Renata Gomes Cordeiro

A Programação Linear na classificação de padrões

Campos dos Goytacazes/RJ

2013

Renata Gomes Cordeiro

A Programação Linear na classificação de padrões

Monografia apresentada ao Curso de Graduação em Ciência da Computação da Universidade Estadual do Norte Fluminense Darcy Ribeiro como requisito para obtenção do título de Bacharel em Ciência da Computação, sob orientação do Profº. Fermín Alfredo Tang Montané.

Tutor: Fermín Alfredo Tang Montané.

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO

Campos dos Goytacazes/RJ

2013

Sumário

1 Introdução

A programação linear é uma das disciplinas que compõem a programação matemática e constitui um dos pilares da pesquisa operacional. As aplicações da programação linear estão presentes em diversos setores, tais como nas indústrias, nos transportes, na saúde, na educação, na administração pública, na computação, etc. É mais comunente aplicada na engenharia de produção, em problemas que buscam a distribuição eficiente de recursos, minimização de gastos e maximização do lucro. O presente trabalho está focado na utilização da programação linear em aplicações que buscam a cliassifcação de dados em determiando padrão, por meio de classificadores gerados através de um modelo de programação lienar.

O método simplex proposto por Danzitg (1963) é um dos métodos mais conhecidos e eficientes para resolver problemas de programação linear. Trata-se de um dos poucos algoritmos que foi implantado comercialmente há mais de 40 anos. Atualmente, está presente em softwares comerciais tais como CPLEX[1] e LINGO[2]. O método simplex tem como principais carcaterísticas o fato de ser matricial, ou seja, aloca os dados a serem calculados em matrizes, de resolver o conjunto de equações, que formam o modelo de programação linear, de forma interativa até que a solução ótima seja obtida e de ser um método determinístico. Um método alternativo, teoricamente superior ao método simplex, é o método dos pontos interiores, proposto por Karmarkar (1984). Na prática, tanto o método simplex, quanto o método dos pontos interiores competem até hoje.

Um dos focos dos estudos das aplicações da programação linear é na utlização de separação de pontos. O modelo gera um hiperplano que separa conjuntos de pontos, sendo que cada conjunto pertence a um padrão. Na classificação de um conjunto é possivel conhecer o padrão ao qual o conjunto pertence através dos hiperplanos. Na computação esse tipo de aplicação pode ser utilizada em várias atividades com o objetivo de classificar dados, como por exemplo:

[1]http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/
[2]http://www.lindo.com/

- Diagnósticos através de imagens de exames médicos, como por exemplo se um tumor é benigno ou maligno.
- Classificação de expressões facias, que podem ser utlizadas em aplicações que buscam uma iteração tranparente homem - máquina.
- Classificação de espécies de plantas através de imagens.
- Classificação de genêros músicas através de arquivos de áudio.

De forma geral, a aplicação da programação linear na separação de pontos pode ser utilizada em abordagens que permitam a caracterização de padrões através de vetores númericos, esse vetores representam os pontos e são denominados vetores de características.

1.1 Objetivos e justificativas

O objetivo do presente trabalho é o estudo da utilização da programação linear, mais especificamente do método simplex, na separação de pontos, que repesentam padrões, através de hiperplanos, que serão utlizados na classificação de um conjunto em um determinado padrão.

Através desse estudo deve ser possível verificar a eficiência da programação linear nesse tipo de aplicação. São realizados testes com vetores gerados aleatóriamente e com dados reais para verificar a eficácia do modelo de programação linear na separação de padrões.

A programação linear possui aplicações em diversas áreas, como: indústria, produção, saúde e computação gráfica. Porém é um método mais comumente utilizado na engenharia de produção em problemas como: alocação de recursos e planejamento de produção. O presente trabalho justifica-se pelo fato de abordar uma aplicação prática dentro da computação, onde a aplicação da programação linear não é tão explorada quanto na engenharia de produção.

1.2 Metodologia

Para o cumprimento do obejtivo final, o trabalho é composto por algumas etapas:

- O estudo do método simplex revisado, suas características, vantagens do ponto de vista computacional
- Obtenção de dados para testes
- A implementação do modelo do problema de programação linear utilizando a linguagem de programação JAVA juntamente com o software CPLEX
- A implementação da etapa de classificação de um conjunto de dados que possui padrão inicialmente desconhecido
- Realização de testes

1.3 Estrutura do trabalho

REFAZER NO FINAL!!!! O presente trabalho apresenta a seguinte estrutura: o capítulo 2 apresenta uma descrição do problema geral de programação linear, os principais métodos de solução e algumas aplicações práticas; o capítulo 3 apresenta de forma detalhada o Método Simplex Revisado, foco deste trabalho; o capítulo 4 apresenta o processo de classificação de imagens e como a programação linear se aplica neste processo; no capítulo 5 serão apresentadas as ; e no capítulo 6 serão apresentadas as conclusões e possíveis trabalhos futuros.

1.4 A Programação Linear e as suas Aplicações

Na pesquisa operacional, a programação linear é uma das técnicas mais utilizadas em problemas de otimização. Os problemas de programação linear geralmente buscam a distribuição eficiente de recursos limitados para atender um determinado objetivo, por isso suas aplicações estão presentes em diversas áreas como computação, administração, indústria e transporte (PAMPLONA, 2005).

Um problema de programação linear é expresso através de um modelo que é composto por equações e inequações lineares. Esse tipo de problema busca a distribuição eficiente de recursos com restrições para alcançar um objetivo, em geral, maximizar lucros ou minimizar custos. Em um problema de programação linear esse objetivo é expresso através de uma equação linear denominada função objetivo. Para a formulação do problema, é necessário também definir os recursos necessários e em que proporção são requeridos. Essas informações são expressas em equações ou inequações lineares, uma para cada recurso. Esse conjunto de equações ou inequações é denominado restrições do modelo (PAMPLONA, 2005).

1.5 Descrição do Problema de Programação Linear

O modelo de um problema de programação linear normalmente é apresentado em uma das formas a seguir (PASSOS, 2009):

$$Max\ z = c^T x$$

$$s.a. \begin{cases} Ax \leq b \\ x \geq 0 \end{cases}$$

ou

$$Min\ z = c^T x$$

$$s.a. \begin{cases} Ax \geq b \\ x \geq 0 \end{cases}$$

Um problema de programação linear com até três variáveis pode ser representado graficamente utilizando três eixos cartesianos. Os problemas com duas variáveis podem ainda ser facilmente resolvidos por meio da representação gráfica (PASSOS, 2009).

A seguir é apresentado um problema com duas variáveis e sua representação. Apesar de, na prática os problemas de programação linear possuirem um número de variáveis muito maior que dois ou três, a visualização gráfica de modelo, mesmo que simples, contribui para o entendimento dos métodos de resolução apresentados nas seções a seguir. No problema exemplo, uma empresa, que fabrica vários produtos, deseja maximizar o lucro na venda de 2 desses produtos (HILLIER; LIEBERMAN, 2006).

$Maximize\ z = 3x_1 + 5x_2$
$Sujeito\ a$
$1x_1 \leq 4 \qquad (a)$
$2x_2 \leq \qquad\quad (b)$
$3x_1 + 2x_2 \leq 8 \quad (c)$
$x_1 \geq 0, x_2 \geq 0$

Onde,

- $\boldsymbol{x_1}$ representa a quantidade do produto 1 produzido em uma semana
- $\boldsymbol{x_2}$ representa a quantidade do produto 2 produzido em uma semana
- $\boldsymbol{z}$ representa o lucro total por semana de produção desses dois produtos (em milhões de dólares), sendo o lucro do produto 1 de 3 milhões e o do produto 2 de 5 milhões.

E as restrições representam as restrições de tempo de cada máquina utilizada no processo de produção,

- A equação **(a)** garante que, durante o processo de produção, cada produto 1 necessita de 1 hora na máquina 1, e a máquina só tem disponível 4 horas por semana
- A equação **(b)** garante que, durante o processo de produção, cada produto 2 necessita de 2 horas na máquina 2, e a máquina só tem disponível 12 horas por semana
- A equação **(c)** garante que, durante o processo de produção, cada produto 1 necessita de 3 horas na máquina 3, e cada produto 2 necessita de 2 horas na máquina 3, e a máquina só tem disponível 8 horas por semana

Graficamente representado o problema ficaria da seguinte forma:

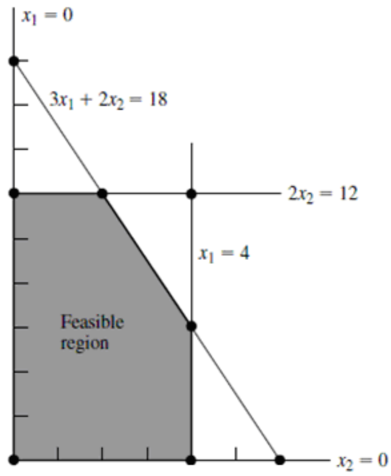


Figura 1: Representação gráfica de um Problema de Programação Linear de duas variáveis

Onde cada reta representa uma restrição do modelo, e a área cinza representa a região viável, ou seja, nessa área estão contidas os valores viáveis de x_1 e x_2 para a maximização do lucro.

Os métodos para resolução de problemas de programação linear buscam esses valores de x_1 e x_2 para a determinação da solução ótima.

1.6 Métodos de Solução

Entre os métodos mais famosos para a resolução de problemas de programação linear estão o método simplex e o método de pontos interiores. Depois da apresentação do método simplex,outros métodos com diferentes abordagens foram propostos (TODD, 2002). Porém, dentre os métodos existentes apenas o método de pontos interiores é atualmente competitivo em relação ao método simplex (BIXBY, 1992 apud MUNARI, 2009). A principal diferença entre esses dois métodos é o que o método simplex caminha pelos vértices da região viável, enquanto o método de pontos interiores caminha pelo interior da região viável (MACULAN; FAMPA, 2006). Além disso, uma outra diferença é que o simplex exige muitas iterações com cálculos simples, enquanto no método de pontos interiores poucas iterações são exigidas, porém com cálculos mais elaborados. Apesar das vantagens do método de pontos interiores em relação ao método estudado neste trabalho, o método simplex possui melhor desempenho na resolução de problemas de pequeno porte em relação ao método de pontos interiores, tornando-se um método indispensável em ferramentas de programação linear.

1.6.1 Método Simplex

O método simplex é um dos algoritmos mais populares para a resolução de problemas de programação linear. Surgiu a mais de 60 anos atrás e foi proposto por George Dantzig.

É um método iterativo, e sua ideia principal consiste no fato de que a cada iteração uma nova solução é encontrada, sempre melhor que a anterior até o ponto em que a solução ótima é obtida. Outra característica do método é o fato de ser matricial, ou seja, os dados a serem calculados são armazenados em matrizes.

Com a utilização do método, foi percebido que a cada iteração eram requeridos muitos cálculos sobre valores que nem sempre importavam para a iteração seguinte, fato que do ponto de vista computacional tornaria o método ineficiente. Esse método é chamado de método simplex padrão ou tabular. A partir desse fato foi desenvolvido o método simplex revisado visando a resolução de problemas de programação linear computacionalmente.

O método simplex surgiu nos Estados Unidos e foi proposto pelo matemático George Dantzig. Quando trabalhava no Pentágono recebeu dos seus colegas o desafio de tentar mecanizar o processo de planejamento. No ano de 1947 Dantzig propôs o método simplex que tornou possível a solução de problemas de otimização de vários tipos, como transporte, produção, alocação de recursos e problemas de escalonamento (LIMA, 2004).

1.7 Princípio Básico do Método

A idéia do método simplex consiste em resolver repetidas vezes um sistema de equações lineares, e assim obter uma sucessão de soluções até encontrar a solução ótima. Ou seja, é um processo onde nos movemos de uma solução viável para outra sempre melhor ou pelo menos não pior.

Um problema de programação linear é sempre constituído de uma função objetivo e várias restrições. Geometricamente, essas restrições resultam em uma forma geométrica, chamada hiperplano, no espaço n-dimensional sendo n o número de variáveis no modelo. E cada vértice desse hiperplano é considerado uma solução viável, como ilustra o exemplo abaixo.

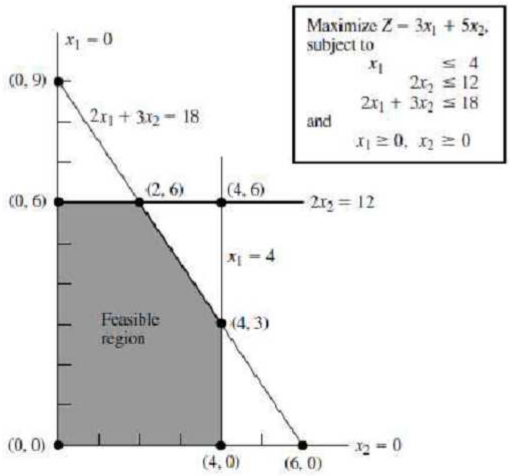


Figura 2: Um modelo de programação linear e a sua respectiva representação gráfica

(HILLIER; LIEBERMAN, 2006)

Na figura 2 apresenta-se um exemplo de um problema de programação linear e sua representação geométrica. Nesse exemplo, de acordo com a representação geométrica do modelo, existem cinco possíveis soluções: x1=0 e x2=0; x1=0 e x2=6; x1=2 e x2=6; x1=4 e x2=3; x1=4 e x2=0. Como o objetivo é maximizar, concluímos que a melhor solução é x1=2 e x2=6, em que z =36. No método Simplex a cada iteração, antes da solução ótima ser obtida, é encontrada uma dessas possíveis soluções que se localizam nos vértices do gráfico.

1.8 Descrição do Método Simplex Revisado

O método simplex revisado surgiu como uma solução para evitar cálculos desnecessários. Esse método foi projetado para problemas a serem solucionados computacionalmente.

No simplex revisado, são armazenados na memória volátil apenas os dados realmente necessários. Além disso, os cálculos são realizados apenas sobre a coluna que é utilizada na iteração, o que evita cálculos com matrizes que poderia acarretar uma imprecisão nos resultados.

Esse método mantém a característica do simplex, que é a troca entre a variável que entra na base e a que sai, além de também exigir certo esforço computacional. Porém sua grande vantagem é a economia de tempo e espaço, que garantida pelo modo como é desenvolvida a solução.

Nesta seção descreve-se os passos do algoritmo simplex revisado. O problema considerado é de maximização. A distinção entre matrizes, vetores e escalares foi feita da seguinte forma: letra maiúscula e negrito para matriz (**MATRIZ**); letra minúscula e negrito para vetor (**vetor**); letra em itálico para escalar (*escalar*).

Considere o seguinte problema de programação linear:

$Maximizar\ \mathbf{cx}$
$Sujeito\ a$
$\mathbf{Ax} \leq \mathbf{b}$
$\mathbf{x} \geq 0$

O vetor $\mathbf{c}$, de coeficientes na função objetivo, é dividido duas componentes: $\mathbf{c_B}$ e $\mathbf{c_N}$, coeficientes das variáveis básicas e não-básicas, respectivamente. Por analogia, o vetor $\mathbf{x}$, de incógnitas do problema, é subdividido em $\mathbf{x_B}$ e $\mathbf{x_N}$. A matriz $\mathbf{A}$, de coeficientes das restrições, é dividida em duas submatrizes: $\mathbf{B}$ e $\mathbf{N}$, coeficientes das variáveis básicas e não-básicas, respectivamente. Os passos do algoritmo são os seguintes.

Passo 1: Calcular o valor das variáveis básicas: $\mathbf{x_b} \ = \ \mathbf{B^{-1}b} \geq 0$

Passo 2: Calcular o vetor multiplicador: $\lambda \ = \ \mathbf{c_B B^{-1}}$

Passo 3: Escolher a variável que entra na base. Para isso, calcula-se: $\mathbf{p} \ = \ \mathbf{c_N} - \lambda \mathbf{N}$

Se $\mathbf{p} \ = \ (\mathbf{c_N} - \lambda \mathbf{N}) \leq 0$ PARAR.

A solução $\mathbf{x_B}$ já é a solução ótima.

Caso contrário, escolher uma coluna de $\mathbf{N}$, coluna $\mathbf{a_k}$, tal que $p_k \ = \ c_n k - \lambda \mathbf{a_k} > 0$

Um critério frequente é escolher a coluna $\mathbf{a_k}$ que resulte no maior valor de p_k.

Então, assumindo $e = k$, k representa o índice da coluna, $\mathbf{a_k}$ representa a coluna de $\mathbf{A}$ candidata a entrar na base.

Passo 4: Determinar a variável que sai da base. Para isso, calcula-se: $\mathbf{y} = \mathbf{B^{-1}a_k}$

Se $\forall i \frac{b_i}{y_i} \leq 0$ PARAR.

A solução é não-limitada. Caso contrário, calcular: $\forall i \underset{\substack{1 \leq i \leq m \\ y_i > 0}}{Min} \left\{ \frac{b_i}{y_i} \right\}$ e guardar o índice correspondente a $s = i$. A variável $(x_b)_s$ sai da base.

Passo 5: Criar a matriz $\mathbf{E}$. $\mathbf{E} = (\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_s} - \mathbf{1}, \gamma, \mathbf{e_s} + \mathbf{1}, ..., \mathbf{e_m})$

Onde, cada coluna da matriz é um vetor unitário com exceção da coluna s, que corresponde ao vetor γ, calculado da seguinte maneira:
$\gamma^{\mathbf{t}} = \left(\frac{-\alpha_{1,e}}{\alpha_{s,e}} \; \frac{-\alpha_{2,e}}{\alpha_{s,e}} \; ... \; \frac{-\alpha_{s-1,e}}{\alpha_{s,e}} \; \frac{1}{\alpha_{s,e}} \; \frac{-\alpha_{s+1,e}}{\alpha_{s,e}} \; ... \; \frac{-\alpha_{m,e}}{\alpha_{s,e}} \right)$

Onde $\alpha_i e$ são os elementos vetor do $\mathbf{y}$.

Passo 6: Calcular nova matriz $\mathbf{B^{-1}}$: $\mathbf{B^{-1}} = \mathbf{EB^{-1}}$

Passo 7: Atualizar os vetores $\mathbf{c_B}$ e $\mathbf{x_B}$.

1.8.1 Método de Pontos Interiores

Em 1984, Karmarkar revolucionou a área de programação linear com a publicação de um algoritmo de complexidade polinomial que apresentou bom desempenho quando aplicado a problemas práticos (MACULAN; FAMPA, 2006). Essa publicação deu origem a um novo campo de pesquisa, chamado de método dos pontos interiores.

O método de pontos interiores tem como principal característica realizar a busca por soluções no interior da região viável do problema, até encontrar a solução ótima (MENEZES, 2008). Em teoria, o método de pontos interiores é melhor que o método simplex, principalmente quando se leva em conta o critério de complexidade de pior caso. O método de pontos interiores possui complexidade polinomial, enquanto o método simplex possui complexidade exponencial No entanto, na prática ambos os métodos concorrem até hoje. Já que o sucesso do método depende da estrutura dos problemas, da esparsidade [3] e da arquitetura dos computadores (MACULAN; FAMPA, 2006).

1.9 Aplicações Práticas

Um problema de programação linear, como já dito anteriormente, busca a otimização na distribuição de recursos sujeitos a restrições. Por isso é considerada uma poderosa ferramenta de apoio a decisão (FROSSARD, 2009) e com utilização em diversas áreas, como: indústria, saúde, computação, produção, etc. As empresas, por exemplo, devem estar

[3]Quando uma matriz possui uma grande proporção de elementos nulos diz-se que é uma matriz esparsa (MUNARI, 2009).

constantemente atentas à competitividade e às restrições existentes com o objetivo de alcançar suas metas, para isso é necessário otimizar os recursos disponíveis (FROSSARD, 2009). Daí a importância da utilização da programação linear empregada em seu exemplo mais geral: maximizar o lucro e minimizar custos. Na definição de modelos desse tipo deve-se considerar o preço de venda e o custo de produção, além de restrições do tipo: quantidade de matéria- prima e mão-de-obra disponíveis, máquinas disponíveis para produção, entre outros (FROSSARD, 2009).

> "Administrar com eficiência os recursos disponíveis na empresa, através do planejamento, controle e execução das atividades relacionadas á utilização destes, é fator fundamental na busca da otimização do resultado global da empresa. A programação linear juntamente com as técnicas de pesquisa operacional, permite identificar o resultado ótimo, considerando todas as restrições impostas no modelo adotado." (FROSSARD, 2009, p. 31)

Na computação, a programação linear é empregada, por exemplo, no processamento de imagens. Além disso é tema de estudos que buscam implementações eficientes de métodos de programação linear, em especial o método simplex, de forma distribuída ou integrada ao banco de dados.

1.9.1 Aplicações em Processamento de Imagens

O termo wavelet refere-se basicamente a um conjunto de funções com forma de pequenas ondas (BLITZKOW, 2008). A decomposição wavelet é uma metodologia de decomposição de uma função ou sinal em um domínio de frequências e espaço, sendo possível investigar a ocorrência de fenômenos localizados no espaço e frequência simultaneamente (PEIXOTO, 2009). A decomposição wavelet em sua versão discreta é utilizada na compreensão de dados sendo útil no processamento de imagens quado é necessário realizar a extração de informações de um sinal e a análise de freqüências, principalmente quando ocorrem rápidas variações na frequência (LEITE, 2007)

A decomposição wavelet tem sido abordada mediante diferentes métodos, como a Transformada de Fourier, por exemplo. Dentre eles estão presentes os métodos que tem como base a programação linear. Um dos primeiros trabalhos nesta área foi proposto por Saunders (2001) utilizando métodos de pontos interiores. Devido ao fato dos problemas lineares obtidos a partir da decomposição wavelet possuirem matrizes muito densas, os autores restringiram a pesquisa apenas ao caso de problemas com dicionários (conjunto de formas de ondas) com uma estrutura especial. O método gera problemas lineares de grande porte, um problema de sinal de onda típico de comprimento 8192, por exemplo, se traduz em um problema linear de tamanho 8192 por 212.992.

No trabalho de Yarmish (2006) o autor resolve um conjunto de problemas de decomposição wavelet, utilizando tanto o método simplex revisado quanto o método simplex padrão. O autor mostra que embora o método simplex revisado seja superior no caso do problema mais geral (problema esparso), a resolução de problemas de decomposição wavelet resulta em um problema denso, onde o método simplex padrão tem melhor desempenho.

Na figura 3 são apresentados vários tipos de wavelets.

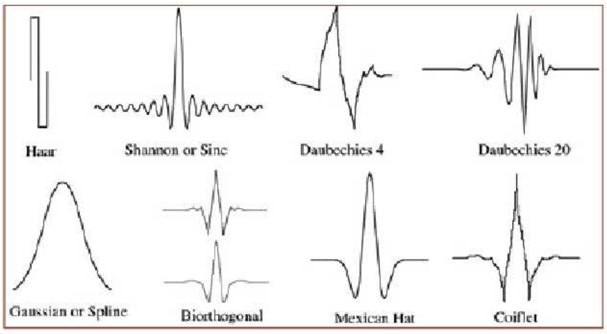


Figura 3: Exemplos de tipos de wavelets

(FUGAL, 2009)

Em seu trabalho Tziritas (2007) diz que problemas de análise de imagens podem ser formulados como problemas de rotulagem, onde a partir de uma imagem segmentada, são realizadas comparações entre pontos vizinhos a fim de rotular um grupo de pontos afins. Uma questão que vem incentivando pesquisas nessas áreas, é como resolver problemas desse tipo de forma eficiente e precisa. O autor propõe a utilização do esquema primal-dual da programação linear, e diz que essa utilização revelou excelentes resultados

1.9.2 Aplicações em Programação Distribuída

Slyke (2009) e Yarmish (2001) demonstram a eficiência da utilização do método simplex padrão de forma distribuída. Esse método é mais utilizado e mais eficiente que o método revisado na resolução de problemas de grande porte, onde existe um maior volume de dados, fazendo mais sentido a utilização da programação distribuída. De acordo com Slyke (2009) e Yarmish (2001) a eficiência do método simplex revisado é afetada pela densidade do problema, ou seja, é uma método mais eficiente para problemas esparsos, os quais são mais comuns. Porém existem aplicações que exigem modelos mais densos, como em processamento de imagens. Nesse caso a implementação de uma algoritmo do método simplex padrão distribuído se mostra mais eficiente. Além disso, não existem algoritmos eficientes do método simplex revisado distribuído. Em conclusão o autor diz que a implementação apresentou bons resultados, especialmente para problemas grandes e densos.

Em Allgöwer (2011) é proposto um algoritmo simplex distribuído para problemas de programação linear degenerados e atribuições multi- agentes. Um problema de programação linear é dito degenerado se na solução uma das variáveis receber valor zero. O objetivo do trabalho é propor um algoritmo simplex para ser implementado em uma distribuição multi- agente, onde os a gentes devem concordar com uma solução ótima ou declarar o problema como inviável, caso não haja solução ótima.

Em seu trabalho Kumar (1994) faz uma análise da performance de um algoritmo simplex paralelo em vários tipos de arquiteturas de rede, como por exemplo, rede hipercubo (Figura 4) e rede de estações de trabalho. Nesse tipo de implementação do método simplex, os dados da matriz são distribuídos entre os processadores que compõem a rede. Como resultado preliminar o autor obteve que na rede hipercubo a velocidade da resolução dos problemas cresceu a medida que mais processadores foram incorporados, inclusive para problemas de grande porte. Já na rede de estações de trabalho, para problemas pequenos a velocidade não teve um aumento proporcional ao aumento do número de processadores, enquanto que para problemas de maior porte a velocidade aumentou de forma proporcional ao número de processadores.

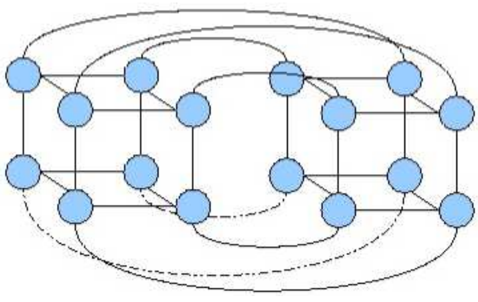


Figura 4: Na rede hipercubo com dimensão d, nesse caso d=4, cada nó está ligado a outros d nós, e o número de processadores é 2^d

1.9.3 Aplicações em Banco de Dados

De acordo com Nagel (2008) dentre as inúmeras ferramentas de programação linear existentes, existe a carência de uma ferramenta que resolva problemas de programação linear de forma integrada a um banco de dados, utilizando procedimentos armazenados em sql pré-compiladas e armazenadas no banco de dados juntamente com os dados. Essa características em uma ferramenta se torna importante, principalmente, em problemas de grande porte, suprimindo o tráfego de dados já que toda a lógica é executada internamente no banco de dados, e apenas o resultado final é enviado para o usuário. Na implementação apresentada por Nagel (2008) os dados do modelo a ser resolvido são armazenados como modelos relacionais, ou seja, em matrizes bi-dimensionais. Apesar das vantagens desse tipo de implementação, o autor chegou a conclusão que o tempo de execução ainda deve ser melhorado.

1.9.4 Aplicação em Saúde

A programação linear também se aplica na área da saúde, em Goldberg (2006) é feita uma proposta de utilização da programação linear no tratamento do câncer. Em um determinado tipo de tratamento são inseridos cateteres na área afetada para introduzir o medicamento necessário, porém o medicamento acaba afetado células saudáveis além das cancerígenas. Como os problemas de programação linear podem ser resolvidos como problemas determinísticos e com solução exata, Goldberg (2006) propõe que a formulação de um problema de otimização para determinar o tempo de permanência dos cateteres, minimizando os desvios na quantidade da dose necessitada pelo paciente. Nos testes realizados, os resultados obtidos não mostraram uma vantagem significativa em relação ao método atualmente utilizado.

1.9.5 Aplicação no reconhecimento de expressões faciais

Em seus trabalhos, Hadid (2005) e Dyer (2005) apresentaram uma abordagem de reconhecimento de expresssões faciais utilizando a programação linear. Em ambos a programação linear foi utilizada na fase de classificação da expressão. Em Dyer (2005) a programação lienar foi utlizada ainda na fase de extração de carcaterísticas, para determinar a quantidade de caracteríticas a serem selecionadas e os principais pontos da face. Enquanto no trabalho de (HADID, 2005) é utilizado o filtro de Gabor na etapa de extração de características. O Filtro de Gabor tem a capacidade de realçar bordas e saliências na imagem. Após a aplicação do filtro na imagem, são realçadas as características salientes presentes no rosto (cantos dos olhos, cantos da boca, narinas, entre outros) (GUIMARãES, 2001). Em ambos os trabalhos foram obtidos ótimos resultados e superiores a outros métodos com os quais foram comparados.

A programação linear é utlizada na etapa após o tratamento da imagem. Primeiramente as expressões são combinadas em pares. No trabalho de (HADID, 2005), por exemplo, busca-se o reconhecimento de sete expressões: raiva, desgosto, medo, felicidade, tristeza, surpresa e neutro. Essas expressões são combinada em pares do tipo: raiva-desgosto, raiva-medo, etc. Formando um total de 21 pares. Após isso o modelo de programação linear, apresentado mais adiante, é utilizado para gerar os calssificares, o modelo deve ser rodado 21 vezes, uma para cada par de expressões. Essa etapa é também chamada de treinamento, pois a partir do classificadores gerados as expressões das imagens a serem analisadas serão determinadas. Essa abordagem também substitui a utilização de redes neurais para o treinamento, uma vantagem é que o modelo necessita ser rodado uma única vez para cada par, enquanto em uma abordagem utilizando redes neurais este processo se trona iterativo.

Na última etapa de determinação da exepressão é utilizda uma estrtura de árvore binária de torneio.

1.9.6 Aplicações na Engenharia de Produção

Dentre as aplicações mais conhecidas da programação linear, encontram-se os problemas de planejamento de produção e controle de estoque e os problemas de mistura. Os problemas do primeiro tipo possuem inúmeras aplicações, desde a alocação de máquinas para atender determinada demanda, até a utilização do estoque para atender a uma mudança imprevisível na demanda e necessidades de contratação de demissão para enfrentar mudanças nas necessidades de mão de obra. Já os problemas de mistura tratam, basicamente, da mistura de diferentes matérias para a produção de produtos, que devem obedecer a algumas especificações e, ao mesmo tempo, minimizar o custo ou maximizar o lucro. (TAHA, 2008). Um exemplo mais prático seria a produção de rações animais onde o objetivo é minimizar o custo de produção da ração composta por dois ingredientes. Estando restrito a quantidade total que deve ser produzida, além das quantidades de nutrientes necessários.

$Minimize\ z = 0,9x_1 + 0,5x_2$

$Sujeito\ a$

$$x_1 + x_2 \geq 90 \ (a)$$
$$0,09x_1 + 0,05x_2 \geq 0,07(x_1 + x_2) \ (b)$$
$$0,02x_1 + 0,06x_2 \geq 0,03(x_1 + x_2) \ (c)$$
$$x_1 \geq 0 \ (d)$$
$$x_2 \geq 0 \ (e)$$

Onde,

- **x1** representa a quantidade do ingrediente 1 que a ração deve conter
- **x2** representa a quantidade do ingrediente 2 que a ração deve conter
- **Z** representa o custo total da produção dessa ração, sendo que o custo (por grama) do ingrediente 1 é de R$ 0,9 e o custo (por grama) do ingrediente 2 é de R$ 0,5

E as restrições representam as restrições das quantidades de nutrientes que a ração deve conter e a quantidade total,

- **(a)** representa a quantidade total de ração que deve ser produzida (em quilos)
- **(b)** representa que a ração deve conter 7% de um determinado nutriente, e 9% de cada grama do ingrediente 1 e 5% de cada grama do ingrediente 2 é composto por esse nutriente.
- **(c)** representa que a ração deve conter 3% de um determinado nutriente, e 2% de cada grama do ingrediente 1 e 6% de cada grama do ingrediente 2 é composto por esse nutriente.
- **(d)** representa que a ração deve conter o ingrediente 1
- **(e)** representa que a ração deve conter o ingrediente 1

A programação linear além de estar presente, é fundamental em diversas áreas, tornando-se uma ferramenta de apoio a decisão e contribuindo para o sucesso de projetos nas áreas em que se aplica.

1.10 Métodos de Classificação

Em problemas de calssificação, dado um conjunto de dados de treinamento onde cada subconjunto pertence a uma entre n classes, o objetivo é que dada uma instância o método de classificação retorne a qual das n classes essa instância pertence. A seguir são apresentados alguns dos métodos mais citados na literatura.

1.10.1 Support Vector Machines

Support Vector Machines ou Máquinas de vetores de suporte (SVM) têm a capacidade de gerar classificadores na etapa de treinamento e de classificar os dados na etapa de teste de acordo com o classificadores gerados. Considerando um problema de classificação com duas classes, uma SVM irá determinar um plano que separe os pontos dessas duas classe, de forma que a distância entre o hiperplano e os pontos seja a máxima possível. Os pontos de cada classe utlizados como referência são denominados vetores de suporte. No caso de conjuntos linearmente inseparáveis, é adicionada uma variável não negativa a equação do hiperplano. De forma que para os pontos que estão do lado correto do hiperplano classificador, essa variável é nula, porém para os pontos que se encontram do lado incorreto do hiperplano, essa variável recebe o valor que pode ser definido como uma medida de quanto o ponto falhou em estar no lado correto do hiperplano e na medida de distância máxima de acordo com os vetores de suporte, nesse caso além de buscar o hiperplano com distância máxima entre o hiperplano e os pontos, deve-se buscar também a minimização do somatório dessas variáveis em cada ponto(GUNN, 1998)(LIMA, 2002).

FIGURA!!!!!!!!!!

No caso em que é o núemro de classes é maior que dois, duas abordagens podem ser utilizadas com as SVM (LORENA, 2003):

- Um contra todos: Nessa abordagem, para K padrões são geradas K SVM. Na criação das SVM, é gerado um hiperplano que separa 1 classe das k-1 classes. Na determinação do padrão de uma instância x, o padrão é aquele que, entre as k SVM obteve mais mais valores de x do lado do hiperplano onde se encontrava o padrão k.
- Todos contra um: Nessa abordagem os padrões são agrupados em pares e uma SVM é gerada para cada par. Um esquema de votação deve ser utilizado para determinar o padrão de uma instância, que deve ser analisada a partir de cada SVM e cada SVM retorna uma classe possível. A classe que obtiver mais pontos da instância é

a classe atribuída.

1.10.2 Árvore de decisão

1.10.3 Bayes

1.10.4 Multi- layer pecceptron

1.10.5 k- nearest neighbor

1.11 Métodos de validação do modelo

Um modelo de classificação pode ser verificado em relação a sua acurácia apartir de uma conjunto de dados ou a sua performance em relação a outros métodos classificadores podem ser verificados. Algumas técnicas utilizadas com esse intuito serão apresentadas a seguir. No presente trabalho a técnica Cross Validation é utilizada com o objetivo de verificar a acurácia, nesse caso, da metodologia utlizada na classificação de padrões (KOHAVI, 1995) (BALDISSEROTTO, 2005).

1.11.1 Handout

Esse é um método simples, onde os dados são divididos em 2 grupos mutuamente exclusivos, sendo um grupo utilizado para treino e o outro para teste. Em geral 2/3 dos dados são utilizados no treinamento e 1/3 na etapa de testes. É mais indicado quando uma grande quantidade de dados está disponível, de forma que os dois grupos possam conter dados suficientes de todos os padrões, nesse caso, suficientes para que o resultado final não seja comprometido pela falta de dados no grupo da etapa de treino e nem no grupo de testes. Apesar de ser um método de fácil implementação, a divisão entre conjunto de treinamento e teste deve ser bem estudada de forma que todos os padrões estejam bem representados nos dois grupos, por isso é um método recomendável quando o conjunto de dados possui um grande número de representações de cada padrão (KOHAVI, 1995) (BALDISSEROTTO, 2005).

1.11.2 Cross Validation

Nesse método, o conjunto de dados também é dividido em cojunto de treinamento e de teste, porém esses dois grupos variam a cada rodada de teste. O tipo de cross validation mais comumente utilizado é o k-fold cross validation, onde os dados são divididos em k conjuntos mutuamente exclusivos e as etapas de treinamento e teste são realizdas K vezes, de forma que a cada rodada um conjunto diferente é utilizado como teste e os outros k-1 conjuntos são utilizados para treinamento. Esse método é recomendável quando o conjunto de dados não é tão grande, pois não existema preocupação de quais dados separa para teste e quais separa para validação, já que a ao final da execução do método todos o dados terão participado dos conjuntos de validação e teste (KOHAVI, 1995) (BALDISSEROTTO, 2005).

1.11.3 Leave-one-out

Nessa técnica, do conjunto com o número total de dados N, são realizadas N rodadas de treinamento e teste, sendo que a cada rodada N-1 dados são utlizados para teste e o dados restante é utlizado para treinamento. Pelas suas características, esse é um método recomendável para conjuntos de dados pequenos já que a quantidade de rodadas de treinamento e teste é igual a quantidade de dados o que o torna um método com alto custo computacional (KOHAVI, 1995) (BALDISSEROTTO, 2005).

1.12 Trabalhos Relacionados

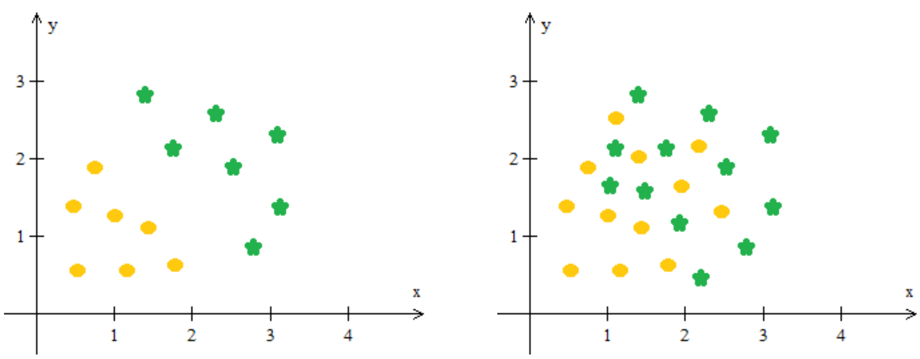
E seu trabalho (LIMA, 2002) busca o classificação de impressões digitais em 5 classe. Esse tipo de classificação tem o próposito de gerenciar grandes bancos de dados de impressões digitais e acelerar o processo de identificação. Foi utilizado um banco contendo 4000 imagens igualmente distribuídas entre as cinco classes. O autor dividiu o banco em dois grupos, utilizando um para treinamento e outro para teste. Após a extração das características das imagens, Máquinas de Vetores de Suporte foram utlizadas na classificação. Na abordagem todos contra um foi obtida uma média de precisão de 91,18%, já na abordagem um contra todos a precisão na classificação das impressões foi de 90,43%.

2 Aplicação da programação linear na separação de pontos

Na tarefa onde o obejtivo é a separação de dois ou mais conjuntos de pontos, sendo que cada conjunto representa um padrão, busca-se um método para que essa separação seja obtida com resultado ótimo. Cada conjunto é o resultado de um conjunto de pontos e cada ponto é é representado por um vetor de características. Em (BENNETT; MANGASARIAN, 1992) é proposto um modelo de programação linear capaz de um gerar um hiperplano de minimiza a soma dos pontos classificados de forma errada. Dessa forma o modelo é capaz de gerar um hiperplano separador para pontos linearmente separáveis e para conjutnos linearmente inseparáveis, o modelo gera o melhor hiperplano de forma que essa média calculada seja mínima. Na figura abaixo estão ilustrados os casos de dois conjuntos linearmente separáveis e dois conjuntos linearmente inseparáveis. COLOCAR FIGURA!!!

2.1 O modelo

Considerando dois conjuntos, no espaço n-dimensional R^n, sendo o conjunto A representado pela matriz m x n, e o conjunto B representado pela matriz k x n. Contextualizando,



(a) (b)

Figura 5: Em 5(a) são apresentados dois conjuntos linearmente separáveis e em 5(b) são apresentados dois conjuntos linearmente inseparáveis k e m são as quantidades de pontos em cada conjunto, e n é a quantidade de características. O modelo gera um hiperplano que separa os pontos dos dois conjuntos. De forma que os m pontos do conjunto A fiquem de um lado do hiperplano e os k pontos do conjunto B fiquem do outro lado do hiperplano. Na figura abaixo o modelo é apresentado. E a seguir é apresentado de forma mais detalhada.

COLOCAR MODELO!!!!

O modelo determina o hiperplano:

$$x\omega = \gamma$$

Onde, é ω o vetor normal ao hiperplano e γ é um escalar. De forma que:

$$A\omega \geq e\gamma$$

e

$$B\omega \leq e\gamma$$

Onde e é um vetor de 1's com dimensão m para o conjunto A e k para o conjunto B. Para conjuntos linearmente separáveis torna-se fácil traçar um hiperplano separador, porém para conjuntos linearmente inseparáveis é necessária uma estratégia para que o hiperplano separador seja ótimo. O modelo busca o melhor hiperplano, para isso gera dois hiperplanos limites somando 1 unidade a equação do hiperplano separador:

$$A\omega \geq e\gamma + e$$

e

$$B\omega \leq e\gamma - e$$

De forma que o hiperplano separador fica localizado exatamente entre esses dois hiperplanos limites. Detalhando o modelo de acordo com Trevisan (2010), considerando x como um vetor em R^n, definimos:

1. $(x_i)_+ = \max_{i=1,2,\ldots,n} \{x_i, 0\}$
2. $\|x\|_1 = \sum_{i=1}^{n} |x_i|$

Podemos escrever o problema de minimização com norma da seguinte forma:

$$\min_{\omega,\gamma} \frac{1}{m} \left\|(-A\omega + e\gamma + e)_+\right\|_1 + \frac{1}{k} \left\|(B\omega - e\gamma + e)_+\right\|_1$$

Seja $g : R^n \mapsto R^m, h : R^n \mapsto R^k$ e S um subconjunto de R^n . Os problemas abaixo possuem soluções idênticas:

$$\min_{x \in S} \|g(x)_+\|_1 + \|h(x)_+\|_1$$

$$\min_{x \in S} e^t y + e^t z$$

$$s.a. \begin{cases} y \geq g(x) \\ \geq h(x) \\ y \geq 0, z \geq 0 \end{cases}$$

Como $g(x)_+ \geq g(x)$ e $h(x)_+ \geq h(x)$ podemos observar que os valores ótimos de y e z podem der dados pelas igualdades $y = g(x)_+$ e $z = h(x)_+$.

NUMERAR AS EQUAÇÕES!!!

$$\min_{\omega,\gamma,y,z} \frac{e^T y}{m} + \frac{e^T z}{k}$$

$$s.a. \begin{cases} -A\omega + e\gamma + e \leq y \\ B\omega - e\gamma + e \leq z \\ y \geq 0, y \geq 0 \end{cases}$$

Dados do Modelo:

m: quantidade de imagens no conjunto correspondente a primeira expressão facial; k: quantidade de imagens no conjunto correspondente a segunda expressão facial; A: Matriz contendo o conjunto de vetores de caraterísticas das imagens correspondentes à primeira expressão facial; B: Matriz contendo o conjunto de vetores de caraterísticas das imagens correspondentes à segunda expressão facial; e: vetor coluna unitário;

Variáveis do Modelo:

ω: coeficientes do hiperplano que separa os conjuntos de imagens em dois grupos; γ: constante do hiperplano que separa os conjuntos de imagens em dois grupos; y: vetor contendo as distâncias de cada imagem do conjunto A ao hiperplano separador mais um; z: vetor contendo as distâncias de cada imagem do conjunto B ao hiperplano separador mais um.

Formulação Matemática:

A função objetivo (1) procura minimizar a somas das médias das distâncias das imagens de ambos conjuntos. A restrição (2) exige que os pontos se encontrem necessariamente abaixo do hiperplano separador mais uma constante unitária. A restrição (3) é análoga a restrição (2) aplicada as imagens da segunda expressão facial. As equações em (4) definem a natureza das variáveis do modelo, como sendo não negativas.

2.2 Exemplo Ilustrativo do Modelo Classificador

Para ilustrar o modelo vamos considerar dois exemplos em R^2 (BENNETT; MANGASARIAN, 1992) desta que a ilustração gráfica fique mais facilmente compreensível.

- Exemplo1:

  Considerando as matrizes a seguir:

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 2 & 1 \\ 2 & 2 \\ 2 & 3 \\ 2 & 4 \\ 3 & 3 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 4 & 1 \\ 4 & 2 \\ 4 & 3 \\ 5 & 2 \\ 5 & 3 \\ 5 & 4 \\ 6 & 2 \end{bmatrix}$$

  Nesse caso, temos: m = 9 (quantidade de pontos da matriz A) e k = 7 (quantidade de pontos da matriz B). Submetendo as matrizes ao modelo temos:

  - Valor da função objetivo = 0
  - ω = [-2 1]
  - γ = -4
  - y = [0 0 0 0 0 0 0 0 0]
  - z = [0 0 0 0 0 0 0]

  Como o valor da função objetivo depende dos vetores y e z, podemos notar que o valor 0 indica que os conjuntos A e B são linearmente separáveis. A seguir é apresentada a representação gráfica dos pontos e das retas geradas pelo modelo.

COLOCAR FIGURA!!!

A partir da representação gráfica podemos perceber que a reta separadora $-2x+y = -4$ se localizou exatamente no meio entre os dois conjuntos com o auxílio das duas retas: $-2x + y = -3$ e $-2x + y = -5$.

- Exemplo2: Vamos considerar agora as matrizes:

$$A = \begin{bmatrix} 3 & 4 \\ 4.3 & 4.5 \\ 4.5 & 2 \\ 3 & 5.5 \end{bmatrix} B = \begin{bmatrix} 4 & 2 \\ 4.5 & 3.5 \\ 5 & 3 \end{bmatrix}$$

  Neste exemplo temos: m = 4 e k = 3. De acordo com o modelo, temos os seguintes valores:

  * Valor da função objetivo = 0.92
  * ω = [-0.91 0.91]
  * γ = -0.82
  * y = [0 0 2.46 0]
  * z = [0 0.91 0]

  Com o valor da função objetivo diferente de zero, temos dois conjuntos linearmente inseparáveis, como ilustrado na figura a seguir.

  COLOCAR FIGURA!!!

  Nesse exemplo notamos que o valor 2.46 no vetor y é referente ao ponto do conjunto A que está localizado do lado errado da reta. E o valor 0.91é referente ao ponto do conjunto B que está localizado mais próximo à reta separadora, apesar do ponto estar localizado do ponto correto da reta ele está localizado acima da reta limite $-0.91x + 0.91y = -1.82$.

Em Hadid (2005) foi realizado um estudo de reconhecimento de expressões faciais, utilizando a base de dados JAFFE (LYONS MIYUKI KAMACHI, 1997), em uma etapa de pre processamento as coordenadas do olhos foram marcadas manualmente e utilizando uam máscara elíptica as imagens foram reduzidas a região da face. Na extraçãod e características foi utilizado o método local binary patterns. O modelo de programação linear proposto em (BENNETT; MANGASARIAN, 1992) é utlizado para gerar os classificadores e na classificação de imagens com expressão desconhecida foi utilizada uma árvore de torneio. Para validação os dados foram divididos em 10 conjuntos, o ciclo de treinamento e teste foi repetido 10 vezes, sendo que em cada ciclo um conjunto foi usado como teste e os outros nove para validação. Foram realizados 20 desses ciclos e uma média da taxa de acertos foi calculada, obtendo uma taxa de 93.8%. Os autores concluíram que o método obteve uma performance excelente comparado a outros estudos de receonhecimento de expressão facial que utilizaram a mesma base de dados.

2.2.1 Etapa de validação

COLOCAR FIGURA do método!!!

COLOCAR FIGURA DA IMPLEMENTAÇÃO!!!

Para testar a metodologia utlizada neste trabalho para classificação de padrões, foi utilizado o método cross validation. Através dos resultados desse método é possível analisar a acurácia da metodologia de classificação, ou seja, a capacidade de classificar uma instância com o seu padrão correto.

A metodologia k-fold cross validation foi implementada na linguagem de programação Java. Os dados são arrumados manualmente em k arquiivos. A implementação seleciona os dados de teste e de validação, e retorna o resultado parcial a cada rodada e o resultado final após os k ciclos de validação e teste.

Referências Bibliográficas

ALLGöWER, M. B. G. N. F. B. F. A Distributed Simplex Algorithm for Degenerate Linear Programs and Multi-Agent Assignments. *Automatica*, Maio 2011.

BALDISSEROTTO, C. *Técnicas de aprendizagem de máquina para previsão de sucesso em implantes dentários*. Dissertação (Mestrado) — Universidade de Pernambuco, 2005.

BENNETT, K. P.; MANGASARIAN, O. L. *Robust Linear Programming Discrimination Of Two Linearly Inseparable Sets*. 1992.

BIXBY, R. E. Implementing the simplex method: The initial basis. *ORSA Journal*, v. 4, p. 267–284, 1992.

BLITZKOW, A. C. B. B. D. *Ondaletas: Histórico e Aplicação*. Maio 2008.

DANZITG, G. *Linear Programming and Extensions*. [S.l.]: Princeton University Press, 1963.

DYER, G. G. C. R. Simultaneous Feature Selection and Classifier Training via Linear Programming: A Case Study for Face Expression Recognition. *IEEE Transactions on systems, man, and cybernetics*, v. 35, n. 3, p. 477–488, Junho 2005.

FROSSARD, A. C. P. Programação Linear: Maximização de Lucro e Minimização de Custos. *Revista Científica da Faculdade Lourenço Filho*, v. 6, n. 1, 2009.

FUGAL, D. L. Wavelets: Beyond Comparison. *SatMagazine*, v. 7, n. 1, p. 35–41, maio 2009.

GOLDBERG, R. A. E. L. J. P. I.-C. J. H. J. F. O. K. Optimization of HDR brachytherapy dose distributions using linear programming with penalty costs. *The International Journal of Medical Physics Research and Practice*, v. 33, n. 11, 2006.

GUIMARãES, D. R. O. N. Sistema Híbrido Inteligente Aplicado ao Reconhecimento de Rostos. In: *Anais do I WORCAP*. [S.l.: s.n.], 2001. p. 45–47.

GUNN, S. R. *Support Vector Machines for Classification and Regression*. 1998.

HADID, X. F. M. P. A. Facial Expression Recognition with Local Binary Patterns and Linear Programming. *Pattern Recognition and Image Analysis*, v. 15, n. 2, p. 546–548, 2005.

HILLIER, F.; LIEBERMAN, G. *Introdução à Pesquisa Operacional*. 8º. ed. [S.l.]: Mc-Graw-Hill, 2006.

KARMARKAR, N. A new polynomial algorithm for linear programming. *Combinatorica*, v. 4, p. 373–395, 1984.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: . [S.l.]: Morgan Kaufmann, 1995. p. 1137–1143.

KUMAR, G. K. V. *Performance and Scalability of the Parallel Simplex Method for Dense Linear Programming Problems An Extended Abstract*. 1994. Disponível em: <http://www.cs.umn.edu/~kumar/papers/simplex.ps>.

LEITE, F. E. A. *Análise Estatística de Padrões Sísmicos: Decomposição em Multiescala*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte Centro de Ciências Exatas e da Terra Departamento de Física Teórica e Experimental, Dezembro 2007.

LIMA, A. R. G. *Máquinas de Vetores Suporte na Classificaçã de Impressões Digitais*. Dissertação (Mestrado) — Universidade Federal do Ceará, 2002.

LIMA, D. de. *A Programação Matemática no Planejamento de Produção na Relação Avícola/ Aviário: Alojamento e Desalojamento de Aves*. Dissertação (Mestrado) — Universidade Federal do Paraná, 2004.

LORENA, A. C. P. L. F. d. C. A. C. *Introdução às Máquinas de Vetores Suporte*. 2003.

LYONS MIYUKI KAMACHI, J. G. M. J. *Japanese Female Facial Expressions (JAFFE), Database of digital images*. 1997.

MACULAN, N.; FAMPA, M. H. C. *Otimização Linear*. [S.l.]: Editora Universidade de Brasília, 2006.

MENEZES, L. de L. P. M. A. F. Implementação de algoritmos simplex e pontos interiores para programação linear. *Estudos*, v. 15, n. 2, p. 225–246, 2008.

MUNARI, J. P. A. *Técnicas Computacionais para uma implementação eficiente e estável de metódos tipo smplex*. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação, ICMP-USP, 2009.

NAGEL, A. K. A. New Developments in Robotics, Automation and Control. In: ______. [S.l.]: InTech, 2008. cap. Linear Programming in Database, p. 339–354.

PAMPLONA, E. de O. *Engenharia Econômica II*. 2005. Disponível em: <http://www.iepg.unifei.edu.br/edson/download/Engecon2/CAP5EE2PLapost.pdf>. Acesso em: 04/06/2012.

PASSOS, A. N. *Estudos em Programação Linear*. Dissertação (Mestrado) — Instituto de Matemática, Estatística e Computação Científica, UNICAMP, 2009.

PEIXOTO, P. da S. *Resolução numérica de EDPs utilizando ondaletas harmônicas*. Dissertação (Mestrado) — Instituto de Matemática e Estátistica da Universidade de São Paulo, Junho 2009.

SAUNDERS, S. S. C. D. L. D. M. A. Atomic Decomposition by Basis Pursuit. *SIAM Review*, v. 43, n. 1, p. 129–159, 2001.

SLYKE, G. Y. R. A Distributed, scaleable simplex method. *The Journal of Supercomputing*, v. 49, n. 3, p. 373 – 381, Setembro 2009.

TAHA, H. A. *Operation Research: An Introduction*. 8. ed. [S.l.]: Pearson, 2008.

TODD, M. J. The many facets of linear programming. *Mathematical Programming*, v. 91, p. 417–436, 2002.

TREVISAN, E. P. *O uso da programação linear na separação de pontos*. Dissertação (Mestrado) — Universidade Estadual de Campinas - UNICAMP, 2010.

TZIRITAS, N. K. G. Approximate Labeling via Graph Cuts Based on Linear Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 29, n. 8, p. 1436–1453, 2007.

YARMISH, G. *A Distributed Implementation of the Simplex Method* . Tese (Doutorado) — Polytechnic University, Março 2001.

YARMISH, G. The Simplex Method Applied to Wavelet Decomposition. In: SCIENCE, U. o. T. a. D. G. R. Dattatreya Department of C. (Ed.). *MATH'06 Proceedings of the 10th WSEAS International Conference on Applied Mathematics*. [S.l.: s.n.], 2006. p. 226–228.