling assumptions, such as outliers and "samples of unknown classes" — they even suggest that unlabeled samples should be used only when the labeled data alone produce a poor classifier. Nigam et al. (2000) suggest several possible difficulties: numerical problems in the EM algorithm, mismatches between the natural clusters in feature space and the assumed classes.

Intrigued by such results, we have conducted extensive tests with simulated problems, and have observed the same pattern of "degradation." The interested reader can again consult (Cozman & Cohen, 2002). Here we present a different test, now with real data. Figure 1 shows the result of learning a Naive Bayes classifier using different combinations of labeled and unlabeled datasets for the Adult classification problem in the UCI repository (using the training and testing datasets in the repository). We see that adding unlabeled data can improve classification when the labeled data set is small (30 labeled data), but degrade performance as the labeled data set becomes larger.

Both Shahshahani and Landgrebe (1994a) and Nigam (2001) are rather explicit in stating that unlabeled data can degrade performance, but rather vague in explaining how to analyze the phenomenon. There are several possibilities: numerical errors, mismatches between the distribution of labeled and unlabeled data, incorrect modeling assumptions. Are unlabeled samples harmful only because of numerical instabilities? Is performance degradation caused by increases in variance, or bias, or both? Can performance degradation occur in the absence of bias; that is, when modeling assumptions are correct? Do we need specific types of models, or very complex structures, to produce performance degradation?

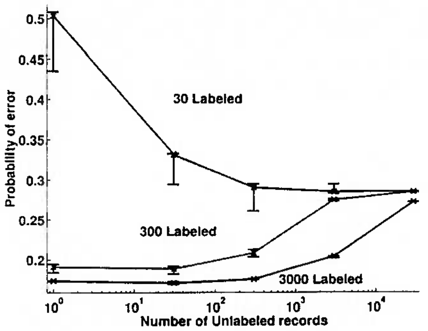


Figure 1. Naive Bayes classifiers generated from the Adult database (bars cover 30 to 70 percentiles).

Our strategy in addressing these questions is to study the asymptotic behavior of exact maximum likelihood estimators under semi-supervised learning. The asymptotic results obtained in the next section allows us to analyze semi-supervised learning without resorting to numerical methods, and to obtain insights that are not clouded by the uncertainties of numerical optimization. We do not deny that numerical problems can happen in practice (see McLachlan and Basford, (1988, Section 3.2) and Corduneanu and Jaakkola (2002)), but we are interested in more fundamental phenomena. The examples in the next section show that performance degradation with unlabeled data would occur even if numerical problems were somehow removed.

4. Asymptotics of Semi-Supervised Learning

In this section we discuss the asymptotic behavior of maximum likelihood estimators in semi-supervised learning. We assume throughout that expectations $E[\log p(C,\mathbf{X})]$, $E[\log p(\mathbf{X})]$, $E[\log p(C,\mathbf{X}|\theta)]$, and $E[\log p(\mathbf{X}|\theta)]$ exist for every θ, and each function attains a maximum at some value of θ in an open neighborhood in the parameter space. Again, we remark that a formal list of assumptions can be found in (Cozman & Cohen, 2003). The assumptions eliminate some important models (such as Cauchy distributions), but they retain the most commonly used distributional models.

To state the relevant results, a Gaussian density with mean μ and variance σ^2 is denoted by $N(\mu,\sigma^2)$, and the following matrices are defined (matrices are formed by running through the indices i and j):
$A_Y(\theta) = E\left[\partial^2 \log p(Y|\theta) / \partial\theta_i\theta_j\right]$,
$B_Y(\theta) = E[(\partial \log p(Y|\theta) / \partial\theta_i)(\partial \log p(Y|\theta) / \partial\theta_j)]$.
We use the following known result (Berk, 1966; Huber,

[2] The workshop at IJCAI2001 witnessed a great deal of discussion on whether unlabeled data are really useful, as communicated to us by George Forman.

1967; White, 1982). Consider a parametric model $F(Y|\theta)$ with the properties discussed in previous sections, and a sequence of maximum likelihood estimates $\hat{\theta}_N$, obtained by maximization of $\sum_{i=1}^{N} \log p(y_i|\theta)$, with an increasing number of independent samples N, all identically distributed according to $F(Y)$. Then $\hat{\theta}_N \to \theta^*$ as $N \to \infty$ for θ in an open neighborhood of θ^*, where θ^* maximizes $E[\log p(Y|\theta)]$. If θ^* is interior to the parameter space, θ^* is a regular point of $A_Y(\theta)$ and $B_Y(\theta^*)$ is non-singular, then $\sqrt{N}\left(\hat{\theta}_N - \theta^*\right) \sim N(0, C(\theta^*))$, where $C_Y(\theta) = A_Y(\theta)^{-1} B_Y(\theta) A_Y(\theta)^{-1}$. This result does not require the distribution $F(Y)$ to belong to the family $F(Y|\theta)$.

In semi-supervised learning, the samples are realizations of $(C, \mathbf{X})$ with probability λ, and of $\mathbf{X}$ with probability $(1 - \lambda)$. Denote by $\tilde{C}$ a random variable that assumes the same values of C plus the "unlabeled" value u. We have $p(\tilde{C} \neq u) = \lambda$. The actually observed samples are realizations of $(\tilde{C}, \mathbf{X})$, and we obtain $\tilde{p}(\tilde{C} = c, \mathbf{X})$ equal to

$$(\lambda p(C = c, \mathbf{X}))^{I_{\{\tilde{C} \neq u\}}(c)} \left((1 - \lambda) p(\mathbf{X})\right)^{I_{\{\tilde{C} = u\}}(c)},$$

where $p(\mathbf{X})$ is a mixture density obtained from $p(C, \mathbf{X})$ (Expression (1)) and $I_\phi(Z)$ is the indicator function (1 if $Z \in \phi$; 0 otherwise). Accordingly, the parametric model adopted for $(\tilde{C}, \mathbf{X})$ is $\tilde{p}(\tilde{C} = c, \mathbf{X}|\theta)$, equal to

$$(\lambda p(C = c, \mathbf{X}|\theta))^{I_{\{\tilde{C} \neq u\}}(c)} \left((1 - \lambda) p(\mathbf{X}|\theta)\right)^{I_{\{\tilde{C} = u\}}(c)}.$$

Using these definitions, we obtain our main result:

Theorem 1 *Consider supervised learning where samples are randomly labeled with probability λ. Adopting previous assumptions, the value of θ^* (the limiting value of maximum likelihood estimates) is:*

$$\arg\max_{\theta} \lambda E[\log p(C, \mathbf{X}|\theta)] + (1 - \lambda) E[\log p(\mathbf{X}|\theta)], \quad (2)$$

where the expectations are with respect to $p(C, \mathbf{X})$. □

Proof. The value θ^* maximizes $E[\log \tilde{p}(\tilde{C}, \mathbf{X}|\theta)]$ (expectation with respect to $\tilde{p}(\tilde{C}, \mathbf{X})$), and $E[\log p(\tilde{C}, \mathbf{X}|\theta)]$ is equal to $E[I_{\{\tilde{C} \neq u\}}(\tilde{C}) (\log \lambda + \log p(C, \mathbf{X}|\theta)) + I_{\{\tilde{C} = u\}}(\tilde{C}) (\log(1 - \lambda) + \log p(\mathbf{X}|\theta))]$; thus the expected value is equal to $\lambda \log \lambda + (1 - \lambda) \log(1 - \lambda) + E[I_{\{\tilde{C} \neq u\}}(\tilde{C}) \log p(C, \mathbf{X}|\theta)] + E[I_{\{\tilde{C} = u\}}(\tilde{C}) \log p(\mathbf{X}|\theta)]$. The first two terms of this expression are irrelevant to maximization with respect to θ. The last two terms are equal to $\lambda E[\log p(C, \mathbf{X}|\theta) \,|\tilde{C} \neq u] + (1 - \lambda) E[\log p(\mathbf{X}|\theta) \,|\tilde{C} = u]$. As we have $\tilde{p}(\tilde{C}, \mathbf{X}|\tilde{C} \neq u) = p(C, \mathbf{X})$ and $\tilde{p}(\mathbf{X}|\tilde{C} = u) = p(\mathbf{X})$ the last expression is equal to $\lambda E[\log p(C, \mathbf{X}|\theta)] + (1 - \lambda) E[\log p(\mathbf{X}|\theta)]$, where the last two expectations are now with respect to $p(C, \mathbf{X})$. Thus we obtain Expression (2). □

Expression (2) indicates that the objective function in semi-supervised learning can be viewed asymptotically as a "convex" combination of objective functions for supervised learning ($E[\log p(C, \mathbf{X}|\theta)]$) and for unsupervised learning ($E[\log p(\mathbf{X}|\theta)]$). Denote by θ^*_λ the value of θ that maximizes Expression (2) for a given λ; use θ^*_l for θ^*_1 and θ^*_u for θ^*_0.[3] We note that, with a few additional assumptions on the modeling densities, Theorem 1 and the implicit function theorem can be used to prove that θ^*_λ is a continuous function of λ. This shows that the "path" followed by the solution is a continuous one, as also assumed by Corduneanu and Jaakkola (2002) in their discussion of numerical methods for semi-supervised learning.

The asymptotic variance in estimating θ under the conditions of Theorem 1 can also be obtained using results in White (1982). The asymptotic variance is ABA, where $A = \left(\lambda A_{(C,\mathbf{X})}(\theta^*) + (1 - \lambda) A_{\mathbf{X}}(\theta^*)\right)^{-1}$ and $B = \left(\lambda B_{(C,\mathbf{X})}(\theta^*) + (1 - \lambda) B_{\mathbf{X}}(\theta^*)\right)$. It can be seen that this asymptotic covariance matrix is positive definite, so asymptotically an increase in N (the number of labeled and unlabeled samples), leads to a reduction in the variance of $\hat{\theta}$. This reduction in variance is true regardless of whether $F(C, \mathbf{X})$ is in $F(C, \mathbf{X}|\theta)$.

Model is correct Suppose first that the family of distributions $F(C, \mathbf{X}|\theta)$ contains the distribution $F(C, \mathbf{X})$; that is, $F(C, \mathbf{X}|\theta_\top) = F(C, \mathbf{X})$ for some $\theta_\top$. When such a condition is satisfied, $\theta^*_l = \theta^*_u = \theta_\top$ given identifiability, and then $\theta^*_\lambda = \theta_\top$ (so maximum likelihood is consistent, bias is zero, and classification error converges to the Bayes error). By following a derivation in Shahshahani and Landgrebe (1994b) for unbiased estimators, we can argue (approximately) that the expected classification error depends on the variance of $\hat{\theta}$. We have $A(\theta^*_\top) = -B(\theta^*_\top)$, thus the asymptotic covariance of the maximum likelihood estimator is governed by the inverse of the Fisher information. Because the Fisher information is a sum of the information from labeled data and the information from unlabeled data (Zhang & Oles, 2000; Cozman & Cohen, 2003), and because the information from unlabeled data is always positive definite, the conclusion is that *unlabeled data must cause a reduction in classification error when the model is correct.* Similar derivations can be found in Ganesalingam and McLachlan (1978) and in Castelli (1994).

Model is incorrect We now study the scenario that is more relevant to our purposes, where the distribution $F(C, \mathbf{X})$ does not belong to the family of distributions

[3] We have to handle a difficulty with $\mathbf{e}(\theta^*_u)$: given only unlabeled data, there is no information to decide the labels for decision regions, and the classification error is 1/2 (Castelli, 1994). To simplify the discussion, we assume that, when $\lambda = 0$, an "oracle" will be available to indicate the labels of the decision regions.

$F(C,\mathbf{X}|\theta)$. In view of Theorem 1, it is perhaps not surprising that unlabeled data can have the deleterious effect discussed in Section 3. Suppose that $\theta_u^* \neq \theta_l^*$ and that $\mathbf{e}(\theta_u^*) > \mathbf{e}(\theta_l^*)$ (we show how this can happen in a later example). If we observe a large number of labeled samples, the classification error is approximately $\mathbf{e}(\theta_l^*)$. If we then collect more samples, most of which unlabeled, we eventually reach a point where the classification error approaches $\mathbf{e}(\theta_u^*)$. So, the net result is that we started with classification error close to $\mathbf{e}(\theta_l^*)$, and by adding a great number of unlabeled samples, classification performance degraded towards $\mathbf{e}(\theta_u^*)$. The basic fact here is that (estimation and classification) biases are directly affected by λ. Hence, a necessary condition for this kind of performance degradation is that $\mathbf{e}(\theta_u^*) \neq \mathbf{e}(\theta_l^*)$; a sufficient condition is that $\mathbf{e}(\theta_u^*) > \mathbf{e}(\theta_l^*)$. If $\mathbf{e}(\theta_l^*)$ is smaller than $\mathbf{e}(\theta_u^*)$, then a labeled dataset can be dwarfed by a much larger unlabeled dataset — the classification error using the whole dataset can be larger than the classification error using only labeled data.

A summary 1) Labeled and unlabeled data contribute to a reduction in variance in semi-supervised learning under maximum likelihood estimation. 2) When the model is correct, the maximum likelihood estimator is unbiased and both labeled and unlabeled data reduce classification error by reducing variance. 3) When the model is incorrect, there may be different asymptotic estimation/classification biases for different values of λ; asymptotic classification error may also be different for different values of λ — an increase in the number of unlabeled samples may lead to a larger estimation bias and a larger classification error.

An example: performance degradation with Gaussian data The previous discussion alluded to the possibility that $\mathbf{e}(\theta_u^*) > \mathbf{e}(\theta_l^*)$ when the model is incorrect. To understand how such a phenomenon can occur, consider an example of obvious practical significance. Consider Gaussian observations (X,Y) taken from two classes c' and c''. We do not know the mixing factor $\eta = p(C = c')$; the data is sampled from a distribution with mixing factor 3/5. We know that X and Y are Gaussian variables: the mean of (X,Y) is $(0,3/2)$ conditional on $\{C = c'\}$, and $(3/2,0)$ conditional on $\{C = c''\}$; variances for X and for Y conditional on C are equal to 1. We believe that X and Y are independent given C, but actually X and Y are *dependent* conditional on $\{C = c''\}$ — the correlation $\rho = E[(X - E[X])(Y - E[Y])|C = c'']$ is equal to $4/5$ (X and Y are in fact independent conditional on $\{C = c'\}$). If we knew the value of ρ, we would obtain an optimal classification boundary on the plane $X \times Y$ (this optimal classification boundary is quadratic). As we assume that ρ is zero, we are generating a Naive-Bayes classifier that approximates $p(C|X,Y)$.

Under the incorrect assumption that $\rho = 0$, the classification boundary is linear: $y = x + 2\log((1-\hat{\eta})/\hat{\eta})/3$, and consequently it is a decreasing function of $\hat{\eta}$. With labeled data we can easily obtain $\hat{\eta}$ (a sequence of Bernoulli trials); then $\eta_l^* = 3/5$ and the classification boundary is given by $y = x - 0.27031$. Note that the (linear) boundary obtained with labeled data is not the best possible linear boundary. We can in fact find the best possible linear boundary of the form $y = x + \gamma$. The classification error can be written as a function of γ that has positive second derivative; consequently the function has a single minimum that can be found numerically (the minimizing γ is -0.45786). If we consider the set of lines of the form $y = x + \gamma$, we see that the farther we go from the best line, the larger the classification error. Figure 2 shows the linear boundary obtained with labeled data and the best possible linear boundary. The boundary from labeled data is "above" the best linear boundary.

Now consider the computation of η_u^*, the asymptotic estimate with unlabeled data. By Theorem 1, we must obtain:

$$\arg\max_{\eta\in[0,1]} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} ((3/5)N([0,3/2]^T,\mathrm{diag}[1,1])+$$

$$(2/5)N([3/2,0]^T,[\begin{smallmatrix} 1 & 4/5 \\ 4/5 & 1 \end{smallmatrix}]))\times$$

$$\log(\eta N([0,3/2]^T,\mathrm{diag}[1,1])+$$

$$(1-\eta)N([3/2,0]^T,\mathrm{diag}[1,1]))dydx.$$

The second derivative of this double integral is always negative (as can be seen interchanging differentiation with integration), so the function is concave and there is a single maximum. We can search for the zero of the derivative of the double integral with respect to η. We obtain this value numerically, $\eta_u^* = 0.54495$. Using this estimate, the linear boundary from unlabeled data is $y = x - 0.12019$. This line is "above" the linear boundary from labeled data, and, given the previous discussion, leads to a larger classification error than the boundary from unlabeled data. We have: $\mathbf{e}(\gamma) = 0.06975$; $\mathbf{e}(\theta_l^*) = 0.07356$; $\mathbf{e}(\theta_u^*) = 0.08141$. The boundary obtained from unlabeled data is also shown in Figure 2.

This example suggests the following situation. Suppose we collect a large number N_l of labeled samples from $p(C,X)$, with $\eta = 3/5$ and $\rho = 4/5$. The labeled estimates form a sequence of Bernoulli trials with probability $3/5$, so the estimates quickly approach η_l^* (the variance of $\hat{\eta}$ decreases as $6/(25N_l)$). If we add a very large amount of unlabeled data to our data, $\hat{\eta}$ approaches η_u^* and the classification error increases.

By changing the "true" mixing factor and the correlation ρ, we can produce other examples where the best linear boundary is between the "labeled" and the "unlabeled"

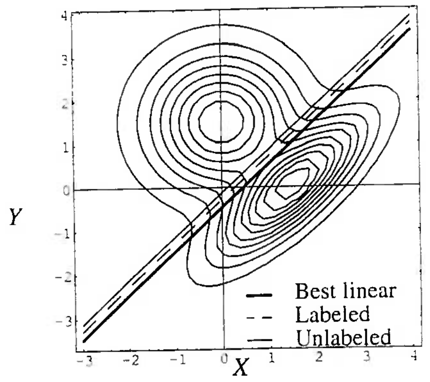


Figure 2. Contour plots of the Gaussian mixture $p(X,Y)$, the best classification boundary of the form $y = x + \gamma$, the linear boundary obtained from labeled data (middle line) and the linear boundary obtained from unlabeled data (upper line).

boundaries, and examples where the "unlabeled" boundary is between the other two. Non-Gaussian examples of degradation can also be easily produced, including examples with univariate models; the interested reader may consult a longer version of this paper (Cozman & Cohen, 2003).

Discussion The obvious consequence of the previous results is that unlabeled data can in fact degrade performance even in simple situations. For degradation to occur, modeling errors must be present — unlabeled data are always beneficial in the absence of modeling errors. The most important fact to understand is that *estimation bias depends on the ratio of labeled to unlabeled data*; this is somewhat surprising as bias is usually taken to be a property of the assumed and the "true" models, and not to be dependent on the data. If the performance obtained with a given set of labeled data is better than the performance with infinitely many unlabeled samples, then at some point the addition of unlabeled data must decrease performance.

5. Learning Classifiers in Practice

To avoid an excessively pessimistic and theoretical tone, we would like to mention some positive practical experience with semi-supervised learning. We have observed that semi-supervised learning of Naive Bayes and TAN classifiers (Friedman et al., 1997), using the EM algorithm to handle unlabeled samples, can be quite successful in classification problems with very large numbers of features and not so large labeled datasets. Text classification and image understanding problems typically fit this pattern; not surprisingly, the best results in the literature are exactly in these applications. A plausible explanation is that such applications contain such a large number of observables that the variance of estimators is very large for the number of available labeled samples, and then the reduction in variance from unlabeled data offsets increases in bias. This agrees with the empirical findings of Shahshahani and Landgrebe (1994b), where unlabeled data are useful as more observables are used in classifiers — while Nigam et al. (2000) suggest that adding observables can worsen the effect of unlabeled data, the opposite should be expected.

However, our experiments indicate that Naive Bayes and TAN are often plagued by performance degradation in relatively common classification problems, for example for the datasets found in the UCI repository. Overall, we have noticed that TAN classifiers have an edge over Naive Bayes classifiers.[4] It is even possible to look at performance degradation as a "signal" that modeling assumptions are incorrect, and to switch from an initial Naive Bayes classifiers to a TAN classifier if performance degradation is observed.

We observe that the most natural way to go beyond Naive Bayes and TAN classifiers is to look for arbitrary Bayesian networks that can represent the relevant distributions; we have had significant success in this direction. Given the many possible approaches to Bayesian network learning, we just mention an interesting approach that has produced excellent results.

We have developed an stochastic structure search algorithm (named SSS) that essentially performs Metropolis-Hastings runs in the space of Bayesian networks; we have observed that this method, while demanding huge computational effort, can improve on TAN classifiers (Cohen et al., 2003). To illustrate these statements, take the Shuttle dataset from the UCI repository. With 43500 labeled samples, a Naive Bayes classifier has classification error of 0.07% (on independent test set with 14500 labeled samples). With 100 labeled samples, a Naive Bayes classifier has classification error of 18%; by adding 43400 unlabeled samples, the resulting Naive Bayes classifier has error of about 30%! A TAN classifier with 100 labeled and 43400 unlabeled samples leads to classification error of about 19%. The SSS algorithm does much better, leading to classification error of only 3.7%. Interestingly, we obtain classification error of just 4.4% by selecting 500 additional labeled samples *randomly* and producing a TAN classifier with EM.

The last observation suggests that active learning should be a profitable strategy in labeled-unlabeled situation McCallum and Nigam (1998). When feasible, active learning can use unlabeled data in a clever and effective manner.

[4] The combination of TAN with EM to handle unlabeled data is described in Meila (1999).

We close by warning the reader that only a careful analysis of unlabeled data can lead to better learning methods. As an example, we can use the insights in this paper to analyze the class of estimators proposed by Nigam et al. (2000). They build Naive Bayes classifiers by maximizing a modified log-likelihood of the form $\lambda' L_l(\theta) + (1-\lambda')L_u(\theta)$ (where L_l is the "likelihood" for labeled data and L_u is the "likelihood" for unlabeled data) while searching for the best possible λ'. There is no reason why this procedure would improve performance, but it may work sometimes: In the Gaussian example in Section 4, if the boundary from labeled data and the boundary from unlabeled data are in different sides of the best linear boundary, we can find the best linear boundary by changing λ' — we can improve on *both* supervised and unsupervised learning in such a situation![5] In any case, one cannot expect to find the best possible boundary just by changing λ'; as an example, consider again the Shuttle dataset from the UCI repository, taking 100 labeled and 43400 unlabeled samples. We have observed a *monotonic* increase in classification error of Naive Bayes classifiers as we vary λ' from 0 to 1: no value of λ' can do better than just using the available labeled data! The good results obtained by Nigam et al. (2000) could either be attributed to the fact that Naive Bayes is the "correct model" in text classification, or to the fact that text classification handles a huge number of features (and the comments in the first paragraph of this section apply).

6. Conclusion

In this paper we have derived and studied the asymptotic behavior of semi-supervised learning based on maximum likelihood estimation (Theorem 1). We have also presented a detailed analysis of performance degradation from unlabeled data, and explained this phenomenon in terms of asymptotic bias. In view of these results, overly optimistic statements in the literature must be ammended. Also, procedures such as Algorithm M are perhaps not reasonable in the presence of modeling errors.

Despite these sobering comments, we note that our techniques can lead to better semi-supervised classifiers in a variety of situations, as argued in Section 5.

We have focused on modeling errors, and not on numerical instabilities. Note first that modeling errors *must* be present for performance degradation to occur. One of our contributions is to connect in a very precise way modeling errors to performance degradation. The connection, as we have argued, comes from an understanding of asymptotic bias. We have on purpose not dealt with two types of modeling errors. First, we have avoided the possibility that labeled and unlabeled data are sampled from different distributions (McLachlan, 1992, pages 42-43); second, we have avoided the possibility that more classes are represented in the unlabeled data than in the labeled data, perhaps due to the scarcity of labeled samples (Nigam et al., 2000). We believe that, by constraining ourselves to simpler modeling errors, we have indicated that performance degradation must be prevalent in practice.

Results in this paper can be extended in several directions. It should be interesting to find necessary and sufficient conditions for a model to suffer performance degradation with unlabeled data. Also, the analysis of bias should be much enlarged, with the addition of finite sample results. Another possible avenue is to look for optimal estimators in the presence of modeling errors (Kharin, 1996). Finally, it would be important to investigate performance degradation in other frameworks, such as support vector machines, co-training, or entropy based solutions (Jaakkola et al., 1999). We conjecture that any approach that incorporates unlabeled data, so as to improve performance when the model is correct, may suffer from performance degradation when the model is incorrect (this fact can be seen in the co-training results of Ghani, (2001, Hoovers-255 dataset)). If we could find an universally robust semi-supervised learning method, such a method would indeed be a major accomplishment.

Regardless of the approach that is used, semi-supervised learning is affected by modeling assumptions in rather complex ways. The present paper should be helpful as a first step in understanding unlabeled data and their peculiarities in machine learning.

Acknowledgements

This work has received continued and substantial support from HP Labs. We thank Alex Bronstein and Marsha Duro for proposing the research on labeled-unlabeled data and for many suggestions and comments during the course of the work. We thank Tom Huang for substantial support to this research; Moises Goldszmidt for valuable suggestions; George Forman for telling us about an IJCAI workshop; Kevin Murphy for the free BNT system; Marina Meila and Íon Muslea for comments in a preliminary version. We coded our own Naive Bayes and TAN classifiers using the libraries of the JavaBayes system (available at `http://www.cs.cmu.edu/~javabayes`).

References

Baluja, S. (1998). Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. *Neural Information Processing Systems*.

[5] Some authors have argued that labeled data should be given more weight (Corduneanu & Jaakkola, 2002), but this example shows that there are no guarantees concerning the supposedly superior effect of labeled data.

Berk, R. H. (1966). Limiting behavior of posterior distributions when the model is incorrect. *Annals of Math. Statistics*, 51–58.

Bruce, R. (2001). Semi-supervised learning using prior probabilities and EM. *Int. Joint Conf. on Artificial Intelligence Workshop on Text Learning.*

Castelli, V. (1994). *The relative value of labeled and unlabeled samples in pattern recognition.* PhD dissertation, Stanford University.

Castelli, V., & Cover, T. M. (1995). On the exponential value of labeled samples. *Pattern Recognition Letters*, *16*, 105–111.

Castelli, V., & Cover, T. M. (1996). The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Trans. on Information Theory*, *42*, 2102–2117.

Cohen, I., Sebe, N., Cozman, F. G., Cirelo, M. C., & Huang, T. (2003). Learning Bayesian network classifiers for facial expression recognition using both labeled and unlabeled data. *Conf. on Computer Vision and Pattern Recognition* (to appear).

Collins, M., & Singer, Y. (2000). Unsupervised models for named entity classification. *Int. Conf. on Machine Learning* (pp. 327–334).

Comité, F. D., Denis, F., Gilleron, R., & Letouzey, F. (1999). Positive and unlabeled examples help learning. *Int. Conf. on Algorithmic Learning Theory* (pp. 219–230). Springer-Verlag.

Cooper, D. B., & Freeman, J. H. (1970). On the asymptotic improvement in the outcome of supervised learning provided by additional nonsupervised learning. *IEEE Trans. on Computers*, *C-19*, 1055–1063.

Corduneanu, A., & Jaakkola, T. (2002). Continuations methods for mixing heterogeneous sources. *Conf. on Uncertainty in Artificial Intelligence* (pp. 111–118).

Cozman, F. G., & Cohen, I. (2002). Unlabeled data can degrade classification performance of generative classifiers. *Int. Conf. of the Florida Artificial Intelligence Research Society* (pp. 327–331). Pensacola, Florida.

Cozman, F. G., & Cohen, I. (2003). The Effect of Modeling Errors in Semi-Supervised Learning of Mixture Models, How Unlabeled Data Can Degrade Performance of Generative Classifiers, at `http://www.poli.usp.br/p/fabio.cozman/Publications/Report/lul.ps.gz`.

Devroye, L., Gyorfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition.* New York: Springer Verlag.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, *29*, 131–163.

Ganesalingam, S., & McLachlan, G. J. (1978). The efficiency of a linear discriminant function based on unclassified initial samples. *Biometrika*, *65*.

Meila, M. (1999). *Learning with mixtures of trees.* PhD dissertation, MIT.

Ghani, R. (2001). Combining labeled and unlabeled data for text classification with a large number of categories. *IEEE Int. Conf. on Data Mining.*

Goldman, S., & Zhou, Y. (2000). Enhancing supervised learning with unlabeled data. *Int. Joint Conf. on Machine Learning.*

Huber, P. J. (1967). The behavior of maximum likelihood estimates under nonstandard conditions. In *Fifth Berkeley Symposium in Mathematical Statistics and Probability*, 221–233.

Jaakkola, T. S., Meila, M., & Jebara, T. (1999). Maximum entropy discrimination. *Neural Information Processing Systems 12.*

Hosmer Jr., D. W. (1973). A comparison of iterative maximum likelihood estimates of the parameters of a mixture of two normal distributions under three different types of sample. *Biometrics*, *29*, 761–770.

Kharin, Y. (1996). *Robustness in statistical pattern recognition.* Kluver Academic Publishers.

McCallum, A., & Nigam, K. (1998). Employing EM and pool-based active learning for text classification. *Int. Conf. on Machine Learning* (pp. 359–367).

McLachlan, G. J. (1992). *Discriminant analysis and statistical pattern recognition.* New York: John Wiley and Sons Inc.

McLachlan, G. J., & Basford, K. E. (1988). *Mixture models: inference and applications to clustering.* New York: Marcel Dekker Inc.

Miller, D. J., & Uyar, H. S. (1996). A mixture of experts classifier with learning based on both labelled and unlabelled data. In *Neural Information Processing Systems* 571–577.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*, 103–144.

Nigam, K. P. (2001). *Using unlabeled data to improve text classification* (Technical Report CMU-CS-01-126). School of Computer Science, Carnegie Mellon University, Pennsylvania.

O'Neill, T. J. (1978). Normal discrimination with unclassified observations. *J. of American Statistical Assoc.*, *73*, 821–826.

Ratsaby, J., & Venkatesh, S. S. (1995). Learning from a mixture of labeled and unlabeled examples with parametric side information. *COLT* (pp. 412–417).

Seeger, M. (2001). *Learning with labeled and unlabeled data* (Technical Report). Institute for Adaptive and Neural Computation, University of Edinburgh, Edinburgh, United Kingdom.

Shahshahani, B. M., & Landgrebe, D. A. (1994a). *Classification of multi-spectral data by joint supervised-unsupervised learning*(Technical Report TR-EE 94-1). School of Electrical Engineering, Purdue University, Indiana.

Shahshahani, B. M., & Landgrebe, D. A. (1994b). The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. on Geoscience and Remote Sensing*, *32*, 1087–1095.

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, *50*, 1–25.

Zhang, T., & Oles, F. (2000). A probability analysis on the value of unlabeled data for classification problems. *Int. Joint Conf. on Machine Learning* (pp. 1191–1198).

Learning Bayesian Network Classifiers for Facial Expression Recognition using both Labeled and Unlabeled Data

Ira Cohen[1], Nicu Sebe[2], Fabio G. Cozman[3], Marcelo C. Cirelo[3], Thomas S. Huang[1]

[1]Beckman Institute, University of Illinois at Urbana-Champaign, IL, USA, {iracohen, huang}@ifp.uiuc.edu

[2]Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands, nicu@liacs.nl

[3]Escola Politécnica, Universidade de São Paulo, São Paulo, Brazil {fgcozman, marcelo.cirelo}@usp.br

Abstract

Understanding human emotions is one of the necessary skills for the computer to interact intelligently with human users. The most expressive way humans display emotions is through facial expressions. In this paper, we report on several advances we have made in building a system for classification of facial expressions from continuous video input. We use Bayesian network classifiers for classifying expressions from video. One of the motivating factor in using the Bayesian network classifiers is their ability to handle missing data, both during inference and training. In particular, we are interested in the problem of learning with both labeled and unlabeled data. We show that when using unlabeled data to learn classifiers, using correct modeling assumptions is critical for achieving improved classification performance. Motivated by this, we introduce a classification driven stochastic structure search algorithm for learning the structure of Bayesian network classifiers. We show that with moderate size labeled training sets and large amount of unlabeled data, our method can utilize unlabeled data to improve classification performance. We also provide results using the Naive Bayes (NB) and the Tree-Augmented Naive Bayes (TAN) classifiers, showing that the two can achieve good performance with labeled training sets, but perform poorly when unlabeled data are added to the training set.

1. Introduction

Since the early 1970s, Ekman has performed extensive studies of human facial expressions [10, 11] and found evidence to support universality in facial expressions. These "universal facial expressions" are those representing happiness, sadness, anger, fear, surprise, and disgust. Ekman's work inspired many researchers to analyze facial expressions using image and video processing. By tracking facial features and measuring the amount of facial movement, they attempt to categorize different facial expressions. Recent work on facial expression analysis has used these "basic expressions" or a subset of them (see Pantic and Rothkrantz's [19] detailed review of many of the research done in recent years). All these methods are similar in that they first extract some features from the images or video, then these features are used as inputs into a classification system, and the outcome is one of the preselected emotion categories. They differ mainly in the features extracted and in the classifiers used to distinguish between the different emotions.

We have developed a real time facial expression recognition system. The system uses a model based non-rigid face tracking algorithm to extract motion features that serve as input to a Bayesian network classifier used for recognizing facial expressions [5]. In our system, as with all other past research in facial expression recognition, learning the classifiers was done using labeled data and supervised learning algorithms. One of the challenges facing researchers attempting to design facial expression recognition systems is the relatively small amount of available labeled data. Construction and labeling of a good database of images or videos of facial expressions requires expertise, time, and training of subjects. Only a few such databases are available, such as the Cohn-Kanade database [14]. However, collecting, without labeling, data of humans displaying expressions is not as difficult. Such data is called unlabeled data. It is beneficial to use classifiers that are learnt with a combination of some labeled data and a large amount of unlabeled data. This paper is focused at describing how to learn to classify facial expressions with labeled and unlabeled data, also known as semi-supervised learning.

Bayesian networks, the classifiers used in our system, can be learned with labeled and unlabeled data using maximum likelihood estimation. One of the main questions is whether adding the unlabeled data to the training set improves the classifier's recognition performance on unseen data. In Section 3 we briefly discuss our recent results demonstrating that, counter to statistical intuition, when the assumed model of the classifier does not match the true data generating distribution, classification performance could *degrade* as more and more unlabeled data are added to the training set. Motivated by this, we propose in Section 4 a classification driven stochastic structure search (SSS) algo-

1063-6919/03 $17.00 © 2003 IEEE

rithm for learning the structure of Bayesian network classifiers. We demonstrate the algorithm's performance using commonly used databases from the UCI repository [2]. In Section 5 we perform experiments with our facial expression recognition system using two databases and show the ability to use unlabeled data to enhance the classification performance, even with a small labeled training set.We have concluding remarks in Section 6.

2. Facial Expression Recognition System

We start with a brief description of our real time facial expression recognition system. The system is composed of a face tracking algorithm which outputs a vector of motion features of certain regions of the face. The features are used as inputs to a Bayesian network classifier.

The face tracking we use in our system is based on a system developed by Tao and Huang [22] called the Piecewise Bézier Volume Deformation (PBVD) tracker. This face tracker uses a model-based approach where an explicit 3D wireframe model of the face is constructed. Once the model is constructed and fitted, head motion and local deformations of the facial features such as the eyebrows, eyelids, and mouth can be tracked. The recovered motions are represented in terms of magnitudes of some predefined motion of various facial features. Each feature motion corresponds to a simple deformation on the face, defined in terms of the Bézier volume control parameters. We refer to these motions vectors as Motion-Units (MU's). The MU's used in the face tracker are shown in Figure 1(a). The MU's are used as the basic features for the classification scheme described in the next sections.

Figure 1: The facial motion measurements

2.1. Bayesian network classifiers

We start with a few conventions that are adopted throughout. The goal here is to label an incoming vector of *features* (MUs) $\mathbf{X}$. Each instantiation of $\mathbf{X}$ is a *record*. We assume that there is a *class variable* C; the values of C are the *labels*, one of the facial expressions. The classifier receives a record $\mathbf{x}$ and generates a label $\hat{c}(\mathbf{x})$. An optimal classification rule can be obtained from the exact distribution $p(C, \mathbf{X})$. However, if the distribution is not known, we have to learn it from expert knowledge or data.

For recognizing facial expression using the features extracted from the face tracking system, we consider probabilistic classifiers that represent the a-posteriori probability of the class given the features, $p(C, \mathbf{X})$, using Bayesian networks [20]. A Bayesian network is composed of a directed acyclic graph in which every node is associated with a variable X_i and with a conditional distribution $p(X_i|\Pi_i)$, where Π_i denotes the parents of X_i in the graph. The directed acyclic graph is the *structure*, and the distributions $p(X_i|\Pi_i)$ represent the *parameters* of the network. We say that the assumed structure for a network, S', is *correct* when it is possible to find a distribution, $p(C, \mathbf{X}|S')$, that matches the distribution that generates data; otherwise, the structure is *incorrect*. We use maximum likelihood estimation to learn the parameters of the network. When there are missing data in our training set, we use the EM algorithm [9] to maximize the likelihood.

A Bayesian network having the correct structure and parameters is also optimal for classification because the a-posteriori distribution of the class variable is accurately represented. A Bayesian network classifier is a *generative* classifier when the class variable is an ancestor (e.g., parent) of some or all features. A Bayesian network classifier is *diagnostic*, when the class variable has non of the features as descendants. As we are interested in using unlabeled data in learning the Bayesian network classifier, we restrict ourselves to generative classifiers and exclude structures that are diagnostic, which cannot be trained using maximum likelihood approaches with unlabeled data [23, 21].

Two examples of generative Bayesian network classifiers are the Naive Bayes (NB) classifier, in which the features are assumed independent given the class, and the Tree-Augmented Naive Bayes classifier (TAN). The NB classifier makes the assumption that all features are conditionally independent given the class label. Although this assumption is typically violated in practice, NB have been used successfully in many classification applications. One of the reasons for the NB success is attributed to the small number of parameters needed to be learnt.

In the structure of the TAN classifier, the class variable is the parent of all the features and each feature has at most one other feature as a parent, such that the resultant graph of the features forms a tree. Using the algorithm presented by Friedman et al. [12], the most likely TAN classifier can be estimated efficiently. When unlabeled data are available, estimating the parameters of the Naive Bayes classifier can be done using the EM algorithm. As for learning the TAN classifier, we learn the structure and parameters using the EM-TAN algorithm, derived from [16].

We have previously used both the NB and TAN classifiers to perform facial expression recognition [6, 5] with good success. However, we used only labeled data for classification. With unlabeled data we show in our experiments that the limited expressive power of Naive Bayes and TAN causes the use of unlabeled data to degrade the performance of our recognition system. This statement will become clear as we describe the properties of learning with labeled and unlabeled data in the next section.

3. Learning a classifier from labeled and unlabeled training data

In this section we discuss properties of classifiers learned with labeled and unlabeled data. In particular, we discuss the possibility that unlabeled data *degrade* classification performance.

Early work proved that unlabeled data lead to improved classification performance, *provided that* the modeling assumptions of the classifier are correct [3, 23]. These have advanced an optimistic view of the labeled-unlabeled problem, where unlabeled data can be profitably used whenever available. However, unlabeled data can also lead to significant degradation in classification performance. A few results in the literature illustrate this possibility. Nigam et al [18] use Naive Bayes classifiers and a large number of features, and report that, when modeling assumptions "are not satisfied, EM may actually degrade rather than improve classifier accuracy" and suggest giving a smaller weight to the unlabeled data. Baluja [1] use unlabeled data to help learn how to determine face orientation. He observed that with Naive Bayes classifiers, unlabeled data sometimes degraded the performance, and proceeded to model the dependencies among the features, finding that such models use better the unlabeled data.

We have conducted an investigation on the effect of unlabeled data and showed that unlabeled data can have deleterious effect when the modeling assumptions are incorrect [8]; here we summarize the main points. We have observed that degradation is not just caused by numerical problems, such as local convergence of the EM algorithm; nor is it just caused by differences between the distribution of labeled data and the distribution of unlabeled data; nor is it just caused by outliers. These explanations do not suffice to clarify why is it that labeled records are routinely seen to improve classification, even in the presence of outliers or incorrect clusters of features, while the same modeling problems lead unlabeled data to degrade classification. This degradation occurs because the asymptotic classification performance of a classifier with incorrect structure can be different when this classifier is learned with fully labeled data and when the classifier is learned with labeled and unlabeled data. Moreover, we proved that there is a fundamental lack of robustness of maximum likelihood estimators when trained with labeled and unlabeled data under incorrect modeling assumptions.

Consider Figure 2 which illustrates the differences in classification bias of classifiers learned from labeled and unlabeled data, where the bias is measured from the Bayes error rate. We simulate the asymptotic case of infinite data.[1]

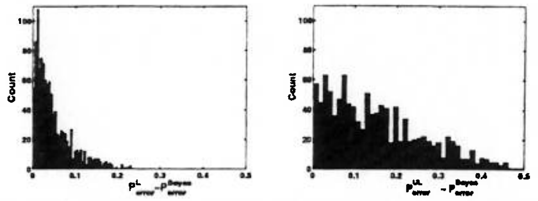


Figure 2: Histogram of classification error bias from the Bayes error rate under incorrect independence assumptions for training with labeled data (left) and training with unlabeled data (right).

We generated 100 different binary classifiers, each with 4 Gaussian distributed features given the class, and not independent of each other. The parameters of each classifier are: the class prior, $\eta = p(C = 0)$, the mean vectors, $\mu_0, \mu_1 \in \Re^4$, and a common covariance matrix $S \in \Re^{4x4}$. The Bayes error rate of the classifiers ranged from $0.7-35\%$, with most being around 10% (the Bayes error was computed analytically using the true parameters of the classifiers).

For each classifier we looked at different combinations of making incorrect independence assumptions, by assuming that features are independent of each other (from one to all features being independent of each other; overall 11 combinations). For example, if we assume that x_1 and x_3 are independent of the rest of the features, the covariance matrix we estimate under this assumption must have the form:

$$\hat{S} = \begin{pmatrix} s_{11} & 0 & 0 & 0 \\ 0 & s_{22} & 0 & s_{24} \\ 0 & 0 & s_{33} & 0 \\ 0 & s_{42} & 0 & s_{44} \end{pmatrix} \qquad (1)$$

thus some elements of the covariance matrix are incorrectly forced to be zero.

For each combination we computed the classification error of two classifiers (trained under the independence assumptions): one simulating training with infinite labeled data and a second trained with infinite unlabeled data. For the labeled data case, since the ML estimation is unbiased, the learned parameters are the true priors, the means, and the elements of the covariance matrix that were not forced to be zero. For unlabeled data, we approximated infinity with $100,000$ training records (which is very large compared to 25, the largest number of parameters estimated in the experiments). We used EM to learn with unlabeled data, with the starting point being the parameter set of the labeled only

[1] Care should be taken when using only unlabeled data in training. As noted by Castelli [3], with unlabeled data it is possible to recover all the parameters of the classifier (under some restrictions, such as identifiability), but a decision on the actual labeling is not possible since we do not know what are the class labels. In the following we assume that we are given this knowledge and therefore are able to perform classification.

classifier, therefore assuring that the difference in the results of the two estimated classifiers do not depend on the starting point of EM. The essential idea of EM when using unlabeled data is to guess pseudo-labels for the unlabeled data based on a previously estimation of the model. These new pseudo-labeled data are then used to create a new classifier, and the process is repeated until a stable solution maximizing the likelihood is found.

Over all, we computed 1100 classification errors for the completely labeled case and 1100 for the unlabeled case. From the errors we generated the classification error bias histograms in Figure 2. The histograms show that the classification bias of the labeled based classifiers tends to be more highly concentrated closer to 0 compared to the unlabeled based classifiers. We also observed that using unlabeled data always resulted in a higher error rate compared to using labeled data. The only exception was when we did not make any incorrect independence assumptions, in which the classifiers trained with unlabeled data achieved the Bayes error rate, as expected. What we understand from these histograms is that when training with labeled data, many classifiers will perform well (although never achieve the optimal Bayes rate). However, classifiers trained with unlabeled data need to be more accurate in their modeling assumptions to achieve good performance and they are a great deal more sensitive to such inaccuracies.

4. Learning the structure of Bayesian network classifiers

The conclusion of the previous section indicates the importance of obtaining the correct structure when using unlabeled data in learning the classifier. If the correct structure is obtained, unlabeled data improve a classifier; otherwise, unlabeled data can actually degrade performance. Somewhat surprisingly, the option of searching for better structures was not proposed by researchers that previously witnessed the performance degradation. Apparently, performance degradation was attributed to unpredictable, stochastic disturbances in modeling assumptions, and not to mistakes in the underlying structure – something that can be detected and fixed.

One attempt to overcome the performance degradation from unlabeled data could be to switch models as soon as degradation is detected. Suppose that we learn a classifier with labeled data only and we observe a degradation in performance when the classifier is learned with labeled and unlabeled data. We can switch to a more complex structure at that point. An interesting idea is to start with a Naive Bayes classifier and, if performance degrades with unlabeled data, switch to a different type of Bayesian network classifier, namely the TAN classifier. If the correct structure can be represented using a TAN structure, this approach will indeed work. However, even the TAN structure is only a small set of all possible structures. Moreover, as the experiments in the next sections show, switching from NB to TAN does not guarantee that the performance degradation will not occur.

A different approach to overcome performance degradation is to use some standard structure learning algorithm, as there are many such algorithms in the Bayesian network literature [12, 7]. A common goal of many existing methods is to find a structure that best fits the joint distribution of all the variables given the data. Because learning is done with finite datasets, most methods penalize very complex structures that might overfit the data, using for example the minimum description length (MDL) score. The difficulty of structure search is the size of the space of possible structures. With finite amounts of data, algorithms that search through the space of structures maximizing the likelihood, can lead to poor classifiers because the a-posteriori probability of the class variable could have a small effect on the score [12]. Therefore, a network with a higher score is not necessarily a better classifier. Friedman et al [12] further suggest changing the scoring function to focus only on the posterior probability of the class variable, but show that it is not computationally feasible.

The drawbacks of likelihood based structure learning algorithms could be magnified when learning with unlabeled data; the posterior probability of the class has a smaller effect during the search, while the marginal of the features would dominate.

4.1. Classification driven stochastic structure search

In this section we propose a method that can effectively search for better structures *with an explicit focus on classification.* We essentially need to find a search strategy that can efficiently search through the space of structures. As we have no simple closed-form expression that relates structure with classification error, it would be difficult to design a gradient descent algorithm or a similar iterative method. Even if we did that, a gradient search algorithm would be likely to find a local minimum because of the size of the search space.

First we define a measure over the space of structures which we want to maximize:

Definition 1 *The* inverse error measure *for structure S is*

$$inv_e(S') = \frac{\frac{1}{p_{S'}(\hat{c}(X) \neq C)}}{\sum_S \frac{1}{p_S(\hat{c}(X) \neq C)}}, \qquad (2)$$

where the summation is over the space of possible structures and $p_S(\hat{c}(X) \neq C)$ is the probability of error of the best classifier learned with structure S.

We use Metropolis-Hastings sampling [17] to generate samples from the inverse error measure, without having to ever compute it for all possible structures. For constructing the Metropolis-Hastings sampling, we define a neighborhood of a structure as the set of directed acyclic graphs

Dataset	Training records		# Test	NB-L	NB-LUL	TAN-L	TAN-LUL	SSS-LUL
	# labeled	# unlabeled						
TAN artificial	300	30000	50000	83.41%	59.21%	90.89%	91.94%	91.05%
Shuttle	500	43000	14500	82.44%	76.10%	81.19%	90.22%	96.26%
Satimage	600	3835	2000	81.65%	77.45%	83.54%	81.05%	83.35%
Adult	6000	24862	15060	83.86%	73.11%	84.72%	80.00%	85.04%

Table 1: Classification accuracy for Naive Bayes, TAN, and stochastic structure search: Naive Bayes classifier learned with labeled data only (NB-L), Naive Bayes classifier learned with labeled and unlabeled data (NB-LUL), TAN classifier learned with labeled data only (TAN-L), TAN classifier learned with labeled and unlabeled data (TAN-LUL), stochastic structure search with labeled and unlabeled data (SSS-LUL).

to which we can transit in the next step. Transition is done using a predefined set of possible changes to the structure; at each transition a change consists of a single edge addition, removal, or reversal. We define the acceptance probability of a candidate structure, S_{new}, to replace a previous structure, S_t as follows:

$$\min\left(1, \left(\frac{inv_e(S_{new})}{inv_e(S_t)}\right)^{\frac{1}{T}} \frac{q(S_t|S_{new})}{q(S_{new}|S_t)}\right) = \min\left(1, \left(\frac{p^t_{error}}{p^{new}_{error}}\right)^{\frac{1}{T}} \frac{N_t}{N_{new}}\right) \quad (3)$$

where $q(S'|S)$ is the transition probability from S to S', T is a temperature factor, and N_t and N_{new} are the sizes of the neighborhoods of S_t and S_{new} respectively; this choice corresponds to equal probability of transition to each member in the neighborhood of a structure. This further creates a Markov chain which is aperiodic and irreducible, thus satisfying the Markov chain Monte Carlo (MCMC) conditions [15]. We summarize our algorithm in Figure 3.

```
1. Fix the network structure to some initial structure, $S_0$.
2. Estimate the parameters of the structure $S_0$ and compute
   the probability of error $p^0_{error}$.
3. Set $t = 0$.
4. Repeat, until a maximum number of iterations is reached
   $(MaxIter)$
   • Sample a new structure $S_{new}$, from the neighborhood of
     $S_t$ uniformly, with probability $1/N_t$.
   • Learn the parameters of the new structure using maxi-
     mum likelihood estimation. Compute the probability of
     error of the new classifier, $p^{new}_{error}$.
   • Accept $S_{new}$ with probability given in Eq. (3).
   • If $S_{new}$ is accepted, set $S_{t+1} = S_{new}$ and
     $p^{t+1}_{error} = p^{new}_{error}$ and change $T$ according to the
     temperature decrease schedule. Otherwise $S_{t+1} = S_t$.
   • $t = t + 1$.
5. return the structure $S_j$, such that $j = \underset{0 \leq j \leq MaxIter}{argmin} (p^j_{error})$.
```

Figure 3: Stochastic structure search algorithm (SSS)

Roughly speaking, T close to 1 would allow acceptance of more structures with higher probability of error than previous structures. T close to 0 mostly allows acceptance of structures that improve probability of error. A fixed T amounts to changing the distribution being sampled by the MCMC, while a decreasing T is a simulated annealing run, aimed at finding the maximum of the inverse error distribution. The rate of decrease of the temperature determines the rate of convergence. Asymptotically in the number of data, a logarithmic decrease of T will guarantee convergence to a global maximum with probability that tends to one [13].

There are two caveats though; first, the logarithmic cooling schedule is very slow and we do not have infinite number of data, second, we never have access to the true probability of error for each structure - we calculate the classification error from a limited pool of training data (denoted by $\hat{p}^S_{error}$). To avoid the problem of overfitting we can take several approaches. Cross-validation can be performed by splitting the labeled training set to smaller sets. However, this approach can significantly slow down the search, and is suitable only if the labeled training set is moderately large. Instead, we use the multiplicative penalty term derived from structural risk minimization to define a modified error term:

$$(\hat{p}^S_{error})^{mod} = \frac{\hat{p}^S_{error}}{1 - c \cdot \sqrt{\frac{h_S(log(2n/h_S)+1) - log(\eta/4)}{n}}}, \quad (4)$$

where h_S is the Vapnik-Chervonenkis (VC) dimension of the classifier with structure S, n is the number of training records, η and c are between 0 and 1. To approximate the VC dimension, we use $h_S \propto N_S$, with N_S the number of (free) parameters in the Markov blanket of the class variable in the network, assuming that all variables are discrete.

To illustrate the performance of SSS algorithm, we performed experiments with some of the UCI machinel learning datasets and an artificially generated data set (a Bayesian network with TAN structure), using relatively small labeled sets and large unlabeled sets (Table 1). The results using the UCI datasets show, to varying degrees, the ability of SSS to utilize unlabeled data. The most dramatic improvement is seen with the Shuttle dataset. The results with the artificially generated data show that SSS was able to achieve almost the same performance as TAN, which had the advantage of a-priori knowledge of the correct structure. We also see that for both NB and TAN, using unlabeled data can cause performance degradation, therefore the idea of switching between these simple models is not guaranteed to work.

5. Facial Expression Recognition Experiments

We test the algorithms for the facial expression recognition system. We initially consider experiments where all the data is labeled. Then we investigate the effect of using both labeled and unlabeled data.

We use two different databases, one collected by Chen and Huang [4] and the Cohn-Kanade database [14]. The first consists of subjects that were instructed to display facial expressions corresponding to six types of emotions. In the Chen-Huang database there are five subjects. For each subjects there are six video sequences per expression, each sequence starting and ending in the Neutral expression. There are on average 60 frames per expression sequence. The Cohn-Kanade database [14] consists of expression sequences of subjects, starting from a Neutral expression and ending in the peak of the facial expression. There are 104 subjects in the database. Because for some of the subjects, not all of the six facial expressions sequences were available to us, we used a subset of 53 subjects, for which at least four of the sequences were available. For each person there are on average 8 frames for each expression.

We measure the accuracy with respect to the classification result of each frame, where each frame in the video sequence was manually labeled to one of the expressions (including Neutral). This manual labeling can introduce some 'noise' in our classification because the boundary between Neutral and the expression of a sequence is not necessarily optimal, and frames near this boundary might cause confusion between the expression and the Neutral.

5.1. Experimental results with labeled data

We start with a person-independent experiment using all the labeled data. For this test we use the sequences of some subjects as test sequences and the sequences of the remaining subjects as training sequences (we leave out one subject in the Chen-Huang database and 10 subjects for the Cohn-Kanade database). This test is repeated five times, each time leaving different subjects out (leave one out cross validation). Table 2 shows the recognition rate of the test for all classifiers. We see that the Naive Bayes classifier performs poorly. However, a significant improvement for both the TAN and the SSS algorithm is obtained, with SSS being significantly better. It should be noted that with a smaller training set, SSS would not have been able to explore many structure and its performance would have probably be the same or worse than NB and TAN.

5.2. Experiments with labeled and unlabeled data

We consider now both labeled and unlabeled data in a person-independent experiment. We first partition the data to a training set and a test set and randomly choose a portion of the training set and remove the labels. This procedure ensures that the distribution of the labeled and the unlabeled sets are the same.

Table 2: Recognition rates (%) for person-independent test

	NB	TAN	SSS
Chen-Huang Database	71.78	80.31	83.62
Cohn-Kandade Database	77.70	80.40	81.80

We train Naive Bayes and TAN classifiers, using just the labeled part of the training data and the combination of labeled and unlabeled data. We use the SSS algorithm to train a classifier using both labeled and unlabeled data (we do not search for the structure with just the labeled part because it is too small for performing a full structure search).

We see in Table 3 that with NB and TAN, even when using only 200 and 300 labeled samples, adding the unlabeled data degrades the performance of the classifiers, and we would have been better off not using the unlabeled data. Using the SSS algorithm, we are able to improve the results and use the unlabeled data to achieve performance which is higher than using just the labeled data with NB and TAN. The fact that the performance is lower than in the case when all the training set was labeled (see Table 2) implies that the relative value of labeled data is higher than of unlabeled data, as was shown by Castelli [3]. However, had there been more unlabeled data, the performance would be expected to improve.

6. Summary and Discussion

In this work, we presented several advances we made in building a real-time system for classification of facial expressions from continuous video input. The facial expression recognition was done using Bayesian networks classifiers. Collecting labeled data of humans displaying expressions is a difficult task and therefore, we were interested in learning the classifiers with both labeled and unlabeled data. One question we asked was whether adding the unlabeled data to the training set improves the classifier's recognition performance on unseen data. We showed that when incorrect modeling assumptions are used, the unlabeled data could have deleterious effect on the classification performance, while the same unlabeled data, under correct modeling assumptions, are theoretically guaranteed to improve the classification performance. With this result we proposed a classification driven stochastic structure search algorithm for learning the structure of the Bayesian network classifiers. We demonstrated the algorithm's performance using standard databases from the UCI repository. Using moderate size labeled training sets and large amount of unlabeled data, our method was able to utilize unlabeled data to improve classification performance.

We tested our classifiers for facial expression recognition using two databases. We compared the results with two

Table 3: Classification results for facial expression recognition with labeled and unlabeled data.

Dataset	Training records		# Test	NB-L	NB-LUL	TAN-L	TAN-LUL	SSS-LUL
	# labeled	# unlabeled						
Cohn-Kanade	200	2980	1000	72.50%	69.10%	72.90%	69.30%	74.80%
Chen-Huang	300	11982	3555	71.25%	58.54%	72.45%	62.87%	74.99%

other Bayesian network classifiers that have been used in our system: Naive Bayes and TAN networks and we showed that the two can achieve good performance with labeled training sets, but perform poorly when unlabeled data are added to the training set. We showed that by searching for the structure driven by the classification error enables us to use the unlabeled data to improve the classification performance.

In conclusion, our main contributions are as follows. We applied Bayesian network classifiers to the problem of facial expression recognition and we proposed a method that can effectively search for the correct Bayesian network structure focusing on classification. We also stressed the importance of obtaining such a structure when using unlabeled data in learning the classifier. If correct structure is used, the unlabeled data improve the classification, otherwise they can actually degrade the performance. Finally, we integrated the classifiers and the face tracking system to build a real time facial expression recognition system.

Acknowledgments

We thank Alex Bronstein and Marsha Duro at HP-Labs for proposing the research on labeled-unlabeled data and for many suggestions and comments during the course of the work on this topic, as their help was critical to the results described here. We coded our own classifiers in the Java language, using the libraries of the JavaBayes system (freely available at http://www.cs.cmu.edu/~javabayes). This work has been supported in part by the National Science Foundation Grants CDA-96-24396 and IIS-00-85980. The work of Ira Cohen has been supported by a Hewlett Packard fellowship.

References

[1] S. Baluja. Probabilistic modelling for face orientation discrimination: Learning from labeled and unlabeled data. In *NIPS*, 1998.

[2] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[3] V. Castelli. *The relative value of labeled and unlabeled samples in pattern recognition.* PhD thesis, Stanford, 1994.

[4] L.S. Chen. *Joint processing of audio-visual information for the recognition of emotional expressions in human-computer interaction.* PhD thesis, University of Illinois at Urbana-Champaign, Dept. of Electrical Engineering, 2000.

[5] I. Cohen, N. Sebe, L.S Chen, A. Garg, and T.S. Huang. Facial expression recognition from video sequences: Temporal and static modeling. *to appear in CVIU special issue on face recognition*, 2003.

[6] I. Cohen, N. Sebe, A. Garg, and T.S. Huang. Facial expression recognition from video sequences. In *ICME*, 2002.

[7] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:308–347, 1992.

[8] F.G. Cozman and I. Cohen. Unlabeled data can degrade classification performance of generative classifiers. In *FLAIRS*, 2002.

[9] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[10] P. Ekman. Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique. *Psychological Bulletin*, 115(2):268–287, 1994.

[11] P. Ekman and W.V. Friesen. *Facial Action Coding System: Investigator's Guide.* Consulting Psychologists Press, 1978.

[12] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131–163, 1997.

[13] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of operational research*, 13:311–329, 1988.

[14] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis, 2000.

[15] D. Madigan and J. York. Bayesian graphical models for discrete data. *Int. Statistical Review*, 63:215–232, 1995.

[16] M. Meila. *Learning with mixture of trees.* PhD thesis, Massachusetts Institute of Technology, 1999.

[17] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[18] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.

[19] M. Pantic and L.J.M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *PAMI*, 22(12):1424–1445, 2000.

[20] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988.

[21] M. Seeger. Learning with labeled and unlabeled data. Technical report, Edinburgh University, 2001.

[22] H. Tao and T.S. Huang. Connected vibrations: A modal analysis approach to non-rigid motion tracking. In *CVPR*, pages 735–740, 1998.

[23] T. Zhang and F. Oles. A probability analysis on the value of unlabeled data for classification problems. In *ICML*, 2000.

Semisupervised Learning of Classifiers: Theory, Algorithms, and Their Application to Human-Computer Interaction

Ira Cohen, *Member, IEEE*, Fabio G. Cozman, Nicu Sebe, *Member, IEEE*, Marcelo C. Cirelo, and Thomas S. Huang, *Fellow, IEEE*

Abstract—Automatic classification is one of the basic tasks required in any pattern recognition and human computer interaction application. In this paper, we discuss training probabilistic classifiers with labeled and unlabeled data. We provide a new analysis that shows under what conditions unlabeled data can be used in learning to improve classification performance. We also show that, if the conditions are violated, using unlabeled data can be detrimental to classification performance. We discuss the implications of this analysis to a specific type of probabilistic classifiers, Bayesian networks, and propose a new structure learning algorithm that can utilize unlabeled data to improve classification. Finally, we show how the resulting algorithms are successfully employed in two applications related to human-computer interaction and pattern recognition: facial expression recognition and face detection.

Index Terms—Semisupervised learning, generative models, facial expression recognition, face detection, unlabeled data, Bayesian network classifiers.

1 Introduction

Many pattern recognition and human computer interaction applications require the design of classifiers. Classifiers are either designed from expert knowledge or from training data. Training data can be either labeled or unlabeled. In many applications, obtaining fully labeled training sets is a difficult task; labeling is usually done using human expertise, which is expensive, time consuming, and error prone. Obtaining unlabeled data is usually easier since it involves collecting data that are known to belong to one of the classes without having to label it. For example, in facial expression recognition, it is easy to collect videos of people displaying expressions, but it is very tedious and difficult to label the video to the corresponding expressions. Learning with both labeled and unlabeled data is known as *semisupervised learning*.

We start with a general analysis of semisupervised learning for probabilistic classifiers. The goal of the analysis is to show under what conditions unlabeled data can be used to improve the classification performance. We review maximum-likelihood estimation when learning with labeled and unlabeled data. We provide an asymptotic analysis of the value of unlabeled data to show that unlabeled data help in reducing the estimator's variance. We show that, when t assumed probabilistic model matches the data generati distribution, the reduction in variance leads to an improved classification accuracy; a situation that has been analyzed before [1], [2]. However, we show that, when the assumed probabilistic model does not match the true data generating distribution, using unlabeled data can be detrimental to the classification accuracy; a phenomenon that has been generally ignored or misinterpreted by previous researchers who observed it empirically before [1], [3], [4]. This new result emphasizes the importance of using correct modeling assumption when learning with unlabeled data.

We also present, in this paper, an analysis of semisupervised learning for classifiers based on Bayesian networks. While, in many classification problems, simple structures learned with just labeled data have been used successfully (e.g., the Naive-Bayes classifier [5], [6]), such structures fail when trained with both labeled and unlabeled data [7]. Bayesian networks are probabilistic classifiers in which the joint distribution of the features and class variables is specified using a graphical model [8]. The graphical representation has several advantages. Among them are the existence of algorithms for inferring the class label, the ability to intuitively represent fusion of different modalities with the graph structure [9], [10], the ability to perform classification and learning without complete data, and, most importantly, the ability to learn with both labeled and unlabeled data. We discuss possible strategies for choosing a good graphical structure and argue that, in many problems, it is necessary to search for such a structure. Most structure search algorithms are driven by likelihood-based cost functions, which are potentially inadequate for classification [11], [12] due to their attempt to maximize the overall likelihood of the data, while largely ignoring the important quantity for classification; the class a posteriori likelihood. As such, we propose a classification driven stochastic structure search algorithm (SSS), which combines both labeled and

- *I. Cohen is with Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304. E-mail: ira.cohen@hp.com.*
- *F.G. Cozman and M.C. Cirelo are with Escola Politecnica, Universidade de São Paulo, Av. Prof. Mello Moraes 2231-Cidade Universitaria, São Paulo, SP Brazil. E-mail: fgcozman@usp.br, marcelo.cirelo@poli.usp.br.*
- *N. Sebe is with the Faculty of Science, University of Amsterdam, The Netherlands. E-mail: nicu@science.uva.nl.*
- *T.S. Huang is with the Beckman Institute, University of Illinois at Urbana-Champaign, 405 N Mathews Ave., Urbana, IL 61801. E-mail: huang@ifp.uiuc.edu.*

Manuscript received 11 June 2003; revised 30 Dec. 2003; accepted 31 Mar. 2004.
Recommended for acceptance by S.K. Pal.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0128-0603.

unlabeled data to train the classifier and to search for a better performing Bayesian network structure.

Following the new understanding of the limitations imposed by the properties of unlabeled data and equipped with an algorithm to overcome these limitations, we apply the Bayesian network classifiers to two human-computer interaction problems: facial expression recognition and face detection. In both of these applications, obtaining unlabeled training data is relatively easy. However, in both cases, labeling of the data is difficult. For facial expression recognition, accurate labeling requires expert knowledge [13] and, for both applications, labeling of a large amount of data is time consuming for the human labeler. We show that Bayesian network classifiers trained with structure search benefit from semisupervised learning in both of these problems.

The remainder of the paper is organized as follows: In Section 2, we discuss the value of unlabeled data and illustrate the possibility of unlabeled data to degrade the classification performance. In Section 3, we propose possible solutions for Bayesian network classifiers to benefit from unlabeled data by learning the network structure. We introduce a new stochastic structure search algorithm and empirically show its ability to learn with both labeled and unlabeled data using data sets from the UCI machine learning repository [14]. In Section 4.1, we describe the components of our real-time face recognition system, including the real-time face tracking system and the features extracted for classification of facial expressions. We perform experiments of our facial expression recognition system using two databases and show the ability to utilize unlabeled data to enhance the classification performance, even with a small labeled training set. Experiments of Bayesian network classifiers for face detection are given in Section 4.2. We have concluding remarks in Section 5.

2 Learning a Classifier from Labeled and Unlabeled Training Data

The goal is to classify an incoming vector of observables $\mathbf{X}$. Each instantiation of $\mathbf{X}$ is a *sample*. There exists a *class variable* C; the values of C are the *classes*. We want to build *classifiers* that receive a sample $\mathbf{x}$ and output a class. We assume 0-1 loss and, consequently, our objective is to minimize the probability of error (*classification error*). If we knew exactly the joint distribution $p(C, \mathbf{X})$, the optimal rule would be to choose the class value with the maximum a posteriori probability, $p(C|\mathbf{x})$ [15]. This classification rule attains the minimum possible classification error, called the *Bayes error*.

We take that the probabilities of $(C, \mathbf{X})$, or functions of these probabilities, are estimated from data and then "plugged" into the optimal classification rule. We assume that a parametric model $p(C, \mathbf{X}|\theta)$ is adopted. An estimate of θ is denoted by $\hat{\theta}$ and we denote throughout by $\hat{\theta}^*$ the asymptotic value of $\hat{\theta}$. If the distribution $P(C, \mathbf{X})$ belongs to the family $p(C, \mathbf{X}|\theta)$, we say the "model is correct"; otherwise, we say the "model is incorrect." We use "estimation bias" loosely to mean the expected difference between $p(C, \mathbf{X})$ and the estimated $p(C, \mathbf{X}|\hat{\theta})$.

We consider the following scenario: A sample $(c, \mathbf{x})$ is generated from $p(C, \mathbf{X})$. The value c is then either revealed and the sample is a *labeled* one or the value c is hidden and the sample is an *unlabeled* one. The probability that any sample is labeled, denoted by λ, is fixed, known, and independent of the samples.[1] Thus, the same underlying distribution $p(C, \mathbf{X})$ generates both labeled and unlabeled data. It is worth noting that we assume the revealed label is correct and is not corrupted by noise; the case of noisy labels has been studied in various works (such as [17], [18], [19], [20], chapter 2 of [21]). Extending our analysis to the noisy labeled case is beyond the scope of this paper.

Given a set of N_l labeled samples and N_u unlabeled samples, we use maximum-likelihood for estimating $\hat{\theta}$. We consider distributions that decompose $p(C, \mathbf{X}|\theta)$ as $p(\mathbf{X}|C, \theta)$ $p(C|\theta)$, where both $p(\mathbf{X}|C, \theta)$ and $p(C|\theta)$ depend explicitly on θ. This is known as a *generative model*. The log-likelihood function of a generative model for a data set with labeled and unlabeled data is:

$$L(\theta) = L_l(\theta) + L_u(\theta) + \log\left(\lambda^{N_l}(1-\lambda)^{N_u}\right), \qquad (1)$$

where

$$L_u(\theta) = \sum_{j=(N_l+1)}^{N_l+N_u} \log\left[p(\mathbf{x}_j|\theta)\right],$$

and

$$L_l(\theta) = \sum_{i=1}^{N_l} \log\left[\prod_{C}(p(C=c'|\theta)p(\mathbf{x}_i|c', \theta)^{I_{\{C=c'\}}(c_i)}\right]$$

with $I_A(Z)$ the indicator function: 1 if $Z \in A$; 0 otherwise. $L_l(\theta)$ and $L_u(\theta)$ are the likelihoods of the labeled and unlabeled data, respectively.

Statistical intuition suggests that it is reasonable to expect an average improvement in classification performance for any increase in the number of samples (labeled or unlabeled). Indeed, the existing literature presents several empirical and theoretical findings that do indicate positive value in unlabeled data. Cooper and Freeman [22] were optimistic enough about unlabeled data so as to title their work as "On the Asymptotic Improvement in the Outcome of Supervised Learning Provided by Additional Nonsupervised Learning." Other early studies, such as [23], [24], [25], further strengthened the assertion that unlabeled data should be used whenever available. Castelli [26] and Ratsaby and Venkatesh [27] showed that unlabeled data are always asymptotically useful for classification. Krishnan and Nandy [19], [20] extended the results of [25] to provide efficiency results for discriminant and logistic-normal models for samples that are labeled stochastically. It should be noted that such previous theoretical work makes the critical assumption that $p(C, \mathbf{X})$ belongs to the family of models $p(C, \mathbf{X}|\theta)$ (that is, the "model is correct").

There has also been recent applied work on semisupervised learning [1], [3], [4], [5], [28], [29], [30], [31], [32]. Overall, these publications advance an optimistic view of the labeled-unlabeled data problem, where unlabeled data can be profitably used whenever available.

However, a more detailed analysis of current applied results does reveal some puzzling aspects of unlabeled data. Researchers have reported cases where the addition of

[1] This is different from [3] and [16], where λ is a parameter that can be set.

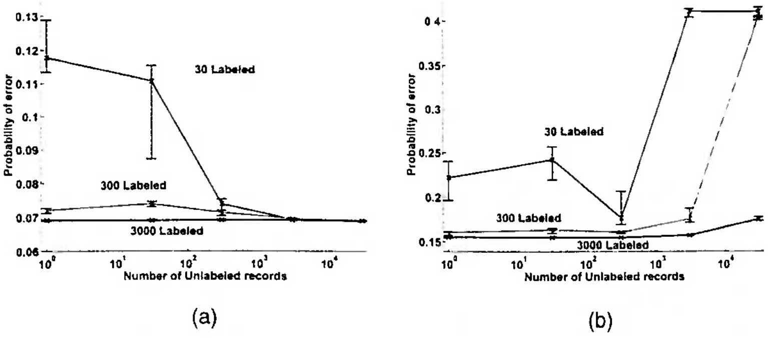


Fig. 1. Naive Bayes classifier from data generated from (a) a Naive Bayes model and (b) a TAN model. Each point summarizes 10 runs of each classifier on testing data; bars cover 30 to 70 percentiles.

unlabeled data degraded the performance of the classifiers when compared to the case in which unlabeled data is not used. These cases were not specific to one type of data, but for different kinds, such as sensory data [1], computer vision [5], and text classification [3], [4].

To explain the phenomenon, we began by performing extensive experiments providing empirical evidence that degradation of performance is directly related to incorrect modeling assumptions [33], [34], [35]. Consider Fig. 1, which shows two typical results. Here, we estimated the parameters of a Naive Bayes classifier with 10 features using the Expectation-Maximization (EM) algorithm [36] with varying numbers of labeled and unlabeled data. Fig. 1 shows classification performance when the underlying model actually has a Naive Bayes structure (Fig. 1a) and when the underlying model is not Naive Bayes (Fig. 1b). The result is clear: When we estimate a Naive Bayes classifier with data generated from a Naive Bayes model, more unlabeled data help; when we estimate a Naive Bayes classifier with data that do not come from a corresponding model, more unlabeled data can degrade performance (even for the case of 30 labeled and 30,000 unlabeled samples!).

To provide a theoretical explanation to the empirical evidence, we derived the asymptotic properties of maximum-likelihood estimators for the labeled-unlabeled case. The analysis, presented in the remainder of this section, provides a unified explanation of the behavior of classifiers for both cases; when the model is correct and when it is not.

2.1 The Value of Unlabeled Data in Maximum-Likelihood Estimation

We base our analysis on the work of White [37] on the properties of maximum-likelihood estimators—properties that hold for the case of model correctness and model incorrectness. In [37], Theorems 3.1, 3.2, and 3.3 showed that, under suitable regularity conditions,[2] maximum-likelihood estimators converge to a parameter set θ^* that minimizes the Kullback-Liebler (KL) distance between the assumed family of distributions, $p(Y|\theta)$, and the true distribution, $p(Y)$. White [37] also shows that the estimator is asymptotically Normal, i.e., $\sqrt{N}(\hat{\theta}_N - \theta^*) \sim \mathcal{N}\ (0, C_Y(\theta))$ as N (the number of samples) goes to infinity. $C_Y(\theta)$ is a covariance matrix equal to $A_Y(\theta)^{-1}B_Y(\theta)A_Y(\theta)^{-1}$, evaluated at θ^*, where $A_Y(\theta)$ and $B_Y(\theta)$ are matrices whose (i,j)th element $(i, j = 1, \ldots, d$, where d is the number of parameters) is given by:

$$\begin{aligned} A_Y(\theta) &= E[\partial^2 \log p(Y|\theta)/\partial\theta_i\theta_j], \\ B_Y(\theta) &= E[(\partial \log p(Y|\theta)/\partial\theta_i)(\partial \log p(Y|\theta)/\partial\theta_j)]. \end{aligned} \tag{2}$$

Using these definitions and general result, we obtain:

Theorem 1. *Consider supervised learning where samples are randomly labeled with probability λ. Assuming identifiability for the marginal distributions of $\mathbf{X}$, then the value of θ^*, the limiting value of maximum-likelihood estimates, is:*

$$\arg\max_{\theta}(\lambda E[\log\, p(C, \mathbf{X}|\theta)] + (1-\lambda)E[\log\, p(\mathbf{X}|\theta)]), \tag{3}$$

where the expectations are with respect to $p(C, \mathbf{X})$. Additionally, $\sqrt{N}(\hat{\theta}_N - \theta^*) \sim \mathcal{N}(0, C_\lambda(\theta))$ as $N \to \infty$, where $C_\lambda(\theta)$ is given by:

$$\begin{aligned} C_\lambda(\theta) &= A_\lambda(\theta)^{-1}B_\lambda(\theta)A_\lambda(\theta)^{-1} \text{ with,} \\ A_\lambda(\theta) &= \left(\lambda A_{(C,\mathbf{X})}(\theta) + (1-\lambda)A_{\mathbf{X}}(\theta)\right) \text{ and} \\ B_\lambda(\theta) &= \left(\lambda B_{(C,\mathbf{X})}(\theta) + (1-\lambda)B_{\mathbf{X}}(\theta)\right), \end{aligned} \tag{4}$$

evaluated at θ^*, where $A_{\mathbf{X}}(\theta)$, $A_{(C,\mathbf{X})}(\theta)$, $B_{\mathbf{X}}(\theta)$, and $B_{(C,\mathbf{X})}(\theta)$ are the A and B defined in (2), with Y replaced by $(C, \mathbf{X})$ or $\mathbf{X}$.

Proof. Denote by $\tilde{C}$ a random variable that assumes the same values of C plus the "unlabeled" value u. We have $p(\tilde{C} \neq u) = \lambda$. The observed samples are realizations of $(\tilde{C}, \mathbf{X})$, so we can write the probability distribution of a sample compactly as follows:

$$\begin{aligned} &\tilde{p}(\tilde{C} = c, \mathbf{X} = \mathbf{x}) = \\ &(\lambda p(C =\text{'}c, \mathbf{X} = \mathbf{x}))^{I_{\{\tilde{C}\neq u\}}(c)}((1-\lambda)p(\mathbf{X} = \mathbf{x}))^{I_{\{\tilde{C}=u\}}(c)}, \end{aligned} \tag{5}$$

where $p(\mathbf{X})$ is a mixture density obtained from $p(C, \mathbf{X})$. Accordingly, the parametric model adopted for $(\tilde{C}, \mathbf{X})$ is:

$$\begin{aligned} &\tilde{p}(\tilde{C} = c, \mathbf{X} = \mathbf{x}|\theta) = \\ &(\lambda p(C = c, \mathbf{X} = \mathbf{x}|\theta))^{I_{\{\tilde{C}\neq u\}}(c)}((1-\lambda)p(\mathbf{X} = \mathbf{x}|\theta))^{I_{\{\tilde{C}=u\}}(c)}. \end{aligned} \tag{6}$$

From White's results stated above, we know that θ^* maximizes $E[\log \tilde{p}(\tilde{C}, \mathbf{X}|\theta)]$ (expectation with respect to $\tilde{p}(\tilde{C}, \mathbf{X})$). We have:

[2] The conditions ensure the existence of the derivatives defined below and the expectations used in Theorem 1.

$$\begin{aligned} E[\log \tilde{p}(\tilde{C},\mathbf{X}|\theta)] &= E\Big[I_{\{\tilde{C}\neq u\}}(\tilde{C})(\log\lambda + \log p(C,\mathbf{X}|\theta)) \\ &\quad + I_{\{\tilde{C}=u\}}(\tilde{C})(\log(1-\lambda) + \log\tilde{p}(\mathbf{X}|\theta))\Big] \\ &= \lambda\log\lambda + (1-\lambda)\log(1-\lambda) \\ &\quad + E\Big[I_{\{\tilde{C}\neq u\}}(\tilde{C})\log p(C,\mathbf{X}|\theta)\Big] \\ &\quad + E\Big[I_{\{\tilde{C}=u\}}(\tilde{C})\log p(\mathbf{X}|\theta)\Big]. \end{aligned}$$

The first two terms of this expression are irrelevant to maximization with respect to θ. The last two terms are equal to

$$\lambda E[\log p(C,\mathbf{X}|\theta)|\tilde{C}\neq u] + (1-\lambda)E[\log p(\mathbf{X}|\theta)|\tilde{C}=u].$$

As we have $\tilde{p}(\tilde{C},\mathbf{X}|\tilde{C}\neq u) = p(C,\mathbf{X})$ and $\tilde{p}(\mathbf{X}|\tilde{C}=u) = p(\mathbf{X})$ ((5)), the last expression is equal to

$$\lambda E[\log p(C,\mathbf{X}|\theta)] + (1-\lambda)E[\log p(\mathbf{X}|\theta)],$$

where the last two expectations are now with respect to $p(C,\mathbf{X})$. Thus, we obtain (3). Expression (4) follows directly from White's theorem and (3), replacing Y by $(C,\mathbf{X})$ and $\mathbf{X}$ where appropriate. □

A few observations can be made from the theorem. First, (3) indicates that semisupervised learning can be viewed asymptotically as a "convex" combination of supervised and unsupervised learning. The objective function for semisupervised learning is a combination of the objective function for supervised learning ($E[\log p(C,\mathbf{X}|\theta)]$) and the objective function for unsupervised learning ($E[\log p(\mathbf{X}|\theta)]$). Second, because the asymptotic covariance matrix is positive definite as $B_Y(\theta)$ is positive definite and $A_Y(\theta)$ is symmetric for any Y, $\theta A(\theta)^{-1}B_Y(\theta)A(\theta)^{-1}\theta^T = w(\theta)B_Y(\theta)w(\theta)^T > 0$, where $w(\theta) = \theta A_Y(\theta)^{-1}$. We see that, asymptotically, an increase in N, the number of labeled and unlabeled samples, will lead to a reduction in the variance of $\hat{\theta}$. Such a guarantee can perhaps be the basis for the optimistic view that unlabeled data should always be used to improve classification accuracy. In the following, we show this view is valid when the model is correct and that it is not always valid when the model is incorrect.

2.2 Model Is Correct

Suppose first that the family of distributions $P(C,\mathbf{X}|\theta)$ contains the distribution $P(C,\mathbf{X})$, that is, $P(C,\mathbf{X}|\theta_\top) = P(C,\mathbf{X})$ for some $\theta_\top$. Under this condition, the maximum-likelihood estimator is consistent, thus $\theta^*_{\lambda=1} = \theta^*_{\lambda=0} = \theta_\top$ given identifiability. Thus, $\theta^*_\lambda = \theta_\top$ for any $0 \leq \lambda \leq 1$.

Shahshahani and Landgrebe [1] suggested using the Taylor expansion of the classification error around $\theta_\top$ to link the decrease in variance associated with unlabeled data to a decrease in classification error. They show that the smaller the variance of the estimator, the smaller the classification error and since the variance of the estimator is smaller as the number of samples increases (labeled or unlabeled), adding the unlabeled data would reduce classification error. A more formal, but less general, argument is presented by Ganesalingam and McLachlan [25] as they compare the relative efficiency of labeled and unlabeled data. Castelli [26] also derives a Taylor expansion of the classification error to study estimation of the mixing factors, $(C = c)$; the derivation is very precise and states all the required assumptions.

2.3 Model Is Incorrect

We now study the more realistic scenario where the distribution $P(C,\mathbf{X})$ does not belong to the family of distributions $P(C,\mathbf{X}|\theta)$. In view of Theorem 1, it is perhaps not surprising that unlabeled data can have the deleterious effect observed occasionally in the literature. Suppose that $\theta^*_u \neq \theta^*_l$ and that $\mathrm{e}(\theta^*_u) > \mathrm{e}(\theta^*_l)$, as in the example in the next section, where $\theta^*_l = \theta^*_{\lambda=1}$ and $\theta^*_u = \theta^*_{\lambda=0}$.[3] If we observe a large number of labeled samples, the classification error is approximately $\mathrm{e}(\theta^*_l)$. If we then collect more samples, most of which are unlabeled, we eventually reach a point where the classification error approaches $\mathrm{e}(\theta^*_u)$. So, the net result is that we started with classification error close to $\mathrm{e}(\theta^*_l)$ and, by adding a large number of unlabeled samples, classification performance degraded. The basic fact here is that estimation and classification bias are affected differently by different values of λ. Hence, a necessary condition for this kind of performance degradation is that $\mathrm{e}(\theta^*_u) \neq \mathrm{e}(\theta^*_l)$; a sufficient condition is that $\mathrm{e}(\theta^*_u) > \mathrm{e}(\theta^*_l)$.

2.3.1 *Example: Bivariate Gaussians with Spurious Correlation*

The previous discussion alluded to the possibility that $\mathrm{e}(\theta^*_u) > \mathrm{e}(\theta^*_l)$ when the model is incorrect. To the skeptical reader, who may still think that this will not occur in practice or that numerical algorithms, such as EM, are the cause of performance degradation, we analytically show how this occurs with an example of obvious practical significance. More examples are provided in [38] and [34].

We will assume that bivariate Gaussian samples (X,Y) are observed. The only modeling error is an ignored dependency between observables. This type of modeling error is quite common in practice and has been studied in the context of supervised learning [39], [40]. It is often argued that ignoring some dependencies can be a positive decision as we may see a reduction in the number of parameters to be estimated and a reduction on the variance of estimates [41].

Example 1. Consider real-valued observations (X,Y) taken from two classes c' and c''. We know that X and Y are Gaussian variables and we know their means and variances given the class C. The mean of (X,Y) is $(0, 3/2)$ conditional on $\{C = c'\}$, and $(3/2, 0)$ conditional on $\{C = c''\}$. Variances for X and for Y conditional on C are equal to 1. We do not know, and have to estimate, the mixing factor $\eta = p(C = c')$. The data is sampled from a distribution with a mixing factor equal to 3/5.

We want to obtain a Naive-Bayes classifier that can approximate $p(C|X,Y)$; Naive-Bayes classifiers are based on the assumption that X and Y are independent given C. Suppose that X and Y are independent conditional on $\{C = c'\}$, but that X and Y are dependent conditional on $\{C = c''\}$. This dependency is manifested by a correlation

$$\rho = E[(X - E[X|C = c''])(Y - E[Y|C = c''])|C = c''] = 4/5.$$

[3] We have to handle a difficulty with $\mathrm{e}(\theta^*_u)$: Given only unlabeled data, there is no information to decide the labels for decision regions and then the classification error is 1/2 [26]. Instead of defining $\mathrm{e}(\theta^*_u)$ as the error for $\lambda = 0$, we could define $\mathrm{e}(\theta^*_u)$ as the error of λ approaching 0.

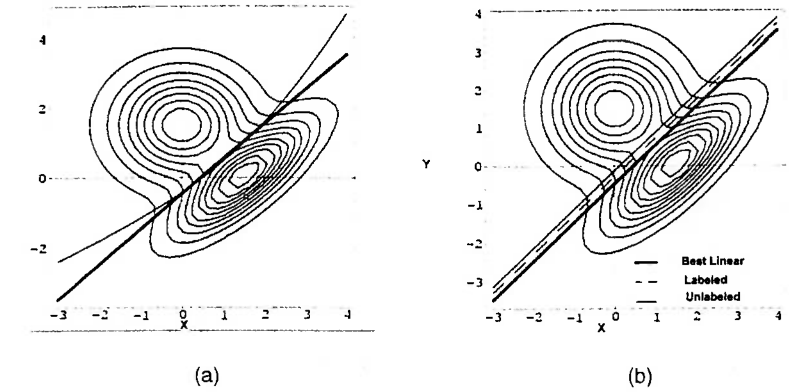


Fig. 2. Graphs for Example 1. (a) Contour plots of the mixture $p(X,Y)$, the optimal classification boundary (quadratic curve), and the best possible classification boundary of the form $y = x + \gamma$. (b) The same contour plots and the best linear boundary (lower line), the linear boundary obtained from labeled data (middle line), and the linear boundary obtained from unlabeled data (upper line); thus, the classification error of the classifier obtained with unlabeled data is larger than that of the classifier obtained with labeled data.

If we knew the value of ρ, we would obtain an optimal classification boundary on the plane $X \times Y$. This optimal classification boundary is shown in Fig. 2 and is defined by the function

$$y = \left(40x - 87 + \sqrt{5265 - 2160x + 576x^2 + 576\log(100/81)}\right)/32.$$

Under the incorrect assumption that $\rho = 0$, the classification boundary is then linear:

$$y = x + 2\log((1-\hat{\eta})/\hat{\eta})/3$$

and, consequently, it is a decreasing function of $\hat{\eta}$. With labeled data, we can easily obtain $\hat{\eta}$ (a sequence of Bernoulli trials); then $\eta_l^* = 3/5$ and the classification boundary is given by $y = x - 0.27031$.

Note that the (linear) boundary obtained with labeled data is not the best possible linear boundary. We can in fact find the best possible linear boundary of the form $y = x + \gamma$. For any γ, the classification error $e(\gamma)$ is

$$\frac{3}{5}\int_{-\infty}^{\infty}\int_{-\infty}^{x+\gamma} N\left(\begin{bmatrix}0\\3/2\end{bmatrix}, \mathrm{diag}[1,1]\right)dydx$$
$$+\frac{2}{5}\int_{-\infty}^{\infty}\int_{x+\gamma}^{\infty} N\left(\begin{bmatrix}3/2\\0\end{bmatrix}, \begin{bmatrix}1 & 4/5\\4/5 & 1\end{bmatrix}\right)dydx.$$

By interchanging differentiation with respect to γ with integration, it is possible to obtain $de(\gamma)/d\gamma$ in closed form. The second derivative $d^2e(\gamma)/d\gamma^2$ is positive when $\gamma \in [-3/2, 3/2]$; consequently, there is a single minimum that can be found by solving $de(\gamma)/d\gamma = 0$. We find the minimizing γ to be

$$(-9 + 2\sqrt{45/4 + \log(400/81)})/4 \approx -0.45786.$$

The line $y = x - 0.45786$ is the best linear boundary for this problem. If we consider the set of lines of the form $y = x + \gamma$, we see that the farther we go from the best line, the larger the classification error. Fig. 2 shows the linear boundary obtained with labeled data and the best possible linear boundary. The boundary from labeled data is "above" the best linear boundary.

Now, consider the computation of η_u^*. Using Theorem 1, the asymptotic estimate with unlabeled data is:

$$\eta_u^* = \arg\max_{\eta\in[0,1]} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \log\Big(\eta N([0,3/2]^T, \mathrm{diag}[1,1])$$
$$+ (1-\eta)N([3/2,0]^T, \mathrm{diag}[1,1])\Big)$$
$$\Big((3/5)N([0,3/2]^T, \mathrm{diag}[1,1])$$
$$+ (2/5)N\left(\begin{bmatrix}3/2\\0\end{bmatrix}, \begin{bmatrix}1 & 4/5\\4/5 & 1\end{bmatrix}\right)\Big)dydx.$$

The second derivative of this double integral is always negative (as can be seen interchanging differentiation with integration), so the function is concave and there is a single maximum. We can search for the zero of the derivative of the double integral with respect to η. We obtain this value numerically, $\eta_u^* \approx 0.54495$. Using this estimate, the linear boundary from unlabeled data is $y = x - 0.12019$. This line is "above" the linear boundary from labeled data and, given the previous discussion, leads to a larger classification error than the boundary from labeled data. We have: $e(\gamma) = 0.06975$, $e(\theta_l^*) = 0.07356$, $e(\theta_u^*) = 0.08141$. The boundary obtained from unlabeled data is also shown in Fig. 2.

This example suggests the following situation. Suppose we collect a large number N_l of labeled samples from $p(C,X)$, with $\eta = 3/5$ and $\rho = 4/5$. The labeled estimates form a sequence of Bernoulli trials with probability 3/5, so the estimates quickly approach η_l^* (the variance of $\hat{\eta}$ decreases as $6/(25N_l)$). If we add a very large amount of unlabeled data to our data, $\hat{\eta}$ approaches η_u^* and the classification error increases.

2.4 Finite Sample Effects

The asymptotic analysis of semisupervised learning suffices to show the fundamental problem that can occur when learning with unlabeled data. The focus on asymptotics is adequate as we want to eliminate phenomena that can vary from data set to data set. If $e(\theta_l^*)$ is smaller than $e(\theta_u^*)$, then a large enough labeled data set can be dwarfed by a much

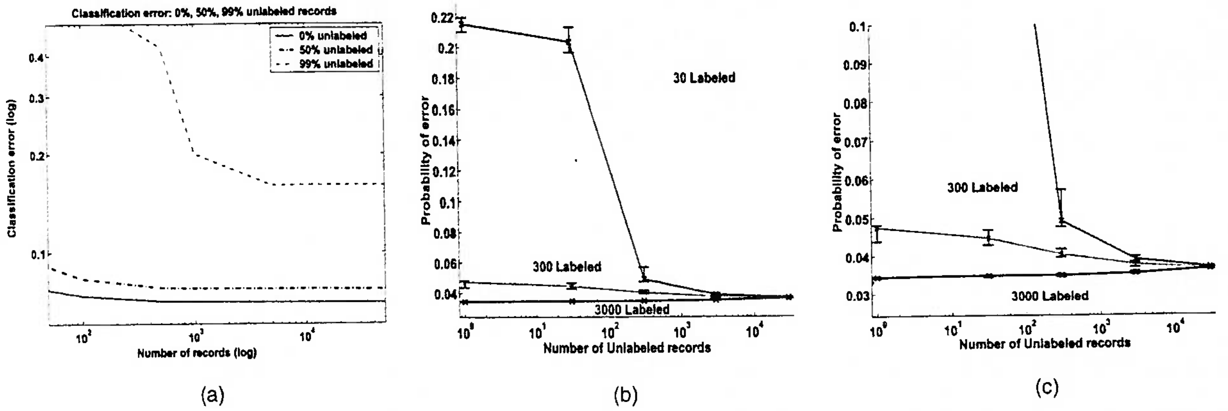


Fig. 3. (a) LU-graphs for the example with two Gaussian observables. Each sample in each graph is the average of 100 trials; classification error was obtained by testing in 10,000 labeled samples drawn from the correct model. (b) Naive Bayes classifiers from data generated from a TAN model (introduced in Section 3) with 49 observables (each variable with two to four values); points in the graphs summarize 10 runs on testing data (bars cover 30 to 70 percentiles). (c) Same graph as (b), enlarged. Note that unlabeled data does lead to a significant improvement in performance when added to 30 or 300 labeled samples. There is performance degradation in the presence of 3,000 labeled samples.

larger unlabeled data set—the classification error using the whole data set can be larger than the classification error using the labeled data only. But, what occurs with finite sample size data sets? We performed extensive experiments with real and artificial data sets of various sizes [7], [34]. Throughout our experiments, we used the EM algorithm to maximize the likelihood ((1)) and we started the EM algorithm with the parameters obtained using labeled data as these starting points can be obtained in closed-form.

To visualize the effect of labeled and unlabeled samples, we suggest that the most profitable strategy is to fix the *percentage* of unlabeled samples (λ) among all training samples. We then plot classification error against the number of training samples. Call such a graph a *LU-graph*.

Example 2. Consider a situation where we have a binary class variable C with values c' and c'' and $p(C = c') = 0.4017$. We also have two real-valued observables X and Y with distributions:

$$P(X|c') = N(2,1), \quad p(X|c'') = N(3,1),$$

$$P(Y|c', x) = N(2,1), \quad p(Y|c'', x) = N(1 + 2x, 1).$$

There is dependency between Y and X conditional on $\{C = c''\}$. Suppose we build a Naive Bayes classifier for this problem. Fig. 3a shows the LU-graphs for 0 percent unlabeled samples, 50 percent unlabeled samples, and 99 percent unlabeled samples, averaging over a large ensemble of classifiers. The asymptotes converge to different values. Suppose then that we started with 50 labeled samples as our training data. Our classification error would be about 7.8 percent, as we can see in the LU-graph for 0 percent unlabeled data. Suppose we added 50 labeled samples, we would obtain a classification error of about 7.2 percent. Now, suppose we added 100 *unlabeled* samples. We would move from the 0 percent LU-graph to the 50 percent LU-graph. Classification error would increase to 8.2 percent! And, if we then added 9,800 unlabeled samples, we would move to the 99 percent LU-graph, with classification error about 16.5 percent—more than twice the error we had with just 50 labeled samples.

It should be noted that, in difficult classification problems, where LU-graphs decrease very slowly, unlabeled data may improve classification performance for certain regions of the LU graphs. Problems with a large number of observables and parameters should require more training data, so we can expect that such problems benefit more consistently from unlabeled data. Figs. 3b and 3c illustrate this possibility for a Naive-Bayes classifier with 49 features. Another possible phenomenon is that the addition of a substantial number of unlabeled samples may reduce variance and decrease classification error, but an additional, much larger, pool of unlabeled data can eventually add enough bias so as to increase classification error. Such a situation is likely to have happened in some of the results reported by Nigam et al. [3], where classification errors go up and down as more unlabeled samples are added.

In summary, semisupervised learning displays an odd failure of robustness: For certain modeling errors, more unlabeled data can degrade classification performance. Estimation bias is the central factor in this phenomenon as the level of bias depends on the ratio of labeled to unlabeled samples. Most existing theoretical results on semisupervised learning are based on the assumption of no modeling error and, consequently, bias has not been an issue so far.

3 Semisupervised Learning for Bayesian Network Classifiers

We now turn our attention to the implication of the previous analysis to Bayesian network classifiers. As stated before, we chose Bayesian network classifiers for several reasons; classification is possible with missing data in general and unlabeled data in particular, the graphical representation is intuitive and can be easily expanded to add different features and modalities, and there are efficient algorithms for inference.

A Bayesian network [8] is composed of a directed acyclic graph in which every node is associated with a variable X_i and with a conditional distribution $p(X_i|\Pi_i)$, where Π_i denotes the parents of X_i in the graph. The joint probability distribution is factored to the collection of conditional probability distributions of each node in the graph as:

$$P(X_1, \dots, X_n) = \prod_{i=1}^{n} p(X_i|\Pi_i).$$

The directed acyclic graph is the *structure* and the distributions $p(X_i|\Pi_i)$ represent the *parameters* of the network. Consider now that data generated by a distribution $p(C, \mathbf{X})$ are collected. We say that the assumed structure for a network, S', is *correct* when it is possible to find a distribution, $p(C, \mathbf{X}|S')$, that matches the data generating distribution $p(C, \mathbf{X})$; otherwise, the structure is *incorrect*.[4,5] Maximum-likelihood estimation is one of the main methods to learn the parameters of the network. When there are missing data in training set, the EM algorithm can be used to maximize the likelihood.

As a direct consequence of the analysis in the previous section, a Bayesian network that has the correct structure and the correct parameters is also optimal for classification because the a posteriori distribution of the class variable is accurately represented. Thus, there is great motivation for obtaining the correct structure when conducting semisupervised learning. Somewhat surprisingly, the option of searching for better structures has not been proposed by researchers who have previously witnessed the performance degradation when learning with unlabeled data. In the following sections, we describe different strategies for learning Bayesian network classifiers with labeled and unlabeled data.

3.1 Switching between Simple Models and Structure Learning

If we observe performance degradation, we may try to find the "correct" structure for our Bayesian network classifier. Alas, learning Bayesian network structure is not a trivial task.

One attempt, perhaps the simplest, to overcome performance degradation from unlabeled data could be to assume a very simple model (such as the Naive Bayes), which is typically not the correct structure, and switch to a more complex model as soon as degradation is detected. One such family of models is the Tree-Augmented Naive-Bayes (TAN) [11]. While such a strategy has no guarantees to find the correct structure, the existence of an efficient algorithm for learning the TAN models, both in the supervised case [11] and in the semisupervised case [7], [42], makes switching to TAN models attractive. However, while both the Naive-Bayes and TAN classifiers have been observed to be successful in the supervised case [41], the same success is not always observed for the semisupervised case (Section 3.3).

When such simple strategies fail, performing unconstrained structure learning is the alternative. There are various approaches for learning the structure of Bayesian networks, using different criteria in an attempt to find the correct structure.

The first class of structure learning methods we consider is the class of independence-based methods, also known as constraint-based or test-based methods. There are several such algorithms [43], [44], [45], all of them can obtain the correct structure if there are fully reliable independence tests available; however, not all of them are appropriate for classification. The Cheng-Bell-Liu algorithms (CBL1 and CBL2) seem particularly well-suited for classification as they strive to keep the number of edges in the Bayesian networks as small as possible and the performance of CBL1 on labeled data only has been reported to surpass the performance of TAN [46]. Because independence-based algorithms like CBL1 do not explicitly optimize a metric, they cannot handle unlabeled data directly through an optimization scheme like EM. To handle unlabeled data, the following strategy was derived (denoted as EM-CBL): Start by learning a Bayesian network with the available labeled data, then use EM to process unlabeled data followed by independence tests with the "probabilistic labels" generated by EM to obtain a new structure. EM is used again in the new structure and the cycle is repeated, until two subsequent networks are identical. It should be noted that such a scheme, however intuitively reasonable, has no convergence guarantees; one test even displayed oscillating behavior.

A second class of structure learning algorithms are score-based methods. At the heart of most score-based methods is the likelihood of the training data, with penalty terms to avoid overfitting. A good comparison of the different methods is found in [47]. Most existing methods cannot, in their present form, handle missing data, in general, and unlabeled data in particular. The structural EM (SEM) algorithm [48] is one attempt to learn structure with missing data. The algorithm attempts to maximize the Bayesian score using an EM-like scheme in the space of structures and parameters; the method performs an always-increasing search in the space of structures, but does not guarantee the attainment of even a local maximum. When learning the structure of a classifier, score-based structure learning approaches have been strongly criticized. The problem is that, with finite amounts of data, the a posteriori probability of the class variable can have a small effect on the score that is dominated by the marginal of the observables, therefore leading to poor classifiers [11], [12]. Friedman et al. [11] showed that TAN surpasses score-based methods for the fully labeled case, *when learning classifiers*. The point is that, with unlabeled data, score-based methods, such as SEM, are likely to go astray even more than has been reported in the supervised case; the marginal of the observables further dominates the likelihood portion of the score as the ratio of unlabeled data increases.

3.2 Classification Driven Stochastic Structure Search (SSS)

Both the score-based and independence-based methods try to find the correct structure of the Bayesian network, but fail to do so because there is not enough data for either reliable independence tests or for a search that yields a good classifier. Consider the following alternative: As we are interested in finding a structure that performs well as a classifier, it would be natural to design algorithms that use classification error as the guide for structure learning. Here, we can further leverage on the properties of semisupervised learning: We know that unlabeled data can indicate incorrect structure through degradation of classification performance and we also know

[4] These definitions follow directly from the definitions of correct and incorrect models described in the previous section.

[5] There is not necessarily a unique correct structure, e.g., if a structure is correct (as defined above), all structures that are from the same Markov equivalent class are also correct since causality is not an issue.

```
Procedure Stochastic structure search (SSS):
• Fix the network structure to some initial structure, $S_0$.
• Estimate the parameters of the structure $S_0$ and compute the probability of error $p^0_{error}$.
• Set $t = 0$.
• Repeat, until a maximum number of iterations is reached ($MaxIter$),
    – Sample a new structure $S_{new}$, from the neighborhood of $S_t$ uniformly, with probability $1/N_t$.
    – Learn the parameters of the new structure using maximum likelihood estimation. Compute the
      probability of error of the new classifier, $p^{new}_{error}$.
    – Accept $S_{new}$ with probability given in Eq.(8).
    – If $S_{new}$ is accepted, set $S_{t+1} = S_{new}$ and $p^{t+1}_{error} = p^{new}_{error}$ and change $T$ according to the
      temperature decrease schedule. Otherwise $S_{t-1} = S_t$.
    – $t = t + 1$.
• Return the structure $S_j$, such that $j = \arg\min_{0\le j\le MaxIter}(p^j_{error})$.
```

Fig. 4. Stochastic structure search algorithm.

that classification performance improves with the correct structure. Thus, a structure with higher classification accuracy over another indicates an improvement toward finding the optimal classifier.

To learn the structure using classification error, we must adopt a strategy for searching through the space of all structures in an efficient manner while avoiding local maxima. In this section, we propose a method that can effectively search for better structures *with an explicit focus on classification*. We essentially need to find a search strategy that can efficiently search through the space of structures. As we have no simple closed-form expression that relates structure with classification error, it would be difficult to design a gradient descent algorithm or a similar iterative method. Even if we did that, a gradient search algorithm would be likely to find a local minimum because of the size of the search space.

First, we define a measure over the space of structures which we want to maximize:

Definition 1. *The inverse error measure for structure S' is*

$$inv_e(S') = \frac{\frac{1}{p_{S'}(\hat{c}(X)\neq C)}}{\sum_S \frac{1}{p_S(c(X)\neq C)}}, \qquad (7)$$

where the summation is over the space of possible structures and $p_S(\hat{c}(X) \neq C)$ is the probability of error of the best classifier learned with structure S.

We use Metropolis-Hastings sampling [49] to generate samples from the inverse error measure without having to ever compute it for all possible structures. For constructing the Metropolis-Hastings sampling, we define a neighborhood of a structure as the set of directed acyclic graphs to which we can transit in the next step. Transition is done using a predefined set of possible changes to the structure; at each transition, a change consists of a single edge addition, removal, or reversal. We define the acceptance probability of a candidate structure, S_{new}, to replace a previous structure, S_t, as follows:

$$\min\left(1, \left(\frac{inv_e(S^{new})}{inv_e(S^t)}\right)^{1/T} \frac{q(S^t|S^{new})}{q(S^{new}|S^t)}\right) = \min\left(1, \left(\frac{p^t_{error}}{p^{new}_{error}}\right)^{1/T} \frac{N_t}{N_{new}}\right), \qquad (8)$$

where $q(S'|S)$ is the transition probability from S to S' and N_t and N_{new} are the sizes of the neighborhoods of S_t and S_{new}, respectively; this choice corresponds to equal probability of transition to each member in the neighborhood of a structure. This choice of neighborhood and transition probability creates a Markov chain which is aperiodic and irreducible, thus satisfying the Markov chain Monte Carlo (MCMC) conditions [50]. We summarize the algorithm, which we name Stochastic Structure Search (SSS), in Fig. 4.

We add T as a temperature factor in the acceptance probability. Roughly speaking, T close to 1 would allow acceptance of more structures with higher probability of error than previous structures. T close to 0 mostly allows acceptance of structures that improve probability of error. A fixed T amounts to changing the distribution being sampled by the MCMC, while a decreasing T is a simulated annealing run, aimed at finding the maximum of the inverse error measures. The rate of decrease of the temperature determines the rate of convergence. Asymptotically in the number of data, a logarithmic decrease of T guarantees convergence to a global maximum with probability that tends to one [51].

The SSS algorithm, with a logarithmic cooling schedule T, can find a structure that is close to the minimum probability of error. There are two caveats though. First, the logarithmic cooling schedule is very slow. We use faster cooling schedules and a starting point which is the best out of either the NB classifier or the TAN classifier. Second, we never have access to the true probability of error for any given structure, p^S_{error}. Instead, we use the empirical error over the training data (denoted as $\hat{p}^S_{error}$).

To avoid the problem of overfitting several approaches are possible. The first is cross-validation; the labeled training data is split to smaller sets and several tests are performed using the smaller sets as test sets. However, this approach can significantly slow down the search and is

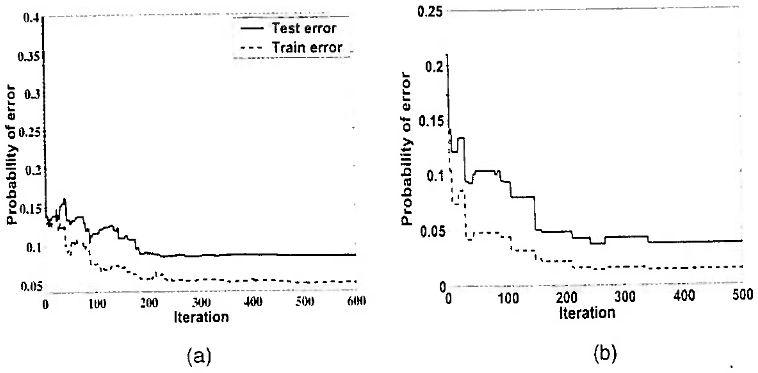


Fig. 5. Train and test error during the structure search for (a) the artificial data and (b) shuttle data for the labeled and unlabeled data experiments.

suitable only if the labeled training set is moderately large. Another approach is to penalize different structures according to some complexity measure. We could use the BIC or MDL complexity measure, but we chose to use the multiplicative penalty term derived from structural risk minimization since it is directly related to the relationship between training error and generalization error. We define a modified error term for use in (7) and (8):

$$(\hat{p}^S_{error})^{mod} = \frac{\hat{p}^S_{error}}{1 - c \cdot \sqrt{\frac{h_S(\log\frac{2n}{h_S}+1)-\log(\eta/4)}{n}}}, \quad (9)$$

where h_S is the Vapnik-Chervonenkis (VC) dimension of the classifier with structure S, n is the number of training records, η and c are between 0 and 1.

To approximate the VC dimension, we use $h_S \propto N_S$, where N_S is the number of (free) parameters in the Markov blanket of the class variable in the network, assuming that all variables are discrete. We point the reader to [52], in which it was shown that the VC dimension of a Naive Bayes classifier is linearly proportional to the number of parameters. It is possible to extend this result to networks where the features are all descendants of the class variable. For more general networks, features that are not in the Markov blanket of the class variable cannot effect its value in classification (assuming there are no missing values for any feature), justifying the above approximation. In our initial experiments, we found that the multiplicative penalty outperformed the holdout method and the MDL and BIC complexity measures.

3.3 Evaluation Using UCI Machine Learning Data Sets

To evaluate structure learning methods with labeled and unlabeled data, we started with an empirical study involving simulated data. We artificially generated data to investigate: 1) whether the SSS algorithm finds a structure that is close to the structure that generated the data and 2) whether the algorithm uses unlabeled data to improve the classification performance. A typical result is as follows: We generated data from a TAN structure with 10 features. The data set consisted of 300 labeled and 30,000 unlabeled records. We first estimated the Bayes error rate by learning with the correct structure and with a very large fully labeled data set. We obtained a classification accuracy of 92.49 percent. We learned one Naive Bayes classifier with only the labeled records and another with both labeled and unlabeled records; likewise, we learned a TAN classifier only with the labeled records and another with both labeled and unlabeled records, using the EM-TAN algorithm, and, finally, we learned a Bayesian network classifier with our SSS algorithm using both labeled and unlabeled records. The results are presented in the first row of Table 1. With the correct structure, adding unlabeled data improves performance significantly (columns TAN-L and EM-TAN). Note that adding unlabeled data degraded the performance from 16 percent error to 40 percent error when we learned the Naive Bayes classifier. The structure search algorithm comes close to the performance of the classifier learned with the correct structure. Fig. 5a shows the changes in the test and train error during the search process. The graph shows the first 600 moves of the search, initialized with the Naive Bayes structure. The error usually decreases as new structures are accepted; occasionally, we see an increase in the error allowed by Metropolis-Hastings sampling.

Next, we performed experiments with some of the UCI data sets, using relatively small labeled sets and large unlabeled sets (Table 1). The results suggest that structure learning holds the most promise in utilizing the unlabeled data. There is no clear "winner" approach, although SSS yields better results in most cases. We see performance degradation with NB for every data set. EM-TAN can sometimes improve performance over TAN with just labeled data (Shuttle). With the Chess data set, discarding the unlabeled data and using only TAN seems the best approach. We have compared two likelihood-based structure learning methods (K2 and MCMC) on the same data sets as well [34], showing that, even if we allow the algorithms to use large labeled data sets to learn the structure, the resultant networks still suffer from performance degradation when learned with unlabeled data.

Illustrating the iterations of the SSS algorithm, Fig. 5b shows the changes in error for the shuttle data set.

4 Learning Bayesian Network Classifiers for HCI Applications

The experiments in the previous section discussed commonly used machine learning data sets. In the next sections, we discuss two HCI applications that could benefit from the use of unlabeled data. We start with facial expression recognition.

TABLE 1
Classification Results (in %) for Naive Bayes, TAN, EM-CBL1, and Stochastic Structure Search

Dataset	Train		Test	NB-L	EM-NB	TAN-L	EM-TAN	EM-CBL1	SSS
	# lab	# unlab							
TAN artificial	300	30000	50000	83.4±0.2	59.2±0.2	90.9±0.1	91.9±0.1	N/A	91.1±0.1
Satimage	600	3835	2000	81.7±0.9	77.5±0.9	**83.5±0.8**	81.0±0.9	**83.5±0.8**	83.4±0.8
Shuttle	100	43400	14500	82.4±0.3	76.1±0.4	81.2±0.3	90.5±0.2	91.8±0.2	**96.3±0.2**
Adult	6000	24163	15060	83.9±0.3	73.1±0.4	84.7±0.3	80.0±0.3	82.7±0.3	**85.0±0.3**
Chess	150	1980	1060	79.8±1.2	62.1±1.5	**87.0±1.0**	71.2±1.4	81.0±1.2	76.0±1.3

xx-L indicates learning only with the available labeled data.

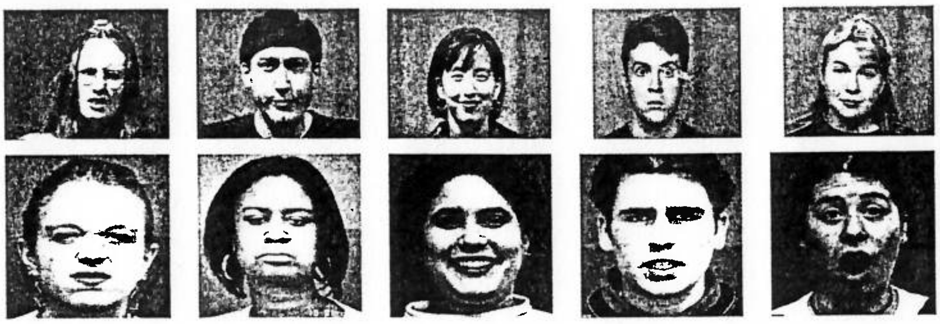


Fig. 6. Examples of images from the video sequences used in the experiment. The top row shows subjects from the Chen-Huang DB, the bottom row shows subjects from the Kanade et al. DB (printed with permission from the researchers).

4.1 Facial Expression Recognition Using Bayesian Network Classifiers

Since the early 1970s, Ekman and his colleagues have performed extensive studies of human facial expressions [53] and found evidence to support universality in facial expressions. These "universal facial expressions" are those representing happiness, sadness, anger, fear, surprise, and disgust. Ekman's work inspired many researchers to analyze facial expressions by means of image and video processing. By tracking facial features and measuring the amount of facial movement, they attempt to categorize different facial expressions. Recent work on facial expression analysis and recognition has used these "basic expressions" or a subset of them. In [54], Pantic and Rothkrantz provide an in-depth review of much of the research done in automatic facial expression recognition in recent years.

One of the challenges facing researchers attempting to design facial expression recognition systems is the relatively small amount of available labeled data. Construction and labeling of a good database of images or videos of facial expressions requires expertise, time, and training of subjects. Only a few such databases are available, such as the Kanade et al. database [55]. However, collecting, without labeling, data of humans displaying expressions is not as difficult. Therefore, it is beneficial to use classifiers that can be learned with a combination of some labeled data and a large amount of unlabeled data. As such we use (generative) Bayesian network classifiers.

We have developed a real-time facial expression recognition system [56]. The system uses a model-based nonrigid face tracking algorithm [57] to extract motion features (seen in Fig. 7) that serve as input to a Bayesian network classifier used for recognizing the different facial expressions. There are two main motivations for using Bayesian network classifiers in this problem. The first is the ability to learn with unlabeled data and infer the class label even when some of the features are missing (e.g., due to failure in tracking because of occlusion). The second motivation is that it is possible to extend the system to fuse other modalities, such as audio, in a principled way by simply adding subnetworks representing the audio features.

4.1.1 Experimental Design

We use two different databases, a database collected by Chen [58] and the Kanade et al. AU code facial expression database [55]. The first is a database of subjects that were instructed to display facial expressions corresponding to the six types of emotions. All the tests of the algorithms are performed on a set of five people, each one displaying six sequences of each one of the six emotions, starting and ending at the neutral expression. The video sampling rate was 30 Hz and a typical emotion sequence is about 70 samples long (~ 2s). Fig. 6 (upper row) shows one frame of each subject.

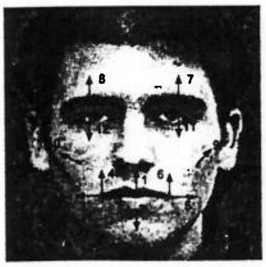


Fig. 7. Motion units extracted from face tracking.

TABLE 2
Summary of the Databases

Database	# of Subjects	Overall # of sequences per expression	# of sequences per subject per expression	average # of frames per expression
Chen-Huang DB	5	30	6	70
Cohn-Kanade DB	53	53	1	8

The Kanade et al. database [55] consists of expression sequences of subjects, starting from a neutral expression and ending in the peak of the facial expression. There are 104 subjects in the database. Because, for some of the subjects, not all of the six facial expressions sequences were available to us, we used a subset of 53 subjects, for which at least four of the sequences were available. For each subject, there is at most one sequence per expression with an average of eight frames for each expression. Fig. 6 (lower row) shows some examples used in the experiments. A summary of both databases is presented in Table 2. We measure the accuracy with respect to the classification result of each frame, where each frame in the video sequence was manually labeled to one of the expressions (including neutral). This manual labeling can introduce some "noise" in our classification because the boundary between neutral and the expression of a sequence is not necessarily optimal and frames near this boundary might cause confusion between the expression and the neutral.

4.1.2 Experimental Results with Labeled Data

We start with experiments using all our labeled data. This can be viewed as an upper bound on the performance of the classifiers trained with most of the labels removed. For the labeled only case, we also compare results with training of an artificial Neural network (ANN) so as to test how Bayesian network classifiers compare with a different kind of classifier for this problem. We perform person independent tests by partitioning the data such that the sequences of some subjects are used as the test sequences and the sequences of the remaining subjects are used as training sequences. Table 3 shows the recognition rate of the test for all classifiers. The classifier learned with the SSS algorithm outperforms both the NB and TAN classifiers, while ANN does not perform well compared to all the others.

4.1.3 Experiments with Labeled and Unlabeled Data

We perform person-independent experiments with labeled and unlabeled data. We first partition the data to a training set and a test set (2/3 training, 1/3 for testing) and choose at random a portion of the training set and remove the labels. This procedure ensures that the distribution of the labeled and the unlabeled sets are the same.

TABLE 3
Recognition Rate (%) for Person-Independent Test

	NB	TAN	SSS	ANN
Chen-Huang Database	71.78	80.31	83.62	66.44
Cohn-Kandade Database	77.70	80.40	81.80	73.81

We then train Naive Bayes and TAN classifiers, using just the labeled part of the training data and the combination of labeled and unlabeled data. We also use the SSS and the EM-CBL1 algorithms to train a classifier, using both labeled and unlabeled data (we do not search for the structure with just the labeled part because it is too small for performing a full structure search).

Table 4 shows the results of the experiments. We see that with NB and TAN when using 200 and 300 labeled samples, adding the unlabeled data degrades the performance of the classifiers and we would have been better off not using the unlabeled data. We also see that EM-CBL1 performs poorly in both cases. Using the SSS algorithm, we are able to improve the results and utilize the unlabeled data to achieve performance which is higher than using just the labeled data with NB and TAN. The fact that the performance is lower than in the case when all the training set was labeled (about 75 percent compared to over 80 percent) implies that the relative value of labeled data is higher than of unlabeled data, as was shown by Castelli [26]. However, had there been more unlabeled data, the performance would be expected to improve.

4.2 Applying Bayesian Network Classifiers to Face Detection

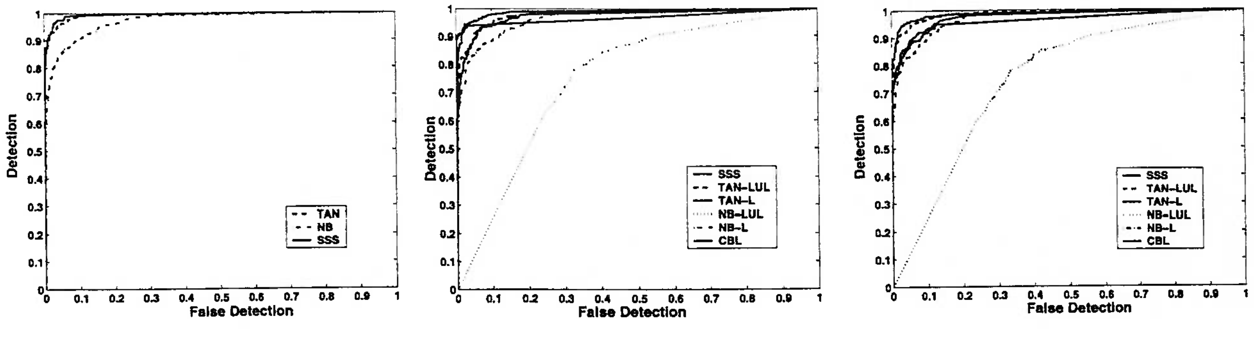
We apply Bayesian network classifiers to the problem of face detection with the purpose of showing that, using our proposed methods, semisupervised learning can be used to learn good face detectors. We take an appearance-based approach, using the intensity of image pixels as the features for the classifier. For learning and defining the Bayesian network classifiers, we must look at fixed size windows and learn how a face appears in such windows, where we assume that the face appears in most of the window's pixels. The goal of the classifier would be to determine if the pixels in a fixed size window are those of a face or nonface.

We note that there have been numerous appearance-based approaches for face detection, many with considerable success (see Yang et al. [59] for a detailed review on the state-of the-art in face detection). However, there has not been any attempt, to our knowledge, to use semisupervised learning in face detection. While labeled databases of face images are available, a universally robust face detector is still difficult to construct. The main challenge is that faces appear very different under different lighting conditions, expressions, with or without glasses, facial hair, makeup, etc. A classifier trained with some labeled images and a large number of unlabeled images would enable incorporating many more facial variations without the need to label huge data sets.

In our experiments, we used a training set consisting of 2,429 faces and 10,000 nonfaces obtained from the MIT CBCL Face database #1 [60]. Each face image is cropped and

TABLE 4
Classification Results for Facial Expression Recognition with Labeled and Unlabeled Data

Dataset	Train		Test	NB-L	EM-NB	TAN-L	EM-TAN	EM-CBL1	SSS
	# lab	# unlab							
Cohn-Kanade	200	2980	1000	72.5±1.4	69.1±1.4	72.9±1.4	69.3±1.4	66.2±1.5	**74.8±1.4**
Chen-Huang	300	11982	3555	71.3±0.8	58.5±0.8	72.5±0.7	62.9±0.8	65.9±0.8	**75.0±0.7**



(a)

(b)

(c)

Fig. 8. ROC curves showing detection rates of faces compared to false detection of faces of the different (SSS, TAN, and NB) classifiers and different ratios of labeled and unlabeled data, (a) with all the data labeled (no unlabeled data), (b) with 95 percent of the data unlabeled, and (c) with 97.5 percent of the data unlabeled.

resampled to an 8×8 window, thus we have a classifier with 64 features. We also randomly rotate and translate the face images to create a training set of 10,000 face images. In addition, we have available 10,000 nonface images. We leave out 1, 000 images (faces and nonfaces) for testing and train the Bayesian network classifier on the remaining 19,000. In all the experiments, we learn a Naive Bayes, a TAN, and two general generative Bayesian network classifiers, the latter using the EM-CBL1 and the SSS algorithms.

To compare the results of the classifiers, we use the receiving operating characteristic (ROC) curves. The ROC curves show, under different classification thresholds, ranging from 0 to 1, the probability of detecting a face in a face image, $P_D = P(\hat{C} = face|C = face)$, against the probability of falsely detecting a face in a nonface image, $P_{FD} = P(\hat{C} = face|C \neq face)$.

We first learn using all the training data being labeled. Fig. 8a shows the resultant ROC curve for this case. The classifier learned with the SSS algorithm outperforms both the TAN and NB classifiers and all perform quite well, achieving about 96 percent detection rates with a low rate of false alarm.

Next, we remove the labels of 95 percent of the training data (leaving only 475 labeled images) and train the classifiers. Fig. 8b shows the resultant ROC curve for this case. We see that the NB classifier using both labeled and unlabeled data performs very poorly. The TAN based on the 475 labeled images and the TAN based on the labeled and unlabeled images are close in performance, thus there was no significant degradation of performance when adding the unlabeled data. The classifier using all data and the SSS outperforms the rest with an ROC curve close to the best ROC curve in Fig. 8a. Fig. 8c shows the ROC curve with only 250 labeled data used. Again, NB with both labeled and unlabeled performs poorly, while SSS outperforms the other classifiers with no great reduction of performance compared to the two other ROC curves. The experiment shows that using structure search, the unlabeled data was utilized successfully to achieve a classifier almost as good as if all the data was labeled.

5 Summary and Discussion

Using unlabeled data to enhance the performance of classifiers trained with few labeled data has many applications in pattern recognition, computer vision, HCII, data mining, text recognition, and more. To fully utilize the potential of unlabeled data, the abilities and limitations of existing methods must be understood.

The main contributions of this paper can be summarized as follows:

1. We have derived and studied the asymptotic behavior of semisupervised learning based on maximum-likelihood estimation. We presented a detailed analysis of performance degradation from unlabeled data, showing that it is directly related to modeling assumptions, regardless of numerical instabilities or finite sample effects.
2. We discussed the implications of the analysis of semisupervised learning on Bayesian network classifiers, namely, the importance of structure when unlabeled data are used in training. We listed the possible shortcomings of likelihood-based structural learning algorithms when learning classifiers, especially when unlabeled data are present.
3. We introduced a classification driven structure search algorithm based on Metropolis-Hastings

[58] L.S. Chen, "Joint Processing of Audio-Visual Information for the Recognition of Emotional Expressions in Human-Computer Interaction," PhD thesis, Univ. of Illinois at Urbana-Champaign, 2000.

[59] M.H. Yang, D. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 1, pp. 34-58, Jan. 2002.

[60] "MIT CBCL Face Database #1," MIT Center for Biological and Computation Learning: http://www.ai.mit.edu/projects/cbcl, 2002.

[61] K. Bennett and A. Demiriz, "Semi-Supervised Support Vector Machines," *Proc. Neural Information and Processing Systems (NIPS),* pp. 368-374, 1998.

[62] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. 11th Ann. Conf. Computational Learning Theory,* pp. 92-100, 1998.

[63] R. Ghani, "Combining Labeled and Unlabeled Data for Multiclass Text Categorization," *Proc. Int'l Conf. Machine Learning (ICML),* pp. 187-194, 2002.

Ira Cohen earned the BSc degree from Ben-Gurion University, Israel, and the MS and PhD degrees from the University of Illinois at Urbana-Champaign in electrical and computer engineering in 1998, 2000, and 2003, respectively. Since 2003, he has been a research scientist at Hewlett-Packard research labs in Palo Alto, California, where he works on machine learning theory with application to computer system and performance modeling. His research interests are in probabilistic models, computer vision, human computer interaction, and system modeling. He is a member of the IEEE.

Fabio G. Cozman received the PhD degree in robotics in 1997 from the School of Computer Science, Carnegie Mellon University. He is an associate professor at the University of São Paulo, Brazil. He has worked on the theory and applications of sets of probability measures, Bayesian networks, and semisupervised learning.

Nicu Sebe received the PhD degree from Leiden University, The Netherlands, in 2001. Currently, he is with the Faculty of Science, University of Amsterdam, The Netherlands, where he is doing research in the areas of multimedia information retrieval and human-computer interaction in computer vision applications. He is the author of the book *Robust Computer Vision-Theory and Applications* (Kluwer, April 2003) and of the upcoming book *Computer Vision: A Machine Learning Approach* (Kluwer, 2004). He was the guest editor of an CVIU Special Issue on Video Retrieval and Summarization (December 2003), was the cochair of the Fifth ACM Multimedia Information Retrieval Workshop, (MIR '03) (in conjunction with ACM Multimedia 2003) and is the cochair of the Human Computer Interaction Workshop, (HCI '04) (in conjunction with ECCV 2004). He also was the technical program chair for the International Conference on Image and Video Retrieval (CIVR 2003). He has published more than 50 technical papers in the areas of computer vision, content-based retrieval, pattern recognition, and human-computer interaction and has served on the program committee of several conferences in these areas. He is a member of the IEEE and ACM.

Marcelo C. Cirelo is an electrical engineer and graduate student at University of São Paulo, with interests in the areas of machine learning and computer vision—particularly in applications that require manipulation of labeled and unlabeled data.

Thomas S. Huang received the BS degree in electrical engineering from National Taiwan University, Taipei, Taiwan, and the MS and ScD degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, Massachusetts. He was on the faculty of the Department of Electrical Engineering at MIT from 1963 to 1973 and on the faculty of the School of Electrical Engineering and director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now the William L. Everitt Distinguished Professor of Electrical and Computer Engineering and a research professor at the Coordinated Science Laboratory, and head of the Image Formation and Processing Group at the Beckman Institute for Advanced Science and Technology and cochair of the Institute's major research theme Human Computer Intelligent Interaction. Dr. Huang's professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 14 books and more than 500 papers in network theory, digital filtering, image processing, and computer vision. He is a member of the National Academy of Engineering, a foreign member of the Chinese Academies of Engineering and Sciences, and a fellow of the International Association of Pattern Recognition, IEEE, and the Optical Society of American, and has received a Guggenheim fellowship, an A.V. Humboldt Foundation Senior US Scientist Award, and a fellowship from the Japan Association for the Promotion of Science. He received the IEEE Signal Processing Society's Technical Achievement Award in 1987 and the Society Award in 1991. He was awarded the IEEE Third Millennium Medal in 2000. Also in 2000, he received the Honda Lifetime Achievement Award for "contributions to motion analysis." In 2001, he received the IEEE Jack S. Kilby Medal. In 2002, he received the King-Sun Fu Prize, International Association of Pattern Recognition, and the Pan Wen-Yuan Outstanding Research Award. He is a founding editor of the *International Journal of Computer Vision, Graphics, and Image Processing* and editor of the Springer series in information sciences, published by Springer Verlag.

sampling and showed that it performs well both on fully labeled data sets and on labeled and unlabeled training sets. As a note for practitioners, the SSS algorithm appears to work well for relatively large data sets and difficult classification problems that are represented by complex structures. Large data sets are those where there are enough labeled data for reliable estimation of the empirical error, allowing search for complex structures, and there are enough unlabeled data to reduce the estimation variance of complex structures.

4. We presented our real-time facial expression recognition system using a model-based face tracking algorithm and Bayesian network classifiers. We showed experiments using both labeled and unlabeled data.
5. We presented the use of Bayesian network classifiers for learning to detect faces in images. We note that, while finding a good classifier is a major part of any face detection system, there are many more components that need to be designed for such a system to work on natural images (e.g., ability to detect at multiscales, highly varying illumination, large rotations of faces, and partial occlusions). Our goal was to present the first step in designing such a system and show the feasibility of the approach when training with labeled and unlabeled data.

Our discussion of semisupervised learning for Bayesian networks suggests the following path: When faced with the option of learning Bayesian networks with labeled and unlabeled data, start with Naive Bayes and TAN classifiers, learn with only labeled data, and test whether the model is correct by learning with the unlabeled data. If the result is not satisfactory, then SSS can be used to attempt to further improve performance with enough computational resources. If none of the methods using the unlabeled data improve performance over the supervised TAN (or Naive Bayes), either discard the unlabeled data or try to label more data, using active learning for example.

Following our investigation of semisupervised learning, there are several important open theoretical questions and research directions:

- Is it possible to find necessary and sufficient conditions for performance degradation to occur? Finding such conditions are of great practical significance. Knowing these conditions can lead to the design of new useful tests that will indicate when unlabeled can be used or when they should be discarded or if a different model should be chosen.
- An important question is whether other semisupervised learning methods, such as transductive SVM [61] or cotraining [62], will exhibit the phenomenon of performance degradation? While no extensive studies have been performed, a few results from the literature suggest that it is a realistic conjecture. Zhang and Oles [2] demonstrated that transductive SVM can cause degradation of performance when unlabeled data are added. Ghani [63] described experiments where the same phenomenon occurred with cotraining. If the causes of performance degradation are similar for different algorithms, it should be possible to present a unified theory for semisupervised learning.
- Are there performance guarantees for semisupervised learning with finite amounts of data, labeled and unlabeled? In supervised learning, such guarantees are studied extensively. PAC and risk minimization bounds help in determining the minimum amount of (labeled) data necessary to learn a classifier with good generalization performance. However, there are no existing bounds on the classification performance when training with labeled and unlabeled data. Finding such bounds can be derived using principles in estimation theory, based on the asymptotic covariance properties of the estimator. Other bounds can be derived using PAC theoretical approaches. The existence of such bounds can immediately lead to new algorithms and approaches, better utilizing unlabeled data.
- Can we use the fact that unlabeled data indicate model incorrectness to actively learn better models? The use of active learning seems promising whenever possible and it might be possible to extend active learning to learn better models, not just enhancement of the parameter estimation.

In closing, this work should be viewed as a combination of three main components. The theory showing the limitations of unlabeled data is used to motivate the design of algorithms to search for better performing structures of Bayesian networks and finally, the successful application to the real-world problems we were interested in solving by learning with labeled and unlabeled data.

ACKNOWLEDGMENTS

The authors would like to acknowledge Alex Bronstein and Marsha Duro of HP-Labs for their support and discussions on the analysis of semisupervised learning. Portions of this work have been done at HP-Labs, Palo-Alto. Ira Cohen has been funded by an HP Fellowship. Financial support has also been granted by HP-Brazil through the Eldorado Research Institute. The authors would like to thank Jilin Tu for his work on the face tracking, Roy Wang and Ming-Hsuan Yang for their help with the face detection database, and Jeffery Cohn and Larry Chen for the facial expression databases. They would like to thank Moises Goldszmidt, Ashutosh Garg, Ismenia Blavatsky, and Michael Lew for discussions on various parts of this work and the reviewers for their helpful comments. They coded our own Naive Bayes and TAN classifiers in the Java language, using the libraries of the JavaBayes system (freely available at http://www.cs.cmu.edu/~javabayes). This work has been partially funded by US National Science Foundation Grant IIS 00-85980.

REFERENCES

[1] B. Shahshahani and D. Landgrebe, "Effect of Unlabeled Samples in Reducing the Small Sample Size Problem and Mitigating the Hughes Phenomenon," *IEEE Trans. Geoscience and Remote Sensing*, vol. 32, no. 5, pp. 1087-1095, 1994.

[2] T. Zhang and F. Oles, "A Probability Analysis on the Value of Unlabeled Data for Classification Problems," *Proc. Int'l Conf. Machine Learning (ICML)*, pp. 1191-1198, 2000.

[3] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM," *Machine Learning*, vol. 39, no. 2, pp. 103-134, 2000.

[4] R. Bruce, "Semi-Supervised Learning Using Prior Probabilities and EM," *Proc. Int'l Joint Conf. AI Workshop Text Learning: Beyond Supervision*, 2001.

[5] S. Baluja, "Probabilistic Modelling for Face Orientation Discrimination: Learning from Labeled and Unlabeled Data," *Proc. Neural Information and Processing Systems (NIPS)*, pp. 854-860, 1998.

[6] R. Kohavi, "Scaling Up the Accuracy of Naive Bayes Classifiers: A Decision-Tree Hybrid," *Proc. Second Int't Conf. Knowledge Discovery and Data Mining*, pp. 202-207, 1996.

[7] I. Cohen, F.G. Cozman, and A. Bronstein, "On the Value of Unlabeled Data in Semi-Supervised Learning Based on Maximum-Likelihood Estimation," Technical Report HPL-2002-140, Hewlett-Packard Labs, 2002.

[8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, Calif.: Morgan Kaufmann, 1988.

[9] A. Garg, V. Pavlovic, and J. Rehg, "Boosted Learning in Dynamic Bayesian Networks for Multimodal Speaker Detection," *Proc. IEEE*, vol. 91, pp. 1355-1369, Sept. 2003.

[10] N. Oliver, E. Horvitz, and A. Garg, "Hierarchical Representations for Learning and Inferring Office Activity from Multimodal Information," *Proc. Int'l Conf. Multimodal Interfaces, (ICMI)*, 2002.

[11] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers," *Machine Learning*, vol. 29, no. 2, pp. 131-163, 1997.

[12] R. Greiner and W. Zhou, "Structural Extension to Logistic Regression: Discriminative Parameter Learning of Belief Net Classifiers," *Proc. Ann. Nat'l Conf. Artificial Intelligence*, pp. 167-173, 2002.

[13] P. Ekman and W. Friesen, *Facial Action Coding System: Investigator's Guide.* Palo Alto, Calif.: Consulting Psychologists Press, 1978.

[14] C.L. Blake and C.J. Merz, "UCI Repository of Machine Learning Databases," Dept. of Information and Computer Sciences, Univ. of California, Irvine, 1998.

[15] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition.* New York: Springer Verlag, 1996.

[16] A. Corduneanu and T. Jaakkola, "Continuations Methods for Mixing Heterogeneous Sources," *Proc. Uncertainty in Artificial Intelligence (UAI)*, pp. 111-118, 2002.

[17] R. Chhikara and J. McKeon, "Linear Discriminant Analysis with Misallocation in Training Samples," *J. Am. Statistical Assoc.*, vol. 79, pp. 899-906, 1984.

[18] C. Chittineni, "Learning with Imperfectly Labeled Examples," *Pattern Recognition*, vol. 12, pp. 271-281, 1981.

[19] T. Krishnan and S. Nandy, "Efficiency of Discriminant Analysis when Initial Samples Are Classified Stochastically," *Pattern Recognition*, vol. 23, pp. 529-537, 1990.

[20] T. Krishnan and S. Nandy, "Efficiency of Logistic-Normal supervision," *Pattern Recognition*, vol. 23, pp. 1275-1279, 1990.

[21] S. Pal and E.A. Pal, *Pattern Recognition from Classical to Modern Approaches.* World Scientific, 2002.

[22] D.B. Cooper and J.H. Freeman, "On the Asymptotic Improvement in the Outcome of Supervised Learning Provided by Additional Nonsupervised Learning," *IEEE Trans. Computers*, vol. 19, no. 11, pp. 1055-1063, Nov. 1970.

[23] D.W. Hosmer, "A Comparison of Iterative Maximum Likelihood Estimates of the Parameters of a Mixture of Two Normal Distributions under Three Different Types of Sample," *Biometrics*, vol. 29, pp. 761-770, Dec. 1973.

[24] T.J. O'Neill, "Normal Discrimination with Unclassified Observations," *J. Am. Statistical Assoc.*, vol. 73, no. 364, pp. 821-826, 1978.

[25] S. Ganesalingam and G.J. McLachlan, "The Efficiency of a Linear Discriminant Function Based on Unclassified Initial Samples," *Biometrika*, vol. 65, pp. 658-662, Dec. 1978.

[26] V. Castelli, "The Relative Value of Labeled and Unlabeled Samples in Pattern Recognition," PhD thesis, Stanford Univ., Palo Alto, Calif., 1994.

[27] J. Ratsaby and S.S. Venkatesh, "Learning from a Mixture of Labeled and Unlabeled Examples with Parametric Side Information," *Proc. Eighth Ann. Conf. Computational Learning Theory*, pp. 412-417, 1995.

[28] T. Mitchell, "The Role of Unlabeled Data in Supervised Learning," *Proc. Sixth Int'l Colloquium Cognitive Science*, 1999.

[29] D.J. Miller and H.S. Uyar, "A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data," *Neural Information and Processing Systems (NIPS)*, pp. 571-577, 1996.

[30] M. Collins and Y. Singer, "Unupervised Models for Named Entity Classification," *Proc. Int'l Conf. Machine Learning*, pp. 327-334, 2000.

[31] F. DeComite, F. Denis, R. Gilleron, and F. Letouzey, "Positive and Unlabeled Examples Help Learning," *Proc. 10th Int'l Conf. Algorithmic Learning Theory*, O. Watanabe and T. Yokomori, eds., pp. 219-230, 1999.

[32] S. Goldman and Y. Zhou, "Enhancing Supervised Learning with Unlabeled Data," *Proc. Int'l Conf. Machine Learning*, pp. 327-334, 2000.

[33] F.G. Cozman and I. Cohen, "Unlabeled Data Can Degrade Classification Performance of Generative Classifiers," *Proc. 15th Int'l Florida Artificial Intelligence Soc. Conf.*, pp. 327-331, 2002.

[34] I. Cohen, "Semisupervised Learning of Classifiers with Application to Human-Computer Interaction," PhD thesis, Univ. of Illinois at Urbana-Champaign, 2003.

[35] F.G. Cozman, I. Cohen, and M. Cirelo, "Semi-Supervised Learning of Mixture Models," *Proc. Int'l Conf. Machine Learning (ICML)*, pp. 99-106, 2003.

[36] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc., Series B*, vol. 39, no. 1, pp. 1-38, 1977.

[37] H. White, "Maximum Likelihood Estimation of Misspecified Models," *Econometrica*, vol. 50, pp. 1-25, Jan. 1982.

[38] F.G. Cozman and I. Cohen, "The Effect of Modeling Errors in Semi-Supervised Learning of Mixture Models: How Unlabeled Data Can Degrade Performance of Generative Classifiers," technical report, Univ. of Sao Paulo, http://www.poli.usp. br/p/fabio.cozman/Publications/lul.ps.gz, 2003.

[39] S.W. Ahmed and P.A. Lachenbruch, "Discriminant Analysis when Scale Contamination Is Present in the Initial Sample," *Classification and Clustering*, pp. 331-353, New York: Academic Press, 1977.

[40] G.J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition.* New York: John Wiley and Sons, 1992

[41] J.H. Friedman, "On Bias, Variance, 0/1-Loss, and the Curse-of-Dimensionality," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 55-77, 1997.

[42] M. Meila, "Learning with Mixture of Trees," PhD thesis Massachusetts Inst. of Technology, Boston, 1999.

[43] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, second ed. Cambridge, Mass.: MIT Press, 2000.

[44] J. Pearl, *Causality: Models, Reasoning, and Inference.* Cambridge, Mass.: Cambridge Univ. Press, 2000.

[45] J. Cheng, R. Greiner, J. Kelly, D.A. Bell, and W. Liu, "Learning Bayesian Networks from Data: An Information-Theory Based Approach," *Artificial Intelligence J.*, vol. 137, pp. 43-90, May 2002.

[46] J. Cheng and R. Greiner, "Comparing Bayesian Network Classifiers," *Proc. Uncertainty in Artificial Intelligence (UAI)*, pp. 101-108, 1999.

[47] T.V. Allen and R. Greiner, "A Model Selection Criteria for Learning Belief Nets: An Empirical Comparison," *Proc. Int'l Conf. Machine Learning (ICML)*, pp. 1047-1054, 2000.

[48] N. Friedman, "The Bayesian Structural EM Algorithm," *Proc. Uncertainty in Artificial Intelligence (UAI)*, pp. 129-138, 1998.

[49] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equation of State Calculation by Fast Computing Machines," *J. Chemical Physics*, vol. 21, pp. 1087-1092, 1953.

[50] D. Madigan and J. York, "Bayesian Graphical Models for Discrete Data," *Int'l Statistical Rev.*, vol. 63, no. 2, pp. 215-232, 1995.

[51] B. Hajek, "Cooling Schedules for Optimal Annealing," *Math. Operational Research*, vol. 13, pp. 311-329, May 1988.

[52] D. Roth, "Learning in Natural Language," *Proc. Int'l Joint Conf. Artificial Intelligence*, pp. 898-904, 1999.

[53] P. Ekman, "Strong Evidence for Universals in Facial Expressions: A Reply to Russell's Mistaken Critique," *Psychological Bulletin*, vol. 115, no. 2, pp. 268-287, 1994.

[54] M. Pantic and L.J. M. Rothkrantz, "Automatic Analysis of Facial Expressions: The State of the Art," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1424-1445, Dec. 2000.

[55] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive Database for Facial Expression Analysis," *Proc. Automatic Face and Gesture Recognition (FG '00)*, pp. 46-53, 2000.

[56] I. Cohen, N. Sebe, A. Garg, and T. S. Huang, "Facial Expression Recognition from Video Sequences," *Proc. Int'l Conf. Multimedia and Expo (ICME)*, pp. 121-124, 2002.

[57] H. Tao and T.S. Huang, "Connected Vibrations: A Modal Analysis Approach to Non-Rigid Motion Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 735-740, 1998.