SIMONE CASTELO BRANCO SIMÕES

AGREGAÇÃO VIA BOOTSTRAP: UMA INVESTIGAÇÃO DE DESEMPENHO EM CLASSIFICADORES ESTATÍSTICOS E REDES NEURAIS, AVALIAÇÃO NUMÉRICA E APLICAÇÃO NO SUPORTE AO DIAGNÓSTICO DE CÂNCER DE MAMA

Recife, fevereiro de 2007

AGREGAÇÃO VIA BOOTSTRAP: UMA INVESTIGAÇÃO DE DESEMPENHO EM CLASSIFICADORES ESTATÍSTICOS E REDES NEURAIS, AVALIAÇÃO NUMÉRICA E APLICAÇÃO NO SUPORTE AO DIAGNÓSTICO DE CÂNCER DE MAMA

SIMONE CASTELO BRANCO SIMÕES

Orientador: Prof. Dr. Wilson Rosa de Oliveira Júnior

Dissertação apresentada ao programa de pós-graduação em biometria como requerimento parcial para obtenção do grau de mestre em biometria pela Universidade Federal Rural de Pernambuco

RECIFE - PE - BRASIL

FEVEREIRO - 2007

Ficha catalográfica
Setor de Processos Técnicos da Biblioteca Central – UFRPE

S593a Simões, Simone Castelo Branco
Agregação via bootstrap: uma investigação de desempenho em classificadores estatísticos e redes neurais, avaliação numérica e aplicação no suporte ao diagnóstico de câncer de mama / Simone Castelo Branco Simões -- 2007.
112 f. : il.

Orientador: Wilson Rosa de Oliveira Júnior
Dissertação (Mestrado em Biometria) - Universidade Federal Rural de Pernambuco. Departamento de Estatística e Informática.
Inclui apêndice e bibliografia

CDD 574. 018 2

1. Reconhecimento de padrões
2. Classificação estatística
3. Rede neural
4. Bagging
5. Bootstrap
6. Câncer de mama
I. Oliveira Júnior, Wilson Rosa de
II. Título

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOMETRIA

AGREGAÇÃO VIA BOOTSTRAP: UMA INVESTIGAÇÃO DE DESEMPENHO EM CLASSIFICADORES ESTATÍSTICOS E REDES NEURAIS, AVALIAÇÃO NUMÉRICA E APLICAÇÃO NO SUPORTE AO DIAGNÓSTICO DE CÂNCER DE MAMA.

SIMONE CASTELO BRANCO SIMÕES

Dissertação julgada adequada para obtenção do título de mestre em Biometria, defendida e aprovada por unanimidade em 27/02/2007 pela banca examinadora.

Orientador:

Prof. Dr. Wilson Rosa de Oliveira Júnior
Universidade Federal Rural de Pernambuco

Comissão Examinadora:

Prof. Dr. Borko D. Stosic
Universidade Federal Rural de Pernambuco

Prof. Dr. Getúlio José Amorim do Amaral
Universidade Federal de Pernambuco

Prof. Dr. Francisco de Assis Tenório de Carvalho
Universidade Federal de Pernambuco

Agradecimentos

A meu bom Deus de todas as horas e momentos.

A toda minha família, em especial a meu pai, mãe e irmã, que sempre buscaram o melhor para mim.

A Dindinho, que sempre esteve ao meu lado, auxiliando, incentivando e, por todo seu amor, carinho e dedicação.

Às minhas companheiras de desafio e desabafo, Cristiane Rocha, Adalmeres e Cristiane Almeida, pelo apoio e alegrias compartilhados. Torço muito pelo sucesso de vocês!

A meu orientador pela confiança demonstrada, amizade e contribuições.

Pela disponibilidade e sugestões sou imensamente grata aos professores Gauss Cordeiro e Francisco Louzada Neto.

Aos participantes da banca examinadora, pela contribuição teórica apresentada como forma de sugestão.

À CAPES, pelo apoio financeiro.

O ritmo das coisas que o destino cria
Não se compõe na pressão da ansiedade
A conquista é merecimento que se agracia
Pela outorga de Deus em sua bondade.
– Jairo Lima

AGREGAÇÃO VIA BOOTSTRAP: UMA INVESTIGAÇÃO DE DESEMPENHO EM CLASSIFICADORES ESTATÍSTICOS E REDES NEURAIS, AVALIAÇÃO NUMÉRICA E APLICAÇÃO NO SUPORTE AO DIAGNÓSTICO DE CÂNCER DE MAMA

Autora: SIMONE CASTELO BRANCO SIMÕES

Orientador: Prof. Dr. Wilson Rosa de Oliveira Júnior

RESUMO

Em reconhecimento de padrões, o diagnóstico médico tem recebido grande atenção. Em geral, a ênfase tem sido a identificação de um melhor modelo de previsão diagnóstica, avaliado de acordo com a habilidade de generalização. Nesse contexto, métodos que combinam classificadores têm se mostrado muito eficazes, podendo ser considerados no melhoramento de desempenho em tarefas diagnósticas que exigem maior precisão. O método bagging, proposto por Breiman (1996), utiliza bootstrap para gerar diferentes amostras do conjunto de treinamento, construindo classificadores com as amostras geradas e combinando as diferentes previsões por voto majoritário. Em geral, estudos empíricos são realizados para avaliar o desempenho do bagging. Nesta dissertação, investigamos a habilidade de generalização do bagging para classificadores estatísticos usuais e a rede perceptron de múltiplas camadas através de simulações estocásticas. Diferentes estruturas de separação das populações são construídas a partir de distribuições específicas consideradas. Adicionalmente, realizamos uma aplicação no suporte ao diagnóstico de câncer de mama. Os resultados foram obtidos utilizando o ambiente de programação, análise de dados e gráficos `R`. Em geral, as simulações realizadas indicam que o desempenho do bagging depende do comportamento de separação das populações. Na aplicação, o bagging mostrou ser eficiente no melhoramento da sensibilidade.

BOOTSTRAP AGREGATING: AN INVESTIGATION OF PERFORMANCE IN STATISTICS AND NEURAL NETWORKS CLASSIFIERS, NUMERICAL EVALUATION AND APPLICATION ON BREAST CANCER DIAGNOSTIC SUPPORT

Author: SIMONE CASTELO BRANCO SIMÕES

Adviser: Prof. Dr. Wilson Rosa de Oliveira Júnior

ABSTRACT

In pattern recognition, the medical diagnosis has received great attention. In general, the emphasis has been to identify one best model for diagnostic forecast, measured according to generalization ability. In this context, ensembles methods have been eficients, can be considered on the improvement of performance in diagnostic tasks that demand greater precision. The bagging method, purposed from Breiman (1996), uses bootstrap to generate different samples of the training set, building classifiers with the generated samples and combining different forecasts for majority vote. In general, empirical estudies are done for evaluate the bagging performance. In this thesis, we investigate the bagging generalization ability for statistical usual classifiers and the multilayer perceptron net through sthocastic simulation. Different structures of separation of populations are build from especific distributions. Additionally, we make an application on diagnostic suport of brest cancer. The results were obtained using R. In general, we observed that bagging performance depends on the population separation behavior. In the application, bagging showed to be efficient on sensibility improvement.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1

Introdução

1.1 Introdução

O diagnóstico constitui uma prática médica essencial. Em geral, um diagnóstico médico refere-se a classificação de um indivíduo como doente ou não-doente. Esse processo de classificação se baseia em algumas características (variáveis ou atributos) dos indivíduos, consideradas indispensáveis para inferir conclusões ou tomar decisões.

As variáveis consideradas indispensáveis para o processo de classificação podem ser identificadas por um perito ou passar por um processo de seleção, responsável pela redução do número de variáveis inicialmente consideradas. Na área médica, em geral, elas são extraídas mediante um sensor, como por exemplo, um eletrocardiograma, um raio-X ou uma mamografia.

Em algumas situações, uma correta decisão diagnóstica é fundamental para garantir um perfeito restabelecimento e até a cura de um paciente quando portador de uma dada doença. Uma exemplo de classificação em que um correto diagnóstico é primordial é em casos de detecção de tumores como sendo malígno ou benígno. Nesse caso, uma correta decisão também pode poupar o paciente de uma série de exames desgastantes desnecessários.

Devido a importância de corretas decisões diagnósticas na prática médica, métodos automáticos ou quantitativos de diagnósticos em sistemas de reconhecimento de padrões vêm sendo bastante explorados em pesquisas científicas. Esses sistemas podem funcionar como suporte ao diagnóstico médico, auxiliando o médico na tomada de decisões. Tais sistemas também são denominados sistemas computacionais de apoio ao diagnóstico médico (Theodoridis & Koutroumbas, 2003).

Um sistema de reconhecimento de padrões é caracterizado pela forma automatizada de um processo de decisão diagnóstica, em geral, baseia-se em experiências passadas para inferir sobre decisões futuras. Métodos estatísticos de classificação, assim como as redes neurais artificiais vêm sendo bastante explorados nesta prática.

Um exemplo onde esses sistemas podem ser empregados é em casos de detecção do câncer de mama por meio da mamografia. Segundo o Instituto Nacional de Câncer (INCA), estudos sobre a efetividade da mamografia relatam a necessidade de exame clínico adicional e, indicam a existência de alguns fatores que podem influenciar esse diagnóstico.

Em tarefas de classificação diagnóstica, o procedimento automático de diagnóstico adotado pode ter seu desempenho medido através da taxa de classificação correta, que mede o nível de eficiência em diagnósticar corretamente um indivíduo.

Além de uma avaliação individual, que tem o intuito de investigar o quão bom é um método de classificação, procedimentos de avaliação de desempenho são freqüentemente realizados de forma comparativa, testando a eficiência de um método em comparação a outros métodos. Na prática, algumas medidas descritivas bastante utilizadas na avaliação de desempenho em procedimentos diagnósticos para duas categorias são sensibilidade e especificidade, além da curva ROC (receiver-operating characteristic).

A ênfase de grande parte dos estudos científicos na área de sistemas de suporte à decisão são essencialmente baseados na identificação de um melhor modelo de previsão. Porém, a utilização de métodos de previsão agregada têm sido recentemente explorados nesta prática. Alguns estudos apontam esses métodos como sendo mais eficazes na habilidade de generalizar. Em geral, a habilidade de generalização é avaliada de acordo com o desempenho em diagnosticar um conjunto de teste independente do conjunto utilizado

no treinamento. Bagging (Breiman, 1996) é uma estratégia de previsão agregada via bootstrap que se propõe a dar suporte à decisão, melhorando o desempenho de alguns classificadores.

Recentes estudos confirmam a eficácia dos métodos de previsão agregados, ver, por exemplo, Breiman (1998), Opitz & Maclin (1999), Dietteerich (2000), Friedman et. al. (2000) e Tan & Gilbert (2003). Entretanto, em geral, estudos empíricos são realizados para avaliar a capacidade de generalização do método bagging. Além disso, a maioria desses estudos não levam em consideração diferentes comportamentos de separação das populações aliado a técnicas estatísticas usuais e rede neural.

O objetivo desta dissertação é investigar o uso de bagging em diferentes cenários, construídos a partir de distribuições específicas, caracterizadas por diferentes formas de separação das populações, aliado a técnicas estatísticas usuais como discriminante linear de Fisher, discriminante quadrático, discriminante logístico, k-vizinhos mais próximos e a rede perceptron de múltiplas camadas. Adicionalmente, realizamos uma aplicação no suporte ao diagnóstico de câncer de mama.

1.2 Organização da Dissertação

Este capítulo inicial é dedicado a apresentação do trabalho. Nele retratamos todo o conteúdo abordado de forma suscinta, apontamos a estrutura de organização da dissertação e descrevemos o suporte computacional utilizado.

No Capítulo 2, introduzimos alguns dos principais conceitos que envolvem o sistema de reconhecimento de padrões. Enfatizamos a tarefa de classificação e descrevemos o processo que envolve a tomada de decisão através de métodos automáticos de diagnóstico como os estatísticos e as redes neurais artificiais.

No Capítulo 3, apresentamos os conceitos básicos que envolvem o processo de reconhecimento de padrões estatístico. Consideramos a teoria de decisão de Bayes e as funções discriminantes.

Do Capítulo 4 ao 7, encontram-se as principais definições metodologicas de técnicas

estatísticas que podem ser utilizadas no suporte ao diagnóstico médico. Mais especificamente, o Capítulo 4 é dedicado ao método não-linear de classificação de análise discriminante quadrática. O Capítulo 5 traz um método não-paramétrico de estimação de densidade, conhecido como método direto de classificação. Os Capítulos 6 e 7 são caracterizados por serem métodos lineares.

O Capítulo 8 é destinado a abordar redes neurais artificiais, também utilizada na construção de classificadores em sistemas de suporte a decisão.

No Capítulo 9 apresentamos uma das técnicas de agregação de classificadores via bootstrap. No Capítulo 10 apresentamos uma avaliação numérica do método de agregação apresentado no Capítulo 9. Avalia-se o comportamento do bagging para os classificadores apresentados anteriormente em relação a quatro comportamentos distintos de separação das populações. Em seguida, o Capítulo 11 é dedicado a uma aplicação a dados reais no suporte ao diagnóstico de câncer de mama. No capítulo final tecemos as conclusões e os comentários.

1.3 Suporte Computacional

As avaliações numéricas, a aplicação e os gráficos foram produzidos utilizando o software `R` (R Development Core Team, 2006) em sua versão 2.3.1 para o sistema operacional `Windows`. O `R` é uma excelente ferramenta e consiste em um ambiente de programação, análise de dados e gráficos. Também possui versão para outros sistemas operacionais (incluindo `linux`) e se encontra disponível gratuitamente em `http://www.R-project.org`. Maiores detalhes sobre o `R` podem ser encontrados em Dalgaard (2002) e Venables & Ripley (2002).

A presente dissertação foi digitada usando o sistema de tipografia LaTeX desenvolvido por Leslie Lamport em 1985, que consiste em uma série de macros ou rotinas do sistema TeX (Knuth, 1986). O LaTeX também é distribuído de forma gratuita.

CAPÍTULO 2

Reconhecimento de Padrões

2.1 Introdução

O reconhecimento de padrões é uma tarefa que nós humanos desempenhamos muito bem. Nós somos capazes de receber informações através dos nossos sentidos e tirar conclusões de maneira rápida, sem consciência do esforço desenvolvido pelo nosso cérebro. Por exemplo, muitos de nós reconhecemos um rosto não visto há muito tempo. Um outro exemplo é a habilidade humana em identificar dígitos e letras, em geral aprendida desde os quatro anos de idade mediante o auxílio de um professor.

Uma característica evidente e de grande importância no reconhecimento de padrões dos humanos é o aprendizado. Na maioria das vezes nós aprendemos a partir de experiências passadas. Para maiores detalhes a respeito do processo de reconhecimento de padrões dos humanos, com ênfase nos mecanismos neurais e do cérebro humano, veja Kandel & Schwartz (1991), Shepherd (1990), Koch & Segev (1989), Kuffler et. al. (1984) e Freeman (1975).

A forma de aprendizado humano inspirou o surgimento de máquinas capazes de desempenhar tarefas de reconhecimento de padrões semelhantes às realizadas pelos humanos. Da mesma forma, o aprendizado de máquina tem como principal característica o apren-

dizado baseado em experiências passadas. Alguns exemplos de tarefas de reconhecimento de padrões que podem ser desempenhadas tanto por humanos como por máquinas, e outras que são puramente tecnológicas podem ser vistas em Ripley (1996).

Mesmo sabendo que nós humanos podemos desempenhar algumas das tarefas de reconhecimento de padrões muito bem, busca-se com a utilização de máquinas formas de reconhecimento de padrões mais precisas, mais baratas ou até mesmo poupar os humanos do trabalho árduo. O alvo do reconhecimento de padrões é tornar claro alguns dos complicados mecanismos de tomada de decisão dos humanos e programar máquinas para reproduzir essas funções (Fukunaga, 1990).

O reconhecimento de padrões tornou-se uma disciplina científica de grande aplicabilidade a partir de 1960, pois desde então era visto apenas como o resultado de pesquisas teóricas na área de Estatística (Theodoridis & Koutroumbas, 2003). Com o avanço tecnológico, houve um aumento na demanda por aplicações práticas. Sua tragetória remonta à aplicações na área militar, onde possui uma longa e respeitável história dentro da engenharia (Ripley, 1996).

Diagnóstico médico, reconhecimento de caracteres escritos a mão, reconhecimento da fala, reconhecimento de peças defeituosas em uma linha de montagem são exemplos de tarefas em reconhecimento de padrões. Um padrão pode ser definido como um conjunto de características, podendo ser extraído a partir de uma imagem, um sinal ou um conjunto de valores. Outros diferentes exemplos de reconhecimento de padrões aplicados a diversas áreas de atividade como agricultura, astronomia, biologia, administração civil, economia, engenharia, geologia, medicina, serviço militar e segurança podem ser vistos em Marques de Sá (2001).

2.2 A Tarefa de Classificação

O reconhecimento de padrões abrange diversas áreas de atividade e é nas tarefas de classificação que se concentram a maioria de suas aplicações (Theodoridis & Koutroumbas, 2003). De forma geral, uma tarefa de classificação consiste em designar um padrão

para uma de duas ou mais categorias (classes ou populações), com base em experiências passadas (exemplos ou conjunto de observações). É considerada uma tarefa de classificação muito evidenciada na área médica, por exemplo, a classificação de um indivíduo como doente ou não-doente.

A ênfase de uma tarefa de classificação é a construção de uma máquina de classificação (classificador) capaz de fazer previsões que auxiliam na tomada de decisão. Um procedimento de classificação é construído com base em um conjunto de observações e o interesse é utilizá-lo para classificar novos padrões.

Como exemplo de tarefa de classificação em reconhecimento de padrões, considere o diagnóstico de câncer de mama através da mamografia. A detecção automatizada de lesões envolve a localização pelo computador de regiões contendo padrões radiológicos suspeitos, porém com a classificação da lesão sendo feita exclusivamente pelo radiologista. Nesse caso, sistemas automáticos de classificação podem funcionar como uma segunda opinião, auxiliando o radiologista na tomada de decisão.

Em geral, a maioria dos sistemas de auxílio ao diagnóstico em mamografia é voltado à detecção de nódulos e microcalcificações. Um exemplo de radiografia da mama com presença de microcalcificação pode ser visualizado na Figura 2.1. As microcalcificações apresentam-se como pequenos grãos de areia polvilhados, indicados por pontos luminosos na figura.

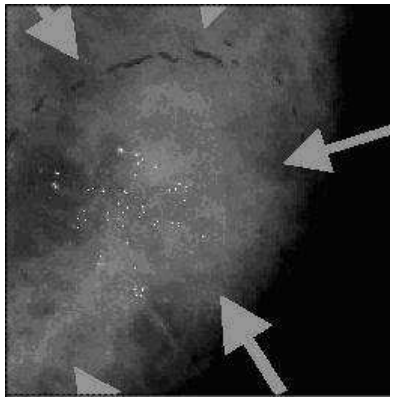


Figura 2.1: Exemplo de uma Radiografia da Mama com microcalcificação.

Em um sistema de reconhecimento de padrões, um classificador pode ser construído a partir de mamografias com lesão suspeita comprovada (tumor malígno ou benígno). Os padrões extraídos dessas mamografias podem constituir indícios para a classificação de novas lesões suspeitas. Usualmente, um sistema utilizado para essa finalidade é chamado de sistema diagnóstico auxiliado por computador ("computer-aided diagnosis - CAD).

O processo que envolve a construção de um classificador consiste, em geral, nas etapas de treinamento e teste. Isso implica na divisão das observações em duas partes independentes: conjunto de treinamento e conjunto de teste. Primeiramente, utilizamos as observações dedicadas ao treinamento para construir o classificador, em seguida o classificador, quando bem projetado, é capaz de designar suficientemente bem (baseado em um critério de avaliação) as observações pertencentes ao conjunto de teste.

Alguns métodos automáticos de reconhecimento de padrões são utilizados com freqüência em tarefas de classificação, dentre eles está a classificação estatística e as redes neurais artificiais. Outros métodos podem ser vistos em Jain et. al. (2000).

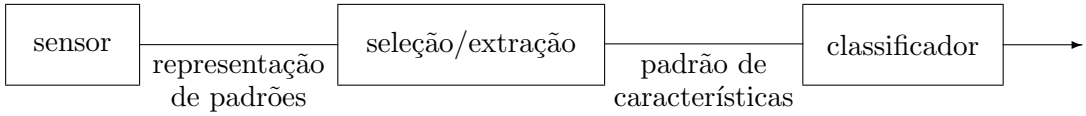
2.2.1 O Processo de Classificação

Em um sistema de reconhecimento de padrões, o processo de classificação baseia-se essencialmente em um conjunto de padrões (atributos ou características), com classes previamente definidas, que sirva de indício para a classificação de novos padrões em uma das demais classes.

Considere um conjunto de observações composto por n vetores de padrões com categorias associadas denotado por $(y_i, \mathbf{x}_i)$, $i = 1, \ldots, n$, onde y_i é uma categoria ($y_i \in \{1, \ldots, g\}$), o padrão $\mathbf{x}_i$ é um vetor p-dimensional de medidas das variáveis $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$. Adicionalmente, considere que n_i observações pertencem a categoria π_i, tal que $\sum_{i=1}^{g} n_i = n$.

Caso não seja possível dispor de um conjunto de observações formado pelos padrões com categorias associadas, pode-se proceder a classificação não-supervisionada, onde as classes são formadas com base nas similaridades dos padrões (Ripley, 1996; Duda et. al.,

Figura 2.2: Procedimento de classificação de padrões.



2001).

Diante do classificador construído, o objetivo passa a ser a designação futura de um novo padrão (padrão ainda não visto pelo classificador) para uma das categorias. Nesse sentido, adotando-se uma regra de decisão que particiona o espaço de medidas em g regiões $\Omega_1, \ldots, \Omega_g$, podemos dizer que se um vetor de padrões pertence a Ω_i, então ele pertence à categoria π_i. Cada região deve ser multiplamente conectada, isto é, ela pode ser composta por diversas regiões conjuntas. Os limites entre as regiões Ω_i são os limites de decisão ou superfícies de decisão.

A Figura 2.2 simplifica a grosso modo o procedimento de classificação de padrões, evidenciando alguns dos estágios a que os dados podem ser submetidos, antecedendo o resultado final. Algumas transformações nos dados são as vezes necessárias e podem ser consideradas em estágios de pré-processamento, seleção de variáveis ou extração de variáveis. Essas transformações atuam nos dados de forma a reduzir a dimensionalidade, removendo informações redundantes ou irrelevantes e, com isso, torna a informação mais apropriada para a classificação.

Ressaltamos que em alguns problemas, medições podem ser feitas diretamente no vetor de características. Nestas situações não existe um estágio de seleção de variáveis automático, essa seleção é feita por um investigador que "conhece" (através de experiências, conhecimento de estudos anteriores relacionados, etc) aquelas variáveis que são importantes para a classificação.

Considere os dados de um estudo de depressão descrito em Afifi & Clark (1996). O conjunto de dados possui 294 observações e corresponde a um estudo epidemiológico de depressão, onde os indivíduos foram entrevistados entre 1979 e 1980. Esse estudo tinha

como objetivo prever estimativas da prevalência e incidência da depressão e identificar fatores e resultados associados com essas condições. Os fatores relacionados incluem fatos cotidianos estressantes, cuidados com a saúde, uso de medicamentos, entre outros.

A Figura 2.3 apresenta o comportamento desses dados para as variáveis idade e renda. Esta figura exibe uma amostra de 50 observações do conjunto de dados original, onde 20 indivíduos são deprimidos e 30 indivíduos são considerados não deprimidos. Utilizaremos estes dados para ilustrar adiante as regiões de classificação para os classificadores abordados neste trabalho.

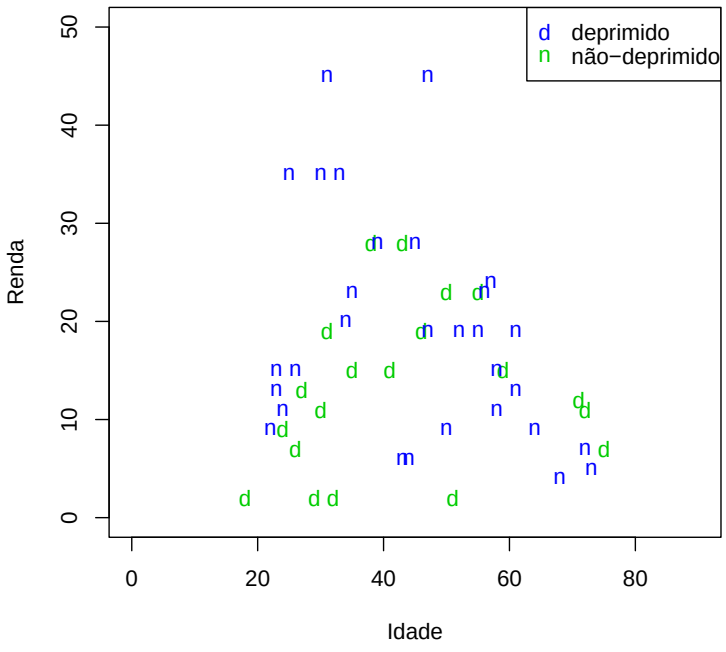


Figura 2.3: Conjunto de dados de depressão referente a 50 observações, segundo idade e renda.

2.3 Classificação Estatística

A ferramenta de reconhecimento de padrões clássica que apresenta uma estrutura mais geral para formular soluções em problemas de reconhecimento de padrões é a Estatística (Bishop, 1995). Essa é a mais antiga das disciplinas utilizadas no reconhecimento de padrões automático a partir de exemplos (Ripley, 1996).

Na classificação estatística, cada padrão é representado em termos de p variáveis ou medidas e é representado como um vetor de características ou atributos $\mathbf{x}$. O objetivo é designar o padrão para uma das g possíveis classes $\pi_1, \ldots, \pi_g$. Dado um conjunto de padrões de treinamento de cada classe, o objetivo da classificação estatística é estabelecer limites de decisão no espaço de características que separem os padrões pertencentes às diferentes categorias.

A classificação estatística é essencialmente baseada no uso de modelos probabilísticos na distribuição do vetor de características das classes com objetivo de derivar funções de classificação. A estimação dessas distribuições é baseada no conjunto de padrões de treinamento, cujas categorias são previamente conhecidas.

Existem variantes dos métodos estatísticos de classificação, que dependem se um modelo paramétrico conhecido está sendo utilizado ou não.

2.4 Redes Neurais

As pesquisas em sistemas baseados no conhecimento, motivadas pelo desenvolvimento de máquinas que inspiram-se no funcionamento do cérebro humano, contribuíram para o surgimento das redes neurais artificiais. Redes neurais artificiais é também uma ferramenta muito utilizada na construção de máquinas de reconhecimento de padrões. Elas são importantes por sua capacidade de generalização e habilidade de lidar com problemas práticos (Ripley, 1996).

As redes neurais artificiais, mais usualmente chamadas de redes neurais, são algumas vezes consideradas como uma extensão das técnicas convencionais de reconhecimento de

padrões estatístico (Bishop, 1995). Adicionalmente, Ripley (1996), Anderson et. al.(1990) e Jain et. al. (2000) também discutem esta relação. Anderson et. al. (1990) apontam que "neural networks are statistics for amateurs... Most NNs conceal the statistics from the user".

A família de redes neurais mais utilizada em tarefa de reconhecimento de padrões é a rede de uma direção (feedforward), que inclui o perceptron de múltiplas camadas (multilayer perceptron) e Funções de bases radiais (Radial-Basis Function). Essas redes são organizadas em camadas conectadas em um mesmo sentido. Outra rede popular é o mapa auto organizável (Self-Organizing Map), ou Rede Kohonen (Kohonen-Network) que é utilizada principalmente no agrupamento de dados (clustering). Contudo, a presente dissertação aborda apenas a rede perceptron de múltiplas camadas. Maiores detalhes a respeito das demais redes podem ser encontrados em Haykin (2001).

Em tarefas de classificação as redes neurais oferecem algumas vantagens tais como, técnicas unificadas e procedimentos flexíveis para encontrar soluções não-lineares (Ripley, 1996). Entretanto, o processo de aprendizado da rede requer, em geral, um alto custo computacional e seus parâmetros não são interpretáveis.

2.5 Combinação de Classificadores

O rápido crescimento tecnológico e o aumento na disponibilidade de computadores mais potentes têm facilitado o uso de diversos métodos mais elaborados de análise e classificação de dados, requeridos principalmente pelo aumento na demanda por aplicações que exigem análises cada vez mais rápidas, precisas e com um baixo custo. Estudos relatam que combinação de classificadores apresentam melhores resultados (em termos de precisão) comparados a resultados obtidos a partir de um único classificador.

Métodos de combinação funcionam, por exemplo, explorando diferentes conjuntos de características, diferentes conjuntos de treinamento, diferentes métodos ou sessões de treinamento, todos resultando em um conjunto de classificadores cujas saídas devem ser combinadas, com o objetivo de melhorar a acurácia global.

A utilização de combinação de classificadores se baseia, essencialmente, no fato de que classificadores individuais podem ter seu desempenho melhorado. Existem muitas razões para combinar múltiplos classificadores para resolver um problema de classificação. Algumas dessas razões são:

1. Devido a diversidade de classificadores, cada um desenvolvido em diferentes contextos e uma representação/descrição inteiramente diferente do mesmo problema. Um exemplo é a identificação de pessoas pela sua voz, rosto, assim como escrita à mão.

2. Quando, em algumas casos mais de um único conjunto de treinamento é utilizado, cada um coletado em momentos diferentes ou em diferentes ambientes. Esses conjuntos de treinamento podem até mesmo utilizar diferentes características.

3. Diferentes classificadores treinados em um mesmo conjunto de dados não devem apenas diferir no seu desempenho global, mas eles também devem mostrar grandes diferenças locais. Cada classificador deve ter sua própria região no espaço de características onde seu desempenho é melhor.

4. Alguns classificadores tais como redes neurais mostram diferentes resultados com diferentes inicializações devido a aleatoriedade ligada ao processo de treinamento. Ao invés de selecionar a melhor rede e descartar as outras, combina-se várias redes, tirando vantagens em todas as tentativas de aprendizado a partir dos dados.

Um esquema de combinação típico consiste de um conjunto de classificadores individuais e um método de combinar que reúne os resultados dos classificadores individuais para tomar a decisão final.

Em geral, na seleção e treinamento de classificadores individuais utiliza-se várias técnicas como repetições e bootstrap para tornar os classificadores um pouco diferenciados, pois é aconselhável a utilização de combinação de classificadores, quando estes são independentes.

Bagging (Breiman, 1996) e Boosting (Freund & Schapire, 1996) são métodos mais usuais de combinação de classificadores. No bagging, diferentes conjuntos de treinamento

são criados por versões bootstrap do conjunto de treinamento original e combinados por maioria de votos, no caso de tarefas de classificação. Boosting é uma técnica utilizada para geração de uma seqüência de conjuntos de treinamento, consiste em atribuir pesos a cada padrão de treinamento, evidenciando sua importância no processo de classificação, e construindo um classificador utilizando uma amostra de treinamento ponderada.

2.6 Avaliação do Desempenho de Classificadores

Existem várias formas de medir o desempenho de classificadores. A forma mais comum utilizada é através da taxa de erro. A taxa de erro consiste no percentual de observações erradamente classificadas com base em um número total de observações.

Na prática, a razão de erro de um sistema de reconhecimento de padrões pode ser estimada utilizando todas as observações da amostra, que são divididas em conjunto de treinamento e conjunto de teste. O classificador é treinado com base na amostra de treinamento, e então ele é avaliado na amostra de teste. O percentual de padrões da amostra de teste erradamente classificados é a estimativa da razão de erro. Outra opção é considerar, além do conjunto de treinamento e de teste, um conjunto de validação, porém esse procedimento é limitado a tamanhos amostrais grandes.

Para determinar o conjunto de treinamento e o conjunto de teste, alguns procedimentos são freqüentemente adotados na prática. Um desses procedimentos é separar o conjunto de dados em treinamento e teste por repetidas vezes, nesse caso considera-se a média da taxa de erro ao final do processo. Essa metodologia foi adotada, por exemplo, por Breiman (1996).

Outra metodologia que pode ser utilizada na investigação de desempenho de classificadores é a validação cruzada. Nesse caso, o conjunto de dados é dividido em v subconjuntos de tamanhos aproximadamente iguais, onde para treinamento do classificador utiliza-se $v-1$ destes, em seguida testa-se no conjunto restante. Esse processo é repetido v vezes, até que todos os v subconjuntos sejam utilizados para teste. Esse procedimento é muito utilizado quando há escassez de exemplos de treinamento.

É comum utilizar na avaliação de desempenho em classificação diagnóstica algumas medidas descritivas como sensibilidade e especificidade, além da curva ROC (receiver-operating characteristic). A curva ROC é uma representação gráfica bi-dimensional da sensibilidade no eixo Y contra a taxa (1 - especificidade) no eixo X ao longo de uma faixa de ponto de corte (Centor, 1991). Por exemplo, indivíduos com mensurações menores ou iguais ao ponto de corte seriam classificados como não-doentes, caso contrário seriam classificados como doentes ou vice-versa. Esses procedimentos são exclusivos para problemas de classificação binária. Outras medidas podem ser consideradas para casos de múltiplas categorias (Webb, 2002).

Quando comparados a classificação resultante do classificador construído com a classificação tida como verdadeira, correspondente ao conjunto de dados previamente categorizado, quatro situações são possíveis para o caso de tarefas de classificação binária (doente e não-doente, por exemplo):

1. Sendo o resultado do classificador sob investigação um verdadeiro-positivo (VP), isto indica que o paciente foi classificado como portador da doença, quando na verdade ele está doente.

2. Sendo o resultado do classificador sob investigação um falso-positivo (FP), isto indica que apesar da resposta do classificador ser positiva quanto a presença da doença, o paciente não está doente.

3. Sendo o resultado do classificador sob investigação um falso-negativo (FN), isto indica que apesar da resposta do classificador ser negativa quanto a presença da doença, o paciente está doente.

4. Sendo o resultado do classificador sob investigação um verdadeiro-negativo (VN), isto indica que o paciente foi classificado como não-doente, quando na verdade ele não está doente.

A matriz resultante da comparação do classificador sob investigação com a classificação previamente definida, pode ser vista na Tabela 2.1.

Tabela 2.1: Matriz de confusão para classificadores que produzem respostas dicotomizadas.

Resultado do classificador	Valores observados na amostra de treinamento	
	Positivo (doente)	Negativo (não-doente)
Positivo	Verdadeiro-positivo (VP)	Falso-positivo (FP)
Negativo	Falso-negativo (FN)	Verdadeiro-negativo (VN)

Neste trabalho, utilizamos duas metodologias como forma de avaliação. Na simulação, calculamos as médias das taxas de erro e_R e e_B, ambas produzidas mediante repetidas gerações aleatórias dos dados em treinamento e teste e, utilizamos o erro ótimo de Bayes como parâmetro de referência para uma avaliação individual dos classificadores. Para os dados reais, também calculamos as médias das taxas de erro e_R e e_B, nesse caso resultantes de repetidas divisões aleatórias do conjunto de dados em treinamento e teste, além das medidas de sensibilidade e especificidade. Sendo e_R o erro simples e e_B o erro bagging. Maiores detalhes sobre as metodologias adotadas ver Capítulos 10 e 11.

Taxa de erro

A taxa de erro constitui a forma mais usual de avaliação de desempenho em tarefas de classificação. Para a matriz de resultados apresentada (2.1), a taxa de erro pode ser obtida através da seguinte expressão:

$$e = \frac{FN + FP}{VP + FN + FP + VN}.$$

Sensibilidade e especificidade

Considerando que uma dada população possa ser dividida em duas categorias de interesse, digamos doentes e não-doentes, a sensibilidade de um modelo pode ser definida como a fração dos pacientes doentes que o modelo é capaz de detectar e, a especificidade

é a proporção de pacientes não-doentes que o modelo é capaz de detectar.

Sendo n_1 o número de indivíduos portadores da doença e n_2 o número de indivíduos sadios, a sensibilidade (S) é estimada pela razão

$$S = \frac{VP}{n_1}$$

e a especificidade (E) é estimada por

$$E = \frac{VN}{n_2}.$$

CAPÍTULO 3

Teoria Estatística Básica de Reconhecimento de Padrões

3.1 Introdução

Duas técnicas básicas utilizadas em classificação são apresentadas neste capítulo. A primeira requer o conhecimento das densidades condicionais das categorias. Como na prática desconhecemos tal função, podemos utilizar valores estimados obtidos a partir de uma amostra com categorias associadas. Os Capítulos 4 e 5 descrevem técnicas para estimação das densidades condicionais das categorias.

A segunda técnica deriva regras de decisão que utilizam os dados para estimar regiões de decisão, sem considerar a metodologia utilizada para obtenção das densidades condicionais das categorias. Esta técnica é considerada nos Capítulos 6, 7 e 8.

3.2 Teoria da Decisão de Bayes

Aqui consideramos uma teoria de decisão elementar que se baseia no conhecimento da função densidade de probabilidade condicional de cada categoria. A partir dessa teoria, uma regra de discriminação pode ser construída com a estimação das funções de densidades condicionais das categorias e o uso do teorema de Bayes. Pode-se considerar um modelo

paramétrico para as funções de densidade e estimar os parâmetros do modelo usando um conjunto de treinamento disponível. Métodos não-paramétricos de estimação de densidade também podem ser adotados, uma técnica a ser considerada pode ser vista no Capítulo 5.

Considere um problema de classificação que consiste em classificar um dado padrão $\mathbf{x}$ em uma de duas categorias (π_1 e π_2). Para tanto assumimos como conhecidas as probabilidades a priori $p(\pi_1)$ e $p(\pi_2)$, e a função densidade de probabilidade condicional das categorias $p(\mathbf{x}/\pi_i)$, $i = 1, 2$, que descreve a distribuição dos vetores de características de cada uma das categorias. Recorrendo ao teorema de Bayes,

$$p(\pi_i/\mathbf{x}) = \frac{p(\mathbf{x}/\pi_i)p(\pi_i)}{p(\mathbf{x})}, \tag{3.1}$$

obtemos a probabilidade a posteriori para cada categoria, onde $p(\mathbf{x}) = \sum_{i=1}^{2} p(\mathbf{x}/\pi_i)p(\pi_i)$.

Um dos critérios adotados na busca de uma regra ótima de classificação é a minimização do erro. Na busca pela minimização do erro em tarefas de classificação, a regra de Bayes consiste em designar $\mathbf{x}$ para uma das duas categorias segundo a regra de classificação:

$$\text{se } p(\pi_1/\mathbf{x}) > p(\pi_2/\mathbf{x}), \mathbf{x} \text{ é classificado como } \pi_1;$$
$$\text{se } p(\pi_1/\mathbf{x}) < p(\pi_2/\mathbf{x}), \mathbf{x} \text{ é classificado como } \pi_2.$$

No caso de igualdade de probabilidades o padrão $\mathbf{x}$ pode ser designado para uma das categorias indefinidamente. Diante do teorema de Bayes apresentado em (3.1), as probabilidades a posteriori $p(\pi_1/\mathbf{x})$ e $p(\pi_2/\mathbf{x})$ da regra de classificação apresentada, podem ser substituídas por $p(\mathbf{x}/\pi_1)p(\pi_1)$ e $p(\mathbf{x}/\pi_2)p(\pi_2)$, respectivamente

Em particular, quando a probabilidade a priori é igual para ambas as categorias a regra de decisão torna-se:

$$\text{se } p(\mathbf{x}/\pi_1) > p(\mathbf{x}/\pi_2), \mathbf{x} \text{ é classificado como } \pi_1;$$
$$\text{se } p(\mathbf{x}/\pi_1) < p(\mathbf{x}/\pi_2), \mathbf{x} \text{ é classificado como } \pi_2.$$

Maiores detalhes da minimização do erro de uma má classificação podem ser vistos em Webb(2002).

Porém, nem sempre a probabilidade de erro de uma má classificação é o melhor critério de minimização a ser adotado, porque é dado igual importância a todos os erros. No entanto, existem casos em que alguns erros devem ter implicações mais sérias que outros. Para esses casos podemos levar em consideração a associação de custos, representando uma perda advinda de uma decisão errada, que consiste em, por exemplo, classificar um padrão $\mathbf{x}$ como pertencente à categoria π_1 dado que ele verdadeiramente pertence à categoria π_2. Esses custos são associados às probabilidades a posteriori, com ponderações mais altas para os casos cujas implicações são mais sérias.

Denote por $c(1/2)$ o custo referente ao erro em designar o padrão $\mathbf{x}$ para a categoria π_1 dado que ele pertence verdadeiramente à categoria π_2 e $c(2/1)$ o custo referente ao erro em designar este padrão para a categoria π_2 dado que ele pertence verdadeiramente à categoria π_1. Ao considerarmos que o erro na classificação de padrões pertencentes a classe π_1 tem conseqüências mais sérias do que o erro em classificar erradamente os padrões que pertencem à classe π_2, devemos escolher $c(2/1) > c(1/2)$. Neste caso os padrões são designados para a classe π_2 se

$$p(\mathbf{x}/\pi_2)p(\pi_2)c(2/1) > p(\mathbf{x}/\pi_1)p(\pi_1)c(1/2),$$

que considerando $p(\pi_1) = p(\pi_2) = 1/2$, temos

$$p(\mathbf{x}/\pi_2) > p(\mathbf{x}/\pi_1)\frac{c(1/2)}{c(2/1)}.$$

Na prática, o processo de associação de custo é muito difícil, pois, em algumas situações, os custos podem ser representados por diferentes fatores, medidos em diferentes unidades (monetária, temporal, etc). Esses custos, conseqüentemente, são vistos como uma opinião subjetiva de um perito.

Apesar de na prática não ser possível se conhecer a distribuição dos dados, a regra de Bayes é muito importante, pois, além de dar uma contribuição teórica ao desenvolvimento

de classificadores, ela é útil na comparação de classificadores em dados simulados, por fornecer uma base de referência, com uma taxa de erro que é a princípio a menor atingível (Ripley, 1996).

3.3 Funções Discriminantes e Fronteiras de Decisão

Uma função discriminante é uma função do padrão $\mathbf{x}$ que conduz a uma regra de classificação. No caso de existirem g categorias, são definidas g funções discriminantes. Cada função discriminante produz um escore discriminante, cuja classificação do padrão $\mathbf{x}$ se dá de acordo com esse escore. O maior escore indica a qual classe o padrão pertence, baseado na seguinte estrutura de decisão para um problema de duas categorias:

$$h(\mathbf{x}) > k, \Rightarrow \mathbf{x} \in \pi_1;$$

$$h(\mathbf{x}) < k, \Rightarrow \mathbf{x} \in \pi_2,$$

sendo k uma constante e $h(\cdot)$ uma função discriminante. No caso de empate, em que $h(\mathbf{x}) = k$, o padrão é designado arbitrariamente a uma das duas classes.

Muitas diferentes formas de funções discriminantes têm sido consideradas na literatura. Elas variam quanto a complexidade da função discriminante. A função $h(\cdot)$ pode ser expressa por uma combinação linear de $\mathbf{x}$ ou até mesmo por uma função multiparamétrica não-linear tal como o perceptron de múltiplas camadas.

Particionando o espaço de características em regiões, em uma tarefa com $g = 2$ classes, as regiões Ω_i e Ω_j são separadas por uma fronteira de decisão no espaço de características multidimensional. No caso de minimização do erro a fronteira de decisão é dada por

$$p(\pi_i/\mathbf{x}) - p(\pi_j/\mathbf{x}) = 0.$$

De um dos lados da fronteira essa diferença é positiva e do outro lado ela é negativa.

Pode ser mais conveniente, algumas vezes, trabalhar com funções discriminantes equivalentes, por exemplo,

$$g_i(\mathbf{x}) = f\{p(\pi_i/\mathbf{x})\},$$

onde $f(\cdot)$ é uma função monotonicamente crescente.

Dado um padrão a ser designado para uma das classes, o critério de decisão consiste em atribuir o padrão à classe correspondente cujo valor da função é maior, ou seja, classifica-se $\mathbf{x}$ em π_i se

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \forall i \neq j.$$

Neste caso o limite de decisão que separa regiões contínuas é descrito por

$$g_i(\mathbf{x}) - g_j(\mathbf{x}) = 0, \ i, j = 1, 2, \ldots, g, \ i \neq j.$$

No Capítulo 4 focalizamos uma família particular de regiões de decisão para o caso específico em que a função densidade é Gaussiana.

CAPÍTULO 4

Análise Discriminante Quadrática

4.1 Introdução

Para este caso específico, nós assumimos que a função de probabilidade da classe π_i $(i = 1, \ldots, g)$ com relação ao padrão $\mathbf{x}$ no espaço de características p-dimensional tem distribuição normal multivarida,

$$p(\mathbf{x}/\pi_i) = (2\pi)^{-p/2}|\Sigma_i|^{-1/2} \exp\left\{\frac{1}{2}(\mathbf{x} - \mu_\mathbf{i})^T \Sigma_i^{-1}(\mathbf{x} - \mu_\mathbf{i})\right\}, \tag{4.1}$$

com média $\mu_i = E(\mathbf{x})$ e matriz de covariância $p \times p$ definida como

$$\Sigma_i = E[(\mathbf{x} - \mu_\mathbf{i})(\mathbf{x} - \mu_\mathbf{i})^T]$$

e $|\Sigma_i|$ indicando o determinante de Σ_i.

Essa é uma das funções densidade mais comuns encontradas na prática. A razão de sua popularidade é em grande parte devido à atratividade computacional que exerce e por seus modelos serem adequados a um grande número de casos (Theodoridis & Koutroumbas, 2003). Assumindo como conhecidas as funções densidade condicionais das classes, agora nos resta estimar os parâmetros com base em uma amostra de treinamento e derivar

limites de decisão que particionam o espaço de características em regiões. Para tanto podemos fazer uso da regra de Bayes de erro mínimo, apresentada no capítulo anterior.

Os limites de decisão são apresentados em forma de funções discriminantes. Eles podem ser caracterizados por formas lineares ou quadráticas, a depender da matriz de covariância dos dados, se ela for comum a todas as classes a função discriminante é linear, caso contrário a função discriminante é quadrática.

4.2 Função Discriminante Quadrática

Relembrando a regra de Bayes de erro mínimo, ao desejar classificar um vetor de padrões $\mathbf{x}$ em uma de g de classes ($\pi_i,\ i = 1, \ldots, g$), o padrão deverá ser designado para a classe no qual a probabilidade a posteriori $p(\pi_i/\mathbf{x})$ é máxima, ou equivalentemente $\log\{p(\pi_i/\mathbf{x})\}$ é máximo.

O processo inicial de classificação consiste na obtenção das probabilidades a posteriori das classes, algumas vezes tida como escore discriminante. De acordo com a regra de Bayes a probabilidade a posteriori é dada por

$$p(\pi_i/\mathbf{x}) = \frac{p(\mathbf{x}/\pi_i)p(\pi_i)}{p(\mathbf{x})}.$$

Cuja regra de classificação consiste em designar $\mathbf{x_0}$ para π_i se

$$\log\{p(\pi_i)p(\mathbf{x_0}/\pi_i)\} > \log\{p(\pi_j)p(\mathbf{x_0}/\pi_j)\}, \quad \forall i \neq j.$$

Os escores discriminante são obtidos por

$$g_i(\mathbf{x_0}) = \log(p(\pi_i/\mathbf{x_0})) = \log(p(\mathbf{x_0}/\pi_i)p(\pi_i)), \tag{4.2}$$

que substituindo a função densidade condicional normal p-variada (4.1) em (4.2), resulta em

$$g_i(\mathbf{x_0}) = -\frac{1}{2}(\mathbf{x_0} - \mu_\mathbf{i})^T\Sigma_i^{-1}(\mathbf{x_0} - \mu_\mathbf{i}) - \frac{1}{2}\log(|\Sigma_i|) - \frac{p}{2}\log(2\pi) + \log\{p(\pi_i))\}. \tag{4.3}$$

A regra de classificação a ser considerada consiste em designar $\mathbf{x_0}$ para π_i se

$$g_i(\mathbf{x_0}) > g_j(\mathbf{x_0}) \quad \forall i \neq j.$$

Baseado em uma amostra de treinamento, as quantidades μ_i e Σ_i são substituídas em (4.3) por valores estimados. Os valores estimados podem ser obtidos por máxima verossimilhança. Os estimadores de máxima verossimilhança de μ_i e Σ_i são, respectivamente,

$$\bar{\mathbf{x}}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} \mathbf{x}_{ik},$$

$$S_i = \frac{1}{n_i} \sum_{k=1}^{n_i} (\mathbf{x}_{ik} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ik} - \bar{\mathbf{x}}_i)^T.$$

O classificador Gaussiano ou regra de discriminação quadrática consiste em designar $\mathbf{x_0}$ para π_i se $g_i > g_j$, para todo $i \neq j$, onde

$$g_i(\mathbf{x_0}) = \log\{p(\pi_i)\} - \frac{1}{2}\log(|S_i|) - \frac{1}{2}(\mathbf{x_0} - \bar{\mathbf{x}}_i)^T S_i^{-1} (\mathbf{x_0} - \bar{\mathbf{x}}_i). \tag{4.4}$$

Se os dados de treinamento foram obtidos mediante amostragem das classes, então uma estimativa da probabilidade a priori $p(\pi_i)$ é n_i/n.

A Figura 4.1 apresenta as regiões de classificação da análise discriminante quadrática para os dados de depressão descritos na Seção 2.2.1.

Considerando que as matrizes de covariâncias são iguais para as g categorias, ou seja $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_g = \Sigma$, a expressão (4.4) define a análise discriminante linear. Maiores detalhes deste método podem ser vistos em Johnson & Wichern (1998).

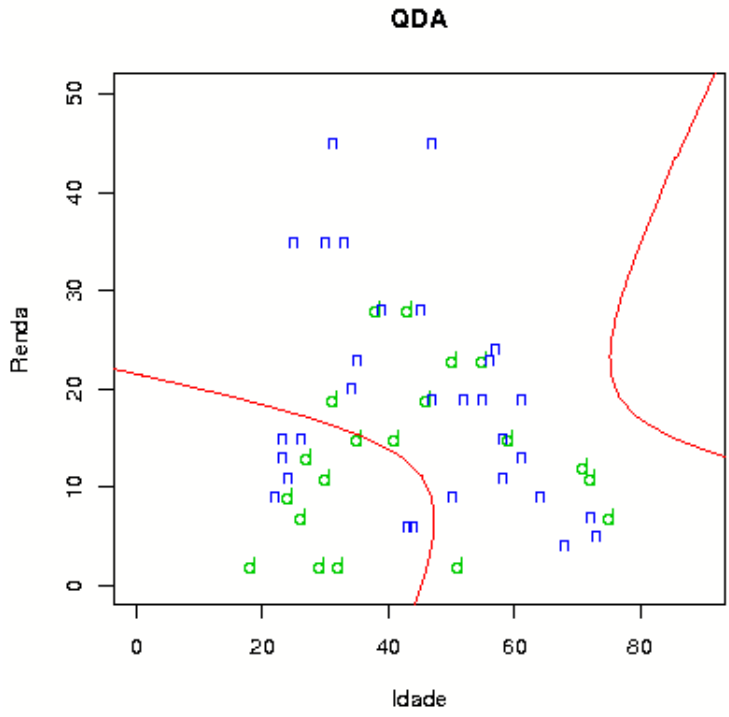


Figura 4.1: Regiões de classificação da função discriminante quadrática para os dados de depressão.

CAPÍTULO 5

K-Vizinhos mais Próximos

5.1 Introdução

Em alguns métodos estatísticos de classificação é necessário conhecer as densidades condicionais das classes. No capítulo anterior, apresentamos um desses métodos (análise discriminante quadrática) caracterizado por estimar a densidade de forma paramétrica. O processo de estimação apresentado parte do pressuposto que a forma funcional específica para o modelo de densidade é Gaussiana. Em seguida, estima-se os parâmetros com base na amostra de treinamento e aplica-se a regra de Bayes. Outro método que requer o conhecimento das densidades condicionais das classes para posterior aplicação da regra de Bayes é o método dos k-vizinhos mais próximos, em inglês k-nearest neighbour (k-nn). Esse é um método não-paramétrico de estimação de densidade que é utilizado, em geral, caso não seja possível assumir uma forma funcional específica para o modelo de densidade. Alguns autores consideram esse o método mais simples dentre as outras técnicas estatísticas, conceitualmente falando (Webb, 2002).

5.2 O Método k-nn

Considerando as técnicas estatísticas de estimação de densidade, onde a probabilidade P de um padrão $\mathbf{x}$, descrito por uma função de densidade $p(\mathbf{x})$, estar situado em uma região $\mathcal{R}$ do espaço de características é, por definição,

$$P = \int_{\mathcal{R}} p(\mathbf{x})\ d\mathbf{x}. \tag{5.1}$$

Suponha que em uma amostra de tamanho n, as observações sejam independentes e identicamente distribuidas de acordo com a lei de probabilidade $p(\mathbf{x})$, então a probabilidade de que k dessas n observações estejam situadas em $\mathcal{R}$ é dada pela lei binomial

$$p_k = \binom{n}{k} p^k (1-p)^{n-k},$$

onde $\mathrm{E}(k) = np$ e $\mathrm{Var}(k) = np(1-p)$.

Sendo k/n a proporção de pontos que atingiram a região $\mathcal{R}$. A média dessa proporção é dada por

$$\mathrm{E}\left(\frac{k}{n}\right) = p$$

e a variância é

$$\mathrm{Var}\left(\frac{k}{n}\right) = \frac{p(1-p)}{n}.$$

Note que k/n é um estimador não viesado e assintoticamente consistente para p (Webb, 2002). Seja k o número de observações da amostra pertencentes a um volume V contido em $\mathcal{R}$ (k é função de $\mathbf{x}$), espera-se que a razão k/n seja uma boa aproximação da probabilidade p, ou seja,

$$p \simeq \frac{k}{n}. \tag{5.2}$$

Sendo $p(\mathbf{x})$ contínua e não havendo variabilidade expressiva sobre a região $\mathcal{R}$, ou seja, para uma pequena região $\mathcal{R}$, pode-se aproximar (5.1) por

$$\int_{\mathcal{R}} p(\mathbf{x})\ d\mathbf{x} \simeq p(\mathbf{x}')\ V, \tag{5.3}$$

onde V é o volume de $\mathcal{R}$ e $\mathbf{x}'$ é um ponto em $\mathcal{R}$.

Relacionando as equações (5.3) e (5.2), pode-se aproximar a densidade $p(\mathbf{x}')$ por

$$\hat{p}(\mathbf{x}') \simeq \frac{k}{nV}. \tag{5.4}$$

Este resultado é o valor estimado da densidade, que poderá ser utilizado na tomada de decisão em problemas de classificação, a partir de uma amostra de dados pré-categorizados.

Aplicando-se o resultado (5.4) em problemas práticos de estimação de densidade existem duas decisões importantes a tomar. A primeira consiste em determinar o valor de k e a outra consiste em determinar o volume V.

As propriedades da estimativa de densidade não precisam necessariamente serem satisfeitas. Note que, por exemplo, a integral da densidade estimada através dos k-vizinhos mais próximos diverge (Duda et. al., 2001).

5.3 Regra de Classificação

Ao utilizar o estimador obtido a partir do resultado intuitivo em (5.4) é conveniente antes proceder a escolha do valor de k e, conseqüentemente, do volume V. A probabilidade k/n é fixada ou, equivalentemente, para uma amostra de tamanho n, fixa-se k e determina-se o volume V que contém as k observações amostrais centradas no ponto $\mathbf{x}'$.

A técnica k-nn, quando utilizada na construção de classificadores, utiliza-se do teorema de Bayes para modelar as densidades condicionais das classes em combinação com as respectivas prioris na obtenção de modelos para as probabilidades a posterioris, que são usadas na tomada de decisão em problemas de classificação.

Obtendo uma estimativa para a densidade, pode-se então derivar uma regra de decisão. Suponha que uma esfera de volume V contenha k_i observações na categoria π_i (tal que $\sum_{i=1}^{g} k_i = k$). Assim, podemos estimar as densidades condicionais das categorias através de

$$\hat{p}(\mathbf{x}/\pi_i) = \frac{k_i}{n_i V}.$$

A densidade não-condicional pode ser estimada por

$$\hat{p}(\mathbf{x}) = \frac{k}{nV},$$

e a probabilidade a priori por

$$\hat{p}(\pi_i) = \frac{n_i}{n}.$$

Com isso, a regra de decisão é designar $\mathbf{x_0}$ para π_i se

$$\hat{p}(\pi_i|\mathbf{x_0}) \geq \hat{p}(\pi_j|\mathbf{x_0}), \forall\ i \neq j.$$

E, utilizando o teorema de Bayes,

$$\frac{k_i}{n_i V}\frac{n_i}{n} \geq \frac{k_j}{n_j V}\frac{n_j}{n}.$$

Assim, aloca-se $\mathbf{x_0}$ em π_i se

$$k_i \geq k_j,\ \forall\ i \neq j.$$

Ou seja, para minimizar o erro de uma má classificação, a regra de decisão consiste em atribuir $\mathbf{x_0}$ a categoria que possui o maior valor correspondente a razão k_i/k, portanto à categoria que possui o maior número de vizinhos dentre os k mais próximos.

A Figura 5.1 apresenta as regiões de classificação do classificador k-nn com diferentes valores de k para os dados de depressão.

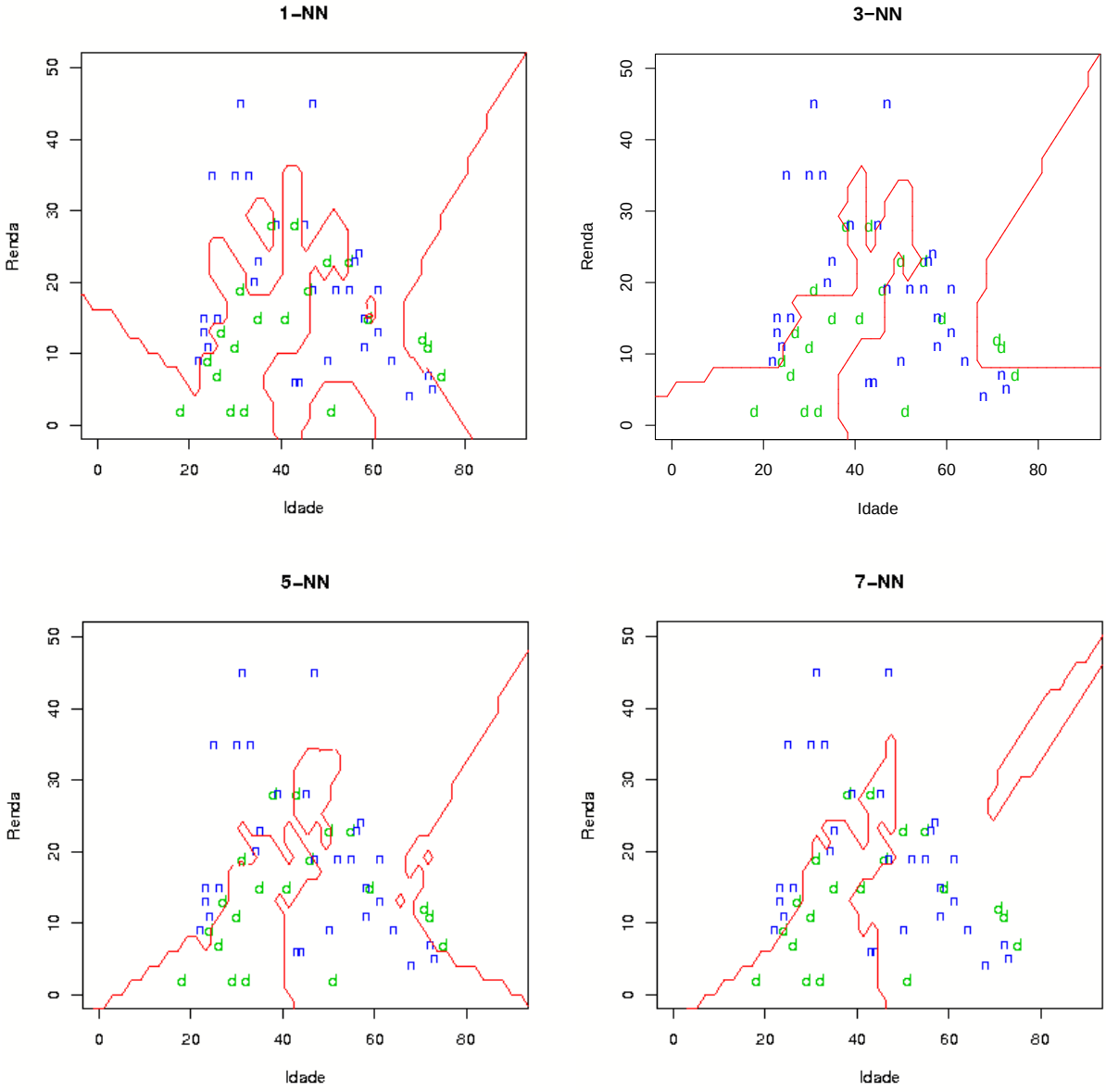


Figura 5.1: Regiões de classificação para os k-vizinhos mais próximos aplicado aos dados de depressão.

A métrica de distância mais freqüentemente utilizada na prática é a distância euclidiana:

$$D(\mathbf{x}, \mathbf{x}_0) = \sqrt{(\mathbf{x} - \mathbf{x}_0)^2}.$$

No contexto de classificação, essa métrica consiste na distância euclidiana entre o ponto $\mathbf{x_0}$ a ser classificado e as demais observações do conjunto de treinamento. Com isso, o processo de classificação consiste em designar $\mathbf{x_0}$ à classe mais freqüente dentre os k-vizinhos mais próximos.

Baseado no conceito de que os parâmetros livres exercem uma certa influência sobre a natureza da estimativa de densidade é de fundamental importância a realização de um procedimento para determinar o valor ótimo de k, para um dado problema de classificação. Este procedimento consiste, em geral, na realização de experimentos, variando o valor de k.

Como não há uma regra explícita para escolha do valor de k, opta-se por escolher o que produzir melhor desempenho na classificação, associado a um menor custo possível. Quanto menor for o valor do k menos tempo computacional é requerido na classificação. Em geral, adota-se um valor ímpar na experimentação, em casos de duas classes, pois assim evita-se casos de empate na hora de designar um novo padrão para uma das duas classes.

CAPÍTULO 6

Análise Discriminante Linear de Fisher

6.1 Introdução

Em comparação com as regras anteriormente citadas, baseadas na regra de decisão bayesiana, na análise discriminante linear de Fisher ao invés de fazermos suposições sobre $p(\mathbf{x}/\pi_i)$, nós fazemos suposições sobre a forma da função discriminante. A função discriminante de Fisher é um exemplo de função discriminante linear, assim como o perceptron em redes neurais. Outro exemplo de função discriminante linear é a função discriminante logística, que corresponde a um modelo linear generalizado, apresentada no capítulo seguinte. Um exemplo de um modelo não-linear é apresentado no Capítulo 8, com a rede perceptron de múltiplas camadas.

Funções discriminantes que são lineares nas características são construídas, tendo por resultado limites lineares de decisão, que utilizando diferentes esquemas de otimização pode dar origem a métodos como a função discriminante linear de Fisher, a ser abordada em seguida.

6.2 Discriminante Linear de Fisher Para Duas Categorias

A análise discriminante proposta por Fisher (1938), transforma observações multivariadas $\mathbf{x}$ em observações univariadas y, tal que os y's derivados das categorias π_i, com $i = 1, \ldots, g$, sejam separados ao máximo. Assim, sua idéia central é reduzir o espaço p-dimensional de variáveis em um espaço de atributos de menor dimensão, de tal forma a separar o máximo possível as categorias. Para tanto, esse método baseia-se em combinações lineares de $\mathbf{x}$ para criar y's. Pode ser utilizado até mesmo se a normalidade multivariada não for aceitável. Porém, o método supõe que as matrizes de covariâncias populacionais sejam iguais, e faz uso de uma matriz de covariância conjunta estimada.

A combinação linear dos $\mathbf{x}$'s assume os valores de $y_{11}, y_{12}, ..., y_{1n_1}$ para as observações da primeira população e os valores $y_{21}, y_{22}, ..., y_{2n_2}$ para as observações da segunda população. A separação desses dois conjuntos de y's univariados é assegurada em termos da diferença entre as médias amostrais das categorias expressas em unidades de desvio padrão, ou seja

$$\frac{|\bar{y}_1 - \bar{y}_2|}{s_y},$$

onde

$$s_y^2 = \frac{\sum_{j=1}^{n_1}(y_{1j} - \bar{y}_1)^2 + \sum_{j=1}^{n_2}(y_{2j} - \bar{y}_2)^2}{n_1 + n_2 - 2}$$

é a estimativa conjunta da variância. O objetivo é selecionar uma combinação linear dos $\mathbf{x}'s$, pelas médias amostrais $\bar{y}_1$ e $\bar{y}_2$, que forneça a separação máxima. A combinação linear, $\hat{y} = \hat{\mathbf{a}}^T\mathbf{x}$, com $\hat{\mathbf{a}}^T = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}^{-1}\mathbf{x}$, maximiza a razão $\frac{(\bar{y}_1 - \bar{y}_2)^2}{s_y^2}$ (Johnson & Wichern, 1998, Cap. 11, pag. 662), onde

$$\mathbf{S} = \left[\frac{n_1 - 1}{(n_1 - 1) + (n_2 - 1)}\right]\mathbf{S}_1 + \left[\frac{n_2 - 1}{(n_1 - 1) + (n_2 - 1)}\right]\mathbf{S}_2,$$

e

$$\mathbf{S}_1 = \frac{1}{n_1 - 1}\sum_{j=1}^{n_1}(\mathbf{x}_{1j} - \bar{\mathbf{x}}_1)(\mathbf{x}_{1j} - \bar{\mathbf{x}}_1)^T, \qquad \mathbf{S}_2 = \frac{1}{n_2 - 1}\sum_{j=1}^{n_2}(\mathbf{x}_{2j} - \bar{\mathbf{x}}_2)(\mathbf{x}_{2j} - \bar{\mathbf{x}}_2)^T$$

com $\bar{\mathbf{x}}_1 = \frac{1}{n_1}\sum_{j=1}^{n_2}\mathbf{x}_{1j}$ e $\bar{\mathbf{x}}_2 = \frac{1}{n_2}\sum_{j=1}^{n_2}\mathbf{x}_{2j}$. Adicionalmente, para todos os valores possíveis de $\hat{\mathbf{a}}$, o máximo da razão $\frac{(\bar{y}_1 - \bar{y}_2)^2}{s_y^2}$ é

$$\mathbf{D}^2 = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2).$$

A solução de Fisher para problemas de separação pode também ser usada para classificar novas observações.

Inicialmente, devemos calcular o ponto médio:

$$\hat{\mathbf{m}} = \frac{\bar{y}_1 + \bar{y}_2}{2}.$$

A partir daí, sabendo que $\bar{y}_1 = \hat{\mathbf{a}}^T\bar{\mathbf{x}}_1$ e $\bar{y}_2 = \hat{\mathbf{a}}^T\bar{\mathbf{x}}_2$, temos

$$\hat{\mathbf{m}} = \frac{1}{2}[(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)'\mathbf{S}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)].$$

A regra de classificação baseda na função discriminante de Fisher consiste para designar $\mathbf{x}_0$ em π_1 se

$$\hat{y}_0 \geq \hat{\mathbf{m}},$$

onde $\hat{y}_0 = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)'\mathbf{S}^{-1}\mathbf{x}_0$. Caso contrário designar $\mathbf{x_0}$ para π_2.

A Figura 6.1 apresenta as regiões de classificação da análise discriminante linear de Fisher para os dados de depressão.

Verifica-se facilmente que o método de Fisher correponde a um caso particular da regra que torna mínimo o erro esperado de uma má classificação. A função discriminante linear de Fisher equivale à função que minimiza o erro quando as matrizes de covariâncias, os custos e as prioris são iguais (Johnson & Wichern, 1998).

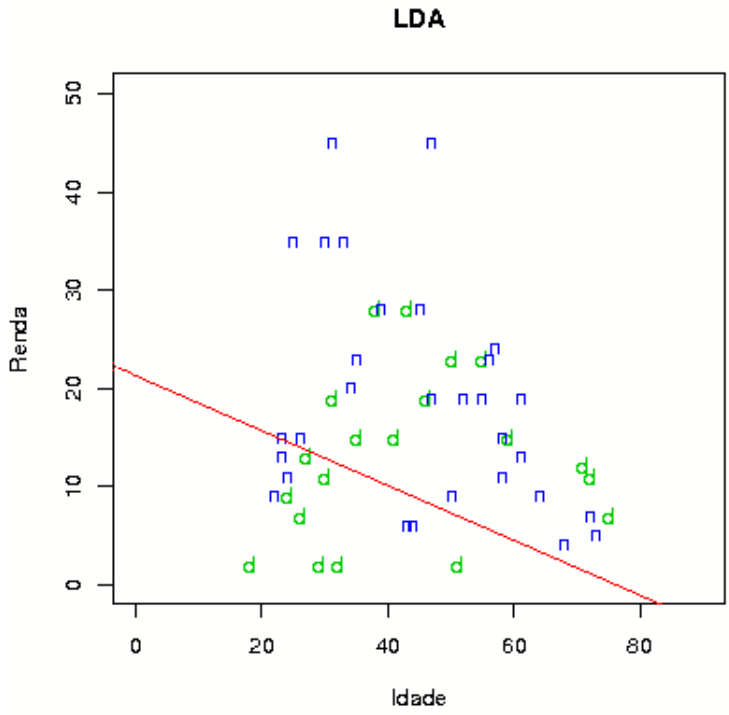


Figura 6.1: Regiões de classificação da função discriminante linear de Fisher para os dados de depressão.

6.3 Discriminante Linear de Fisher Para Mais de Duas Categorias

Fisher (1938), também propôs uma extensão do método discriminante, apresentado na seção anterior, para mais de duas populações. Similarmente, para g grupos o método também supõe que as matrizes de covariâncias populacionais são iguais, ou seja $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_g = \Sigma$. Adicionalmente, considera-se que Σ tem posto completo.

Considere a seguinte combinação linear

$$Y = \mathbf{a}^T\mathbf{X}, \tag{6.1}$$

cujo valor esperado é

$$\mu_{iY} = \mathrm{E}(Y) = \mathbf{a}^T \mathrm{E}(\mathbf{X}/\pi_i) = \mathbf{a}^T \mu_i,$$

que corresponde à categoria π_i. A variância de (6.1) é

$$\mathrm{Var}(Y) = \mathbf{a}^T \mathrm{Cov}(\mathbf{X})\mathbf{a} = \mathbf{a}^T \Sigma \mathbf{a},$$

para todas as populações. Note que o valor esperado μ_{iY} muda quando a população no qual $\mathbf{X}$ é selecionado muda. Defina a média global por

$$\bar{\mu}_Y = \frac{1}{g}\sum_{i=1}^{g} \mu_{iY} = \frac{1}{g}\sum_{i=1}^{g} \mathbf{a}^T \mu_i = \mathbf{a}^T \left(\frac{1}{g}\sum_{i=1}^{g} \mu_i \right) = \mathbf{a}^T \bar{\mu},$$

onde $\bar{\mu} = \frac{1}{g}\sum_{i=1}^{g} \mu_i$. Dessa forma considere a seguinte razão:

$$\begin{aligned} \frac{\sum_{i=1}^{g}(\mu_{iY} - \bar{\mu}_Y)^2}{\sigma_Y^2} &= \frac{\sum_{i=1}^{g}(\mathbf{a}^T\mu_i - \mathbf{a}^T\bar{\mu})^2}{\mathbf{a}^T\Sigma\mathbf{a}} = \frac{\mathbf{a}^T\left(\sum_{i=1}^{g}(\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^T\right)\mathbf{a}}{\mathbf{a}^T\Sigma\mathbf{a}} \\ &= \frac{\mathbf{a}^T B_\mu \mathbf{a}}{\mathbf{a}^T\Sigma\mathbf{a}}, \end{aligned} \tag{6.2}$$

onde $B_\mu = \sum_{i=1}^{g}(\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^T$. Esta razão mede a variabilidade entre os grupos em relação à variabilidade dentro dos grupos. O interesse reside em obter $\mathbf{a}$ que maximize a razão (6.2).

Entretanto, Σ e μ_i são desconhecidos. Seja X_i uma matriz de dados $n_i \times p$, cuja j-ésima linha é denotada por $\mathbf{x}_{ij}^T$. Os vetores de médias amostrais são dados por

$$\bar{\mathbf{x}}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \mathbf{x}_{ij}.$$

E a média global amostral é

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{g} n_i \bar{\mathbf{x}}_i}{\sum_{i=1}^{g} n_i} = \frac{\sum_{i=1}^{g}\sum_{j=1}^{n_i} \mathbf{x}_i}{\sum_{i=1}^{g} n_i},$$

que consiste em um vetor $p \times 1$ de média das n observações da amostra.

Considere o estimador da matriz B_μ como sendo

$$B = \sum_{i=1}^{g} n_i(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T.$$

Em seguida, um estimador de Σ é baseado na matriz

$$W = \sum_{i=1}^{g}(n_i - 1)\mathbf{S}_i = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\bar{\mathbf{x}}_{ij} - \bar{\mathbf{x}}_i)(\bar{\mathbf{x}}_{ij} - \bar{\mathbf{x}}_i)^T.$$

Conseqüentemente, $W/(n_1 + n_2 + ... + n_g - g) = \mathbf{S}$ é uma estimativa de Σ.

Sabendo que W corresponde a constante $(n_1 + n_2 + \ldots + n_g - g)$ multiplicada por $\mathbf{S}$, então o vetor $\hat{\mathbf{a}}$ que maximiza a razão $\hat{\mathbf{a}}^T B\hat{\mathbf{a}}/\hat{\mathbf{a}}^T S\hat{\mathbf{a}}$ também maximiza $\hat{\mathbf{a}}^T B\hat{\mathbf{a}}/\hat{\mathbf{a}}^T W\hat{\mathbf{a}}$. Além disso, nós podemos apresentar o máximo de $\hat{\mathbf{a}}$ como autovetores $\hat{\mathbf{e}}_i$ de $\mathbf{W^{-1}B}$, porque se $\mathbf{W^{-1}B}\hat{\mathbf{e}} = \hat{\lambda}\hat{\mathbf{e}}$, então $\mathbf{S}^{-1}B\hat{\mathbf{e}} = \hat{\lambda}(n_1 + n_2 + ... + n_g - g)\hat{\mathbf{e}}$.

Seja $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_s > 0$ $(s \leq \min(g-1, p))$ os autovalores de $\mathbf{W}^{-1}B$ e $\hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_s$ seus correspondentes autovetores (tal que $\hat{\mathbf{e}}^T S\hat{\mathbf{e}} = 1$). Assim, o vetor de coeficientes $\hat{\mathbf{a}}$ que maximiza a razão

$$\frac{\hat{\mathbf{a}}^T B\hat{\mathbf{a}}}{\hat{\mathbf{a}}^T W\hat{\mathbf{a}}} = \frac{\hat{\mathbf{a}}^T \left(\sum_{i=1}^{g}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T\right)\hat{\mathbf{a}}}{\hat{\mathbf{a}}^T \left(\sum_{i=1}^{g}\sum_{j=1}^{n_i}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T\right)\hat{\mathbf{a}}}$$

é dado por $\hat{\mathbf{a}}_1 = \hat{\mathbf{e}}_1$. A combinação linear $\hat{\mathbf{a}}_k^T = \hat{\mathbf{e}}_k^T\mathbf{x}$ é chamada de k-ésimo discriminante amostral, $k \leq s$.

Uma regra de classificação baseada nos primeiros $r \leq s$ discriminantes amostrais consiste em designar $\mathbf{x_0}$ para π_k se

$$\sum_{j=1}^{r}(\hat{y}_j - \bar{y}_{kj})^2 = \sum_{j=1}^{r}[\hat{\mathbf{a}}_j^T(\mathbf{x_0} - \bar{\mathbf{x}}_k)]^2 \leq \sum_{j=1}^{r}[\hat{\mathbf{a}}_j^T(\mathbf{x_0} - \bar{\mathbf{x}}_i)]^2, \quad \forall i \neq k, r \leq s.$$

CAPÍTULO 7

Análise Discriminante Logística

7.1 Introdução

A regressão logística consiste em um caso particular dos modelos lineares generalizados (McCullagh & Nelder, 1989), utilizados nos casos que a variável de interesse apresenta apenas duas categorias ou que foram de alguma forma dicotomizadas, podendo ser empregada na classificação de uma observação em uma de duas categorias. Quando utilizada na classificação, esta técnica é conhecida como análise discriminante logística.

7.2 Função Discriminante Logística

As variáveis com duas categorias, que podem ser classificadas como sucesso ou fracasso representando as possibilidades de respostas como, por exemplo, 0 e 1, podem ser caracterizadas pela distribuição de Bernoulli.

Assumindo que a variável resposta Y_i segue uma distribuição bernoulli com parâmetro p_i, $Y_i \sim \text{Ber}(p_i)$, ou seja, a densidade de Y_i é

$$f(y_i, p_i) = p_i^{y_i}(1 - p_i)^{1-y_i}. \tag{7.1}$$

Por definição, temos que $P(Y_i = 1) = p_i$ e $P(Y_i = 0) = 1 - p_i$. Com isso,

$$\mathrm{E}(Y_i) = p_i,$$

$$\mathrm{Var}(Y_i) = p_i(1 - p_i).$$

Supondo que a relação entre a média de Y_i e o padrão $\mathbf{x}_i$ seja dado por

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}, \tag{7.2}$$

onde $\beta_0, \beta_1, \ldots, \beta_p$ são os parâmetros desconhecidos.

Pode-se mostrar que, a partir de (7.2), temos

$$p_i = \frac{e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}}}{1 + e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}}} \tag{7.3}$$

e

$$1 - p_i = \left[1 + e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}}\right]^{-1}. \tag{7.4}$$

Seja $y_1, \ldots, y_n$ uma amostra aleatória onde $y_i \sim \mathrm{Ber}(p_i)$, neste caso, a função de verossimilhança é dada por

$$L(p_i, \mathbf{y}) = \prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i},$$

e a log-verossimilhança é

$$\ell(p_i, \mathbf{y}) = \sum_{i=1}^{n} y_i \log p_i + \sum_{i=1}^{n}(1 - y_i)\log(1 - p_i),$$

que desenvolvendo obtemos

$$\ell(p_i, \mathbf{y}) = \sum_{i=1}^{n} y_i \log\left(\frac{p_i}{1-p_i}\right) + \sum_{i=1}^{n}\log(1 - p_i). \tag{7.5}$$

Substituindo (7.3) e (7.4) em (7.5) encontramos

$$\ell(\beta, \mathbf{y}) = \sum_{i=1}^{n} y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) - \sum_{i=1}^{n} \log[1 + \exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})]. \quad (7.6)$$

O método de máxima verossimilhança consiste em maximizar a função definida em (7.6), que expressa a probabilidade dos dados observados em função dos parâmetros do modelo.

O estimador de máxima verossimilhança do vetor $\beta = (\beta_0, \ldots, \beta_p)$ para o modelo de regressão logístico não apresenta forma fechada, portanto deve ser estimado numericamente através de algum método de otimização não-linear (Newton-Raphson, BFGS, Score de Fisher, dentre outros).

Encontrado o estimador de máxima verossimilhança, $\hat{\beta} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)$, o i-ésimo valor ajustado é dado por:

$$\hat{p}_i = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \ldots + \hat{\beta}_p x_{ip}}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \ldots + \hat{\beta}_p x_{ip}}}.$$

Considerando o ponto de corte igual a 0.5, uma regra de classificação consiste em designar o padrão $\mathbf{x_0}$ para a classe π_1 se $\hat{p}_i < 0.5$ e atribuir a classe π_2 caso contrário.

A Figura 7.1 apresenta um exemplo das regiões de classificação da análise discriminante logística. Para esta ilustração, novamente utilizamos os dados de depressão descritos na Seção 2.2.1.

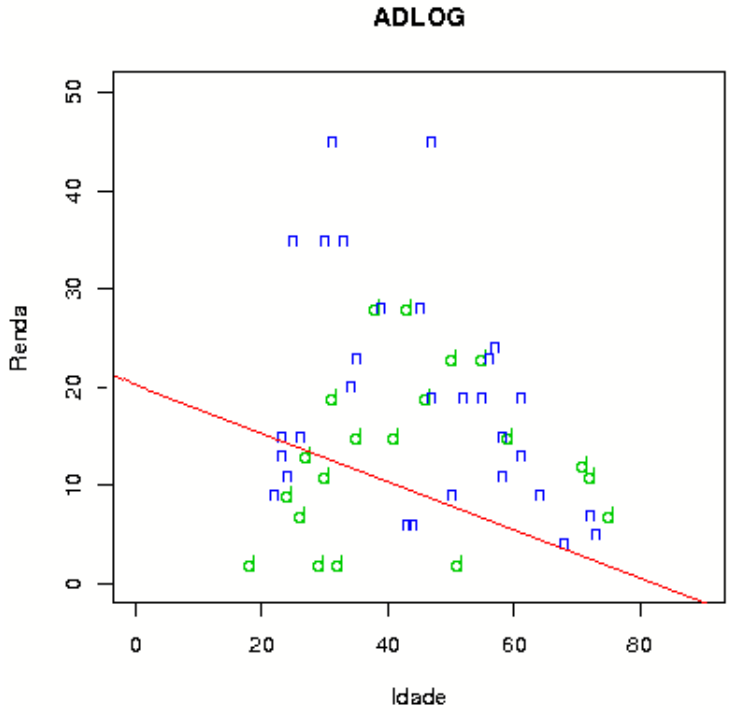


Figura 7.1: Regiões de classificação da função discriminante logística para os dados de depressão.

CAPÍTULO 8

Redes Neurais

8.1 Introdução

Redes neurais é uma técnica proposta para resolução de problemas em diversos campos de aplicação, tais como reconhecimento de padrões e modelagem. Seu funcionamento se baseia no funcionamento do cérebro humano. As redes neurais se assemelham ao cérebro em dois aspectos: pela habilidade em aprender a partir dos padrões de características com ou sem categorias associadas e pela capacidade de generalização.

A era moderna das redes neurais teve início com o trabalho pioneiro de McCulloch & Pitts (1943). Neste artigo, os autores descreveram um cálculo lógico das redes neurais que unificava os estudos de neurofisiologia e da lógica matemática.

As redes neurais podem ser utilizadas em diversas áreas, por exemplo na área de diagnóstico ou prognóstico médico (Anagnostopoulos & Maglogiannis, 2006), na área financeira (West et. al., 2005), etc.

Uma rede neural é caracterizada principalmente por três fatores: arquitetura da rede, tipo de treinamento e função de ativação.

8.2 Arquiteturas de Redes

A arquitetura da rede neural se refere a organização dos neurônios e os tipos de conexões entre eles. Aqui apresentamos as principais arquiteturas utilizadas em problemas de classificação.

Em relação ao número de camadas, podemos ter uma rede com duas camadas, que consiste em uma rede com uma camada de entrada que se conecta diretemente com a camada de saída, ou uma rede com múltiplas camadas, onde existe uma ou mais camadas entre a camada de entrada e a de saída. No enfoque de classificação, as redes neurais que contém apenas duas camadas caracterizam-se pela sua capacidade de resolver apenas problemas linearmente separáveis. Redes com múltiplas camadas são recomendadas em problemas de classificação não-linearmente separáveis.

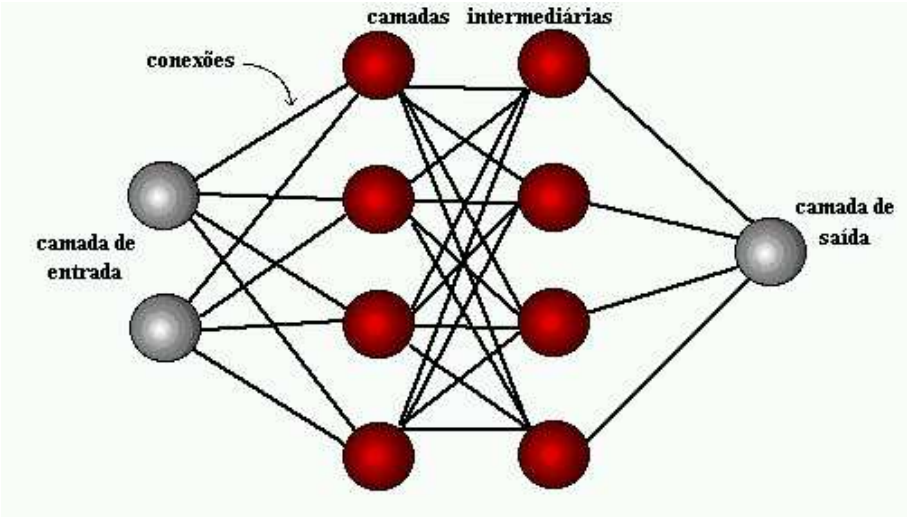


Figura 8.1: Arquitetura de uma rede feedforward completamente conectada.

Os neurônios da rede podem estar conectados das seguintes formas:

1. *Feedforward*: os neurônios de uma camada estão conectados aos neurônios da camada seguinte, não havendo realimentação (a comunicação é de forma unidirecional) nem conexões entre neurônios da mesma camada.

2. *Feedback*: os neurônios de uma camada podem estar conectados aos neurônios das camadas anteriores.

Quanto à conectividade, as redes neurais podem ser caracterizadas como parcialmente conectadas ou completamente conectadas.

A Figura 8.1, apresenta a arquitetura de uma rede feedforward completamente conectada. Esse é um exemplo de um perceptron de múltiplas camadas com duas camadas ocultas e uma camada de saída.

8.3 Função de Ativação

A função de ativação é uma função que aplicada à combinação linear entre as variáveis de entrada e os pesos que chegam a um determinado neurônio, resulta em um valor de saída.

As funções de ativação mais usualmente utilizadas são:

1. Função limiar:

$$\varphi(v) = \begin{cases} 1, v \geq 0 \\ 0, v < 0 \end{cases}$$

2. Função sigmóide:

$$\varphi(v) = \frac{1}{1 + \exp(-av)},$$

onde a é o parâmetro de inclinação da função.

3. Tangente Hiperbólica:

$$\varphi(v) = \tanh(v).$$

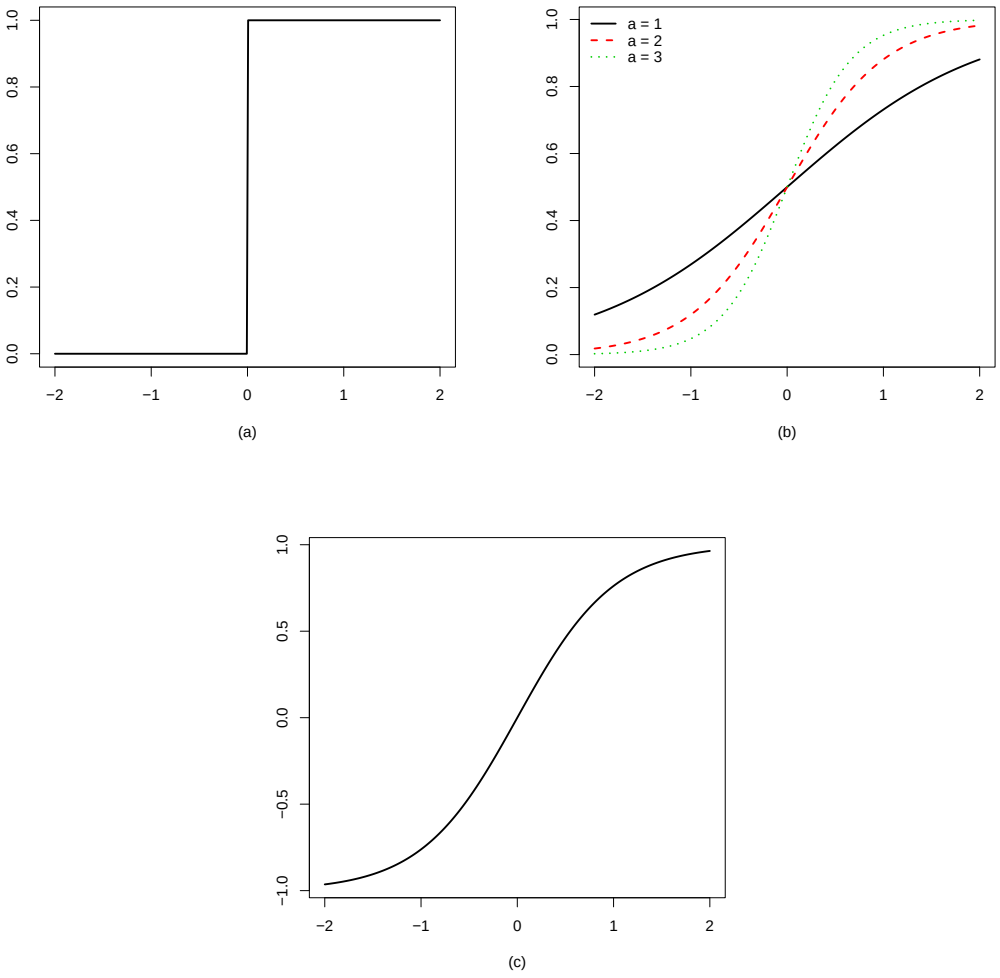


Figura 8.2: Funções de ativação: (a) limiar, (b) sigmóide e (c) tangente hiperbólica.

A função de ativação limiar assume somente dois valores 0 ou 1. A função sigmóide assume valores no intervalo $(0, 1)$. Por outro lado a função de ativação tangente hiperbólica toma valores de -1 a 1. É necessário que $\varphi(\cdot)$ seja continuamente diferenciável.

Uma rede neural pode possuir diferentes funções de ativação para diferentes neurônios. Porém, em geral, as redes utilizam a mesma função de ativação para os neurônios de uma mesma camada. Os gráficos das funções de ativação são apresentados na Figura 8.2.

8.4 Algoritmo de Treinamento

Após especificação da arquitetura da rede neural, torna-se necessário definir o algoritmo de treinamento da rede. Basicamente, o treinamento da rede neural consiste em um problema de minimizaçao não-linear sem restrições, em que os pesos sinápticos da rede são iterativamente modificados para minimizar o erro médio quadrático entre a resposta desejada a partir dos dados de entrada e a saída obtida no neurônio de saída. Do ponto de vista estatístico, o treinamento da rede neural seria estimar os parâmetros do modelo considerando-se um conjunto de dados. Vários métodos para treinamento supervisionado de redes neurais são propostos, entretanto o mais popularmente utilizado para esse tipo de treinamento é o algoritmo de retropropagação (em inglês, backpropagation).

Os algoritmos de aprendizagem são ferramentas organizadas dedicadas ao desempenho de funções específicas para solução de problemas em redes neurais. Basicamente, os algoritmos de aprendizagem diferem entre si pela forma como é realizado o ajuste dos pesos sinápticos dos neurônios. Neste capítulo apresentamos o tipo de aprendizado utilizado nesta dissertação, o aprendizado por correção de erro. Maiores detalhes sobre outros tipos de aprendizado podem ser vistos em Haykin (2001).

A aplicação do algoritmo de retropropagação requer a escolha dos seguintes parâmetros: número de iterações do algoritmo, critério de parada, pesos iniciais e taxa de aprendizado. A escolha dessas quantidades pode ser decisiva para a capacidade de generalização da rede.

8.5 Perceptron de Múltiplas Camadas

Uma rede do tipo feedforward, onde emprega-se a retropropagação do erro (back-propagation) como algoritmo de treinamento é comumente referida como perceptron de múltiplas camadas (MLP).

A MLP é formada por uma camada de entrada, uma camada de saída (onde a solução do problema é obtida), e uma ou mais camadas intermediárias. Cada camada em uma MLP contém um ou mais elementos de processamento (neurônios). As unidades que ligam os neurônios são chamadas de pesos sinápticos. O conjunto das conexões entre os neurônios da rede formam o vetor de pesos sinápticos. Na camada de entrada da rede deve existir p elementos de processamento, isto é, um para cada variável independente.

Além do número de neurônios da camada de entrada, para implementarmos uma rede neural devemos determinar o número de camadas escondidas e o número de neurônios a serem considerados na camada de saída. Estes aspectos afetam o desempenho da rede neural, devendo ser cuidadosamente escolhidos.

Em geral, utiliza-se uma camada intermediária na rede. O que requer maior esforço para construção da rede neural é quanto a definição do número de neurônios dessa camada intermediária. Alguns critérios podem ser adotados, a forma mais comum é através da capacidade preditiva da rede. Em geral, realiza-se vários testes, utilizando diferentes números de neurônios, para escolha da rede.

Não existe um critério geral que permita definir o número de neurônios na camada escondida. Porém, sabemos que redes neurais com poucos neurônios escondidos são preferíveis, visto que elas, em geral, possuem bom poder de generalização, reduzindo o problema de sobreajuste (overfitting) (Haykin, 2001). Entretanto, redes com poucos neurônios escondidos podem não possuir a habilidade suficiente para modelar e aprender os dados em problemas complexos, podendo ocorrer underfitting, ou seja, a rede não converge durante o treinamento.

Em tarefas de classificação costuma-se utilizar o número de neurônios na camada de saída igual ao número de categorias. No caso de classificação binária, pode-se optar por

utilizar apenas um nerônio de saída.

Considere uma rede que possui J neurônios na camada escondida. O j-ésimo neurônio da camada escondida recebe as entradas multiplicadas pelos pesos sinápticos e somadas com uma constante, denominada viés. Denote por a_j o resultado desta operação.

Seja $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p)^T$ o vetor de entradas da rede, onde $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T$, para $i = 1, \ldots, p$ e n corresponde ao tamanho da amostra de treinamento. Assim, a_j é dado por

$$a_j = \omega_{0j} + \sum_{i=1}^{p} \omega_{ij}\mathbf{x}_i, \qquad j = 1, \ldots, J,$$

onde ω_{ij} corresponde ao peso sináptico da conexão entre o j-ésimo neurônio da camada escondida e sua entrada $\mathbf{x}_i$. ω_{0j} é o respectivo viés. Este resultado é levado à função de ativação, que tem como finalidade ativar ou inibir o próximo neurônio. Com isso, a saída v_j do j-ésimo neurônio da camada escondida é

$$v_j = \varphi_j(a_j) = \varphi_j(\omega_{0j} + \sum_{i=1}^{p} \omega_{ij}\mathbf{x}_i), \qquad j = 1, \ldots, J, \tag{8.1}$$

onde $\varphi_j(\cdot)$ é a função de ativação do neurônio j. O resultado obtido em (8.1) é passado para os neurônios da próxima camada. Assim, o k-ésimo neurônio da camada de saída é da forma

$$y_k(\mathbf{x}, \boldsymbol{\omega}) = \varphi_k \left\{ \omega_{0k} + \sum_{j=1}^{J} \omega_{jk}\varphi_j(\omega_{0j} + \sum_{i=1}^{p} \omega_{ij}\mathbf{x}_i) \right\}, \qquad k = 1, \ldots, q, \tag{8.2}$$

onde ω_{jk} é o peso sináptico da conexão entre o k-ésimo neurônio da camada de saída e o j-ésimo neurônio da camada escondida, sendo ω_{0k} seu correspondente viés. φ_k corresponde a função de ativação utilizada no k-ésimo neurônio da camada de saída e q é o número de neurônios na camada de saída. No contexto de classificação a saída (8.2) corresponde ao resultado da final da classificação.

8.5.1 Algoritmo de Retropropagação do Erro

O objetivo do processo de aprendizado é ajustar os parâmetros livres da rede (pesos sinápticos) de maneira a minimizar uma dada função de erro. Uma alternativa consiste em minimizar o erro médio quadrático, definido por

$$e = \frac{1}{n}\sum_{i=1}^{n}[y_i - y(\mathbf{x}_i, \boldsymbol{\omega})]^2,$$

onde y_i é a i-ésima saída desejada e $y(\mathbf{x}_i, \boldsymbol{\omega})$ é a i-ésima saída da rede.

No treinamento com o algoritmo de retropropagação, a rede opera em uma seqüência de duas fases. Na primeira fase um padrão é apresentado à camada de entrada da rede. O padrão de entrada se propaga pela rede, camada por camada até que a resposta seja obtida na camada de saída. Na fase seguinte, a saída obtida pela rede é comparada à saída desejada para esse padrão particular e um possível erro é propagado a partir da camada de saída até a camada de entrada. Os pesos sinápticos são modificados conforme o erro é retropropagado. Este processo iterativo é baseado no método do gradiente descendente. A cada passo do treinamento cada peso sináptico ω_{ij} é adicionado por

$$\Delta\omega_{ij}^{(t)} = -\eta\frac{\partial e^{(t)}}{\partial\omega_{ij}^{(t)}},$$

onde η é a taxa de aprendizado, que consiste em uma constante no intervalo $(0, 1)$ e que mede a magnitude das mudanças dos pesos sinápticos. Assim, os pesos ω_{ij} são atualizados da seguinte forma

$$\omega_{ij}^{(t+1)} = \omega_{ij}^{(t)} + \Delta\omega_{ij}^{(t)}, \tag{8.3}$$

onde t representa a t-ésima iteração do algoritmo.

O método do gradiente descendente, em geral, apresenta convergência lenta e pode ser bastante sensível a escolha da taxa de aprendizado.

O parâmetro taxa de aprendizado tem grande influência durante o processo de treinamento da rede neural. Uma taxa de aprendizado muito baixa torna o aprendizado da

rede muito lento, ao passo que uma taxa de aprendizado muito alta provoca oscilações no treinamento e impede a convergência do processo de aprendizado (Haykin, 2001). Uma alternativa consiste em diminuir progressivamente a taxa de aprendizado durante o treinamento. O processo de treinamento do perceptron de múltiplas camadas pode requerer um tempo de treinamento consideravelmente longo.

Rumelhart & McClelland (1986) sugerem uma maneira de aumentar a taxa de aprendizado sem levar à oscilação. A modificação é realizada através da inclusão do momento μ em (8.3), da seguinte forma

$$\omega_{ij}^{(t+1)} = \omega_{ij}^{(t)} + \Delta\omega_{ij}^{(t)} + \mu\omega_{ij}^{(t-1)}.$$

Esta modificação pode acelerar o processo de aprendizado.

Existe uma série de variações propostas para o algoritmo de retropropagação. Uma modificação conhecida é o Quickprop (Fahlman, 1989). O qual utiliza um procedimento de busca em linha por η para cada parâmetro.

A Figura 8.3 apresenta regiões de classificação do perceptron de múltiplas camadas para os dados de depressão. As regiões variam quanto ao número de neurônios considerados nas camadas intermediárias da rede. A taxa de aprendizado utilizada em todas as redes foi de 0,001. Os números de neurônios considerados em cada rede foram 2(a), 5(b), 10(c) e 15(d).

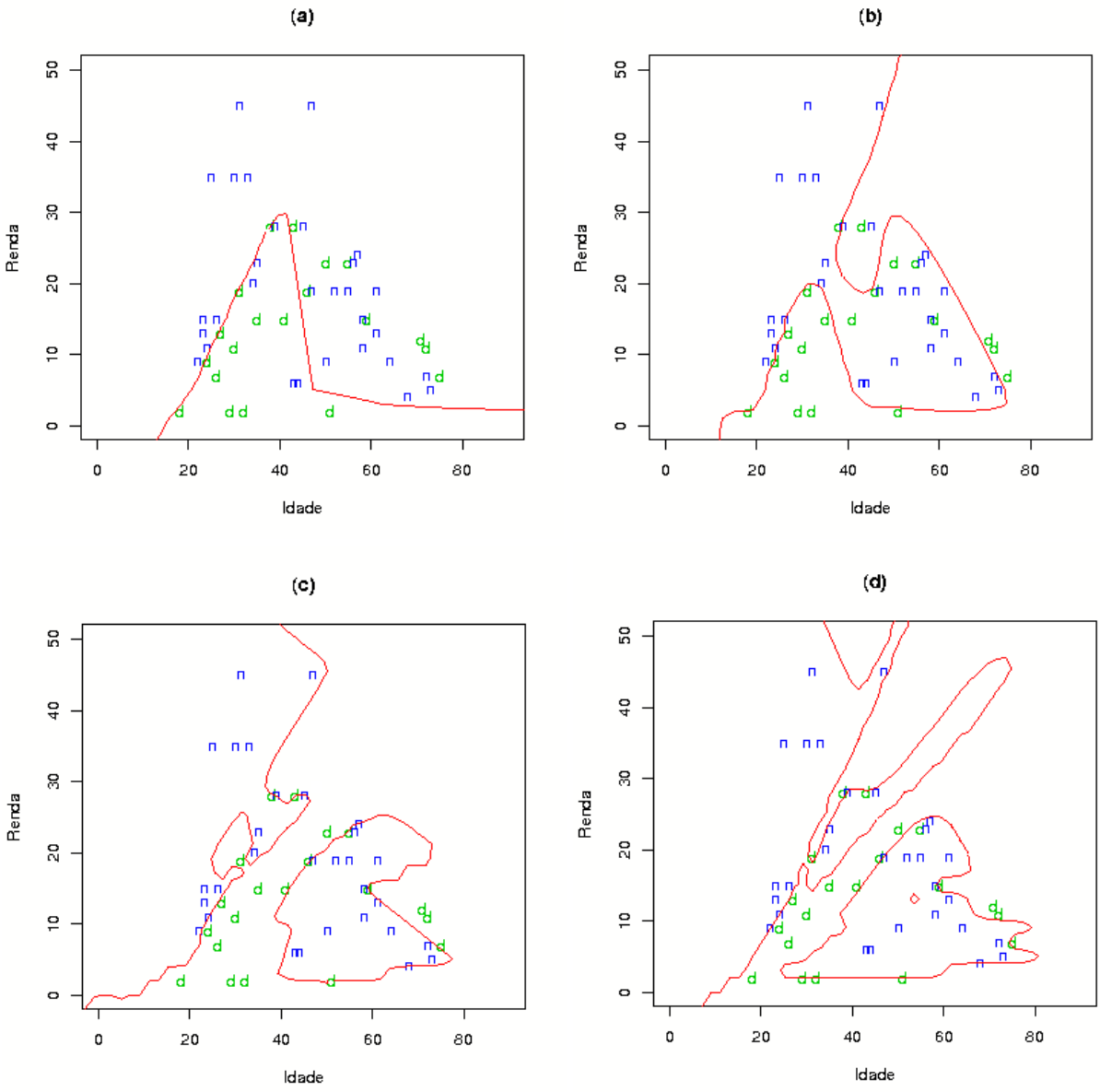


Figura 8.3: Regiões de classificação para o perceptron de múltiplas camadas aplicado aos dados de depressão.

CAPÍTULO 9

Agregação via Bootstrap

9.1 Introdução

Com o intuito de melhorar o desempenho global de alguns métodos automáticos de classificação, alguns procedimentos têm sido sugeridos. Bagging (Breiman, 1996) e boosting (Freund & Schapire, 1996) são alguns deles. Em ambos os procedimentos, múltiplas versões do conjunto de treinamento são geradas e a classificação resultante oriunda dos classificadores contruídos são combinadas.

Neste estudo, consideramos particularmente o procedimento bagging. Para alguns métodos automáticos de classificação, o procedimento bagging mostra-se eficaz no melhoramento de desempenho. Ganhos substanciais foram obtidos a partir do uso de redes neurais e árvores de classificação como métodos automáticos de classificação (Breiman, 1996).

9.2 Bagging

Bagging (**b**ootstrap **agg**regat**ing**) foi proposto inicialmente por Breiman (1996) com enfoque em classificação e regressão.

No contexto de classificação, bagging consiste em um método de previsão que gera múltiplas versões do conjunto de treinamento e treina um classificador em cada amostra gerada. As amostras são geradas através de réplicas bootstrap do conjunto de treinamento. Cada classificador produzido é aplicado a um padrão de teste $\mathbf{x}$ que é classificado por maioria de votos. Em caso de empate, a decisão de classificação é arbitrária. No contexto de regressão recorre-se a uma média das versões agregadas. Como o enfoque deste trabalho está no contexto de classificação, maiores detalhes de bagging em regressão pode ser visto em Breiman (1996).

Em tarefas de classificação, consideramos um conjunto de treinamento $\mathcal{L}$ dado por $(y_i, \mathbf{x}_i),\ i = 1, \ldots, n$, onde o y_i corresponde a categoria da i-ésima observação. Dado um padrão de entrada $\mathbf{x}$, estima-se y por $\phi(\mathbf{x}, \mathcal{L})$. Seja $\mathcal{L}_k$ uma seqüência de conjuntos de treinamento, onde cada conjunto de treinamento contém n observações independentes de mesma função de distribuição de $\mathcal{L}$, o objetivo é usar $\mathcal{L}_k$ para obter um classificador melhor do que o classificador contruído mediante único conjunto de treinamento $\phi(\mathbf{x}, \mathcal{L})$. Ou seja, o interesse consiste em utilizar a seqüência de classificadores $\phi(\mathbf{x}, \mathcal{L}_k)$.

Na prática não dispomos das réplicas de $\mathcal{L}$, apenas dispomos do conjunto de treinamento individual de $\mathcal{L}$. Uma alternativa é tomar repetidas amostras bootstrap $\mathcal{L}^{(b)}$ de $\mathcal{L}$, e formar $\phi(\mathbf{x}, \mathcal{L}^{(B)})$. Tome $\phi_B(\mathbf{x})$ como sendo a categoria majoritária de $\phi(\mathbf{x}, \mathcal{L}^{(B)})$.

$\mathcal{L}^{(B)}$ contém réplicas do conjunto de treinamento, cada réplica consistindo de n observações retiradas aleatoriamente com reposição de $\mathcal{L}$. $\mathcal{L}^{(b)}$, $b = 1, \ldots, B$, são réplicas bootstrap que aproximam a distribuição de $\mathcal{L}$. Maiores detalhes sobre a técnica de reamostragem bootstrap ver Efron & Tibshirani (1993). Dado um conjunto de treinamento $\mathcal{L}$ e um padrão $\mathbf{x}$, o algoritmo bagging é dado por

1. Para $b = 1, \ldots, B$, faça

   (a) Retire uma amostra bootstrap $\mathcal{L}^{(b)}$ de tamanho n com reposição do conjunto de treinamento $\mathcal{L}$;

   (b) Construa um classificador, $\phi(\mathbf{x}, \mathcal{L})$, baseado em $\mathcal{L}^{(b)}$.

2. Classificar o padrão $\mathbf{x}$ de teste utilizando $\phi(\mathbf{x}, \mathcal{L}^{(b)})$, $b = 1, \ldots, B$, e designar $\mathbf{x}$ para a classe mais freqüente.

Se o interesse reside na avaliação de desempenho de um dado classificador agregado via bootstrap, baseado em n observações, pode-se fazer uso do seguinte algoritmo:

O conjunto de dados é dividido aleatoriamente em conjunto de treinamento $\mathcal{L}$ e conjunto de teste $\mathcal{T}$

1. Para $b = 1, \ldots, B$, faça

   (a) Retire uma amostra bootstrap $\mathcal{L}^{(b)}$ de tamanho n com reposição do conjunto de treinamento $\mathcal{L}$;

   (b) Construa um classificador, $\phi(\mathbf{x}, \mathcal{L})$, baseado em $\mathcal{L}^{(b)}$.

2. Classificar um padrão $\mathbf{x}$ de teste utilizando $\phi(\mathbf{x}, \mathcal{L}^{(b)})$, $b = 1, \ldots, B$, e designar $\mathbf{x}$ para a classe mais freqüente.

3. A divisão aleatória dos dados é repetida r vezes.

Um fator que indica o não melhoramento em termos de acurácia pelo bagging é a estabilidade do procedimento na construção de ϕ. Se modificações em $\mathcal{L}$, isto é, se uma réplica $\mathcal{L}$, produz pequenas alterações em ϕ, então ϕ_B deve estar próximo de ϕ. Melhoramentos ocorrem pela instabilizade de procedimentos onde uma pequena alteração em $\mathcal{L}$ pode resultar em uma grande alteração em ϕ. A instabilidade foi estudada em Breiman (1996) por meio de estudos empíricos e foi percebida na utilização de redes neurais, árvores de classificação e árvores de regressão. Por outro lado, o método dos k-vizinhos mais próximos se mostrou estável, assim com a análise discriminante linear.

CAPÍTULO 10

Avaliação Numérica

10.1 Detalhes Metodológicos

Através de simulação de Monte Carlo investigamos o desempenho do método de agregação via bootstrap para os seguintes classificadores:

- Análise discriminante linear de Fisher(lda);
- Análise discriminante quadrática (qda);
- Análise discriminante logística (adlog);
- k-vizinhos mais próximos (k-nn);
- Perceptron múltiplas camadas (mlp).

Utilizando o software livre `R`, estimamos os parâmetros dos métodos de análise discriminate linear de Fisher e análise discriminate quadrática através das funções `lda` e `qda`, respectivamente, do pacote `MASS`. Os parâmetros do modelo logístico são estimados a partir da função `glm` do pacote `stats`. O método k-nn é obtido utilizando a função `knn` do pacote `class`. Os pesos sinápticos das redes neurais foram obtidos utilizando a função `nnet` do pacote `nnet`.

Os dados são gerados de acordo com quatro cenários distintos, baseados em Duong (2004), que ilustram diferentes estruturas de separação das categorias, considerando apenas duas categorias, descritos na Tabela 10.1. No cenário A, a separação das populações é bastante evidente, podendo se dar até mesmo de forma linear. Neste cenário, as populações têm distribuição normal bivariada com matrizes de covariâncias distintas. Separações não-lineares e com sobreposição das categorias são ilustradas nos cenários B, C e D. No cenário B, não consideramos as populações como normais e as matrizes de covariâncias são iguais. Para o cenário C, a população π_1 não segue uma distribuição normal, enquanto que a população π_2 tem distribuição normal bivariada, com comportamentos distintos em relação a variabilidade. Por fim, no cenário D as populações possuem distribuição normal bivariada, com matrizes de covariâncias idênticas. As regiões de contorno dos cenários são exibidas na Figura 10.1.

Em cada cenário utilizamos $r = 1000$ réplicas de Monte Carlo. Consideramos os seguintes tamanhos amostrais: $n = 100, 500$ e 1000. Para cada réplica, geramos $n/2$ observações X da categoria π_1 e $n/2$ observações X da categoria π_2, ou seja, com $p(\pi_1) = p(\pi_2) = 1/2$, de acordo com a Tabela 10.1. Adicionalmente, em cada réplica de Monte Carlo geramos um conjunto de treinamento $\mathcal{L}$ e um conjunto de teste $\mathcal{T}$, sendo $\mathcal{L} = \mathcal{T} = n/2$.

Para cada classificador, o algoritmo bagging é aplicado em cada amostra gerada, retirando B réplicas bootstrap, $B = 10, 25, 50$ e 100, de $\mathcal{L}$ e calculando a taxa de erro pela classificação majoritária em $\mathcal{T}$. A média da taxa de erro bagging para as 1000 réplicas é denotada por e_B. Para cada classificador, denotamos por e_R a média da taxa de erro das réplicas, onde o classificador foi treinado utilizando $\mathcal{L}$ e avaliado utilizando $\mathcal{T}$. Essa quantidade é considerada como uma quantidade básica, onde nenhum método de melhoramento é aplicado ao classificador. Calculamos também a média da taxa de erro para o classificador de Bayes para todos os cenários e tamanhos amostrais avaliados.

Para cada cenário, os parâmetros livres dos classificadores mlp e k-nn foram selecionados por simulações preliminares de forma a minimizar a taxa de erro. Para o classificador k-nn, optou-se por escolher $k = 3$ para os cenários A, B e C, e $k = 16$ para o cenário D.

Adicionalmente, a distância euclidiana foi utilizada como métrica de distância.

As redes neurais utilizadas possuem uma camada intermediária. No cenário A utilizamos uma rede que possui dois neurônios, taxa de aprendizado $\eta = 0,01$. No cenário B a rede possui nove neurônios e $\eta = 0,1$. No cenário C utilizamos uma rede com seis neurônios e $\eta = 0,1$. Por fim, no cenário D utilizamos uma rede com três neurônios e $\eta = 0,01$. Em todas as redes neurais utilizamos a função de ativação sigmóide.

Tabela 10.1: Cenários sob estudo.

Cenário	Estrutura
A	$\pi_1 : X \sim \mathcal{N}\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} \frac{4}{9} & \frac{14}{45} \\ \frac{14}{45} & \frac{4}{9} \end{bmatrix}\right)$
	$\pi_2 : X \sim \mathcal{N}\left(\begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} \frac{4}{9} & 0 \\ 0 & \frac{4}{9} \end{bmatrix}\right)$
B	$\pi_1 : X \sim \frac{1}{2}\mathcal{N}\left(\begin{bmatrix} -\frac{3}{2} \\ -\frac{3}{2} \end{bmatrix}, \begin{bmatrix} \frac{4}{5} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{4}{5} \end{bmatrix}\right) + \frac{1}{2}\mathcal{N}\left(\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \begin{bmatrix} \frac{4}{5} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{4}{5} \end{bmatrix}\right)$
	$\pi_2 : X \sim \frac{1}{2}\mathcal{N}\left(\begin{bmatrix} \frac{3}{2} \\ \frac{3}{2} \end{bmatrix}, \begin{bmatrix} \frac{4}{5} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{4}{5} \end{bmatrix}\right) + \frac{1}{2}\mathcal{N}\left(\begin{bmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}, \begin{bmatrix} \frac{4}{5} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{4}{5} \end{bmatrix}\right)$
C	$\pi_1 : X \sim \frac{1}{2}\mathcal{N}\left(\begin{bmatrix} -\frac{3}{2} \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{3}{10} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{10} \end{bmatrix}\right) + \frac{1}{2}\mathcal{N}\left(\begin{bmatrix} \frac{3}{2} \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{3}{10} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{10} \end{bmatrix}\right)$
	$\pi_2 : X \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{4}{5} & \frac{2}{5} \\ \frac{2}{5} & 1 \end{bmatrix}\right)$
D	$\pi_1 : X \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \frac{2}{3} & \frac{1}{5} \\ \frac{1}{5} & \frac{4}{9} \end{bmatrix}\right)$
	$\pi_2 : X \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ -1 \end{bmatrix}, \begin{bmatrix} \frac{2}{3} & \frac{1}{5} \\ \frac{1}{5} & \frac{4}{9} \end{bmatrix}\right)$

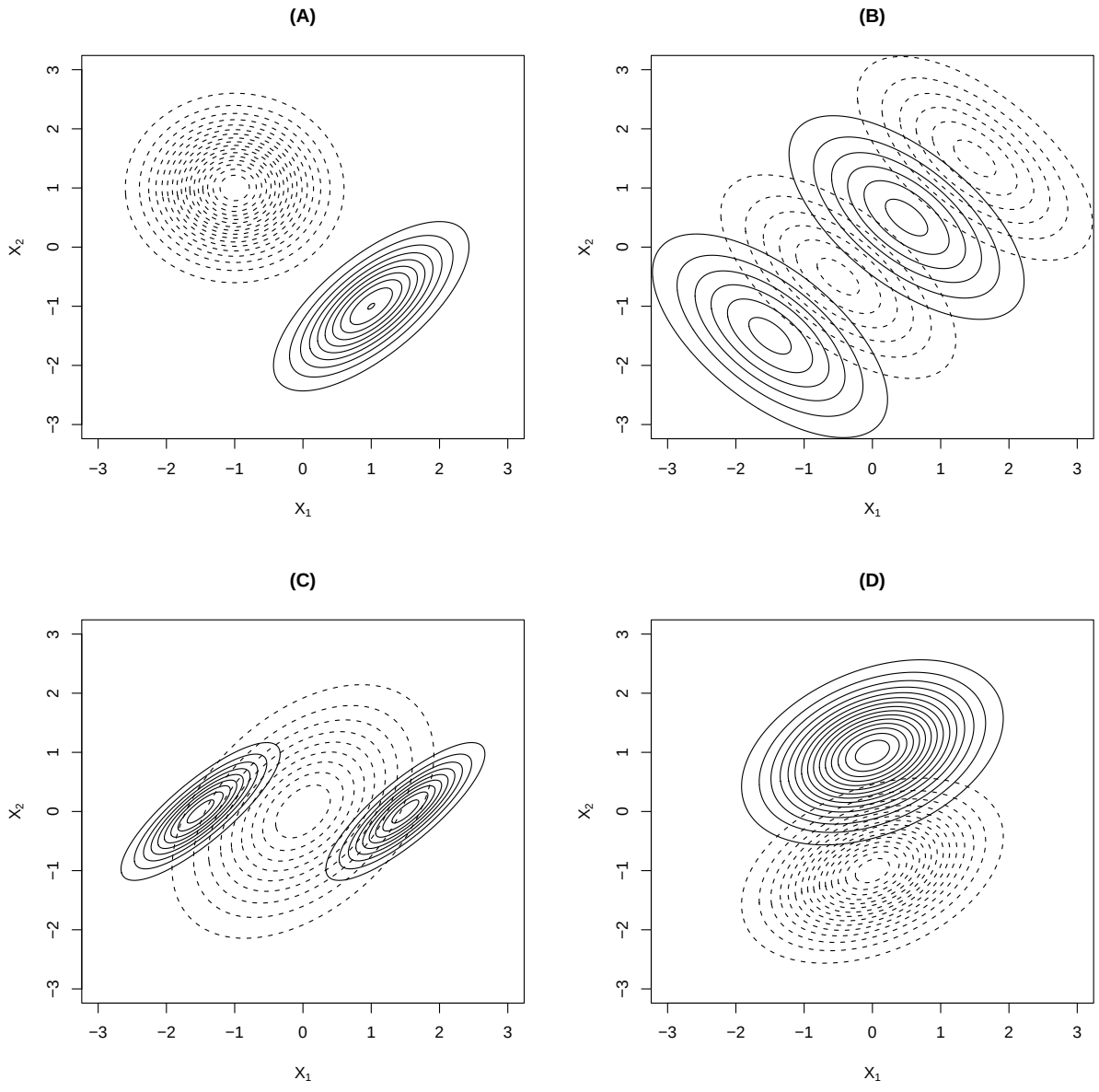


Figura 10.1: Cenários sob estudo.

10.2 Resultados

As tabelas 10.2, 10.3, 10.4 e 10.5 apresentam os resultados das simulações para os cenários A, B, C e D respectivamente. Para cada cenário, apresentamos os tamanhos amostrais investigados e o nome das respectivas técnicas de classificação utilizadas, correspondendo a primeira e segunda colunas. As taxas de erro são apresentadas nas demais colunas. A terceira coluna apresenta a média de erro e_R produzida a partir da geração dos dados de treinamento e teste para as 1000 réplicas de Monte Carlo. Na coluna seguinte a menor média de erro e_B é apresentada, também resultante das 1000 repetição na geração dos dados de treinamento e teste e, de acordo com o número de amostras bootstrap consideradas. Nesse caso, o número de amostras bootstrap que deram origem às menores médias de erro bagging são apresentadas ao lado da média correspondente. Por fim, a última coluna mostra se houve aumento ou decaimento na média de erro bagging com relação à média de erro das repetições - um valor negativo representa um decrescimento na média de erro produzida ao utilizar o bagging, indicando um melhoramento de desempenho.

Sob normalidade multivariada e matrizes de covariâncias distintas, cenário A, o uso de análise discriminante quadrática resultou em taxas de erro ligeiramente menores que os demais métodos em todos os tamanhos amostrais avaliados. Por exemplo, para $n = 100$, a média da taxa de erro e_R do classificador qda foi de 0,43%, enquanto que os classificadores lda, adlog, k-nn e mlp apresentaram taxas de erro médias de 0,87%, 0,75%, 0,49% e 0,48%, respectivamente (Tabela 10.2). Este classificador apresentou médias das taxas de erro mais próximas das taxas médias do erro ótimo de Bayes.

No cenário B (Tabela 10.3), a separação das populações é de forma não-linear e as matrizes de covariâncias são iguais. Neste cenário, como era previsto, os métodos não-lineares de separação apresentaram, em geral, melhores desempenhos. Por exemplo, para $n = 500$ os classificadores lineares lda e adlog apresentaram taxas de erro médias e_R de 43,96% e 35,20%, respectivamente. Por outro lado, para os classificadores não-lineares k-nn e mlp e_R foi de 18,74% e 20,68%, respectivamente. Adicionalmente, neste cenário

a suposição de normalidade não é válida e, apesar de ser um classificador não-linear a análise discriminante quadrática apresentou baixo desempenho.

De acordo com a Figura 10.1(c), pode-se verificar que, assim como no cenário B, o cenário C também é caracterizado pela forma não-linear de separação das categorias. Neste cenário, classificadores lineares apresentam dificuldades em separar as categorias de forma satisfatória. Dentre os classificadores não-lineares avaliados, o classificador qda apresentou, em geral, baixo desempenho, exceto para o tamanho amostral $n = 100$. O classificador k-nn foi, em geral, ligeiramente superior a rede neural proposta.

Para o cenário D, de forma geral, os classificadores apresentaram médias da taxa de erro muito próximas. Sendo o desempenho da análise discriminante linear de Fisher ligeiramente superior, o que provavelmente deve-se ao fato das matrizes de covariâncias serem iguais para ambas as categorias.

Para o classificador lda, o uso de bagging produz uma queda na taxa de erro e_R em todos os cenários. Entretanto, a maior diminuição encontrada foi de 3,97% e ocorreu no cenário C com $n = 100$.

Nos cenários B, C e D, o algoritmo bagging produz uma ligeira melhora na taxa e_R para o classificador qda. Todavia, no cenário A, houve um ligeiro aumento na taxa de erro com $n = 500$, o acréscimo foi de 0,14%.

Com relação ao classificador adlog, bagging produziu melhoras na taxa de erro nos cenários A, C e D. A maior redução na taxa e_R foi de 11,77% e ocorreu no cenário A com $n = 100$. Todavia, no cenário C não houve redução alguma, pelo contrário, houve um ligeiro acréscimo, de no máximo 0,61% com $n = 100$.

O classificador k-nn se mostra, em geral eficiente com bagging. A maior redução foi de 4,28%, obtida no cenário B, com $n = 1000$. Um acréscimo na taxa de erro ocorreu no cenário A, com $n = 100$.

Para o classificador mlp, a média da taxa de erro e_B foi inferior à taxa média e_R em todos os cenários e tamanhos amostrais, exceto no cenário D, $n = 1000$, onde $e_B = e_R = 5,53\%$.

Para todos os classificadores avaliados, o percentual de redução obtido através do

bagging, em geral, decresce à medida que o tamanho da amostra aumenta.

As Figuras 10.2, 10.3 e 10.4 exibem o comportamento das médias das taxas de erro bagging e_B para os quatro cenários avaliados e tamanhos amostrais $n = 100, 500$ e 1000, respectivamente. Pode-se notar, de acordo com estes gráficos, que não há evidência de que o número de réplicas bootstrap extraídas do conjunto de treinamento interfere na taxa de erro média e_B. Em geral, o uso de 100 réplicas bootstrap resultou nas menores taxas de erro agregadas.

Em geral, podemos concluir que a estrutura de separação das populações pode evidenciar ganhos em acurácia pelos classificadores utilizando o bagging. No caso em que a separação se dá até mesmo de forma linear (Cenário A), os classificadores lineares têm seus desempenhos melhorados não sendo necessário para isto mais de cinqüênta réplicas bootstrap e tamanhos amostrais grandes (1000). No caso em que a separação linear é inviável, os classificadores não-lineares, em geral, também melhoram em acurácia com bagging. Essa melhora se mostra menos evidenciada no Cenário D, onde os erros encontra-se muito próximos do erro ótimo de Bayes.

Tabela 10.2: Diferenças entre as menores taxas de erro de bagging comparadas a média dos erros das repetições (cenário A).

Tamanho da Amostra	Classificador	e_R	$e_B(B)$	$((e_B/e_R)-1)\%$
$n = 100$	lda	$0,00872$	$0,00866(25)$	$-0,69$
	qda	$0,00428$	$0,00424(50)$	$-0,93$
	adlog	$0,00748$	$0,00660(50)$	$-11,76$
	k-nn	$0,00488$	$0,00490(50)$	$0,41$
	mlp	$0,00484$	$0,00462(50)$	$-4,55$
	Bayes		$0,00296$	
$n = 500$	lda	$0,00864$	$0,00860(50)$	$-0,37$
	qda	$0,00292$	$0,00293(50)$	$0,14$
	adlog	$0,00463$	$0,00431(25)$	$-6,91$
	k-nn	$0,00412$	$0,00400(100)$	$-2,92$
	mlp	$0,00375$	$0,00370(50)$	$-1,39$
	Bayes		$0,00278$	
$n = 1000$	lda	$0,00821$	$0,00818(25)$	$-0,41$
	qda	$0,00298$	$0,00298(50)$	$0,00$
	adlog	$0,00402$	$0,00390(50)$	$-2,84$
	k-nn	$0,00378$	$0,00363(50)$	$-3,92$
	mlp	$0,00353$	$0,00348(50)$	$-1,30$
	Bayes		$0,00292$	

Tabela 10.3: Diferenças entre as menores taxas de erro de bagging comparadas a média dos erros das repetições (cenário B).

Tamanho da Amostra	Classificador	e_R	$e_B(B)$	$((e_B/e_R)-1)\%$
$n = 100$	lda	$0,42174$	$0,41472(10)$	$-1,66$
	qda	$0,41964$	$0,41502(10)$	$-1,10$
	adlog	$0,36848$	$0,37072(100)$	$0,61$
	k-nn	$0,22534$	$0,22340(100)$	$-0,86$
	mlp	$0,41342$	$0,40110(10)$	$-2,98$
	Bayes		$0,14550$	
$n = 500$	lda	$0,43957$	$0,43705(10)$	$-0,57$
	qda	$0,43551$	$0,43245(10)$	$-0,70$
	adlog	$0,35200$	$0,35283(100)$	$0,24$
	k-nn	$0,18744$	$0,18047(100)$	$-3,72$
	mlp	$0,20680$	$0,15798(100)$	$-23,61$
	Bayes		$0,14740$	
$n = 1000$	lda	$0,44518$	$0,44362(10)$	$-0,35$
	qda	$0,44247$	$0,44036(10)$	$-0,48$
	adlog	$0,34951$	$0,34995(100)$	$0,13$
	k-nn	$0,18398$	$0,17610(100)$	$-4,28$
	mlp	$0,21001$	$0,15181(100)$	$-27,71$
	Bayes		$0,14755$	

Tabela 10.4: Diferenças entre as menores taxas de erro de bagging comparadas a média dos erros das repetições (cenário C).

Tamanho da Amostra	Classificador	e_R	$e_B(B)$	$((e_B/e_R)-1)\%$
$n = 100$	lda	$0,49582$	$0,47612(100)$	$-3,97$
	qda	$0,23222$	$0,23018(100)$	$-0,88$
	adlog	$0,50990$	$0,50380(100)$	$-1,20$
	k-nn	$0,17664$	$0,17356(100)$	$-1,74$
	mlp	$0,33740$	$0,25394(100)$	$-24,74$
	Bayes		$0,11544$	
$n = 500$	lda	$0,49420$	$0,48571(100)$	$-1,72$
	qda	$0,20313$	$0,20286(100)$	$-0,13$
	adlog	$0,49776$	$0,49730(25)$	$-0,09$
	k-nn	$0,15060$	$0,14495(100)$	$-3,75$
	mlp	$0,15356$	$0,13715(100)$	$-10,69$
	Bayes		$0,11712$	
$n = 1000$	lda	$0,49648$	$0,49215(25)$	$-0,87$
	qda	$0,19978$	$0,19951(100)$	$-0,14$
	adlog	$0,49918$	$0,49899(10)$	$-0,04$
	k-nn	$0,14673$	$0,14061(100)$	$-4,17$
	mlp	$0,13891$	$0,12972(50)$	$-6,61$
	Bayes		$0,11682$	

Tabela 10.5: Diferenças entre as menores taxas de erro de bagging comparadas a média dos erros das repetições (cenário D).

Tamanho da Amostra	Classificador	e_R	$e_B(B)$	$((e_B/e_R)-1)\%$
$n = 100$	lda	$0,05788$	$0,05756(50)$	$-0,55$
	qda	$0,06078$	$0,06036(100)$	$-0,69$
	adlog	$0,06538$	$0,06502(100)$	$-0,55$
	k-nn	$0,06604$	$0,06482(100)$	$-1,85$
	mlp	$0,06426$	$0,06298(100)$	$-1,99$
	Bayes		$0,05472$	
$n = 500$	lda	$0,054764$	$0,05468(100)$	$-0,15$
	qda	$0,054836$	$0,05477(100)$	$-0,12$
	adlog	$0,057424$	$0,05719(100)$	$-0,41$
	k-nn	$0,057716$	$0,05714(100)$	$-1,01$
	mlp	$0,057120$	$0,05709(50)$	$-0,06$
	Bayes		$0,05372$	
$n = 1000$	lda	$0,05353$	$0,05348(100)$	$-0,09$
	qda	$0,05377$	$0,05376(100)$	$-0,02$
	adlog	$0,05621$	$0,05610(25)$	$-0,19$
	k-nn	$0,05660$	$0,05609(100)$	$-0,90$
	mlp	$0,05526$	$0,05535(50)$	$0,17$
	Bayes		$0,05301$	

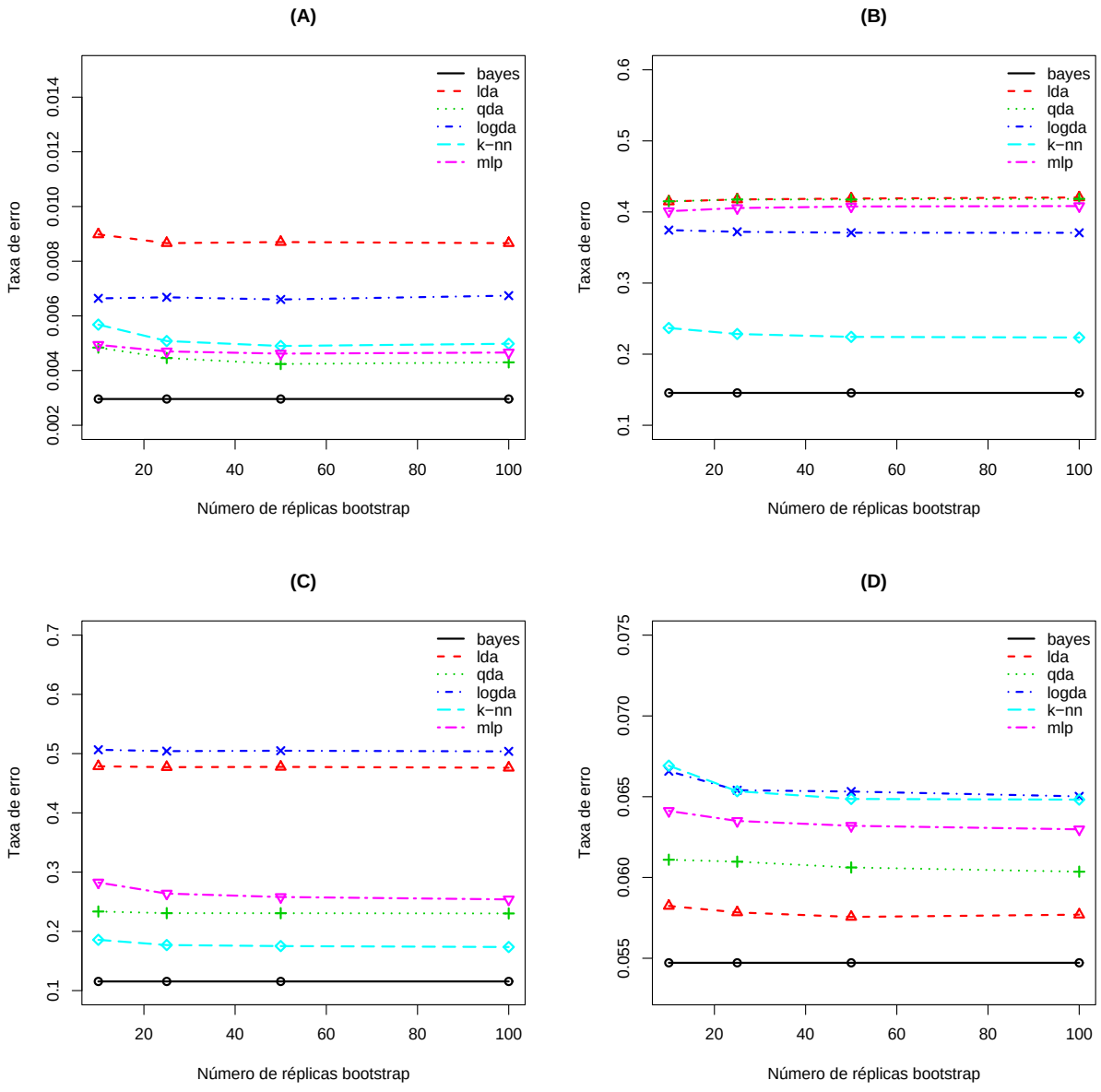


Figura 10.2: Médias das taxas de erro versus número de iterações, $n = 100$.

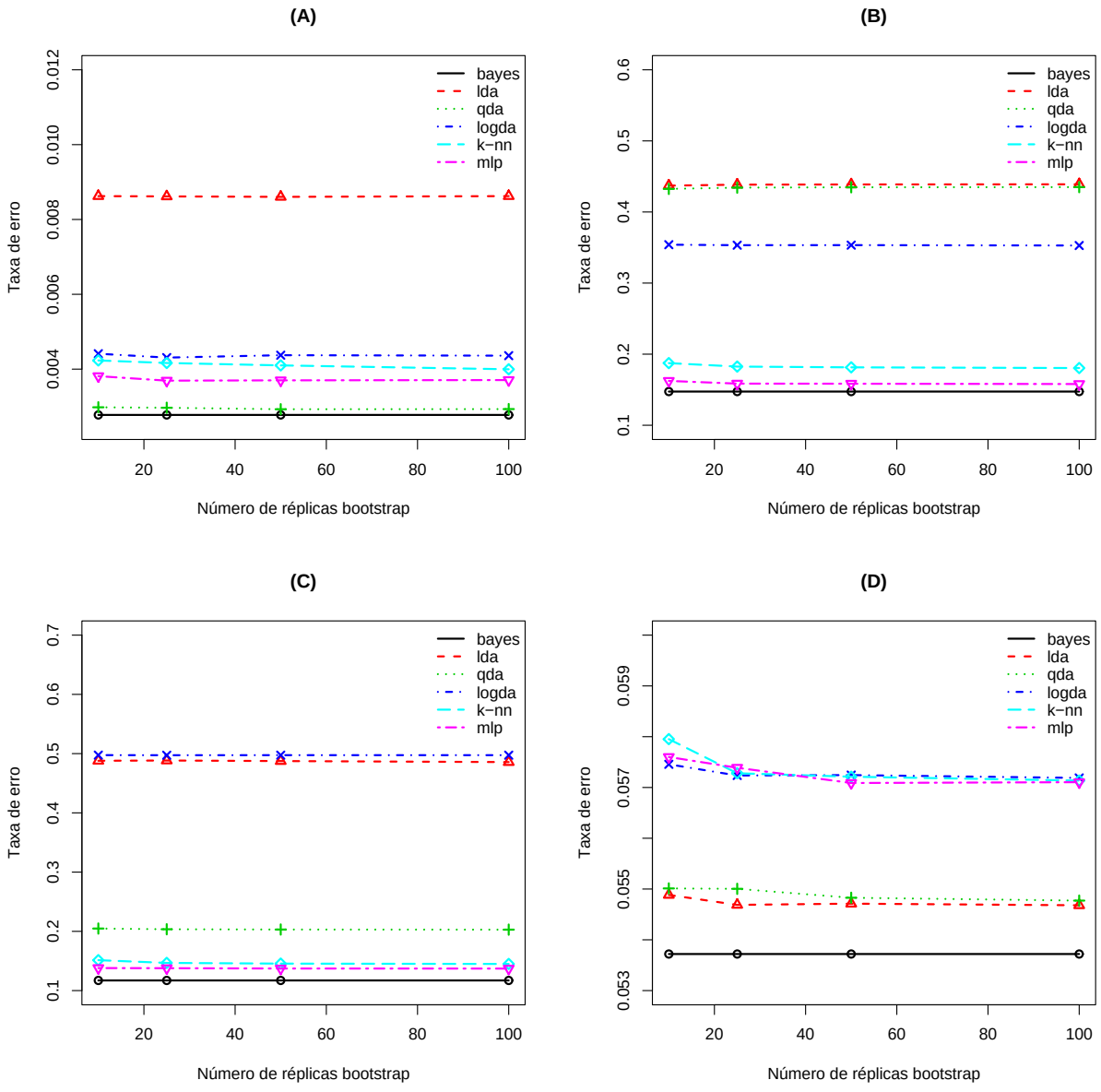


Figura 10.3: Médias das taxas de erro versus número de iterações, $n = 500$.

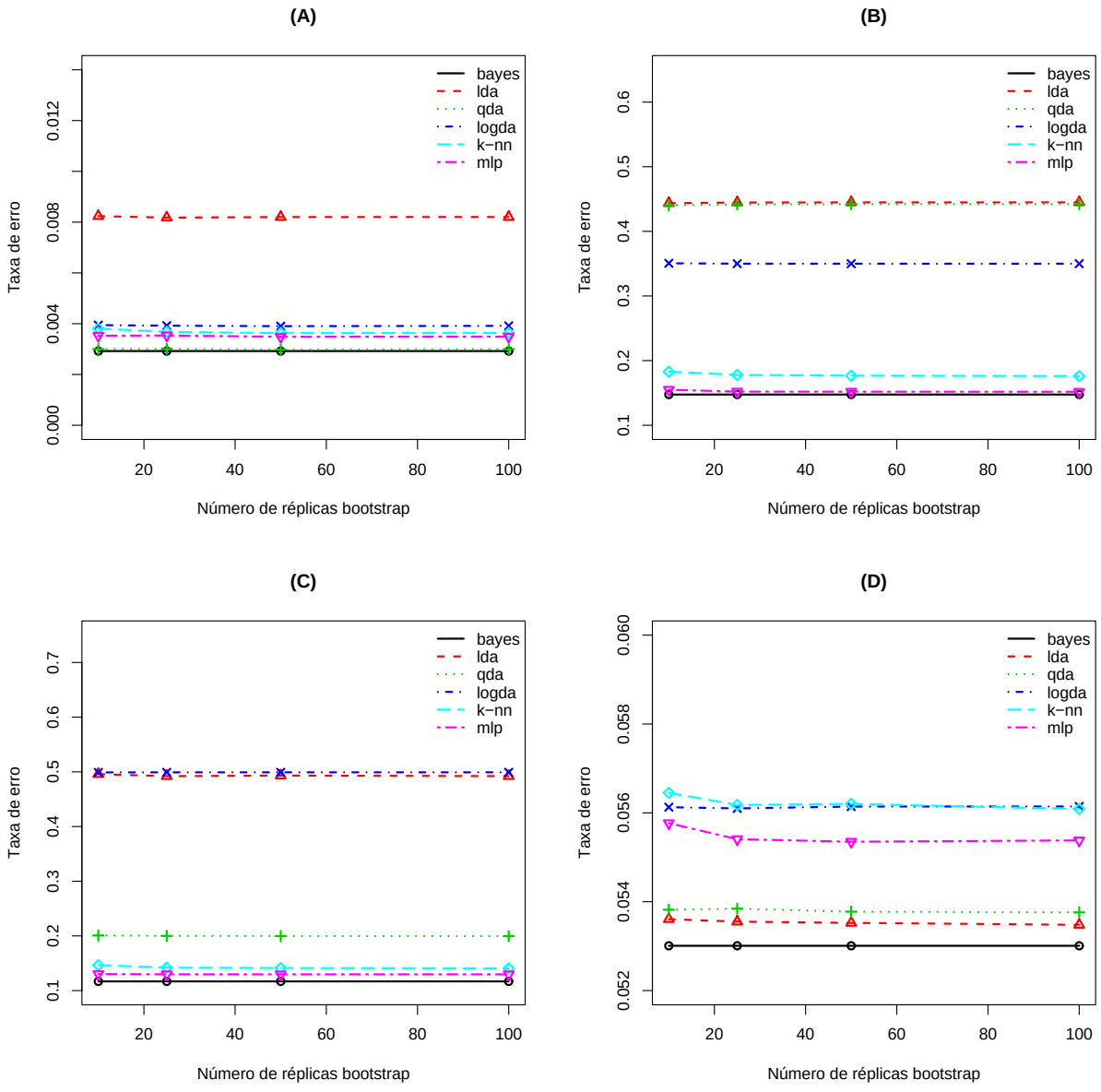


Figura 10.4: Médias das taxas de erro versus número de iterações, $n = 1000$.

CAPÍTULO 11

Suporte ao Diagnóstico de Câncer de Mama

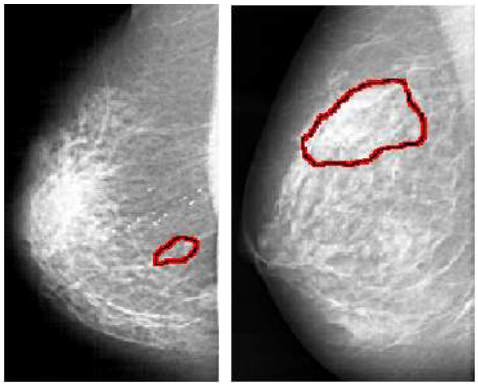
Neste capítulo apresentamos uma aplicação aos dados *Wisconsin Diagnostics Breast Cancer* (WDBC). O conjunto de dados contém 569 observações as quais 357 correspondem a casos benignos e 212 são casos malígnos. Existem 30 variáveis explicativas computadas a partir de características observadas em mamografias digitais. Estes dados estão disponíveis no repositório UCI (`http://www.ics.uci.edu/~mlearn/MLRepository`).

11.1 Introdução

Segundo o Instituto Nacional de Câncer (INCA, `http://www.inca.gov.br`), dentre as formas mais eficazes para detecção precoce do câncer de mama está a mamografia. A mamografia é a radiografia da mama que permite a detecção precoce do câncer, por ser capaz de mostrar lesões em fase inicial, muito pequenas (de milímetros). É realizada em um aparelho de raio X apropriado, chamado mamógrafo. Nele, a mama é comprimida de forma a fornecer melhores imagens, e, portanto, melhor capacidade de diagnóstico. O desconforto provocado é discreto e suportável. Porém, ainda segundo o INCA estudos sobre a efetividade da mamografia sempre utilizam o exame clínico como exame adicional e, indicam que existe dependência de fatores tais como: tamanho e localização da lesão,

densidade do tecido mamário (mulheres mais jovens apresentam mamas mais densas), qualidade dos recursos técnicos e habilidade de interpretação do radiologista.

Em casos de suspeita de câncer, o passo seguinte é a realização de uma biópsia. Na biópsia, a paciente é submetida a uma pequena cirurgia para retirada de parte do nódulo suspeito ou de sua totalidade, que é posteriormente encaminhada para exame. Uma paciente pode precisar ser submetida a inúmeras biopsias. Para evitar tal procedimento doloroso, dispomos de técnicas automátizadas de diagnóstico em sistemas de reconhecimento de padrões, que auxiliam o médico na tomada de decisão, funcionando como uma segunda opinião. Esse processo pode poupar a paciente de uma série de exames desgastantes desnecessários.



(a) (b)

Figura 11.1: Radiografias da mama com lesões suspeitas identificadas: (a) tumor benígno e (b) tumor malígno.

A detecção de lesões envolve a localização pelo computador de regiões contendo padrões radiológicos suspeitos, porém com a classificação da lesão realizada exclusivamente pelo radiologista. Sistemas para auxílio à detecção têm sido desenvolvidos principalmente para imagens de tórax e de mama. Segundo a literatura especializada, de 30% a 50% dos casos de câncer de mama detectados por meio de mamografia apresentam agrupamentos de microcalcificações associados (Giger, 1999). Além disso, estudos demonstraram que 26% dos casos de câncer de mama não-palpável apresentam nódulos associados na mamografia e 18% apresentam nódulos e microcalcificações (Sickles, 1986). Por isso, a maioria dos

sistemas de auxílio ao diagnóstico em mamografia é voltado para a detecção de nódulos e microcalcificações (Giger, 1999).

A Figura 11.1 ilustra duas radiografias da mama. Onde (a) apresenta um caso de tumor benígno e (b) apresenta um caso de tumor malígno, disponível em `http://marathon.csee.usf.edu/Mammography/Database.html`.

11.2 Detalhes Metodológicos

Para cada classificador, as médias das taxas de erro simples e bagging, denotadas por e_R e e_B, respectivamente, foram obtidas utilizando divisões aleatórias do conjunto de dados em treinamento e teste. As taxas de erro e_B foram obtidas construindo classificadores usuais, a partir de B réplicas bootstrap geradas do conjunto de treinamento e agregando as previsões dos mesmos. Este processo foi repetido 100 vezes, como descrito no algoritmo apresentado no Capítulo 9 para $r = 100$. A média da taxa de erro e_R corresponde ao classificador treinado com o conjunto de treinamento e avaliado no conjunto de teste. Calculamos a sensibilidade para as 100 divisões aleatórias do WDBC e para os classificadores gerados via bootstrap, denotadas por S_R e S_B, respectivamente. Da mesma forma, a especificidade para cada caso são denotadas por E_R e E_B, respectivamente.

11.3 Resultados

A Tabela 11.1 exibe as médias das taxas de erro, sensibilidade e especificidade dos classificadores em estudo. Calculamos a taxa de erro e_B, $B = 10, 25, 50$ e 100, e a menor taxa obtida para cada classificador é apresentada nesta tabela.

Para o classificador baseado nos k-vizinhos mais próximos, tomamos, baseado em avaliações iniciais, $k = 3$ e 5. A distância euclidiana foi utilizada como métrica.

Construímos dois perceptrons de múltiplas camadas, ambos com uma única camada intermediária, utilizando o algoritmo de retropropagação do erro e a função de ativação sigmóide. A primeira rede neural tem taxa de aprendizado $\eta = 0,3$ e sete neurônios na

Tabela 11.1: Taxas de erro, sensibilidade e especificidade para o conjunto de dados WDBC.

Classificador	B	Taxa de erro			Sensibilidade		Especificidade	
		e_R	e_B	$((e_B/e_R)-1)\%$	S_R	S_B	E_R	E_B
lda	50	0,0705	0,0710	0,7092	0,9957	0,9958	0,8634	0,8623
qda	100	0,0478	0,0488	2,0921	0,9661	0,9686	0,9384	0,9339
adlog	100	0,0648	0,0575	-11,2654	0,9616	0,9790	0,9089	0,9060
3-nn	25	0,0904	0,0903	-0,1106	0,9651	0,9677	0,8541	0,8518
5-nn	100	0,0898	0,0898	-0,1112	0,9675	0,9706	0,8528	0,8499
mlp(30-7-1) $\eta = 0,3$	100	0,0793	0,0782	-1,38713	0,9593	0,9722	0,8822	0,8715
mlp(30-3-1) $\eta = 0,1$	100	0,0899	0,0704	-21,6908	0,9134	0,9746	0,9069	0,8846

camada intermediária. A segunda rede possui três neurônios e tem taxa de aprendizado de $\eta = 0,1$. Ambas as redes não utilizam coeficiente de momento. Os pesos foram iniciados aleatoriamente no intervalo $(-0,1; 0,1)$. O critério de parada utilizado consiste em interromper o processo ao final de 5000 iterações ou quando o valor da função no passo t menos o valor da função no passo $t-1$ for menor que 10^{-8}, o que ocorrer primeiro. A escolha da estrutura das redes utilizadas foi baseada em avaliações iniciais.

O classificador qda apresentou melhor desempenho em relação aos demais classificadores, apresentando taxa de erro média para as 100 partições do WDBC igual a 4,78%. Para esse classificador, a aplicação do algoritmo bagging não resultou ganho em acurácia. A menor taxa média de erro bagging ocorreu quando utilizamos 100 réplicas do conjunto de treinamento, porém o algoritmo produziu um acréscimo de 2,09% em relação ao erro da repetição.

O uso do algoritmo bagging foi eficaz no melhoramento de desempenho para os classificadores adlog (queda de 11,26% em e_R) e mlp (queda de 1,39% e 21,69% em e_R). Os classificadores que utilizam os k-vizinhos mais próximos como método de classificação (3-nn e 5-nn) apresentaram uma redução de menos de 1% em e_R.

Bagging produziu uma melhora da sensibilidade de todos os classificadores, em particular para o classificador mlp, a sensibilidade foi de 91,34%, enquanto que para o classificador agregado S_B foi de 97,46%. Por outro lado, bagging produziu uma queda na

especificidade dos classificadores avaliados.

CAPÍTULO 12

Conclusões

Como descrevemos anteriormente, métodos automáticos de diagnóstico vêm sendo bastante explorados na prática médica. Em geral, na prática, busca-se por um melhor modelo, aquele que melhor represente o problema em questão. Porém, problemas complexos de classificação podem ser melhor representados por métodos de classificação mais elaborados. Uma alternativa consiste em utilizar métodos de combinação de classificadores tais como bagging, que pode ser utilizado como suporte à decisão com o proposito de melhorar o desempenho de classificadores utilizados de forma convencional.

Em geral, estudos empíricos são realizados para avaliar a capacidade de generalização dos principais métodos de agregação. O objetivo desta dissertação foi investigar o uso de bagging em diferentes cenários ou distribuições das populações através de simulações estocásticas.

A avaliação numérica revelou que o desempenho do bagging depende do comportamento de separação das categorias. No caso em que a separação se dá até mesmo de forma linear (Cenário A), por exemplo, todos os classificadores lineares têm seus desempenhos melhorados. No caso em que a separação linear é inviável, os classificadores não-lineares também melhoram em acurácia com bagging. Adicionalmente, as simulações realizadas indicam que bagging melhora classificadores estáves, tais como análise discriminate. En-

tretanto, esta melhora mostrou ser, na maioria das vezes, pequena e algumas vezes o uso de bootstrap não resultou em uma queda na taxa de erro, ou provocou um aumento na taxa de erro para classificadores estáveis.

O classificador baseado em redes neurais artificiais apresentou melhores resultados quanto a redução na taxa de erro. Para este classificador, o uso de bagging não apresentou, em geral, taxas de erro bagging maiores que as taxas de erro simples e_R tanto nas simulações estocásticas quanto na aplicação a dados reais.

Adicionalmente, a aplicação realizada mostrou que bagging produziu uma redução na taxa de erro para os classificadores baseados na análise discriminante logística e redes neurais artificais, além de produzir um melhoramento na sensibilidade para todos os classificadores em estudo.

Como sugestão de trabalhos futuros, podem ser investigados o bagging aplicado a outras redes neurais utilizadas em tarefas de classificação como a rede RBF, além de variantes do algoritmo de retropropagação do erro. Adicionalmente, outras distribuições das populações podem ser investigadas.

APÊNDICE A

Programa em R

Neste apêndice apresentamos os programas que deram origem aos resultados numéricos inseridos nesta dissertação. Os resultados das simulações referentes aos erros médios para as 1000 réplicas de Monte Carlo foram obtidos mediante o programa apresentado em A.1. As taxas de erro por bootstrap foram obtidas a partir do programa `prog_sim_bagging.R` apresentado em A.2. Ambos os programas foram escritos na linguagem `R`.

A.1 Estimação do Erro via Repetição do Algoritmo

```
######################################################################
# PROGRAMA: prog_sim_rep.R
# USO: Avalia o desempenho de lda, qda, logistica,
#      k-nn e mlp, com relacao a taxa de erro, sob diferentes
#      cenarios via simulacao de Monte Carlo.
######################################################################

rm(list=ls(all=TRUE)) # Remove os objetos


# Pacotes necessarios
library(mvtnorm) # geracao de normal multivariada
library(MASS)    # lda, qda
library(ks)      # geracao de normal mista
```

```
library(class)   # knn
library(AMORE)   # mlp

# Classificador
class = "mlp"

# Cenario de Simulacao
cenario <- "B"

# Numero de replicas de Monte Carlo
r <- 1000

# Tamanho do conjunto de treinamento
L <- 500

# Tamanho do conjunto de teste
T <- 500

# Tamanho da amostra
N <- L + T


# numero de vizinhos mais proximos
kv <- 3

# numero de neuronios na camada intermediaria
nn <- 3

# Taxa de aprendizado
eta <- 0.01


# vetor de medias da taxa de erros da repeticoes para r repeticoes de Monte Carlo
erro_rep <- matrix(NA, r, 1)


########################### Cenario 1 ################################
if (cenario == "A"){
mu1 <- c(1,-1)
mu2 <- c(-1,1)
sigma1 <- rbind(c(4/9, 14/45), c(14/45,4/9))
sigma2 <- rbind(c(4/9, 0), c(0,4/9))
}
```

```
#########################################################################

########################### Cenario 2 ###################################
if (cenario == "B"){
mu1 <- rbind(c(-3/2,-3/2), c(1/2,1/2))
mu2 <- rbind(c(3/2,3/2), c(-1/2,-1/2))
sigma1 <- rbind(rbind(c(4/5, -1/2), c(-1/2,4/5)), rbind(c(4/5, -1/2), c(-1/2,4/5)))
sigma2 <- rbind(rbind(c(4/5, -1/2), c(-1/2,4/5)), rbind(c(4/5, -1/2), c(-1/2,4/5)))
props1 <- c(1/2,1/2)
props2 <- c(1/2,1/2)
}
#########################################################################

########################### Cenario 3 ###################################
if (cenario == "C"){
mu1 <- rbind(c(-3/2, 0), c(3/2,0))
mu2 <- c(0,0)
sigma1 <- rbind(rbind(c(3/10, 1/4), c(1/4, 3/10)), rbind(c(3/10, 1/4), c(1/4, 3/10)))
sigma2 <- rbind(c(4/5, 2/5), c(2/5,1))
props1 <- c(1/2,1/2)
}
#########################################################################

########################### Cenario 4 ###################################
if (cenario == "D"){
mu1 <- c(0,1)
mu2 <- c(0,-1)
sigma1 <- rbind(c(2/3, 1/5), c(1/5,4/9))
sigma2 <- rbind(c(2/3, 1/5), c(1/5,4/9))
}
#########################################################################


# laço de Monte Carlo
for (i in (1:r)){

set.seed(i, kind = "Marsaglia-Multicarry")

if (cenario == "A"){
################ Gerando o conjunto de dados ########################
X1 <- rmvnorm(n = N/2, mu1, sigma1)
X2 <- rmvnorm(n = N/2, mu2, sigma2)
X <- rbind(X1, X2)
```

```
Y <- c(rep(0, N/2), rep(1, N/2))
#########################################################################
}
if (cenario == "B"){
################ Gerando o conjunto de dados ######################
X1 <- rmvnorm.mixt(n = N/2, mu1, sigma1, props1)
X2 <- rmvnorm.mixt(n = N/2, mu2, sigma2, props2)
X  <- rbind(X1, X2)
Y  <- c(rep(0, N/2), rep(1, N/2))
#########################################################################
}

if (cenario == "C"){
################ Gerando o conjunto de dados ######################
X1 <- rmvnorm.mixt(n = N/2, mu1, sigma1, props1)
X2 <- rmvnorm(n = N/2, mu2, sigma2)
X  <- rbind(X1, X2)
Y  <- c(rep(0, N/2), rep(1, N/2))
#########################################################################
}

if (cenario == "D"){
################ Gerando o conjunto de dados ######################
X1 <- rmvnorm(n = N/2, mu1, sigma1)
X2 <- rmvnorm(n = N/2, mu2, sigma2)
X  <- rbind(X1, X2)
Y  <- c(rep(0, N/2), rep(1, N/2))
#########################################################################
}

################## Cria data frame ################################
dados <- data.frame(Y, X)
#########################################################################

treinamento <- c(sample(1:(N/2), (L/2)), sample(((N/2)+1):N,(L/2)))

###################### Classificadores Simples ######################

if(class == "lda"){
################ Classificacao com lda ############################
ADL <- lda(Y ~.,  dados, subset = treinamento)
pred <- table(predict(ADL, dados[-treinamento, ])$class, dados[-treinamento,1])
erro_rep[i] <- (pred[1,2] + pred[2,1])/T
```

```
#########################################################################
}

if(class == "qda"){
#################### Classificacao com qda ##########################
ADQ <- qda(Y ~., dados, subset = treinamento)

pred <- table(predict(ADQ, dados[-treinamento, ])$class, dados[-treinamento,1])
erro_rep[i] <- (pred[1,2] + pred[2,1])/T
#########################################################################
}

if(class == "adlog"){
#################### Classificacao com adlog ########################
ADLOG <- glm(Y ~.,  family = binomial(link = logit), dados, subset = treinamento)
pred <- predict(ADLOG, dados[-treinamento, 2:3])

pred1 <- matrix(T,1)

# Transforma as probs em 0 ou 1
for(j in 1:T)
{
        if(pred[j] < 0.5)
        {
            pred1[j] <- 1
        }
        if (pred[j] > 0.5)
        {
            pred1[j] <- 2
        }
}
pred2 <- table(pred1, dados[-treinamento,1])

if((pred2[1,1] + pred2[1,2]) == T){
       if(pred2[1,1] == (T/2)){
    erro_rep[i] <- 0.5
    }
    }
    else{
    erro_rep[i] <- (pred2[1,2] + pred2[2,1])/T
    }

#########################################################################
```

```
}

if(class == "knn"){
#################### K - NN ########################
KNN <- knn(dados[treinamento,2:3], dados[-treinamento,2:3], dados[treinamento,1], k = kv)
pred <- table(KNN, dados[-treinamento,1])
erro_rep[i] <- (pred[1,2] + pred[2,1])/T
#####################################################
}

if(class == "mlp"){
#################### Classificacao com MLP ########################
rede <- nnet(dados[treinamento, 2:3], dados[treinamento,1], size = nn, rang = 0.1,
                decay = eta, maxit = 5000)

pred <- predict(rede, dados[-treinamento,2:3])

pred1 <- matrix(T,1)

# Transforma as probs em 1 ou 2
for(j in 1:T)
{
        if(pred[j] < 0.5)
        {
            pred1[j] <- 1
        }
        if (pred[j] > 0.5)
        {
            pred1[j] <- 2
        }
}

pred2 <- table(pred1, dados[-treinamento,1])

if((pred2[1,1] + pred2[1,2]) == T){
       if(pred2[1,1] == (T/2)){
    erro_rep[i] <- 0.5
    }
    }
    else{
    erro_rep[i] <- (pred2[1,2] + pred2[2,1])/T
    }
```

```
########################################################################
}

}
# Fim do laco de Monte Carlo

mean(erro_rep) # Media erro da repeticao de MC
```

A.2 Estimação do Erro via Agregação por Bootstrap

```
#####################################################################
# PROGRAMA: prog_sim_bagging.R
# USO: Avalia o desempenho de bagging lda, qda, logistica,
#      k-nn e mlp, com relacao a taxa de erro, sob diferentes
#      cenarios via simulacao de Monte Carlo.
#####################################################################

rm(list=ls(all=TRUE)) # Remove os objetos

# Pacotes necessarios
library(mvtnorm) # geracao de normal multivariada
library(MASS)    # lda, qda
library(ks)      # geracao de normal mista
library(class)   # knn
library(nnet)   # mlp

# Classificador
class <- "mlp"

# Cenario de Simulacao
cenario <- "A"

# numero de replicas bootstrap
B <- 10

# Tamanho do conjunto de treinamento
L <- 50

# Tamanho do conjunto de teste
T <- 50

# Tamanho da amostra
N <- L + T



# numero de neuronios na camada intermediaria
nn <- 2

# Taxa de aprendizado
eta <- 0.01
```

```
# numero de vizinhos mais proximos
kv <- 3

# Numero de replicas de Monte Carlo
r = 1000

# vetor da taxa de erros bagging para r repeticoes de Monte Carlo
erro_bagging   <- matrix(NA, r, 1)


########################### Cenario 1 ###################################
if (cenario == "A"){
mu1 <- c(1,-1)
mu2 <- c(-1,1)
sigma1 <- rbind(c(4/9, 14/45), c(14/45,4/9))
sigma2 <- rbind(c(4/9, 0), c(0,4/9))
}
#########################################################################

########################### Cenario 2 ###################################
if (cenario == "B"){
mu1 <- rbind(c(-3/2,-3/2), c(1/2,1/2))
mu2 <- rbind(c(3/2,3/2), c(-1/2,-1/2))
sigma1 <- rbind(rbind(c(4/5, -1/2), c(-1/2,4/5)), rbind(c(4/5, -1/2), c(-1/2,4/5)))
sigma2 <- rbind(rbind(c(4/5, -1/2), c(-1/2,4/5)), rbind(c(4/5, -1/2), c(-1/2,4/5)))
props1 <- c(1/2,1/2)
props2 <- c(1/2,1/2)
}
#########################################################################

########################### Cenario 3 ###################################
if (cenario == "C"){
mu1 <- rbind(c(-3/2, 0), c(3/2,0))
mu2 <- c(0,0)
sigma1 <- rbind(rbind(c(3/10, 1/4), c(1/4, 3/10)), rbind(c(3/10, 1/4), c(1/4, 3/10)))
sigma2 <- rbind(c(4/5, 2/5), c(2/5,1))
props1 <- c(1/2,1/2)
}
#########################################################################

########################### Cenario 4 ###################################
if (cenario == "D"){
mu1 <- c(0,1)
```

```
mu2 <- c(0,-1)
sigma1 <- rbind(c(2/3, 1/5), c(1/5,4/9))
sigma2 <- rbind(c(2/3, 1/5), c(1/5,4/9))
}
#########################################################################


# laço de Monte Carlo
for (i in (1:r)){

# garante que as amostras tomadas aqui sejam as mesmas das geradas
# no programa prog_sim_rep.R
set.seed(i, kind = "Marsaglia-Multicarry")

if (cenario == "A"){
################ Gerando o conjunto de dados ########################
X1 <- rmvnorm(n = N/2, mu1, sigma1)
X2 <- rmvnorm(n = N/2, mu2, sigma2)
X <- rbind(X1, X2)
Y <- c(rep(0, N/2), rep(1, N/2))
#########################################################################
}
if (cenario == "B"){
################ Gerando o conjunto de dados ########################
X1 <- rmvnorm.mixt(n = N/2, mu1, sigma1, props1)
X2 <- rmvnorm.mixt(n = N/2, mu2, sigma2, props2)
X  <- rbind(X1, X2)
Y  <- c(rep(0, N/2), rep(1, N/2))
#########################################################################
}

if (cenario == "C"){
################ Gerando o conjunto de dados ########################
X1 <- rmvnorm.mixt(n = N/2, mu1, sigma1, props1)
X2 <- rmvnorm(n = N/2, mu2, sigma2)
X  <- rbind(X1, X2)
Y  <- c(rep(0, N/2), rep(1, N/2))
#########################################################################
}

if (cenario == "D"){
################ Gerando o conjunto de dados ########################
X1 <- rmvnorm(n = N/2, mu1, sigma1)
```

```
X2 <- rmvnorm(n = N/2, mu2, sigma2)
X  <- rbind(X1, X2)
Y  <- c(rep(0, N/2), rep(1, N/2))
#########################################################################
}

################## Cria data frame ###################################
dados <- data.frame(Y, X)
#########################################################################

treinamento <- c(sample(1:(N/2), (L/2)), sample(((N/2)+1):N,(L/2)))


# matriz de resultados
resultados_bootstrap <- matrix(NA, T, B)

# Laco para as B replicas de Bootstrap
for (b in 1:B)
{

# Amostra Bootstrap(com reposicao do treinamento)
treina_bootstrap <- sample(treinamento, replace = TRUE)

#### LDA ####
if(class == "lda"){
################# Classificacao com lda ##############################
ADL <- lda(Y ~.,  dados, subset = treina_bootstrap)
resultados_bootstrap[,b] <- predict(ADL, dados[-treinamento, ])$class
#########################################################################
}

if(class == "qda"){
################### Classificacao com qda ############################
ADQ <- qda(Y ~., dados, subset = treina_bootstrap)
resultados_bootstrap[,b] <- predict(ADQ, dados[-treinamento,])$class
#########################################################################
}


if(class == "adlog"){
################### Classificacao com adlog ##########################
ADLOG <- glm(Y ~.,  family = binomial(link = logit), dados, subset = treina_bootstrap)
pred_adlog <- predict(ADLOG, dados[-treinamento, 2:3])
```

```
# Transforma as probs em 0 ou 1
for(j in 1:T)
{
        if(pred_adlog[j] < 0.5)
        {
            resultados_bootstrap[j,b] <- 1
        }
        if (pred_adlog[j] > 0.5)
        {
            resultados_bootstrap[j,b] <- 2
        }
}

########################################################################
}

if(class == "knn"){
#################### K - NN ########################
KNN <- knn(dados[treina_bootstrap,2:3], dados[-treinamento,2:3], dados[treina_bootstrap,1], k = kv)
resultados_bootstrap[,b] <- KNN
#######################################################
}

if(class == "mlp"){
#################### Classificacao com MLP ########################
rede <- nnet(dados[treina_bootstrap, 2:3], dados[treina_bootstrap,1], size = nn, rang = 0.1,
                decay = eta, maxit = 5000)

pred_mlp <- predict(rede, dados[-treinamento,2:3])



# Transforma as probs em 0 ou 1
for(j in 1:T)
{
        if(pred_mlp[j] < 0.5)
        {
            resultados_bootstrap[j,b] <- 1
        }
        if (pred_mlp[j] > 0.5)
        {
            resultados_bootstrap[j,b] <- 2
```

```
        }
}

#########################################################################
}

}
# Fim do laco para as B replicas de Bootstrap


# Classificacao por voto majoritario
cont1 <- matrix (0, T, 1)
cont2 <- matrix (0, T, 1)

for (k in 1:T)
{
    for(j in 1:B)
    {
        if(resultados_bootstrap[k,j] == 1)
        {
                cont1[k] <- cont1[k] + 1
        }
        if (resultados_bootstrap[k,j] == 2)
        {
            cont2[k] <- cont2[k] + 1
        }

    }

}

resultado_classificacao <- matrix(NA, T,1)
for (k in 1:T)
{
           if(cont1[k] > cont2[k])
           {
            resultado_classificacao[k] <- 1
           }

           if(cont1[k] < cont2[k])
           {
            resultado_classificacao[k] <- 2
           }
```

```
            # Empate aleatorio
            if(cont1[k] == cont2[k])
            {
                 r <- runif(1)
                 if(r > 0.5)
                 {
                  resultado_classificacao[k] <- 1
                 }
                 if(r <= 0.5)
                 {
                 resultado_classificacao[k] <- 2
                 }
             }

}

predictbag <- table(resultado_classificacao, dados[-treinamento,1])

if((predictbag[1,1] + predictbag [1,2]) == T){
       if(predictbag[1,1] == (T/2)){
    erro_bagging[i] <- 0.5
    }
    }
    else{
    erro_bagging[i] <- (predictbag[1,2] + predictbag [2,1])/T
    }

}
# Fim do laco de Monte Carlo

mean(erro_bagging)  # Media erro bagging
```

APÊNDICE B

Descrição do conjunto de Dados

Neste apêndice apresentamos uma descrição do conjunto de dados utilizado na aplicação. Esta descrição foi extraída em sua totalidade do repositório UCI de dados (`http://www.ics.uci.edu/~mlearn/MLRepository`) e encontra-se disponível juntamente com conjunto de dados WDBC.

B.1 Wisconsin Diagnostic Breast Cancer

```
1. Title: Wisconsin Diagnostic Breast Cancer (WDBC)

2. Source Information

a) Creators:

    Dr. William H. Wolberg, General Surgery Dept., University of
    Wisconsin,  Clinical Sciences Center, Madison, WI 53792
    wolberg@eagle.surgery.wisc.edu

    W. Nick Street, Computer Sciences Dept., University of
    Wisconsin, 1210 West Dayton St., Madison, WI 53706
    street@cs.wisc.edu  608-262-6619
```

Olvi L. Mangasarian, Computer Sciences Dept., University of
Wisconsin, 1210 West Dayton St., Madison, WI 53706
olvi@cs.wisc.edu

b) Donor: Nick Street

c) Date: November 1995

3. Past Usage:

first usage:

W.N. Street, W.H. Wolberg and O.L. Mangasarian
Nuclear feature extraction for breast tumor diagnosis.
IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science
and Technology, volume 1905, pages 861-870, San Jose, CA, 1993.

OR literature:

O.L. Mangasarian, W.N. Street and W.H. Wolberg.
Breast cancer diagnosis and prognosis via linear programming.
Operations Research, 43(4), pages 570-577, July-August 1995.

Medical literature:

W.H. Wolberg, W.N. Street, and O.L. Mangasarian.
Machine learning techniques to diagnose breast cancer from
fine-needle aspirates.
Cancer Letters 77 (1994) 163-171.

W.H. Wolberg, W.N. Street, and O.L. Mangasarian.
Image analysis and machine learning applied to breast cancer
diagnosis and prognosis.
Analytical and Quantitative Cytology and Histology, Vol. 17
No. 2, pages 77-87, April 1995.

W.H. Wolberg, W.N. Street, D.M. Heisey, and O.L. Mangasarian.
Computerized breast cancer diagnosis and prognosis from fine
needle aspirates.
Archives of Surgery 1995;130:511-516.

W.H. Wolberg, W.N. Street, D.M. Heisey, and O.L. Mangasarian.
Computer-derived nuclear features distinguish malignant from

benign breast cytology.
Human Pathology, 26:792--796, 1995.

See also:
http://www.cs.wisc.edu/~olvi/uwmp/mpml.html
http://www.cs.wisc.edu/~olvi/uwmp/cancer.html

Results:

- predicting field 2, diagnosis: B = benign, M = malignant
- sets are linearly separable using all 30 input features
- best predictive accuracy obtained using one separating plane
  in the 3-D space of Worst Area, Worst Smoothness and
  Mean Texture. Estimated accuracy 97.5% using repeated
  10-fold crossvalidations. Classifier has correctly
  diagnosed 176 consecutive new patients as of November
  1995.

4. Relevant information

Features are computed from a digitized image of a fine needle
aspirate (FNA) of a breast mass. They describe
characteristics of the cell nuclei present in the image.
A few of the images can be found at
http://www.cs.wisc.edu/~street/images/

Separating plane described above was obtained using
Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree
Construction Via Linear Programming." Proceedings of the 4th
Midwest Artificial Intelligence and Cognitive Science Society,
pp. 97-101, 1992], a classification method which uses linear
programming to construct a decision tree. Relevant features
were selected using an exhaustive search in the space of 1-4
features and 1-3 separating planes.

The actual linear program used to obtain the separating plane
in the 3-dimensional space is that described in:
[K. P. Bennett and O. L. Mangasarian: "Robust Linear
Programming Discrimination of Two Linearly Inseparable Sets",
Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

```
    ftp ftp.cs.wisc.edu
    cd math-prog/cpo-dataset/machine-learn/WDBC/

5. Number of instances: 569

6. Number of attributes: 32 (ID, diagnosis, 30 real-valued input features)

7. Attribute information

1) ID number
2) Diagnosis (M = malignant, B = benign)
3-32)

Ten real-valued features are computed for each cell nucleus:

    a) radius (mean of distances from center to points on the perimeter)
    b) texture (standard deviation of gray-scale values)
    c) perimeter
    d) area
    e) smoothness (local variation in radius lengths)
    f) compactness (perimeter^2 / area - 1.0)
    g) concavity (severity of concave portions of the contour)
    h) concave points (number of concave portions of the contour)
    i) symmetry
    j) fractal dimension ("coastline approximation" - 1)

Several of the papers listed above contain detailed descriptions of
how these features are computed.

The mean, standard error, and "worst" or largest (mean of the three
largest values) of these features were computed for each image,
resulting in 30 features.  For instance, field 3 is Mean Radius, field
13 is Radius SE, field 23 is Worst Radius.

All feature values are recoded with four significant digits.

8. Missing attribute values: none

9. Class distribution: 357 benign, 212 malignant
```

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Afifi A. A. & Clark V. (1996). *Computed-Aided Multivariate Analysis.* 3. ed., Chapman & Hall, London.

[2] Anagnostopoulos, I. & Maglogiannis, I. (2006). Neural Network-based diagnostic and prognostic estimations in breast cancer microscopic instances. *Med Bio Eng Comput.* 44, 773-784.

[3] Anderson, J. A., Pellionisz, A. & Rosenfeld, E. (1990). *Neurocomputing 2: Directions for Research.* MA: The MIT Press, Cambridge.

[4] Bishop, C. M.(1995). *Neural Networks for Pattern Recognition.* Oxford University Press, New York.

[5] Breiman, L. (1998). Arcing Classifiers. *The Annals of Statistics* , Vol. 26, 801-849.

[6] Breiman, L. (1996). Bagging Predictors. *Machine Learning*, Vol. 24. 123-140.

[7] Centor, R. M. (1991). Signal Detectability: The use of ROC curves and their Analyses. *Medical Decision Making.* 11, 54 - 115.

[8] Dalgaard, P. (2002). *Introductory Statistics with R*, Springer, New York.

[9] Dietterich, T. G. (2000). Ensemble Methodos Machine Learning. Procceding of the First International Workshop on Multiple Classifier Systems, Italy, *LNCS*, Vol. 1857: 1-15.

[10] Duda, R. O., Hart, P. E. & Stork, D. G. (2001). *Pattern Classification.* 2. ed. Wiley, New York.

[11] Duong, T. (2004). Bandwidth Selectors for Multivariate Kernel Density Estimation. Phd Thesis.

[12] Efron, B. and Tibshirani, R. (1993) *An Introduction to the Bootstrap.* New York, Chapman & Hall.

[13] Fahlman, S. E. (1989). Faster-learning Variations on Back-propagation: An Empirical Study. *Proceedings of the 1988 Coonnectionist Models Summer School, Pittsburg.* 38-51. CA: Morgan Kaufmann, San Mateo.

[14] Fisher, R. A. (1938). The Statistical Utilization of Multiple Measurements. *Annals of Eugenics.* 8, 376-386.

[15] Freeman, W. J. (1975). *Mass Action in the Nervous System.* Academic Press, New York.

[16] Freud, Y. & Schapire, R. E. (1996) Experiments with a new boosting algorithm. Proceedings of the Thirteenth International Conference on Machine Learning. 148-156.

[17] Friedman, I., Hastie, T., Tibshirani, R. (2000). Additive Logistic Regression: a Statistical View of Boosting. *The Annals of Statistics.*, Vol. 28, 337-407.

[18] Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition.* 2. ed., Academic Press, Boston.

[19] Giger, ML. (1999). Computer-aided diagnosis. *Categorial Course in Breast Imaging.* 249-72.

[20] Haykin, S.(2001). *Redes Neurais: Princípios e Práticas.* 2. ed., Bookman, Porto Alegre.

[21] Jain, A.K., Duin, R.P.W. & Mao, J. (2000). Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* vol 22, 4-37.

[22] Johnson, R. A. & Wichern, D. W. (1998). *Applied Multivariate Statistical Analysis.* 4. ed., Prentice Hall, New Jersey.

[23] Kandel, E. R. & Schwartz, J. H. (1991). *Principles of Neural Science.* 3. ed., Elsevier, New York.

[24] Knuth, D. (1986). *The TEXbook.* Adisson-Wesley, New York.

[25] Koch, C. & Segev, I. (1989). *Methods in Neuronal Modeling: From Synapses to Networks.* MA: MIT Press, Cambridge.

[26] Kuffler, S. W., Nicholls, J. G. & Martin, A. R. (1984). *From Neuron to Brain: A Cellular Approach to the Function of the Nervous System.* 2. ed., MA: Sinauer Associates, Suderland.

[27] Marques de Sá, J.P.(2001). *Pattern Recognition: Concepts, Methods and Applications.* Springer.

[28] McCullagh, P. & Nelder, J. A. (1989). *Generalized Linear Models.* 2. ed., Chapman & Hall, London.

[29] McCulloch, W. S. & Pitts, W. (1943). A Logical Calculus of Ideas Immanent In Nervous Activity. *Bulletin of Mathematical Biophysics.* 5, 115-133.

[30] Opitz, D. & Maclin, R. (1999). Popular Ensemble Methods: An Empirical Evaluated of Bagging and Boosting. *Journal of Artificial Intelligence Research*, vol. 11: 169-198.

[31] R Development Core Team (2006). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, `http://www.R-project.org`.

[32] Ripley, B.D.(1996). *Pattern Recognition and Neural Networks.* University Press, Cambridge.

[33] Ripley, B. D. (1993) Statistical aspects of neural networks. *Networks and Chaos - Statistical and Probabilistic Aspects*, eds O. E. Barndorff-Nielsen, J. L. Jensen and W. S. Kendall, pp. 40-123. London: Chapman & Hall.

[34] Rumelhart, D. E. & McClelland, J. L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Foundations. Vol. 1, MA: The MIT Press, Cambridge.

[35] Shepherd, G. M. (1990). *The Synaptic Organization of the Brain.* 3. ed., Oxford University Press, New York.

[36] Sickles, E.A. (1986). Mammographic features of 300 consecutive nonpalpable breast cancers. *AJR.* vol 3, 146-661.

[37] Swets, J. A. & Pickett, R. M. (1982) *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory.* Academic Press, New York.

[38] Tan A. C. & Gilbert D. (2003). Ensemble Machine Learning on Gene Expression Data for Carcer Classification. *Applied Bioinformatics.* Vol. 2, S75-S83.

[39] Theodoridis, S., Koutroumbas, K. (2003). *Pattern Recognition.* 2. ed., Elsevier Academic Press, San Diego.

[40] Venables W. N., Ripley, B.D.(2002). *Modern Applied Statistics with S.* 4. ed., Springer, New York.

[41] Webb, Andrew.(2002). *Statistical Pattern Recognition.* 2. ed., Wiley.

[42] West D., Dellana S. & Qian J. (2005). Neural Network Ensemble Strategies for Financial Decision Applications. *Computers & Operations Research* , Vol. 32. 2543-2559.