Referências Bibliográficas

Agrawal, R., Imielinski, T and Swami, A.. Database Mining: A performance perspective, *IEEE Trans. Knowledge Data Eng.*,v. 5, Dec. 1993.

Aha, D. W., "Heart Disease Databases", link do banco de dados: http://www.ics.uci.edu/pub/machinelearning-databases/heart-disease/heart-disease.names (atualizado até Outubro de 2001).

Aitchison, J. & Dunsmore, I. R.. *Statistical Prediction Analysis*. Cambridge University Press, 1975.

Anderson, J. A.. Diagnosis by logistic discriminant function: further practical problems and results. *Appl. Statist*., 23, 1974, p. 397-404.

Battiti, R.. Using mutual information for selecting features in supervised Neural net learning. *IEEE Trans. Neural Networks*, v. 5, 1994, p. 537-550.

Bishop, C. M.. Neural Networks for Pattern Recognittion. Oxford. Clarendon Press. 1995.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C.. *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.

Cover, T. M., J. Thomas, A.. *Elements of Information Theory*. New York: Wiley, 1991.

Cybenko, G..Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, v.2, 1989, p. 303-314.

Daberllay, G., Klan, P.. An information-theoretic Adaptive Method for Time Series Forecasting. *Neural Networks World*, 1997, p. 227-238

Daberllay, G., Slama, M.. Forecasting the Short-Term Demand for Electricity: Do Neural Networks Stand a Better Chance? *International Journal of Forecasting*, v. 16, 2000, p. 71-83

De Castro, L. N. & Von Zuben, F. J.. In Improving Pruning Technique with Restart for the Kohonen Self_Organizing Feature Map, Proc. Do IJCNN, 3, (1999a), pp.1916-1919.

De Castro, L. N. & Von Zuben, F. J., “Neural Networks with Adaptive Activation Functions: A Second Order Approach”, Proc. do SCI/ISAS’99, 3, (1999b), pp. 574-581.

De Castro, L. N., Iyoda, E. M., Santos, E. P. & Von Zuben F. J.. "Redes Neurais Construtivas: Uma Abordagem Comparativa", Anais do IV CBRN, 1999, pp. 102-107.

Djavan, B., Remzi, M., Zlotta, A., Seitz, C., Snow, P., Marberger, M.. Novel Artificial Neural Network for Early detection of Prostate Cancer – *Journal of Clinical Oncology*, v.20, n. 4 (February 15), 2002, p. 921-929.

Draper, N. R. And Smith, H.. Applied Regression Analysis. 2nd. New York: Wiley, 1981.

Duda, R.O. & Hart, P.E.. *Pattern Classification and Scene Analysis.* New York. n Wiley, 1973.

Fisher, R. A.. "The use of measurements in taxonomic problems" – *Annals of Eugenics*, v. 7:176-184, 1936.

Foresee, F. D., and M. T. Hagan, "Gauss-Newton approximation to Bayesian regularization," *Proceedings of the 1997 International Joint Conference on Neural Networks*, 1997.

Fraser, A. M. & Swinney, H. L.. "Independent coordinates for strange attractors from mutual information", *Phys Rev.*, v. 33, n. 2, 1986.

Fukunaga, K.. *Introduction to Statistical Pattern Recognition*. Academic Press, New York. 1972.

Hagan, M. T., Menhaj, M.. "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, v. 5, n. 6, 1994, p.989-993.

Haykin, S.. *Neural Networks: a comprehensive foundadtion*, Prentice-Hall, 1998.

Hermans, J. & Habbema, J. D. F.. Comparison of five methods to estimate posterior probabilities. *EDV in Medizin und Biologie*, 6, 1975, p. 14-19.

Ho, C.S., Chou, J.S.. Fuzzy ARTRON: A General-purpose Classifier Empowered by Fuzzy ART and Error Back-propagation Learning, *Journal of Information Science and Engineering 17, 683-695*, 2001.

Hu, Z.H., Li, Y.G., Cai, Y.Z., Xu, X.M.. An Empirical Comparison of Ensemble Classification Algorithms with Support Vector Machines, *Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai*, 26-29 August 2004.

Johnson A. R. & Wichern D. W. *Applied multivariate statistical analysis*, 4th. Ed., Prentice Hall, 1998.

Joliffe, I. T.. *Principal Component Analysis*. New York: Springer-Verlag, 1986.

Kattan, M.. Statistical Prediction Models, Artificial Neural Networks, and the Sophism “I Am A Patient, Not a Statistic”. *Journal of Clinical Oncology*, v.20, n. 4(February 15), 2002, p. 885-887.

Krusinska, E. & Liebhart, J.. Robust discriminant functions in assisting medical diagnosis: application to the chronic obturative lung disease data. *Biometrical Journal*, 32, 1990, p. 915-929.

Kwak, N. and Choi, C.. Input Feature Selection for Classification Problems. *IEEE Trans. Neural Networks*, v. 13, no.1, 2002, p. 143-159.

Kwok, T. Y. & Yeung, D. Y.. Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems, *IEEE Trans. On Neural Networks*, 8(3), 1997, pp. 630-645.

Levenberg, K.. "A method for the solution of certain problems in least squares", *Quarterly of Applied Mathematics 2*, 1944, p. 164–168.

MacKay, D. J. C., "Bayesian interpolation," *Neural Computation*, v. 4, n. 3, 1992, p. 415-447.

Marquardt, D.. "An Algorithm for Least Squares Estimation of Nonlinear Parameters", *SIAM J. Appl. Math*. v. 11, 1963, p. 431-441.

McCulloch, W. S. & Pitts, W.. "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*. v. 5, 1943, p. 115-133.

Medeiros, M.C. & Pedreira, C.E.. “What are the effects of forecasting linear time series with neural networks?”, *Engineering Intelligent Systems*, v.9, p.237-242, 2001.

Nguyen, D., Widrow, B., "The truck backer-upper: An example of self-learning in neural networks," *Proceedings of the International Joint Conference on Neural Networks*, vol 2, 1989, pp. 357-363.

Ohno-Machado, L.. Methodological Review Modelong Medical Prognosis:Survival Analysis Techniques. *Journal of Biomedical Informatics* 34, 2002, p. 428-439.

Ohno-Machado, L. & Musen, M. A. Modular Neural Networks for Medical Prognosis: Quantifying the Benefits of Combining Neural Networks for Survival Prediction. Knowledge Systems Laboratory, *Medical Computer Science*, February, 1996.

Principe, J. C., Euliano, N. R. and Lefebvre, W. C.. Neural and Adaptive Systems: Fundamentals Through Simulations, John Wiley. 2000.

Quinlan, R.. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

Setiono, R., Liu, H.. Neural Network Feature Selector. *IEEE Trans. Neural Networks*, v. 8, 1997, p. 654-661.

Shannon, C. E. & Weaver, W.. *The Mathematical Theory of Communication*. Urbana, IL: Univ. Illinois Press, 1949.

Titterington, D. M., Murray, G. D., Murray, L. S., Spiegelhalter, D. J., Skene, A.M., Habemma, J. D. F. & Gelpke, G. J.. Comparison of discrimination techniques applied to a complex data set of head injuried patients (com discussão). J.R. *Statist. Soc*. A, 144, 1981, p. 145-175.

White, H and Racine, J. (2001). "Statistical Inference, The Bootstrap and Neural Network Modeling with Application to Foreign Exchanges Rates". IEEE Transactions on Neural Networks, vol. 12, 1-19.

Yao, X. and Liu, Y. (1999). Neural networks for breast cancer diagnosis, *Proc. of the 1999 Congress on Evolutionary Computation, Vol. 3*, IEEE Press, Piscataway, NJ, USA, 1999, p. 1760-1767.

Anexo A – Seleção de Variáveis sob Informação Mútua com Distribuição Uniforme de Informação (MIFS-U)

A.1 Introdução

A seleção de variáveis de entrada constitui uma das fases mais importantes em problemas de classificação. As variáveis de entrada podem ser classificadas como pertinentes, irrelevantes ou redundantes, e o que se pretende é selecionar somente aquelas que sejam pertinentes (Kwak & Choi, 2002). Nesta dissertação utilizou-se o algoritmo de Seleção de Variáveis sob Informação Mútua com Distribuição Uniforme de Informação (MIFS-U). O algoritmo proposto é aplicado em um grupo de indivíduos do banco de dados público intitulado "Heart Disease Database" (Base de Dados pública de Doença Cardíaca) (Aha, atualizado em 2001), diagnosticados nas cidades de Cleveland e Long Beach, nos Estados Unidos.

A seleção de variáveis tem um papel fundamental na classificação de sistemas como Redes Neurais. Problemas de seleção de variáveis foram pesquisados por vários autores como Battiti (1994), Joliffe (1986) e Agrawal *et al* (1993). Um dos métodos mais populares para lidar com este problema é a análise de componentes principais (PCA) (Joliffe, 1986). Porém, caso se queira preservar os dados originais, este método não é desejável. Recentemente, uma das contribuições mais importantes trata do método de árvore de decisão. Os atributos pertinentes são descobertos um a um iterativamente (Quinlan, 1993) (Breiman *et al*, 1984). Setiono e Lui (1997) propuseram um algoritmo de seleção de variáveis baseado em uma árvore de decisão excluindo as variáveis de entrada da Rede Neural uma a uma e treinando novamente a Rede repetidamente. O classificador com poda dinâmica (CDP) (Agrawal *et al.*,1993) também se baseia numa árvore de decisão a qual faz uso da informação mútua da entrada com a saída. Este método é eficiente e encontra regras mapeando entrada e saída, mas requer muita memória. O seletor de variáveis (Battiti, 1994) também usa a informação mútua entre entrada e saída como o CDP. A regressão stepwise (Drapper & Smith, 1981)

é também considerada uma técnica padrão na seleção de variáveis fazendo uso do teste F como critério de parar a seleção.

O algoritmo de Seleção de Variáveis sob Informação Mútua com Distribuição Uniforme de Informação (MIFS-U) investiga a limitação do seletor de variáveis proposto por (Battiti,1994) e se propõe superar esta limitação e melhorar o desempenho no processo de seleção de variáveis.

Feitas estas considerações, nas seções seguintes serão introduzidos alguns conceitos básicos da teoria da informação que serão usados na aplicação do algoritmo usado nesta dissertação.

A.2
Entropia e Informação Mútua

Sistemas de classificação em Redes Neurais mapeiam variáveis de entrada em classes de saída. Neste processo, existem variáveis que são importantes e variáveis irrelevantes, isto é, com pouca informação relativa à saída. Para resolver o problema de seleção de variáveis, tem-se que achar entradas que contenham muita informação sobre a saída e é necessária uma ferramenta para medir essa informação. A teoria da informação fornece um método para medir a informação de variáveis aleatórias: a entropia e a informação mútua (Shannon *et al*, 1949, Cover *et al*, 1991).

A entropia é uma medida de incerteza de variáveis aleatórias. Seja uma variável aleatória discreta com função de densidade de probabilidade (pdf) p(x). A entropia de X é definida como:

$$H(X) = -\sum_{x \in \chi} p(x) \log p(x) \tag{A.1}$$

Para duas variáveis aleatórias discretas e com pdf conjunta p(x,y), a entropia é definida como:

$$H(X,Y) = -\sum_{X \in \chi} \sum_{y \in \gamma} p(x,y) \log p(x,y) \tag{A.2}$$

Quando certas variáveis são conhecidas e outras não, a incerteza é medida pela entropia condicional:

$$H(Y \mid X) = \sum_{x \in \chi} p(x) H(Y \mid X = x) = -\sum_{x \in \chi} p(x) \sum_{y \in \gamma} p(y \mid x) \log p(y \mid x) =$$
$$= -\sum_{x \in \chi} \sum_{y \in \gamma} p(x, y) \log p(y \mid x) \qquad \text{(A.3)}$$

A entropia conjunta e a entropia condicional têm a seguinte relação:

$$\begin{aligned} H(X,Y) &= H(X) + H(Y|X) \\ &= H(Y) + H(X|Y) \end{aligned} \qquad \text{(A.4)}$$

Esta relação é conhecida como regra da cadeia e implica que a entropia total das variáveis aleatórias X e Y é a entropia de X mais a entropia restante de Y dado X.

A informação contida em duas variáveis aleatórias é definida como a informação mútua entre duas variáveis aleatórias.

$$I(X,Y) = \sum_{x \notin \chi} \sum_{y \in \gamma} p(x, y) \log \frac{p(x, y)}{p(x)\, p(y)} \qquad \text{(A.5)}$$

Se a informação mútua entre duas variáveis aleatórias é grande (pequena), significa que as duas variáveis são muito (pouco) relacionadas. Se a informação mútua é próxima de zero, as duas variáveis aleatórias são independentes.

A informação mútua e a entropia têm as seguintes relações:

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) \\ I(X,Y) &= H(Y) - H(Y|X) \\ I(X,Y) &= H(X) + H(Y) - H(X,Y) \\ I(X,Y) &= I(Y,X) \\ I(X,X) &= H(X) \end{aligned} \qquad \text{(A.6)}$$

Para variáveis aleatórias contínuas, a entropia diferencial e a informação mútua são definidas como:

$$H(X) = -\int p(x)\log p(x)dx$$
$$I(X,Y) = \iint p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy \quad \text{(A.7)}$$

Salienta-se que é praticamente impossível achar exatamente esta função de densidade de probabilidade e executar sua integração. Por esse motivo, dividi-se o espaço da variável de entrada contínua em várias partições discretas e calcula-se a entropia e a informação mútua usadas nas definições para os casos discretos. O erro inerente ao processo de conversão de variáveis contínuas para variáveis discretas é um valor constante que depende do número de partições em que se divide o espaço contínuo (Fraser & Swinney, 1986).

A.3
Algoritmo de Seleção de Variáveis

A.3.1
O Problema de FRn - k

No processo de selecionar variáveis de entrada, é desejável reduzir o número de entradas excluindo variáveis irrelevantes ou redundantes dos dados. Este conceito é formalizado selecionando k variáveis de um conjunto de n variáveis chamado de problema de "redução de variável" (Battiti, 1994). O processo será apresentado a seguir:

[FRn - k]: Dado um conjunto inicial de n variáveis, encontre o subconjunto com k<n variáveis que são "a máxima informação" sobre a classe de saída.

Como visto na seção anterior, a informação mútua entre duas variáveis aleatórias é a quantidade de informação comum entre essas variáveis. O problema de selecionar variáveis de entrada pode ser resolvido calculando a informação mútua (IM) entre variáveis de entrada e classes de saída. Se a informação mútua entre variáveis de entrada e classes de saída pudesse ser obtida com precisão, o problema FRn - k poderia ser reformulado como segue:

[FRn - k]: Dado um conjunto inicial F com n variáveis e C classes de saída, ache o subconjunto $S \subset F$ com k variáveis que minimizam H(C|S), isto é, que maximizam a informação mútua I(C;S).

O algoritmo de seleção que usa informação mútua é como segue:

1) (Inicialização) conjunto F ← "conjunto inicial com n variáveis", S ← "conjunto vazio" (a seta "←" sinaliza que F recebe "conjunto inicial com n variáveis" e S recebe "conjunto vazio").

2) (Cálculo da IM com a classe de saída), $\forall\ \phi_i \in F$, computa-se $I(C; \phi_i)$.

3) (Seleção da primeira variável) procura-se a variável ϕ_i que maximiza $I(C; \phi_i)$, e faz-se: $F \leftarrow F-\{\phi_i\}$, $S \leftarrow \phi_i$.

4) Repete-se até que o número desejado de variáveis seja selecionado.

a) (Cálculo da IM conjunta entre variáveis), $\forall\ \phi_i \in F$, computa-se $I(C; \phi_i; S)$.

b) (Seleção da próxima variável) escolhe-se a variável $\phi_i \in F$ que maximiza $I(C; \phi_i; S)$ e faz-se $F \leftarrow F-\{\phi_i\}$, $S \leftarrow \phi_i$.

5) Saída do conjunto S contém as variáveis selecionadas.

Este algoritmo é inviabilizado pelo tamanho do vetor de variáveis no cálculo de $I(C; \phi_i; S)$.

A.3.2
Seleção de Variáveis sob Informação Mútua (MIFS)

O algoritmo MIFS é similar ao algoritmo de seleção anterior com exceção do Passo 4. Em vez de se calcular $I(C;\phi_i;S)$, usa-se somente as seguintes informações mútuas: $I(C; \phi_i)$ e $I(\phi_i; \phi_s)$ (Battiti,1994). No MIFS, o passo 4 do algoritmo de seleção é substituído como segue:

4) Repete-se até o número desejado de variáveis a serem selecionadas.

a) (Cálculo da IM entre variáveis) para todos os pares de variáveis $(\phi_i; \phi_s)$ com $\phi_i \in F$, $\phi_s \in S$, calcula-se $I(\phi_i; \phi_s)$.

b) (Seleção da próxima variável) escolhe-se a variável com $\phi_i \in F$ que maximiza $I(C; \phi_i) - \beta \sum_{\phi s \in S} I(\phi_i; \phi_s)$ e faz-se $F \leftarrow F-\{\phi_i\}$, $S \leftarrow \phi_i$.

Nesse ponto, β é o parâmetro de redundância. Se β=0, o algoritmo seleciona variáveis na ordem da informação mútua entre variáveis de entrada e saída. A

redundância entre as variáveis de entrada nunca é refletida. Quando $\beta>0$ a redundância é reduzida.

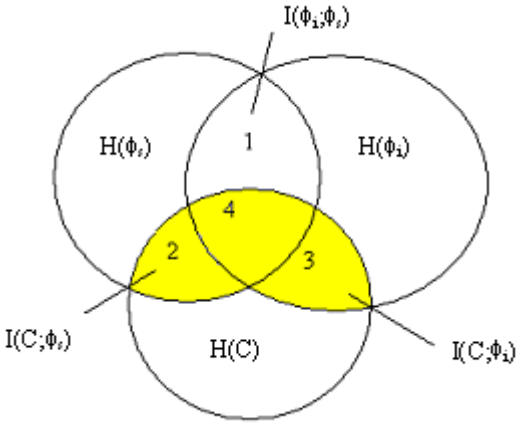


Figura A1 – Relação entre Variáveis de Entrada e Classe de Saída

A relação entre variáveis de entrada e saída pode ser representada na figura A1. O algoritmo de seleção usa a informação mútua para escolher a variável ϕ_i que maximiza a informação mútua conjunta $I(C; \phi_i; \phi_s)$ que são as áreas 2, 3, e 4. Como $I(C; \phi_s)$ (área 2 e 4) é comum para todas as variáveis não selecionadas ϕ_i, calculando-se a informação mútua conjunta $I(C; \phi_i; \phi_s)$, o algoritmo seleciona a variável que maximiza a área 3. Por outro lado, o algoritmo MIFS seleciona a variável que maximiza $I(C; \phi_i) - \beta\, I(\phi_i; \phi_s)$. Para $\beta=1$, isto corresponde a área 3 subtraída da área 1.

Entretanto, se uma variável a ser selecionada é fortemente relacionada com alguma variável já selecionada, a área 1 é grande e isto pode degradar o desempenho do algoritmo. Por isto, o MIFS pode não trabalhar bem em problemas não-lineares.

A.3.3
Seleção de Variáveis sob Informação Mútua com Distribuição Uniforme de Informação (MIFS-U)

O algoritmo maximiza a informação mútua $I(C; \phi_i; \phi_s)$ (áreas 2, 3, e 4 na figura A1) através da seguinte expressão:

$$I(C; \phi_I; \phi_s) = I(C; \phi_s) + I(C; \phi_i \mid \phi_s) \qquad (A.8)$$

Onde $I(C; \phi_i \mid \phi_s)$ representa a informação mútua restante entre a classe de saída C e a variável ϕ_i para um dado ϕ_s. Isto corresponde à área 3 na figura A1, onde a área 2 mais a área 4 representa $I(C; \phi_s)$. Como $I(C; \phi_s)$ é comum para todas as variáveis candidatas a serem selecionadas pelo algoritmo, não há nenhuma necessidade de se calcular essa informação mútua. Assim o algoritmo tenta achar a variável que maximiza $I(C; \phi_i \mid \phi_s)$ (área 3). Porém, calcular $I(C; \phi_i \mid \phi_s)$ requer tanto trabalho quanto calcular $I(C; \phi_i; \phi_s)$. O algoritmo então faz uma aproximação de $I(C; \phi_i \mid \phi_s)$ com $I(\phi_i; \phi_s)$ que são relativamente fáceis de se calcular. A informação mútua condicional $I(C; \phi_i \mid \phi_s)$ pode ser representada como:

$$I(C; \phi_i \mid \phi_s) = I(C; \phi_i) - \{ I(\phi_s; \phi_i) - I(\phi_s; \phi_i \mid C)\} \tag{A.9}$$

Onde $I(\phi_s, \phi_i)$ corresponde às áreas 1 e 4 e $I(\phi_s; \phi_i \mid C)$ corresponde à área 1. Assim, o termo $I(\phi_s; \phi_i) - I(\phi_s; \phi_i \mid C)$ corresponde à área 4 na figura A1. O termo $I(\phi_s; \phi_i \mid C)$ significa a informação mútua entre a variável já selecionada ϕ_s e o candidato ϕ_i para uma determinada classe. Se condicionando pela classe C a razão entre a entropia de ϕ_s e a informação mútua entre ϕ_s e ϕ_i não mudar, a seguinte relação pode ser escrita:

$$\frac{H(\phi_s \mid C)}{H(\phi_s)} = \frac{I(\phi_s, \phi_i \mid C)}{I(\phi_s, \phi_i)} \tag{A.10}$$

Onde $I(\phi_s ; \phi_i \mid C)$ pode ser representada por:

$$I(\phi_s, \phi_i \mid C) = \frac{H(\phi_s \mid C)}{H(\phi_s)} I(\phi_s; \phi_i) \tag{A.11}$$

Usando a equação acima e A9 tem-se que:

$$I(C;\phi_i \mid \phi_s) = I(C;\phi_i) - \left(1 - \frac{H(\phi_s \mid C)}{H(\phi_s)}\right) I(\phi_s;\phi_i) =$$
$$= I(C;\phi_i) - \frac{I(C;\phi_s)}{H(\phi_s)} I(\phi_s;\phi_i) \qquad \text{(A.12)}$$

A condição A10 é mais significativa quando sua distribuição é uniformemente distribuída ao longo da região de $H(\phi_s)$ da figura A1. Por essa razão refere-se ao algoritmo como MIFS-U.

Sendo assim, o passo (4) do algoritmo revisto segue:

4) Repete-se até que o número desejado de variáveis seja selecionado.

a) (Cálculo da entropia), $\forall \phi_s \in S$ computa-se $H(\phi_s)$.

b) (Cálculo da IM entre variáveis) para todos os pares de variáveis (ϕ_i, ϕ_s) com $\phi_i \in F$ *e* $\phi_s \in S$, calcula-se $I(\phi_s; \phi_i)$.

c) (Seleção da próxima variável) escolhe-se uma variável $\phi_i \in F$ que maximize $I(C;\phi_i) - \beta \sum_{\phi_s \in S} (I(C;\phi_s)/H(\phi_s)) I(\phi_i;\phi_s)$; e faz-se $F \leftarrow F-\{\phi_i\}$, $S \leftarrow \phi_i$.

Se $\beta=0$, o algoritmo seleciona variáveis na ordem da informação mútua entre variáveis de entrada e saída. A redundância entre as variáveis de entrada nunca é refletida. Quando $\beta>0$, o algoritmo exclui as variáveis redundantes mais eficazmente. Em geral, pode-se fixar $\beta=1$ (Breiman *et al*, 1984). Para todas as experiências desta dissertação fixou-se $\beta=1$.

Anexo B – Redes Neurais Artificiais

B.1
Introdução

As Redes Neurais Artificiais são uma metodologia, na fronteira da estatística com a inteligência artificial, eficiente e capaz de resolver uma gama de problemas importantes. Na área médica podem ser encontradas diversas aplicações, tais como Kattan (2002), Djavan *et al* (2002), Ohno-Machado (2002), Ohno-Machado & Musen (1996) e Yao & Liu (1999) dentre outros. Na literatura, alguns livros se destacam como de suma importância: Haykin (1998), Bishop (1995), Duda and Hart (1973) dentre outros.

A motivação original desta metodologia foi a tentativa de se modelar a rede de neurônios humanos visando compreender o funcionamento do cérebro. Portanto, como o próprio nome da metodologia revela, sua motivação inicial foi a de realizar tarefas complexas que o cérebro executa com elevada efetividade (por exemplo: reconhecimento de padrões, percepção e controle motor) através da simulação de seu funcionamento.

Segundo Haykin (1998), uma Rede Neural Artificial (RNA) é um sistema de processamento massivamente paralelo, composto por unidades simples com capacidade natural de armazenar conhecimento e disponibilizá-lo para uso futuro.

Do ponto de vista neurofisiológico, muito pouco se conhece sobre o funcionamento dos neurônios e suas conexões o que compromete a fidelidade destes modelos em fisiologia. As RNAs assemelham-se ao cérebro em dois aspectos:

- Elas extraem conhecimento do ambiente através de um processo de *aprendizagem* ou *treinamento*; e
- Os pesos das conexões entre os neurônios, conhecidos como *pesos sinápticos*, são utilizados para armazenar o conhecimento adquirido.

A natureza das RNAs faz com que seu estudo seja multidisciplinar, envolvendo pesquisadores de diversas áreas, como neurofisiologia, psicologia, física, computação, engenharia, estatística, entre outras.

Cientistas da área de computação têm em vista a construção de computadores dotados de processamento paralelo e distribuído, buscando superar as limitações impostas pelos computadores atuais, que realizam processamento serial simbólico.

Inspirados na habilidade apresentada pelos seres humanos e outros animais no desempenho de funções como o processamento de informação sensorial e a capacidade de interação com ambientes pouco definidos, os engenheiros, por exemplo, estão preocupados em desenvolver sistemas artificiais capazes de desempenhar tarefas semelhantes. Habilidades como capacidade de processamento de informação incompleta ou imprecisa e generalização são propriedades desejadas em tais sistemas.

McCulloch & Pitts (1943) projetaram a estrutura que é conhecida como a unidade básica de uma Rede Neural. Estes pesquisadores propuseram um modelo de neurônio como uma unidade de processamento binária e provaram que estas unidades são capazes de executar várias operações lógicas (OU, AND, etc.). Este modelo, apesar de muito simples, fornece uma grande contribuição para as discussões sobre a construção dos primeiros computadores digitais, permitindo a criação dos primeiros modelos matemáticos de dispositivos artificiais que buscavam analogias biológicas. Matematicamente, um neurônio pode ser expresso por:

$$y = f(u) = f(x_1 w_1 + x_2 w_2 + ... + x_n w_n) = f(w^T x) \qquad \text{(B.1)}$$

Onde y é a saída do neurônio, u é a ativação do neurônio, f(.) sua função de ativação, x_i (i = 1,...,n) é o i-ésimo componente do vetor $\boldsymbol{x}$ de entradas, e w_i (i = 1,...,n) é o i-ésimo componente do vetor $\boldsymbol{w}$ de pesos do neurônio.

B.2
Variáveis Principais

A comparação com a neurofisiologia foi apenas uma motivação original da qual pouco sobrou além do nome da ferramenta. No exterior, em especial nos E.U.A, já encontra-se grande aplicabilidade para as redes neurais fora dos muros acadêmicos sendo que, no Brasil, começa-se a perceber seu grande potencial. Conceitualmente, uma rede neural artificial é um dispositivo tanto capaz de processar informação de forma distribuída quanto de incorporar conhecimento através de exemplos. Trata-se, portanto, de um processador capaz de extrair conhecimento experimental disponibilizando-o para uso prático (tomada de decisões, por exemplo).

As Redes Neurais artificiais têm sido desenvolvidas como generalizações de modelos matemáticos de cognição humana ou neurobiologia, assumindo que:

- O processamento da informação ocorre com o auxílio de vários elementos chamados *neurônio*s;
- Os sinais são propagados de um elemento a outro através de *conexõe*s;
- Cada conexão possui um *peso* associado, que, em uma Rede Neural típica, pondera o sinal transmitido; e
- Cada neurônio (ou unidade) possui uma *função de ativação* (geralmente não-linear), que tem como argumento a soma ponderada dos sinais de entrada, para determinar sua saída.

Uma grande vantagem de usar-se uma Rede Neural é a capacidade de resolver problemas sem a necessidade de definição de listas de regras ou de modelos explícitos. Isto possibilita tratar de situações onde é difícil criar modelos adequados da realidade ou situações com freqüentes mudanças no ambiente.

Atenta-se que grande parte desta sua adeqüabilidade funcional deve-se à sua capacidade em inferir relações não-lineares complexas. Frente a estas suas propriedades, hoje, pode-se observar sua aplicabilidade principalmente nas áreas de classificação de padrões (em um sentido amplo) e de previsão.

Uma Rede Neural caracteriza-se pela capacidade de extrair conhecimento experimental e por disponibilizar este conhecimento para uso prático. Apesar da plausibilidade biológica ter sido apenas uma motivação original cabe aqui uma

comparação. O cérebro desenvolve a função de, a partir da observação de dados (*input*), extrair informação disponibilizando-a para a tomada de decisões.

Sabe-se que o conhecimento é adquirido através de um processo de aprendizado. O mesmo acontece com as Redes Neurais artificiais. A informação é armazenada em "densidades de conexão" conhecidas como "pesos sinápticos" (ou simplesmente pesos). O processo de aprendizado de uma Rede se dá através de um algoritmo que deve ser capaz de ajustar iterativamente os pesos de modo que se atinja o objetivo proposto.

A Rede Neural aprende, então, o ambiente através de um processo iterativo de modificação dos pesos de interligação, a partir de estímulos fornecidos pelo ambiente. O tipo de aprendizado é determinado pelo modo com que se promove a adaptação dos parâmetros e isso pode ser feito de dois modos:

1) Aprendizado Supervisionado – usa-se um conjunto de pares, entrada e saída, previamente conhecidos que representam a realidade;

2) Aprendizado Não Supervisionado – não se usa um conjunto de exemplos previamente conhecidos. Uma medida da qualidade da representação do ambiente pela Rede é estabelecida e os parâmetros são modificados de modo a otimizar esta medida. Este tipo de aprendizado é muito utilizado na área de reconhecimento de padrões.

Salienta-se que, nesta dissertação, utilizou-se o aprendizado supervisionado, ou seja, escolheu-se as variáveis referentes a sintomas e ou sinais, dados clínicos e laboratoriais como uma medida da representação do ambiente em estudo.

Em síntese, uma Rede Neural pode ser caracterizada por três aspectos principais: (1) padrão de conexões entre as unidades *(arquitetura* ou *estrutur*a), (2) método de determinação dos pesos das conexões *(algoritmo de treinamento* ou *aprendizage*m) e (3) *função de ativaçã*o.

B.2.1 Arquitetura

A forma pela qual os neurônios de uma RNA estão estruturados (interconectados) está intimamente relacionada ao algoritmo de aprendizagem a ser utilizado para treiná-la. Em geral é possível distinguir três classes fundamentais de arquiteturas: *Redes feedforward de uma única camad*a, *Redes*

feedforward de múltiplas camadas e *Redes recorrente*s, sendo de interesse desta dissertação os dois primeiros casos.

B.2.1.1
Redes *Feedforward* de Uma Única Camada

No caso mais simples de Redes em camadas *(layer*s), tem-se uma camada de entrada com neurônios cujas saídas alimentam a última camada da rede. Geralmente, os neurônios de entrada são propagadores puros, ou seja, eles simplesmente repetem o sinal de entrada em sua saída distribuída. Por outro lado, as unidades de saída costumam ser unidades processadoras, como apresentado na Figura B1. A propagação de sinais nesta Rede é puramente unidirecional *(feedforwar*d): os sinais são propagados apenas da entrada para a saída, e nunca de forma reversa. Esta arquitetura está ilustrada na Figura B1(a) e a direção de propagação dos sinais na Figura B1(b).

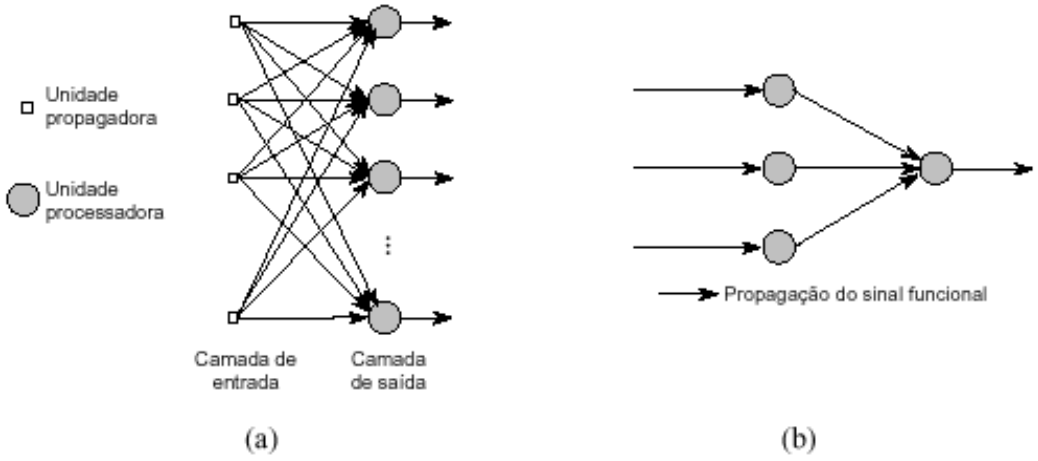


Figura B1 – Redes Neurais tipo *feedforward* com uma Única Camada de Unidades Processadoras. (a) Arquitetura (b) Sentido de Propagação do Sinal Funcional

B.2.1.2
Redes *Feedforward* de Múltiplas Camadas

A segunda classe de Rede *feedforward* se distingue pela presença de uma ou mais camadas intermediárias ou escondidas (camadas em que os neurônios são efetivamente unidades processadoras, mas não correspondem à camada de saída). Adicionando-se uma ou mais camadas intermediárias, aumenta-se o poder

computacional de processamento não-linear e armazenagem da rede. O conjunto de saídas dos neurônios de cada camada da Rede é utilizado como entrada para a camada seguinte. A Figura B2(a) ilustra uma Rede feedforward de múltiplas (duas) camadas intermediárias.

As Redes feedforward de múltiplas camadas, são geralmente treinadas usando o algoritmo de retro-propagação do erro *(error backpropagation*), embora existam outros algoritmos de treinamento. Este algoritmo requer a propagação direta *(feedforwar*d) do sinal de entrada através da rede, e a retro-propagação (propagação reversa, ou *backpropagatio*n) do sinal de erro, como ilustrado na Figura B2(b).

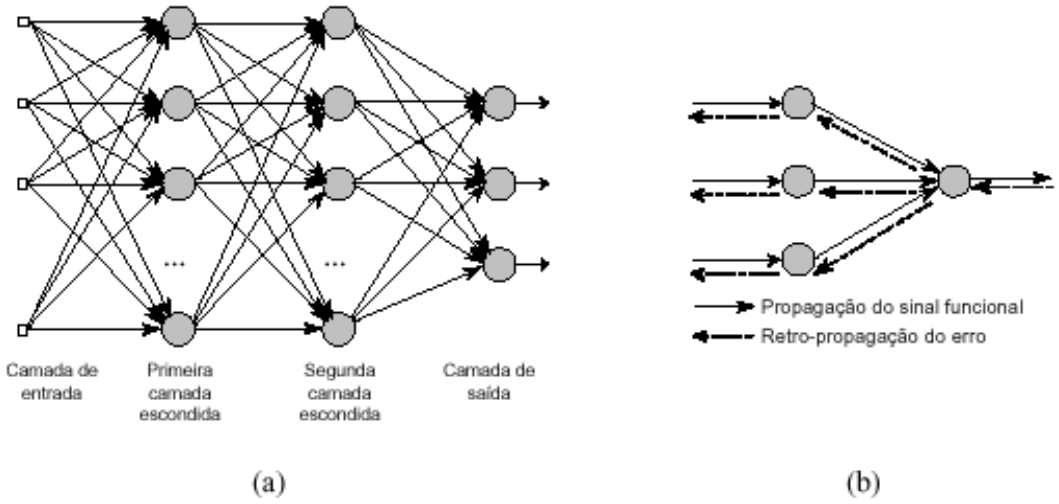


Figura B2 – Redes Neurais tipo *feedforward* com Múltiplas Camadas. (a) Arquitetura (b) Sentido de Propagação do Sinal Funcional e do Sinal de Erro

B.2.2
Métodos de Estimação

A capacidade de *aprendizagem* é uma das características das RNAs. Uma Rede Neural aprende, basicamente, através de um processo iterativo de ajuste de pesos e limiares (*bias*). Atualmente, existem processos mais sofisticados de aprendizagem (ou *treinamen*to), que são capazes de ajustar não apenas os pesos da rede, mas também sua arquitetura e as funções de ativação dos neurônios (Kwok & Yeung, 1997, de Castro *et al*., 1999a,b; de Castro *et al*, 1999).

Segundo Haykin (1998), *Aprendizagem* (ou *treinamen*to) é o processo pelo qual os parâmetros livres de uma Rede Neural são adaptados, através de um

mecanismo de apresentação de estímulos fornecidos pelo ambiente no qual a Rede está inserida. O tipo de treinamento é definido pela forma na qual os parâmetros são modificados.

Esta definição de aprendizagem implica na seguinte seqüência de eventos:

- Apresentação de estímulos à Rede Neural;
- Alteração dos parâmetros livres da rede; e
- Novo padrão de resposta ao ambiente.

Os principais paradigmas de aprendizagem são: (1) supervisionada, (2) não-supervisionada, e (3) por reforço.

B.2.2.1
Aprendizagem Supervisionada

Trata-se de um paradigma de aprendizagem, no qual um *supervisor* possui conhecimento sobre o ambiente em que a Rede está inserida. Este conhecimento está representado sob a forma de um conjunto de amostras de *entrada-saída*. O ambiente, por sua vez, é desconhecido. A Figura B3 ilustra esta abordagem. Os parâmetros da Rede são ajustados pela combinação do sinal de entrada com um sinal de erro, que é a diferença entre a saída desejada e a fornecida pela Rede.

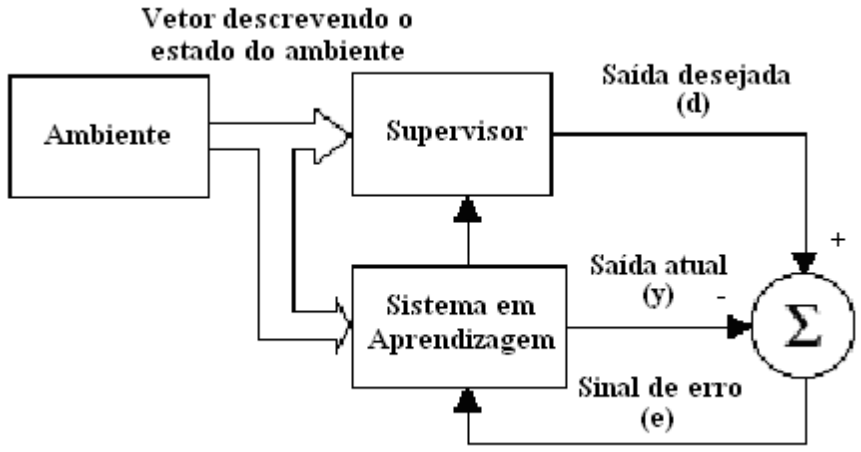


Figura B3 – Diagrama de Blocos do Processo de Aprendizagem Supervisionada

Seja *t* o índice que denota tempo discreto ou, mais precisamente, o intervalo de tempo do processo iterativo responsável pelo ajuste de pesos do neurônio k. O único sinal de saída $y_k(t)$, do neurônio k, é comparado com uma *saída desejad*a, denominada $d_k(t)$.

Conseqüentemente, um sinal de erro $e_k(t)$ é produzido:

$$e_k(t) = d_k(t) - y_k(t) \tag{B.2}$$

B.2.2.1.1
Aprendizagem com Regularização Bayesiana

Um *perceptron* calcula a combinação linear dos dados de entrada de uma rede e os submete a uma função de ativação (linear ou não) produzindo uma saída. Um perceptron de múltiplas camadas (MLP), pode ser definido como um modelo não-linear que aproxima as realizações de um processo estocástico por uma função $G : Xx\Psi \to \Re$ onde $X \subset \Re^n$ e Ψ é um subconjunto compacto de dimensão finita de $\Re^p$, sendo p o número de pesos da rede. Estas definições são atendidas pela especificação de uma rede com uma única camada oculta de neurônios (White & Racine, 2001):

$$y_i = G(x,\psi) + \varepsilon_i = \alpha_0 + \sum_{h=1}^{H} \alpha_h F(\gamma_0 + \sum_{i=1}^{I} \gamma_{hi} x_i) + \varepsilon_i \tag{B.3}$$

Onde $(x,\psi) \in XxΨ$ sendo $x = [x_1, x_2, ...x_I]$ vetores de variáveis independentes e ψ o vetor de parâmetros $\psi = [\alpha', \gamma']$, composto pelos vetores de pesos da camada de saída e da camada oculta respectivamente. Os parâmetros α_0 *e* γ_0 são respectivamente o bias para a camada de saída (intercepto) e o *bias* para a camada oculta. A aplicação $F(x,\psi) \to \Re$ contínua para todo $x \in X$, chamada função de ativação é a função logística:

$$F(x) = (1 + e^{-x})^{-1} \tag{B.4}$$

Os MLP são modelos não-lineares que para um dado número de neurônios na camada oculta e um tamanho suficiente da amostra podem aproximar qualquer função, em outras palavras, um MLP é um aproximador universal (Cybenko, 1989).

O aprendizado ou treinamento de uma rede neural tem tipicamente por objetivo reduzir a soma dos quadrados dos erros (Foresee & Hagan, 1997):

$$\hat{\psi} = \arg\min_{\psi} Q_1(\psi) = \arg\min_{\psi} \sum_{t=1}^{N} (y_t - G(x,\psi))^2 \tag{B.5}$$

Assim como outros modelos flexíveis não-lineares, as RNAs podem sofrer de *"overfitting"*. Este problema ocorre quando se utiliza um número excessivo de neurônios na camada oculta, que levarão a uma perda da capacidade de generalização (fora-da-amostra). Em contrapartida, se o número de neurônios em excesso for reduzido, teremos perda da capacidade de aproximar o processo gerador dos dados (Medeiros e Pedreira, 2001).

Atualmente, diversas metodologias são utilizadas para solucionar o problema de *"overfitting"* (Haykin, 1998). Nesta dissertação foi utilizado o procedimento desenvolvido por Mackay (1992), chamado de Regularização Bayesiana, que consiste em adicionar um termo de penalização (regularização) à função objetivo, de forma que o algoritmo de estimação faça com que os parâmetros irrelevantes convirjam para zero, reduzindo assim o número de parâmetros efetivos utilizados no processo.

Seguindo a notação utilizada por Medeiros e Pedreira (2001), o problema de estimação passa a ser definido como:

$$\hat{\psi} = \arg\min_{\psi} Q_T(\psi) = \arg\min_{\psi} \sum_{t=1}^{N} (\eta Q_1(\psi) - \phi Q_2(\psi))^2 \tag{B.6}$$

Onde a função de penalização é a soma do quadrado dos parâmetros:

$$Q_2(\psi) = \sum_{h=0}^{H} \alpha_h^2 + \sum_{h=0}^{H} \sum_{i=0}^{I} \gamma_{hi}^2 \tag{B.7}$$

O problema de regularização é otimizar a função objetivo de forma a encontrar valores para os parâmetros ϕ *e* η. Este problema de otimização requer o cálculo da matriz Hessiana como pode ser visto em Mackay (1992). O algoritmo desenvolvido por Foresee e Hagan (1997) propõe a aproximação da matriz

Hessiana pelo algoritmo de Levenberg-Marquardt (Levenberg, 1944; Marquardt, 1963), reduzindo o custo computacional.

A aproximação é feita utilizando os seguintes passos (Foresee e Hagan, 1997):

i) Faz-se $\phi = 0 \; e \; \eta = 1$ e utiliza-se o método de Nguyen-Widrow (1989) para inicializar os parâmetros;

ii) Faz-se uma estimativa (um passo) do algoritmo de Levenberg-Marquardt minimizando $Q_1(\psi)$;

iii) Calcula-se o número efetivo de parâmetros $\hat{\delta} = \dim(\psi) - 2\eta.tr(\hat{H})^{-1}$ onde $\hat{H}$ é aproximação da matriz Hessiana feita pelo algoritmo de Levenberg-Marquardt: $\hat{H} = \nabla^2 Q_T(\psi) \approx 2\eta J^T J + 2\phi I_N$ onde J é matriz jacobiana dos erros;

iv) Calcula-se as novas estimativas para $\phi \; e \; \eta$, $\hat{\eta} = \dfrac{T - \hat{\delta}}{2Q_1(\psi)} \; e \; \hat{\phi} = \dfrac{\hat{\delta}}{2Q_2(\psi)}$, onde T é o número de observações;

v) Repete-se os passos ii, iii e iv até a convergência.

Uma discussão detalhada do uso da Regularização Bayesiana, em combinação com o treinamento de Levenberg-Marquardt (Levenberg, 1944; Marquardt, 1963; Hagan, 1994), pode ser encontrada em Foresee & Hagan (1997).

A Regularização Bayesiana está implementada na função *trainbr* encontrada no *toolbox* do Matlab. Uma característica deste algoritmo é que ele fornece uma medida de quantos parâmetros da Rede (pesos e vieses) estão sendo, efetivamente, usados pela rede. Quando se usa a função *trainbr*, é importante deixar o algoritmo trabalhar até o efetivo número de parâmetros convergir. O treinamento pode parar com a mensagem "Máximo MU alcançado". Isto é uma indicação de que o algoritmo verdadeiramente convergiu. Você também pode saber que o algoritmo convergiu quando a soma dos quadrados dos erros (MSE) e a soma dos quadrados dos pesos (MSW) são relativamente constantes em várias repetições.

B.2.2.1.2
Avaliação do Modelo

A essência da aprendizagem por retropopagação do erro é codificar um mapeamento de entrada-saída (representado por um conjunto de exemplos rotulados) nos pesos sinápticos e limiares de um perceptron de múltiplas camadas. Espera-se que a Rede torne-se bem treinada de modo que aprenda o suficiente do passado para generalizar no futuro. Desta perspectiva, o processo de aprendizagem se transforma em uma escolha de parametrização da Rede para esse conjunto de dados (Haykin, 1998).

Quando o número de exemplos rotulados disponíveis, N, for severamente limitado, pode-se usar a forma extrema de validação cruzada múltipla conhecida como o "método deixe um de fora" (*leave-one-out method*, Haykin, 1998 e Bishop, 1995). Neste caso, N-1 exemplos são usados para treinar o modelo, e o modelo é validado testando-o sobre o exemplo deixado de fora. O experimento é repetido para um total de N vezes, cada vez deixando de fora um exemplo diferente para a validação. O erro quadrado na validação é então a média sobre as N tentativas do experimento (Haykin, 1998, Bishop, 1995). No caso do banco de dados desta dissertação, o uso do critério de *leave-one-out* seria o mais indicado. Porém, para que se pudesse estabelecer uma comparação com pesquisas previamente realizadas e encontradas na literatura, e também aplicar a metodologia proposta nesta dissertação, o conjunto de dados disponível foi dividido aleatoriamente em um conjunto de treinamento e em um conjunto de teste, na proporção de 2/3 das amostras para treinamento e 1/3 para teste (generalização). Desta forma, foi utilizada a mesma proporção das pesquisas previamente realizadas encontradas na literatura, às quais foram comparados os resultados aqui obtidos. Foram utilizados 50 conjuntos de treinamento/teste diferentes nos experimentos aqui realizados para que se ter uma boa amostragem estatística, e a média dos resultados desses 50 conjuntos foi o resultado final obtido.

Anexo C – Análise Discriminante

C.1
Introdução

A Análise Discriminante, como método estatístico multivariado, compõe um conjunto de técnicas destinadas a tratar problemas de classificação. Esta técnica surgiu com o objetivo de se distinguir estatisticamente dois ou mais grupos de indivíduos, previamente definidos a partir de variáveis conhecidas para todos os membros dos grupos. Isto é, pretende-se discriminar grupos de indivíduos definidos a priori com base num critério pré-definido, a partir da informação recolhida sobre os indivíduos desses grupos.

Esta técnica de análise multivariada é empregada para descobrir as variáveis que distinguem os membros de um grupo dos de outro, de modo que, conhecidas as variáveis de um novo indivíduo, se possa identificar a que grupo ele pertence. Neste sentido, a Análise Discriminante tem um importante campo de aplicação em problemas de diagnóstico médico. Podem ser encontrados exemplos em Hermans & Habbema (1975), Anderson (1974), Aitchison & Dunsmore (1975), Titteringtn *et al* (1981) e Krusinska & Liebhart (1990), dentre outros.

Em geral, o objetivo da Análise Discriminante é encontrar a separação máxima entre os grupos através da maximização da diferença entre as médias dos grupos relativamente aos desvios padrão no interior de cada grupo. A idéia central é substituir as variáveis originais, em geral numerosas e correlacionadas, por uma combinação linear cujos valores diferenciem ao máximo os indivíduos de seus grupos. Para classificar um indivíduo, calcula-se o valor da combinação linear para seus atributos e verifica-se se este é menor ou maior que um valor limite calculado, de forma a minimizar a probabilidade de erro de classificação (Johnson & Wichern , 1998).

C.2
Relação com o Modelo de Regressão Múltipla

Do ponto de vista algébrico, o modelo de Análise Discriminante para 2 grupos é um caso especial de um modelo de regressão múltipla, onde a variável dependente assume valores discretos. Considere o modelo a seguir:

$$g_{(i)} = b^T X_{(i)} + b_0 + e_{(i)} \quad \text{(C.1)}$$

Onde $\mathbf{g_{(i)}}$ é a classe do individuo i (0 ou 1), **b** é o vetor de coeficientes da regressão**,** $\mathbf{b_0}$ é o intercepto e $\mathbf{e_{(i)}}$ é o erro do modelo. Em condições ideais, a estimação por mínimos quadrados de **b** produz um vetor na mesma direção do eixo determinado pelo método de Fisher (1936), descrito mais adiante.

C.3
Hipótese do Modelo de Análise Discriminante

Para que a regra de classificação fornecida pela Análise Discriminante seja ótima, tornando a probabilidade de erro de classificação mínima, é necessário que os dados atendam as seguintes condições:

- As variáveis explicativas ($\mathbf{X_{(i)}}$) tenham distribuição normal multivariada.
- A matriz de covariância dos grupos seja a mesma. Por outro lado, quanto mais diferentes as médias, mais fácil será discriminar os grupos.

C.4
Método de Fisher para 2 Grupos

O método de Fisher (1936) busca a combinação linear que maximiza a razão da distância ao quadrado entre as projeções dos centróides dos dois grupos para a variância das projeções. Seja **b** o vetor com os pesos da combinação linear. Então, $\mathbf{y_{(i)}}$, o valor da projeção de $\mathbf{X_{(i)}}$ sobre **b**, é dado por:

$$y_{(i)} = b^T X_{(i)} \quad \text{(C.2)}$$

Sejam $\overline{X}_1$ *e* $\overline{X}_2$ os centróides dos grupos 1 e 2, respectivamente. Então, as médias das projeções sobre **b** são dadas por:

$$\bar{y}_1 = b^T \bar{x}_1 \quad \text{e} \quad \bar{y}_2 = b^T \bar{x}_2 \tag{C.3}$$

Supondo que as matrizes de covariância de ambos os grupos são iguais e denotadas por $\mathbf{C_x}$, tem-se que a variância das projeções para qualquer dos grupos é dada por:

$$S_y^2 = b^T C_x \tag{C.4}$$

O objetivo do método é encontrar **b** que maximiza a proporção entre a diferença das médias de $\bar{y}_1$ *e* $\bar{y}_2$ ao quadrado e a variância de **y**, dada por:

$$\Delta = \frac{\left(b^T(\bar{x}_1 - \bar{x}_2)\right)^2}{b^T C_x b} \tag{C.5}$$

A solução deste problema é dada por:

$$b = S^{-1}(\bar{x}_1 - \bar{x}_2) \tag{C.6}$$

Onde **S** é a matriz de covariância (comum aos dois grupos) estimada por:

$$S = \frac{1}{n_1 + n_2 - 2}(x_1^T x_1 + x_2^T x_2) \tag{C.7}$$

E onde x_1 e x_2 são as matrizes de dados referentes aos grupos 1 e 2, com, respectivamente, n_1 e n_2 indivíduos.

A chamada função discriminante de Fisher é dada por:

$$Y_{(i)} = b^T X_{(i)} = (\bar{x}_1 - \bar{x}_2)^T S^{-1} X_{(i)} \tag{C.8}$$

A função discriminante de Fisher pode ser usada para construir uma regra de decisão: uma observação $\mathbf{X_{(i)}}$ será classificada no grupo cuja média está mais próxima. Isto é, se $| y_{(i)} - \bar{y}_1 | < | y_{(i)} - \bar{y}_2 |$ então $\mathbf{X_{(i)}}$ é classificado no grupo 1.

C.5
Probabilidades a Priori e Função de Custo

É comum em situações práticas que se tenha uma probabilidade *a priori* do indivíduo pertencer a um ou outro grupo. Ao mesmo tempo, também ocorre que as conseqüências de um erro de classificação sejam diferentes segundo o tipo de erro.

A função discriminante de Fisher pode incorporar estes parâmetros. Para isso determina-se **b** que minimiza o custo esperado médio dos erros de classificação.

Se $\mathbf{p_i}$ é a probabilidade *a priori* do individuo pertencer ao grupo **i** e $\mathbf{C_{ji}}$ é o custo de classificar um individuo da classe **j** como do grupo **i**, então, a regra ótima de classificação é: se $| b^T (X_{(i)} - \bar{x}_1) | < k | b^T (X_{(i)} - \bar{x}_2) |$ então $X_{(i)}$ é classificado no grupo 1, onde $k=p_2C_{12}/p_1C_{21}$.

C.6
Método *Stepwise*

Por vezes, o investigador vê-se confrontado com um conjunto de informações, sob a forma de variáveis, superior ao necessário para se obter uma distinção satisfatória. Neste caso é possível utilizar um método discriminante *stepwise* (Drapper & Smith, 1981), o qual começa por selecionar as variáveis que mais contribuem para a distinção entre grupos, e em seguida vai incluindo e/ou retirando variáveis nas funções discriminantes, uma a uma, de acordo com um critério que pode ser definido pelo próprio analista.

O método *stepwise* começa por escolher a variável que mais diferencia os grupos de acordo com o critério pré-estabelecido. A segunda variável a ser escolhida é a que, juntamente com a primeira, maximiza o aumento do critério discriminante, e assim por diante. De acordo com este processo, variáveis já escolhidas nas etapas anteriores podem ser retiradas e novas introduzidas, se tais variáveis contribuírem para um aumento do critério definido.

No passo final, ou se verifica que todas as variáveis foram selecionadas ou, então, que as que foram rejeitadas não teriam qualquer contribuição adicional para a distinção entre os grupos.

C.7
Λ de Wilks

A estatística Λ de Wilks é definida como a razão entre a dispersão das médias dos grupos e a variância total e é expressa por:

$$\lambda_{(i)} = \frac{1}{\text{var}(X_{(i)})} \left[\frac{\sum_{j=1}^{k} n_j \, \text{var}(X_{j_{(i)}})}{n} \right] \qquad \text{(C.9)}$$

Por ser uma estatística inversa, a primeira variável a ser escolhida é a que produz o menor valor de Λ. É possível aproximar esta estatística a um teste para a diferença de médias entre os grupos com distribuição F. Depois de feita esta aproximação, a variável a ser introduzida é a que provoca um maior acréscimo no valor de F.

C.8
Estatística F

Uma outra medida, mais formal, da importância de uma variável para a diferenciação entre dois grupos é dada pela estatística F-parcial. A interpretação desta estatística está ligada ao problema de determinar se o grupo a que pertence o indivíduo influencia o valor da variável $\mathbf{X_{(i)}}$, descontadas as contribuições de outras variáveis.

A estatística F-parcial é calculada em função do coeficiente de determinação, R^2, da regressão associada ao modelo:

$$F_{p-1,n-1} = \frac{R^2/(p-1)}{(1-R^2)/(n-p)} \qquad \text{(C.10)}$$

Esta estatística é o critério de entrada e saída de variáveis mais utilizado. Os valores ideais do nível de significância de entrada e saída de uma variável do modelo devem ser definidos pelo usuário. Tradicionalmente, o F-parcial para incluir uma variável deve ser maior do que para excluí-la.

Anexo D – Árvores de Decisão e Algoritmo C4.5

D.1
Árvores de Decisão

O processo de tomada de decisão pode conter vários passos para se obter um resultado. Há várias ferramentas que auxiliam nesse processo. Uma dessas ferramentas é denominada "Árvore de Decisão". Esse modelo é interessante porque pode ser utilizado em diversas áreas de atuação.

Uma árvore de decisão é um modelo de uma função no qual é determinado o valor de uma variável e, com base nesse valor, alguma ação é executada. A ação pode ser para a escolha de uma outra variável ou para a emissão da saída do valor da função.

Cada ação executada depende do valor atual da variável que está sendo testada e de todas as ações anteriores que foram executadas. Em uma árvore de decisão uma variável só é testada uma vez para evitar testes redundantes.

As árvores de decisão são normalmente construídas a partir da descrição de um problema e dão uma visão gráfica da tomada de decisão necessária.

Especificam:

• Variáveis a serem testadas

• Ações a serem executadas

• Ordem da tomada de decisão

É uma técnica que permite visualizar a estrutura de decisão de um processo. Os ramos da árvore correspondem a cada uma das possibilidades lógicas. É também uma boa forma de se esquematizar a estrutura lógica e para se obter a confirmação de que a lógica expressada está correta. De forma clara e objetiva, permite a leitura da combinação das circunstâncias que levam a cada ação.

As árvores de decisão são mais fáceis de ler e compreender quando o número de condições e ações é pequeno. Se existir um número considerável de condições e ações, a árvore de decisão torna-se muito grande e complicada.

D.2
Algoritmo C4.5

Este algoritmo pode auxiliar na execução de tarefas que muitas vezes consistem simplesmente em fornecer a classificação de um caso apresentado. O C4.5 é um algoritmo que pode ser utilizado nessas tarefas, que algumas vezes constam apenas de uma decisão de sim ou não. Muitas vezes as decisões seguem um padrão, por isso muitos especialistas fazem a classificação de um caso olhando para modelos anteriores cujo resultado é conhecido. O C4.5 é capaz de aprender, olhando para um conjunto desses casos, como eles são classificados e a partir daí fazer uma predição para novos casos. O C4.5 age como um especialista, classificando os casos desconhecidos.

Um exemplo clássico, apresentado no livro “C4.5: Programs for Machine Learning” (Quinlan, 1993), da utilização do algoritmo para resolução de um problema através de árvores e tabelas ou regras de decisão, é apresentado a seguir.

Primeiramente, a tabela de decisões a serem tomadas em um jogo de golfe, dependendo das condições climáticas, é apresentada:

Tabela D1 – Tabela de Decisões – Jogo de Golfe

Variáveis				**Desfecho**
Previsão	**Temperatura (F°)**	**Umidade (%)**	**Vento**	**Jogar (positivo) / Não Jogar (negativo)**
ensolarado	85	85	não	Não Jogar
ensolarado	80	90	sim	Não Jogar
encoberto	83	78	não	Jogar
chovendo	70	96	não	Jogar
chovendo	68	80	não	Jogar
chovendo	65	70	sim	Não Jogar
encoberto	64	65	sim	Jogar
ensolarado	72	95	não	Não Jogar
ensolarado	69	70	não	Jogar
chovendo	75	80	não	Jogar
ensolarado	75	70	sim	Jogar
encoberto	72	90	sim	Jogar
encoberto	81	75	não	Jogar
chovendo	71	80	sim	Não Jogar

Esta tabela pode ser resumida na árvore gerada pelo algoritmo C4.5 mostrada a seguir:

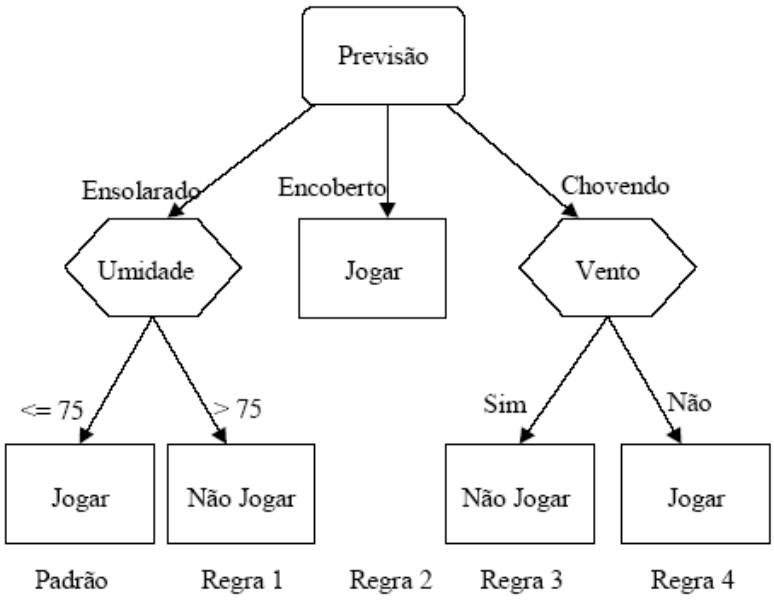


Figura D1 – Árvore de Decisão – Jogo de Golfe

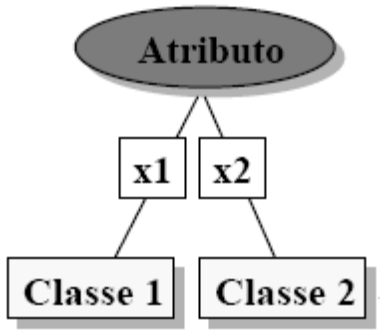
Para que se possa entender o funcionamento de um processo de tomada de decisão, através de uma árvore de decisão e utilizando o algoritmo C4.5, é apresentado um exemplo a seguir, onde todos os passos estão detalhados.

O exemplo em questão relaciona o uso de lentes de contato e o tipo de lentes (sem lentes, lentes gelatinosas, lentes duras), com algumas variáveis, denominadas: “Idade” (jovem, média, sênior), “Prescrição” (Míope, Hipermíope), “Astigmático” (não, sim) e “Lágrimas” (normal, reduzida). A tabela de variáveis e desfecho deste exemplo é dada a seguir:

Tabela D2 – Conjunto de Treinamento – Lentes de Contato

Variáveis				**Desfecho**
Idade	**Prescrição**	**Astigmático**	**Lágrimas**	**Lentes**
Jovem	Míope	Não	Normal	Lentes gelatinosas
Jovem	Míope	Não	Reduzida	Sem lentes
Jovem	Míope	Sim	Normal	Lentes duras
Jovem	Míope	Sim	Reduzida	Sem lentes
Jovem	Hipermíope	Não	Normal	Lentes gelatinosas
Jovem	Hipermíope	Não	Reduzida	Sem lentes
Jovem	Hipermíope	Sim	Normal	Lentes duras
Jovem	Hipermíope	Sim	Reduzida	Sem lentes
Média	Míope	Não	Normal	Lentes gelatinosas
Média	Míope	Não	Reduzida	Sem lentes
Média	Míope	Sim	Normal	Lentes duras
Média	Míope	Sim	Reduzida	Sem lentes
Média	Hipermíope	Não	Normal	Lentes gelatinosas
Média	Hipermíope	Não	Reduzida	Sem lentes
Média	Hipermíope	Sim	Normal	Sem lentes
Média	Hipermíope	Sim	Reduzida	Sem lentes
Sênior	Míope	Não	Normal	Sem lentes
Sênior	Míope	Não	Reduzida	Sem lentes
Sênior	Míope	Sim	Normal	Lentes duras
Sênior	Míope	Sim	Reduzida	Sem lentes
Sênior	Hipermíope	Não	Normal	Lentes gelatinosas
Sênior	Hipermíope	Não	Reduzida	Sem lentes
Sênior	Hipermíope	Sim	Normal	Sem lentes
Sênior	Hipermíope	Sim	Reduzida	Sem lentes

De uma forma resumida, a estrutura de uma árvore de decisão é caracterizada da seguinte maneira:



← Cada **nodo** representa um atributo da base de dados;

← Cada **galho** representa um valor (ou faixa de valor) do atributo;

← Cada **folha** representa uma classe.

Figura D2 – Estrutura de uma árvore de decisão

Existem essencialmente dois tipos de medidas de correlação entre duas variáveis aleatórias: *linear* e *não-linear*. Entretanto, não é confiável assumir sempre correlação linear entre variáveis que representam características do mundo real. Medidas de correlação linear podem não ser capazes de capturar correlações que não são lineares na natureza.

Entre as medidas de correlação não-lineares, muitas são baseadas no conceito de *entropia* de Teoria da Informação, uma medida de incerteza de uma variável aleatória. A entropia de uma variável X é definida como:

$$H(X) = -\sum_{i} P(x_i)\log(P(x_i)) \tag{D.1}$$

A entropia de X depois de observados os valores de uma outra variável Y é definida como:

$$H(X \mid Y) = -\sum_{j} P(x_j)\sum_{i} P(x_i \mid y_j)\log(P(x_i \mid y_j)) \tag{D.2}$$

Onde:

$P(x_i)$ → Probabilidades a priori para todos os valores de X;

$P(x_i \mid y_j)$ → Probabilidades a posteriori de X dados os valores de Y.

O valor pelo qual a entropia de X decresce reflete a informação adicional sobre X proveniente de Y e é chamada de *ganho de informação* (Quinlan, 1993), e dado por:

$$Ganho(X \mid Y) = H(X) - H(X \mid Y) \quad (D.3)$$

De acordo com esta medida, a característica Y é considerada mais correlatada com a variável X do que com a variável Z se:

$$Ganho(X \mid Y) > Ganho(Z \mid Y) \quad (D.4)$$

A raiz a ser escolhida para a árvore de decisão é a variável que apresenta o maior ganho de informação com relação ao desfecho. Assim, para o exemplo em questão, temos:

Ganho (Lentes,Idade) = 0,039397
Ganho (Lentes,Prescrição) = 0,0395113
Ganho (Lentes,Astigmático) = 0,3770057
Ganho (Lentes,Lágrimas) = 0,548954

A variável "Lágrimas" apresentou o maior ganho de informação com relação ao desfecho "Lentes". Adiciona-se então a raiz da árvore:

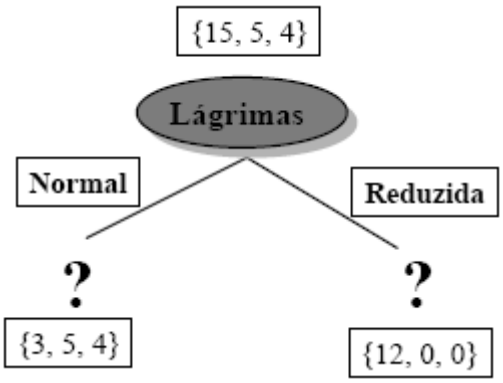


Figura D3 – Árvore de decisão – Lentes de Contato – Esquema 1

A notação "{15,5,4}", significa: 15 amostras com indicação "Sem lentes", 5 amostras com identificação "Lentes gelatinosas", e 4 amostras com identificação "Lentes duras". Ou seja, lê-se: {Sem lentes, Lentes gelatinosas, Lentes duras}. O mesmo ocorre para {3,5,4} e {12,0,0}.

Como todas as 12 amostras onde “Lágrimas” = “Reduzida” apresentam desfecho igual a “Sem lentes” – {12,0,0} –, pode-se então adicionar uma folha com esta classe à árvore, ficando esta da seguinte forma:

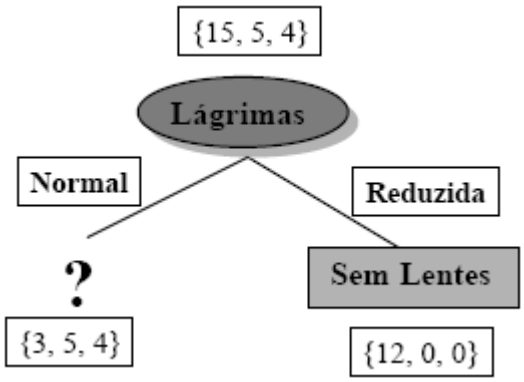


Figura D4 – Árvore de decisão – Lentes de Contato – Esquema 2

Deve-se agora estruturar o galho da árvore com “Lagrimas” = “Normal”. O próximo atributo a ser selecionado é aquele, dentre os 3 que restam, que apresenta o maior ganho de informação com relação ao desfecho, devendo este ganho ser calculado em relação ao subconjunto restante, ou seja, o subconjunto com “Lágrimas” = “Normal”. A tabela com este subconjunto é mostrada a seguir:

Tabela D3 – Subconjunto de Treinamento para “Lágrimas” = “Normal”

Variáveis				**Desfecho**
Idade	**Prescrição**	**Astigmático**	**Lágrimas**	**Lentes**
Jovem	Míope	Não	**Normal**	Lentes gelatinosas
Jovem	Hipermíope	Não	**Normal**	Lentes gelatinosas
Média	Míope	Não	**Normal**	Lentes gelatinosas
Média	Hipermíope	Não	**Normal**	Lentes gelatinosas
Sênior	Míope	Não	**Normal**	Sem lentes
Sênior	Hipermíope	Não	**Normal**	Lentes gelatinosas
Jovem	Míope	Sim	**Normal**	Lentes duras
Jovem	Hipermíope	Sim	**Normal**	Lentes duras
Média	Míope	Sim	**Normal**	Lentes duras
Média	Hipermíope	Sim	**Normal**	Sem lentes
Sênior	Míope	Sim	**Normal**	Lentes duras
Sênior	Hipermíope	Sim	**Normal**	Sem lentes

O ganho de informação de cada variável com relação ao subconjunto mencionado é dado a seguir:

Ganho (Lentes,Idade) = 0,3879185

Ganho (Lentes,Prescrição) = 0,1787706

Ganho (Lentes,Astigmático) = 1,0043723

A variável “Astigmático” apresentou agora o maior ganho de informação com relação ao desfecho “Lentes”. Adiciona-se então este nodo à árvore:

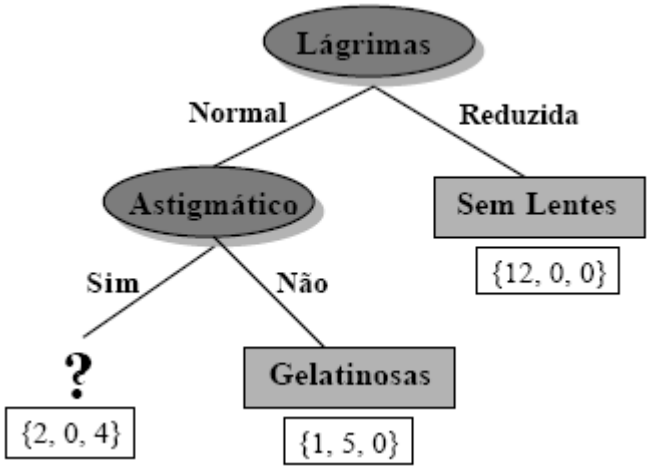


Figura D5 – Árvore de decisão – Lentes de Contato – Esquema 3

Como a maioria das amostras do galho “Astigmático” = “Não” (5 em 6 amostras) pertence à classe “Lentes gelatinosas”, esta folha foi então adicionada a este galho. Para o galho “Astigmático” = “Sim”, utiliza-se o seguinte subconjunto:

Tabela D4 – Subconjunto de Treinamento para “Lágrimas” = “Normal” e “Astigmático” = “Sim”

Variáveis				**Desfecho**
Idade	**Prescrição**	**Astigmático**	**Lágrimas**	**Lentes**
Jovem	Hipermíope	**Sim**	**Normal**	Lentes duras
Média	Hipermíope	**Sim**	**Normal**	Sem lentes
Sênior	Hipermíope	**Sim**	**Normal**	Sem lentes
Jovem	Míope	**Sim**	**Normal**	Lentes duras
Média	Míope	**Sim**	**Normal**	Lentes duras
Sênior	Míope	**Sim**	**Normal**	Lentes duras

Da mesma forma, avalia-se o ganho de informação de cada uma das 2 variáveis restantes com relação ao desfecho sob o subconjunto mencionado, onde se conclui que a variável “Prescrição” fornece um maior ganho de informação com relação ao desfecho do que a variável “Idade”. Tem-se agora o seguinte esquema:

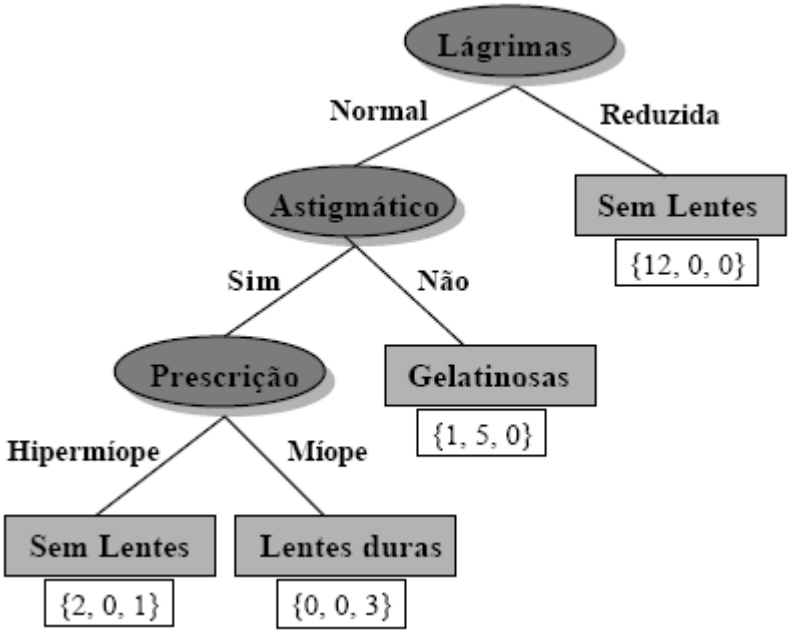


Figura D6 – Árvore de decisão – Lentes de Contato – Esquema Final

Através das análises realizadas, verificou-se que a variável "Idade" não é necessária na classificação, visto que as demais variáveis promovem uma boa divisão das classes, sendo as mais relevantes para a determinação do desfecho. Com a árvore de decisão montada, a derivação de regras é dada da seguinte forma:

1) Se ("Lágrimas" = "Reduzida") ou ("Lágrimas" = "Normal" e "Astigmático" = "Sim" e "Prescrição" = "Hipermíope"), então ("Lentes" = "Sem lentes");

2) Se ("Lágrimas" = "Normal" e "Astigmático" = "Não"), então ("Lentes" = "Lentes gelatinosas");

3) Se ("Lágrimas" = "Normal" e "Astigmático" = "Sim" e "Prescrição" = "Míope"), então ("Lentes" = "Lentes duras").

Pode-se fazer uma avaliação da árvore de decisão através da taxa de acerto, dada pela divisão do número de acertos pelo número total de amostras, ou através da taxa de erro, dividindo-se o número de erros pelo número total de amostras. Para o conjunto de treinamento do exemplo em questão, temos:

Taxa de Acerto = 22 / 24 ➔ Taxa de Acerto = 91,67 %

Taxa de Erro = 02 / 24 ➔ Taxa de Erro = 8,33 %

Com a árvore de decisão montada, pode-se aplicar ao conjunto de teste, com amostras que não foram utilizadas no treinamento, as regras de derivação criadas, podendo-se também avaliar a árvore com este conjunto de teste utilizando-se as taxas de acerto/erro, conforme foi calculado para o conjunto de treinamento.