

UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

MAGNUS ALENCAR DA CRUZ

AVALIAÇÃO DE REDES NEURAIIS COMPETITIVAS
EM TAREFAS DE QUANTIZAÇÃO VETORIAL:
UM ESTUDO COMPARATIVO

FORTALEZA
2007

MAGNUS ALENCAR DA CRUZ

**AVALIAÇÃO DE REDES NEURAIS COMPETITIVAS
EM TAREFAS DE QUANTIZAÇÃO VETORIAL:
UM ESTUDO COMPARATIVO**

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Engenharia de Teleinformática, da Universidade Federal do Ceará, como parte dos requisitos exigidos para obtenção do grau de Mestre em Engenharia de Teleinformática.

Área de concentração: Sinais e Sistemas

Orientador: Prof. Dr. Guilherme de Alencar Barreto

FORTALEZA

2007

Dedico este trabalho a minha esposa

Sabrina Mota

e aos meus pais

José Maria e Maria Yone

pelo constante apoio, incentivo e admiração.

Agradecimentos

A Jesus Cristo, acima de tudo.

Ao meu orientador, Prof. Guilherme de Alencar Barreto, a quem sou grato pela orientação, paciência e confiança depositada.

Aos membros da banca examinadora, pelas valiosas sugestões na defesa.

Aos meus amigos do pequeno grupo, pela ajuda em todas as horas.

Aos colegas de laboratório, por estarem sempre prontos a ajudar, proporcionando excelente ambiente de trabalho.

Ao Prof. João César Moura Mota, pelo apoio durante esta jornada.

Aos professores e funcionários do Departamento de Engenharia de Teleinformática que de forma direta ou indireta participaram do desenvolvimento deste trabalho.

Ao Atlântico e a DATAPREV, empresas que me deram suporte financeiro e de tempo.

Em especial à Sabrina Mota, minha esposa, pelo amor, carinho, incentivo, admiração e apoio incondicional.

Resumo

Esta dissertação tem como principal meta realizar um estudo comparativo do desempenho de algoritmos de redes neurais competitivas não-supervisionadas em problemas de quantização vetorial (QV) e aplicações correlatas, tais como análise de agrupamentos (*clustering*) e compressão de imagens. A motivação para tanto parte da percepção de que há uma relativa escassez de estudos comparativos sistemáticos entre algoritmos neurais e não-neurais de análise de agrupamentos na literatura especializada. Um total de sete algoritmos são avaliados, a saber: algoritmo K -médias e as redes WTA, FSCL, SOM, Neural-Gas, FuzzyCL e RPCL.

De particular interesse é a seleção do número ótimo de neurônios. Não há um método que funcione para todas as situações, restando portanto avaliar a influência que cada tipo de métrica exerce sobre algoritmo em estudo. Por exemplo, os algoritmos de QV supracitados são bastante usados em tarefas de *clustering*. Neste tipo de aplicação, a validação dos agrupamentos é feita com base em índices que quantificam os graus de compacidade e separabilidade dos agrupamentos encontrados, tais como Índice Dunn e Índice Davies-Bouldin (DB). Já em tarefas de compressão de imagens, determinado algoritmo de QV é avaliado em função da qualidade da informação reconstruída, daí as métricas mais usadas serem o erro quadrático médio de quantização (EQMQ) ou a relação sinal-ruído de pico (PSNR). Empiricamente verificou-se que, enquanto o índice DB favorecem arquiteturas com poucos protótipos e o Dunn com muitos, as métricas EQMQ e PSNR sempre favorecem números ainda maiores.

Nenhuma das métricas supracitadas leva em consideração o número de parâmetros do modelo. Em função disso, esta dissertação propõe o uso do critério de informação de Akaike (AIC) e o critério do comprimento descritivo mínimo (MDL) de Rissanen para selecionar o número ótimo de protótipos. Este tipo de métrica mostra-se útil na busca do número de protótipos que satisfaça simultaneamente critérios opostos, ou seja, critérios que buscam o menor erro de reconstrução a todo custo (MSE e PSNR) e critérios que buscam clusters mais compactos e coesos (Índices Dunn e DB). Como consequência, o número de protótipos obtidos pelas métricas AIC e MDL é geralmente um valor intermediário, i.e. nem tão baixo quanto o sugerido pelos índices Dunn e DB, nem tão altos quanto o sugerido pelas métricas MSE e PSNR.

Outra conclusão importante é que não necessariamente os algoritmos mais sofisticados do ponto de vista da modelagem, tais como as redes SOM e Neural-Gas, são os que apresentam melhores desempenhos em tarefas de *clustering* e quantização vetorial. Os algoritmos FSCL e FuzzyCL são os que apresentam melhores resultados em tarefas de quantização vetorial, com a rede FSCL apresentando melhor relação custo-benefício, em função do seu menor custo computacional. Para finalizar, vale ressaltar que qualquer que seja o algoritmo escolhido, se o mesmo tiver seus parâmetros devidamente ajustados e seus desempenhos devidamente avaliados, as diferenças de performance entre os mesmos são desprezíveis, ficando como critério de desempate o custo computacional.

Palavras-chave: redes neurais competitivas, aprendizado não-supervisionado, validação de agrupamentos, quantização vetorial, robustez ao ruído.

Abstract

The main goal of this master thesis was to carry out a comparative study of the performance of algorithms of unsupervised competitive neural networks in problems of vector quantization (VQ) tasks and related applications, such as cluster analysis and image compression. This study is mainly motivated by the relative scarcity of systematic comparisons between neural and non-neural algorithms for VQ in specialized literature. A total of seven algorithms are evaluated, namely: K-means, WTA, FSCL, SOM, Neural-Gas, FuzzyCL and RPCL.

Of particular interest is the problem of selecting an adequate number of neurons given a particular vector quantization problem. Since there is no widespread method that works satisfactorily for all applications, the remaining alternative is to evaluate the influence that each type of evaluation metric has on a specific algorithm. For example, the aforementioned vector quantization algorithms are widely used in clustering-related tasks. For this type of application, cluster validation is based on indexes that quantify the degrees of compactness and separability among clusters, such as the Dunn Index and the Davies-Bouldin (DB) Index. In image compression tasks, however, a given vector quantization algorithm is evaluated in terms of the quality of the reconstructed information, so that the most used evaluation metrics are the mean squared quantization error (MSQE) and the peak signal-to-noise ratio (PSNR). This work verifies empirically that, while the indices Dunn and DB or favors architectures with many prototypes (Dunn) or with few prototypes (DB), metrics MSE and PSNR always favor architectures with well bigger amounts.

None of the evaluation metrics cited previously takes into account the number of parameters of the model. Thus, this thesis evaluates the feasibility of the use of the Akaike's information criterion (AIC) and Rissanen's minimum description length (MDL) criterion to select the optimal number of prototypes. This type of evaluation metric indeed reveals itself useful in the search of the number of prototypes that simultaneously satisfies conflicting criteria, i.e. those favoring more compact and cohesive clusters (Dunn and DB indices) versus those searching for very low reconstruction errors (MSE and PSNR). Thus, the number of prototypes suggested by AIC and MDL is generally an intermediate value, i.e. nor so low as much suggested for the indexes Dunn and DB, nor so high as much suggested one for metric MSE and PSNR.

Another important conclusion is that sophisticated models, such as the SOM and Neural-Gas networks, not necessarily have the best performances in clustering and VQ tasks. For example, the algorithms FSCL and FuzzyCL present better results in terms of the the quality of the reconstructed information, with the FSCL presenting better cost-benefit ratio due to its lower computational cost. As a final remark, it is worth emphasizing that if a given algorithm has its parameters suitably tuned and its performance fairly evaluated, the differences in performance compared to others prototype-based algorithms is minimum, with the coputational cost being used to break ties.

Key-words: competitive neural networks, unsupervised learning, cluster validation, vector quantization, robustness to noise.

Lista de Figuras

1.1 Sistema de codificação baseado em quantização vetorial.	6
1.2 Projeção das redes competitivas nas diversas áreas.	7
2.1 Projeção implementada pela rede SOM	22
2.2 Ilustração do vetor erro de quantização e_q . Os círculos abertos ('o') simbolizam os vetores de dados, enquanto os círculos fechados ('•') simbolizam os vetores de pesos (centróides). Figura extraída de (FROTA, 2005) sob permissão do autor.	27
3.1 Metodologia de aplicação de um critério interno.	33
3.2 Metodologia de Aplicação de Critérios Externos	34
3.3 Metodologia de Aplicação de Critérios Relativos.	36
4.1 Conjunto de dados experimentais.	45
4.2 Posicionamento de $K = 2$ protótipos da rede WTA após 50 épocas.	47
4.3 Posicionamento de $K = 3$ protótipos da rede WTA após 50 épocas.	48
4.4 Posicionamento de $K = 4$ protótipos da rede WTA após 50 épocas.	48
4.5 Posicionamento de $K = 5$ protótipos da rede WTA após 50 épocas.	49
4.6 Posicionamento de $K = 4$ protótipos da rede FSCL após 50 épocas ($z = 0, 1$).	50
4.7 Posicionamento de $K = 4$ protótipos da rede FSCL após 50 épocas ($z = 3$).	51
4.8 Posicionamento de $K = 4$ protótipos da rede RPCL ($\gamma = 0, 05$).	52
4.9 Posicionamento de $K = 4$ protótipos da rede RPCL ($\gamma = 0, 1$).	53
4.10 Erro médio de quantização da rede RPCL em função de K para $\gamma = 0, 05$ e $0, 1$	53
4.11 Posicionamento de $K = 2, 3$ e 4 protótipos da rede SOM-1D.	54
4.12 Posicionamento de $K = 5, 9, 16$ e 25 protótipos da rede SOM-1D.	55
4.13 Posicionamento de $K = 4, 9, 16$ e 25 protótipos da rede SOM-2D.	56

4.14	Os gráficos dos índices Dunn e DB para $K = 4, 9, 16$ e 25 protótipos das redes SOM-1D e SOM-2D.	57
4.15	Curvas do erro médio de quantização da rede SOM para topogias 1D e 2D com $K = 4$ protótipos.	58
4.16	Posicionamento de $K = 2$ protótipos do algoritmo K -médias após treinamento.	59
4.17	Posicionamento de $K = 3$ protótipos do algoritmo K -médias após treinamento.	59
4.18	Posicionamento de $K = 4$ protótipos do algoritmo K -médias após treinamento.	60
4.19	Posicionamento de $K = 5$ protótipos do algoritmo K -médias após treinamento.	61
4.20	Posicionamento de $K = 4$ protótipos da rede FuzzyCL após treinamento ($z = 1, 1$).	62
4.21	Posicionamento de $K = 4$ protótipos da rede FuzzyCL após treinamento ($z = 2$).	62
4.22	Posicionamento de $K = 2$ protótipos da rede Neural-Gas após treinamento.	63
4.23	Posicionamento de $K = 3$ protótipos da rede Neural-Gas após treinamento.	64
4.24	Posicionamento de $K = 4$ protótipos da rede Neural-Gas após treinamento.	64
4.25	Posicionamento de $K = 5$ protótipos da rede Neural-Gas após treinamento.	65
4.26	Gráfico do Erro de Quantização em função do número de protótipos.	66
4.27	Índice Dunn por número de protótipos.	67
4.28	Índice Davies-Bouldin por número de protótipos.	67
4.29	Melhor posicionamento dos protótipos da rede SOM-1D.	68
4.30	Melhor posicionamento dos protótipos da rede WTA segundo índices Dunn e DB.	69
4.31	Melhor posicionamento dos protótipos da rede FSCL segundo índices Dunn e DB.	69
4.32	Melhor posicionamento dos protótipos da rede RPCL segundo índice Dunn.	70
4.33	Melhor posicionamento dos protótipos do K -médias segundo índices Dunn e DB.	70
4.34	Melhor posicionamento dos protótipos da rede FuzzyCL segundo índices Dunn e DB.	71
4.35	Melhor posicionamento dos protótipos da Neural-Gas segundo índice DB.	71
5.1	Figura ilustrativa mostrando a entrada dos dados (<i>pixels</i>) no processo de quantização vetorial usando as redes neurais competitivas.	75

5.2	Imagem original do macaco Mandrill em 256×256 pixels e 8 bits.	75
5.3	Gráfico do erro médio de quantização em função do número de protótipos. . . .	76
5.4	(a) Imagem do macaco Mandrill reconstruída pela rede RPCL com $K = 4$ protótipos. (b) Imagem do macaco Mandrill reconstruída pela rede RPCL com $K = 256$ protótipos.	77
5.5	Gráficos da métrica PSNR versus número de protótipos.	78
5.6	Gráfico do Índice Dunn versus número de protótipos.	79
5.7	(a) Gráfico da evolução do índice Dunn para a rede WTA. (b) Imagem do macaco Mandrill reconstruída pela rede WTA com $K = 256$ protótipos.	80
5.8	Gráfico do Índice Davies-Bouldin versus número de protótipos.	81
5.9	(a) Gráfico da evolução do índice DB para o algoritmo K -médias. (b) Imagem do macaco Mandrill reconstruída pelo algoritmo K -médias com $K = 4$ protótipos. (c) Imagem do macaco Mandrill reconstruída pelo algoritmo K -médias com $K = 256$ protótipos.	82
5.10	(a) Imagem do macaco Mandrill reconstruída pela rede FSCL com $K = 4$ protótipos. (b) Imagem do macaco Mandrill reconstruída pela rede FSCL com $K = 256$ protótipos.	84
5.11	Gráfico do Índice AIC por número de protótipos.	85
5.12	(a) Imagem do macaco Mandrill reconstruída pela rede FSCL com $K = 16$ protótipos. (b) Imagem do macaco Mandrill reconstruída pela rede FuzzyCL com $K = 16$ protótipos.	86
5.13	Curvas do índice MDL versus número de protótipos.	87
5.14	(a) Imagem do macaco Mandrill reconstruída pela rede FuzzyCL com $K = 8$ protótipos. (b) Imagem do macaco Mandrill reconstruída pela rede SOM com $K = 8$ protótipos.	88
5.15	Curvas do índice FPE versus número de protótipos para as redes FSCL e FuzzyCL.	89
5.16	Imagens reconstruídas pelas redes FSCL com $K = 256$ protótipos e FuzzyCL com $K = 64$ protótipos segundo o critério FPE.	90
5.17	Curvas do índice BIC versus número de protótipos (redes FSCL e FuzzyCL).	90

5.18	Imagens reconstruídas pelas redes FSCL com $K = 32$ protótipos e FuzzyCL com $K = 16$ protótipos segundo o critério BIC.	91
5.19	Curvas do índice MDL versus número de protótipos para as redes FSCL e FuzzyCL.	92
5.20	Imagens reconstruídas pelas redes FSCL com $K = 64$ protótipos e FuzzyCL com $K = 32$ protótipos segundo o critério MDL.	93
5.21	Imagem <i>Peppers</i> sem ruído usada no treinamento com resolução 256×256 pixels.	94
5.22	(a) Imagem <i>Peppers</i> com ruído gaussiano ($\sigma = 5$). (b) Imagem <i>Peppers</i> com ruído gaussiano ($\sigma = 15$). (c) Reconstrução da imagem ruidosa ($\sigma = 5$) pela rede SOM com $K = 128$ protótipos. (d) Reconstrução da imagem ruidosa ($\sigma = 15$) pela rede SOM com $K = 128$ protótipos.	95
5.23	(a) Imagem Lena sem ruído usada no treinamento. (b) Imagem Lena com ruído sal-e-pimenta usada no teste.	96
5.24	(a) Imagem reconstruída pela rede SOM com $K = 32$ protótipos. (b) Imagem reconstruída pela rede SOM com $K = 64$ protótipos. (c) Imagem reconstruída pela rede SOM com $K = 128$ protótipos. (d) Imagem reconstruída pelo algoritmo K -médias $K = 32$ protótipos. (e) Imagem reconstruída pelo algoritmo K -médias $K = 64$ protótipos. (f) Imagem reconstruída pelo algoritmo K -médias $K = 128$ protótipos.	97
5.25	Curva do índice PSNR versus a probabilidade de comutação de <i>bit</i>	98
5.26	(a) Imagem reconstruída pela rede WTA para $P = 0, 1$. (b) Imagem reconstruída pela rede WTA para $P = 0, 2$. (c) Imagem reconstruída pela rede WTA para $P = 0, 3$. (d) Imagem reconstruída pela rede WTA para $P = 0, 4$. (e) Imagem reconstruída pela rede WTA para $P = 0, 5$	99
A.1	Erro de Quantização para $K = 2$ protótipos.	108
A.2	Erro de quantização com $K = 3$ protótipos.	109
A.3	Erro de quantização para $K = 4$ protótipos.	109
A.4	Erro de quantização para $K = 5$ protótipos.	110
A.5	Erro de quantização para $K = 6$ protótipos.	110

A.6	Erro de quantização para $K = 7$ protótipos.	111
A.7	Erro de quantização para $K = 8$ protótipos.	111
A.8	Erro de quantização para $K = 9$ protótipos.	112
A.9	Erro de quantização para $K = 10$ protótipos.	112
B.1	Aplicação Java de Redes Neurais com <i>Eclipse IDE de background</i> (fundo). . .	114
C.1	Representação de um Canal Binário Simétrico.	118

Lista de Tabelas

1.1	Comparativo em números de trabalhos entre redes competitivas e não-competitivas no ICANN'07.	8
5.1	Valores do Erro de Quantização para cada algoritmo por número de protótipos.	83
5.2	Valores do Índice Dunn para cada algoritmo por número de protótipos.	83
5.3	Valores do Índice Davies-Bouldin para cada algoritmo por número de protótipos.	84

Lista de Símbolos

$\{\mathbf{x}^\mu\}_{\mu=1}^N$	conjunto de vetore
$\{\mathbf{w}_i\}_{i=1}^K$	conjunto de protótipos ou vetores-código
\mathbf{x}^μ	vetor entrada pertencente à $\{\mathbf{x}^\mu\}_{\mu=1}^N$
\mathbf{w}_i	vetor-código pertencente à $\{\mathbf{w}_i\}_{i=1}^K$
Q	mapeamento de $\{\mathbf{x}^\mu\}_{\mu=1}^N$ em $\{\mathbf{w}_i\}_{i=1}^K$
K	número de grupos ou <i>clusters</i>
N	número de entradas de dados
S_i	um grupo de N_i exemplos para os quais o protótipo \mathbf{w}_i é o mais próximo
N_i	número de exemplos selecionados para o conjunto S_i
i	i -ésimo protótipo de $\{\mathbf{w}_i\}_{i=1}^K$
μ	μ -ésimo vetor de $\{\mathbf{x}^\mu\}_{\mu=1}^N$
$d(\mathbf{x}^\mu, \mathbf{w}_i)$	distorção entre o \mathbf{x}^μ e \mathbf{w}_i
\mathbf{b}_{i^*}	regra ótima para codificação binária
i^*	índice do protótipo mais próximo (menor distorção) de \mathbf{x}^μ
\mathbf{w}_{i^*}	o vetor-código de menor distorção
$C(\mathbf{x}^\mu)$	regra de codificação
$D(\mathbf{b}_{i^*})$	regra de decodificação
R	taxa de codificação do quantizador vetorial
ε	limiar de distorção maior que zero
W_0	conjunto inicial para $\{\mathbf{w}_i\}_{i=1}^K$
W_t	conjunto $\{\mathbf{w}_i\}_{i=1}^K$ na t -ésima iteração
t	iteração atual
D	função de distorção ou função custo
$D(n)$	valor do erro de quantização na n -ésima rodada de ajuste dos protótipos
$\ \cdot\ $	denota a norma euclidiana
$ u $	denota o valor absoluto de u
V_i	conjunto de dados em K regiões de Voronoi, $i = 1, \dots, K$
$K \ll N$	K bem menor que N
\mathbb{R}^n	conjunto dos números reais no espaço de n dimensões
$\mathbf{x}(t)$	um vetor de entrada da rede na iteração t

n	indicativo do número de dimensões do espaço amostral
$i^*(t)$	índice do neurônio vencedor na rede
$\mathbf{w}_i(t)$	o vetor de pesos associado ao neurônio i
$\mathbf{w}_{i^*}(t)$	neurônio vencedor
η	passo de aprendizagem
\mathbf{w}_i^o	valor ótimo do vetor de pesos \mathbf{w}_i
E	representação de uma expressão
η_0	valor inicial de η
η_T	valor final de η
T	número máximo de iterações
$f_i(t)$	elemento ponderador da distância euclidiana na rede FSCL
z	representação de uma constante
$i^r(t)$	índice do rival do neurônio vencedor
γ	controlador da penalização para a rede RPCL
$h(i^*, i; t)$	função de vizinhança da rede SOM
$\vartheta(t)$	raio de influência da função de vizinhança
$\mathbf{r}_i(t)$	posições do neurônio i no arranjo geométrico da rede
$\mathbf{r}_{i^*}(t)$	posições do neurônio i^* no arranjo geométrico da rede
Φ	uma projeção não-linear do espaço de entrada contínuo χ
χ	espaço dos dados, i.e., $\chi \subset \mathbb{R}^n$
\mathcal{A}	espaço de saída discreto da projeção Φ
$h_\lambda(k, t)$	função de vizinhança da rede Neural-Gas
k	posição do neurônio na lista ordenada
$\lambda(t)$	fator ponderador que decai com o tempo da função $h_\lambda(k, t)$
$\mathbf{u}_{\mathbf{w}_i}^\mu$	probabilidade de um vetor \mathbf{x}^μ ser associado ao <i>cluster</i> \mathbf{w}_i
$\mathbf{e}_q(t)$	vetor de erros de quantização
$\mathbf{e}_q(\mathbf{x}(t))$	erro de quantização associado ao vetor $\mathbf{x}(t)$
H_0	hipótese nula
Γ	estatística Hubert
$\hat{\Gamma}$	estatística Γ normalizada
M	matriz de proximidade do conjunto de dados
X	uma matriz $n \times n$
(i, j)	distância entre os pontos (vs_i, vs_j) , i.e., cada célula da matriz X
(vs_i, vs_j)	pontos representativos dos <i>clusters</i> dos padrões x_i e x_j ,
$D(K)$	função do índice Dunn

$\delta(S_i, S_j)$	uma função de dissimilaridade entre S_i e S_j
$\Delta(S_i)$	distância intra- <i>cluster</i> de S_i
S_i	uma representação para um <i>cluster</i>
$R_{i,j}$	uma medida de similaridade entre S_i e S_j
e_{S_j}	erro médio para o <i>cluster</i> S_j
$d(S_i, S_j)$	distância euclideana entre os centros dos <i>clusters</i> S_i e S_j
p	dimensão da quantização vetorial

Lista de Siglas

CL	<i>Competitive Learning</i>
WTA	<i>Winner-take-all</i>
FSCL	<i>Frequency-Sensitive Competitive Learning</i>
SOM	<i>Self-Organizing Map</i>
NGA	Neural-Gas
FuzzyCL	<i>Fuzzy Competitive Learning</i>
RPCL	<i>Rival Penalized Competitive Learning</i>
DB	Davies-Bouldin
PSNR	<i>Peak signal-to-noise ratio</i>
AIC	<i>Akaike's Information Criterion</i>
MDL	<i>Minimum description length</i>
FPE	<i>Final Prediction Error</i>
BIC	<i>Bayesian Information Criterion</i>
KDD	<i>Knowledge Discovery in Database</i>
QV	Quantização Vetorial
LBG	Linde Buzo Gray
RBF	<i>Radial Basis Function</i>
MLP	<i>MultiLayer Perceptron</i>
AR	Auto-Regressivos
RSS	<i>residual sum of squares</i>
MSE	<i>Mean-Squared Error</i>
EQ	Erro de Quantização
MSQE	<i>mean squared quantization error</i>
EQM	erro quadrático médio de quantização
ART	<i>Adaptive Resonance Theory</i>
ICANN'07	<i>Internatinal Conference on Artificial Neural Networks 2007</i>

Sumário

Resumo	iii
Abstract	iv
Lista de Figuras	v
Lista de Tabelas	x
Lista de Símbolos	xi
Lista de Siglas	xiv
<hr/>	
1 Introdução	1
1.1 Redes Neurais Competitivas Não-Supervisionadas	1
1.2 Análise de Agrupamentos	3
1.3 Quantização Vetorial	4
1.4 Motivações da Dissertação	8
1.5 Objetivos da Dissertação	10
1.5.1 Objetivo Geral	10
1.5.2 Objetivos Específicos	11
1.6 Organização do Restante do Documento	11
2 Redes Neurais Competitivas	13
2.1 Introdução	13

2.2	Algoritmo <i>K</i> -Médias <i>Batch</i>	14
2.3	Algoritmo <i>K</i> -Médias Seqüencial	15
2.4	Redes Neurais Não-Supervisionadas Competitivas	16
2.5	Rede <i>Winner-Take-All</i>	17
2.6	Rede <i>Frequency-Sensitive Competitive Learning</i>	19
2.7	Rede <i>Rival Penalized Competitive Learning</i>	19
2.8	Rede Auto-Organizável de Kohonen	20
2.9	Rede Neural-Gas	23
2.10	Agrupamentos Fuzzy	24
2.10.1	Algoritmo <i>K</i> -Médias <i>Batch</i> Nebuloso	25
2.10.2	Rede <i>Fuzzy Competitive Learning</i>	26
2.11	Sobre Aplicações de Redes Neurais Competitivas	26
2.12	Conclusão	28
3	Seleção de Protótipos em Análise de Agrupamentos e Quantização Vetorial	29
3.1	Objetivos da Validação de Agrupamentos	29
3.2	Critérios de Validação de Agrupamentos	31
3.2.1	Critérios Internos e Externos	32
3.2.2	Simulação de Monte Carlo	34
3.2.3	<i>Bootstrapping</i>	35
3.2.4	Critérios Relativos	35
3.3	Índices Baseados em Critérios Relativos	36
3.3.1	Estatística Γ de Hubert Modificada	37
3.3.2	Família de índices Dunn	37
3.3.3	Índice Davies-Bouldin (DB)	38
3.3.4	Silhuetas	39
3.4	Critérios de Informação	40

3.4.1	Critério do Erro Final de Predição	41
3.4.2	Critério de Informação de Akaike	41
3.4.3	Critério de Informação Bayesiana	42
3.4.4	Critério do Comprimento Mínimo de Descrição	42
3.4.5	Crítérios de Informação em Quantização Vetorial	42
3.5	Conclusão	43
4	Resultados - Análise de Agrupamentos	44
4.1	Introdução	44
4.2	Metodologia de Simulação	44
4.2.1	Resultados - Rede WTA	47
4.2.2	Resultados - Rede FSCL	49
4.2.3	Resultados - Rede RPCL	51
4.2.4	Resultados - Rede SOM	54
4.2.5	Resultados - Algoritmo K -médias	58
4.2.6	Resultados - Rede FuzzyCL	61
4.2.7	Resultados - Rede Neural-Gas	63
4.3	Validação de Agrupamentos via Erro de Quantização	66
4.4	Validação via Índices Dunn e Davies-Bouldin	66
4.5	Melhores Agrupamentos segundo os Índices Dunn e DB	68
4.6	Conclusão	72
5	Resultados - Quantização Vetorial	73
5.1	Introdução	73
5.2	Metodologia de Simulação	73
5.3	Experimentos de Seleção Inicial de Modelos	75
5.3.1	Seleção via Erro de Quantização	76

5.3.2	Seleção via Razão Sinal-Ruído de Pico	77
5.3.3	Seleção via Índices de Validação de Agrupamentos	78
5.3.4	Seleção via Critérios de Informação	85
5.4	Critérios de Informação em Quantização Vetorial: Uma Nova Abordagem . . .	88
5.5	Testes de Robustez ao Ruído	93
5.5.1	Ruído Gaussiano na Imagem <i>Peppers</i>	94
5.5.2	Ruído Sal-e-Pimenta na Imagem <i>Lena</i>	96
5.5.3	Ruído no Canal na Imagem <i>Mandrill</i>	98
6	Conclusões e Perspectivas	100
	Referências Bibliográficas	103
	Apêndice A – Avaliação do Erro de Quantização Durante o Treinamento	108
	Appendix B – Aplicação Java	114
	Apêndice C – Canais de Comunicação	116
C.1	Modelos de canal	117
C.2	Canal Binário Simétrico	118

1 Introdução

Este capítulo apresenta as principais razões que levaram ao desenvolvimento desta dissertação, assim como seus objetivos. Ao final deste capítulo faz-se também uma breve descrição sobre os tópicos associados aos demais capítulos do documento.

1.1 Redes Neurais Competitivas Não-Supervisionadas

Redes neurais competitivas não-supervisionadas constituem uma classe de redes neurais artificiais usada para construir uma representação estatística compacta de um conjunto de dados de entrada não-rotulados. Os primeiros modelos de redes neurais competitivas surgiram a partir de estudos que investigavam hipóteses sobre a organização dos neurônios em determinadas áreas do córtex cerebral - por exemplo, o córtex visual - a partir de estímulos sensoriais e como esta organização neuronal está relacionada ao aprendizado (GROSSBERG, 1976b; RUMELHART; ZIPSER, 1985; GROSSBERG, 1987).

De acordo com Muszkat (2006), uma dessas hipóteses está fundamentada no paradigma cognitivo que entende o cérebro como um ecossistema em que os próprios neurônios vivem em situação de competição e organização ditadas pelos estímulos sensoriais provenientes do ambiente em que o portador do cérebro está inserido. Segundo o próprio Muszkat [p. 42]:

Nesse novo paradigma, a palavra de ordem é plasticidade: os 35 mil genes associados ao cérebro confrontam-se com os trilhões de sinapses sujeitas à modulação e à mediação ambientais. Há competição por substrato entre os vários neurônios, e as células neuronais se diferenciam dependendo da necessidade e do local em que se encontram, com suficiente diversidade e fluidez para fazer emergir uma mente autônoma e auto-reflexiva da própria estrutura cerebral e de suas múltiplas conexões.

Pode-se afirmar, portanto, que a competição entre neurônios é um dos princípios fundamentais

da auto-organização cerebral, esta definida genericamente como o processo pelo qual padrões de conectividade e de atividade neuronal altamente estruturados e ordenados emergem a partir de um estado inicial de aparente não-estruturação. Esta habilidade do cérebro de se auto-organizar permite, por exemplo, associar a modificação da estrutura física do cérebro por meio criação e eliminação de conexões sinápticas entre neurônios com a capacidade de aprendizado. Permite também associar a atividade neuronal com a mente. Portanto, de acordo com a visão dominante em neurociência, conexões sinápticas são o repositório do conhecimento no cérebro, a auto-organização da atividade neural é o mecanismo pelo qual conhecimento elementar (na forma de conexões) é combinado para dar forma ao que costumou-se chamar de pensamento, ou simplesmente, a mente (von der Malsburg, 2003).

Com base na metáfora do cérebro como um sistema auto-organizável, modelos de redes neurais competitivas tentam reproduzir dois níveis de auto-organização: (i) a formação de padrões de conexões e (ii) a formação de padrões de atividade. Para implementá-los, há basicamente duas abordagens. A primeira constrói modelos no domínio do tempo contínuo, em que os modelos são formulados por meio de equações diferenciais (KELSO, 1995), sendo a teoria da sinérgica (HAKEN, 2004) e a teoria do campo de redes neurais auto-organizáveis (AMARI, 1983) duas das principais representantes desta linha de pesquisa. A segunda abordagem formula modelos no domínio do tempo discreto, tendo experimentado grande avanço nos últimos 20 anos em função da popularização do computador digital. Os principais representantes desta linha de pesquisa são os modelos competitivos propostos por Kohonen (KOHONEN, 1982, 1990). Há ainda modelos que foram inicialmente propostos em tempo contínuo, mas que tiveram versões posteriormente adaptadas para tempo discreto, tais como os modelos da família ART (*Adaptive Resonance Theory*) (CARPENTER; GROSSBERG, 2003).

Historicamente, o trabalho de Stark et al. (1962) introduziu o primeiro algoritmo de aprendizagem competitiva na literatura, muito embora possa-se argumentar que trabalhos anteriores de Frank Rosenblatt (ROSENBLATT, 1958, 1959, 1962) sobre “aprendizagem espontânea” mereçam esta honraria. Outros trabalhos pioneiros em aprendizagem competitiva e auto-organização foram produzidos nos anos 1970 por von der Malsburg (1976) e Grossberg (1976a). Modelos competitivos para visão foram propostos na mesma época por Fukushima (1975, 1980).

Qualquer que seja a abordagem adotada para construir uma rede neural competitiva, a idéia básica da competição entre neurônios é a seguinte: neurônios da camada de saída competem entre si pelo direito de responder, ou seja, de permanecerem ativos para um determinado estímulo de entrada. Ao final da competição, apenas um neurônio (ou um pequeno grupo de neurônios) estará ativo em resposta àquela informação de entrada. Em geral, ao longo do processo de ajuste

das conexões sinápticas, um neurônio atuará como um detector de características (*feature detector*) (RUMELHART; ZIPSER, 1985), ou seja, ele passará a indicar, por meio de seu disparo, a ocorrência de um padrão de entrada que possua um determinado conjunto de características próprias do grupo a que o padrão de entrada pertence. Isto é possível por que os neurônios de uma rede competitiva extraem propriedades estatísticas (médias) do conjunto de padrões de entrada.

Apesar da origem associada à modelagem de fenômenos estudados pela ciência cognitiva e pela neurociência, alguns pesquisadores começaram a notar que redes neurais competitivas guardavam forte semelhança com ferramentas computacionais e estatísticas comumente usada para análise de dados de uma maneira geral. Em especial, duas áreas de aplicação receberam contribuições na forma de algoritmos de redes neurais competitivas: análise de agrupamento (*clustering*) e quantização vetorial. Estas aplicações são detalhadas nas próximas seções.

1.2 Análise de Agrupamentos

A análise de agrupamentos (*clustering*) tem sua origem no campo da estatística multivariada (HAIR et al., 2005) e engloba uma grande número de técnicas qualitativas ou quantitativas de cuja a finalidade primária é agregar objetos com base nas características que eles possuem. Esta é uma tarefa de natureza intrinsecamente não-supervisionada, uma vez que o conjunto de dados de entrada é composto por vetores de dados não-rotulados, ou seja, vetores para os quais não existe uma classe associada. Nesse caso, os algoritmos empenham-se em descobrir padrões de regularidade estatística escondidos nos dados, padrões estes que não são de fácil determinação por parte do usuário devido à elevada dimensionalidade dos dados.

Análise de agrupamentos é uma área que há muito deixou de ser campo de estudo exclusivo da estatística multivariada, tendo recebido contribuições importantes da área de aprendizado de máquinas, tais como redes neurais artificiais. Daí existirem diversas abordagens para formação de agrupamentos, tais como: determinística, probabilística, baseada em otimização por métodos de gradiente, computação evolucionária, conjuntos nebulosos e hierárquica (XU; BRERETON, 2005; JAIN et al., 1999). Cada abordagem utiliza uma maneira diferente para a identificação e representação dos agrupamentos. Não é meta dessa dissertação fazer um apanhado de toda a gama de abordagens e algoritmos para formação e análise de agrupamentos, mas sim focar naqueles baseados em vetores-protótipos por possuírem equivalentes na área de redes neurais competitivas.

De modo mais formal, a análise de agrupamentos consiste na separação de uma popula-

ção de entidades (objetos ou indivíduos), representados numericamente por vetores de características (*feature vectors*), em determinados subgrupos ou categorias, a fim de se identificar e representar a estrutura organizacional subjacente a cada subgrupo.

Em outras palavras, algoritmos de *clustering* separam as entidades, agrupando-os com base nas características que esses possuem. Entidades pertencentes a um mesmo grupo (*cluster*) são mais similares entre si de acordo com alguma medida de similaridade pré-definida, enquanto que entidades pertencentes a grupos diferentes têm uma similaridade menor (WEBB, 2002; EVERITT, 1993).

Algoritmos de agrupamentos são usados tanto para inferir a maneira como os dados estão organizados e relacionados em cada grupo, quanto para prover vetores-protótipos, também chamados de vetores-referência ou centróides, que servem como elemento representativo dos agrupamentos obtidos. Assim, em resumo, cada agrupamento é composto por um subconjunto (amostra) da população total de dados e representado por um vetor-protótipo.

Redes neurais competitivas podem ser utilizadas na análise de agrupamentos. A idéia é fazer com que o algoritmo neural aprenda, de modo auto-organizado, a representar as características estatísticas de um conjunto de dados não-rotulados. Para uma rede com um número fixo de neurônios, isto significa dizer que cada um de seus vetores de conexões sinápticas passariam a desempenhar o papel de vetor-protótipo para um subconjunto (agrupamento) específico dos dados disponíveis. Uma vez construída esta representação dos dados, as saídas do neurônio indicariam a que agrupamento ele pertence. O número de subconjuntos é determinado pelo número de neurônios da rede.

Uma outra área de aplicação em que redes neurais competitivas têm sido utilizadas com sucesso é conhecida como quantização vetorial (*vector quantization*). Na realidade, há uma ligação estreita entre análise de agrupamentos e quantização vetorial, conforme será mostrado a seguir.

1.3 Quantização Vetorial

Quantização vetorial (QV) é um problema cuja a origem está na engenharia de telecomunicações, mais especificamente na transmissão codificada de informação através de canais de comunicação ruidosos e/ou com banda-passante limitada (GRAY, 1984). Os algoritmos de QV são bastante similares aos algoritmos de análise de agrupamentos. A maior diferença reside no tipo de vetor de entrada que é apresentado ao algoritmo.

Em análise de agrupamentos a entrada é representada por vetores de características cujas componentes refletem propriedades dos objetos a serem agrupados. Em quantização vetorial, por sua vez, os vetores são construídos a partir de amostras de sinais de voz atrasadas no tempo ou grupos de pixels extraídos de imagens digitais (NASRABADI; KING, 1988; RAMAMURTHI; GERSHO, 1986; GERSHO; CUPERMAN, 1983; ABUT et al., 1982). Isto significa que as componentes do vetor são bastante correlacionadas, tanto temporalmente, quanto espacialmente. Esta redundância ocupa banda-passante do canal desnecessariamente. A fim de eliminar redundâncias algoritmos de quantização vetorial são freqüentemente usados para fins de compressão de informação (GERSHO; GRAY, 1992).

A idéia básica de QV é reduzir a redundância no grupo original de dados (voz ou imagem) através da construção de um conjunto de vetores denominados vetores-código (*codevectors*). O conjunto de vetores-código é chamado de dicionário (*codebook*). Vetores-código são os equivalentes funcionais dos protótipos (centróides) dos algoritmos de agrupamento e dos vetores de pesos das redes competitivas.

Matematicamente, o problema de quantização vetorial pode ser definido como um mapeamento Q de um vetor de entrada \mathbf{x}^μ pertencente ao espaço euclidiano, $\{\mathbf{x}^\mu\}_{\mu=1}^N$, em um vetor \mathbf{w}_i pertencente a um subconjunto finito $\{\mathbf{w}_i\}_{i=1}^K$ (MADEIRO et al., 2004), ou seja,

$$Q : \{\mathbf{x}^\mu\}_{\mu=1}^N \rightarrow \{\mathbf{w}_i\}_{i=1}^K \quad (1.1)$$

Deste modo, a idéia central é formar K grupos de tal forma que as distâncias entre os N ($N \gg K$) elementos do conjunto de dados de entrada $\{\mathbf{x}^\mu\}_{\mu=1}^N$ e um dos protótipos $\{\mathbf{w}_i\}_{i=1}^K$ dos grupos seja mínima (FLECK, 2004). Seja S_i um grupo de N_i exemplos para os quais o vetor-referência \mathbf{w}_i é o mais próximo, segundo uma métrica de distância, ou seja

$$S_i = \{\mathbf{x}^\mu \mid d(\mathbf{x}^\mu, \mathbf{w}_i) < d(\mathbf{x}^\mu, \mathbf{w}_j), \forall j \neq i\}. \quad (1.2)$$

A busca de uma melhor solução para a determinação dos grupos é definida pela localização dos protótipos, o que é feito através de um processo iterativo que inclui a reavaliação dos protótipos $\{\mathbf{w}_i\}_{i=1}^K$ e o cálculo de um erro de reconstrução. Portanto, em quantização vetorial, os algoritmos são utilizados não para encontrar agrupamentos de dados propriamente ditos, mas sim para encontrar um conjunto de vetores-códigos que produzam o menor erro de reconstrução.

Segundo (MADEIRO et al., 2004), em um sistema de codificação de sinais baseado em quantização vetorial, conforme apresentado na Figura 1.1, um quantizador vetorial pode ser visto como a combinação de duas funções: um codificador de fonte e um decodificador de

fonte. Dado um vetor $\mathbf{x}^\mu \in \{\mathbf{x}^\mu\}_{\mu=1}^N$ da fonte a ser codificada, o codificador calcula a distância $d(\mathbf{x}^\mu, \mathbf{w}_i)$ entre o vetor de entrada (vetor a ser quantizado) e cada vetor-código $\mathbf{w}_i, i = 1, 2, \dots, K$ do conjunto $\{\mathbf{w}_i\}_{i=1}^K$.

A regra ótima para codificação é a regra do vizinho mais próximo, na qual uma representação binária do índice i^* , denotada por \mathbf{b}_{i^*} , é transmitida ao decodificador de fonte se o vetor-código \mathbf{w}_{i^*} corresponder à menor distância, isto é, se \mathbf{w}_{i^*} for o vetor-código que apresentar a maior similaridade com \mathbf{x}^μ dentre todos os vetores-código do dicionário. Em outras palavras, o codificador usa a regra de codificação $C(\mathbf{x}^\mu) = \mathbf{b}_{i^*}$ se $d(\mathbf{x}^\mu, \mathbf{w}_{i^*}) < d(\mathbf{x}^\mu, \mathbf{w}_j), \forall j \neq i^*$.

Ao receber a representação binária \mathbf{b}_{i^*} , o decodificador de fonte, que dispõe de uma cópia do conjunto $\{\mathbf{w}_i\}_{i=1}^K$, simplesmente procura pelo i^* -ésimo vetor-código (ou vetor-protótipo) e produz o vetor \mathbf{w}_{i^*} como a reprodução (versão quantizada) de \mathbf{x}^μ . Em outras palavras, é utilizada a seguinte regra de decodificação: $D(\mathbf{b}_{i^*}) = \mathbf{w}_{i^*}$.

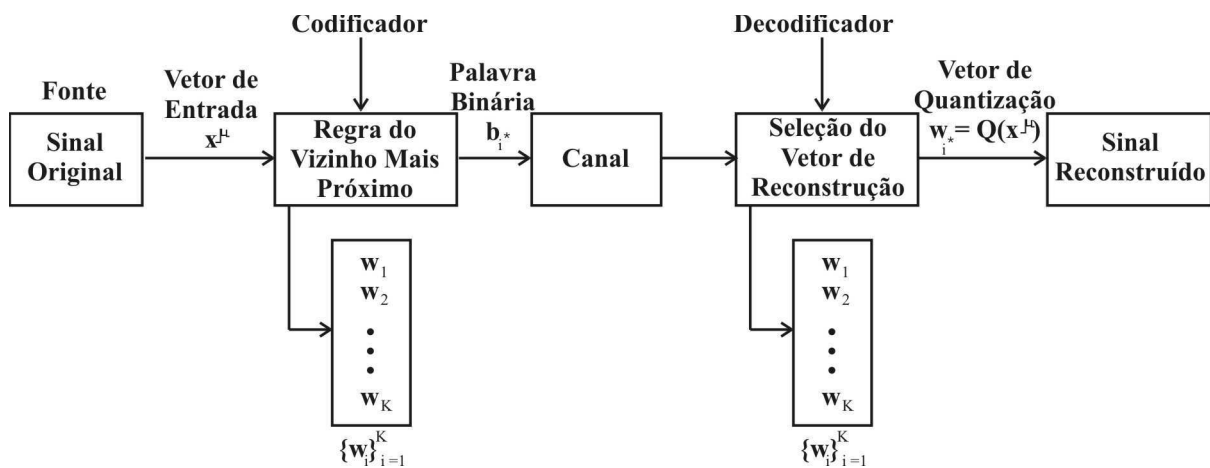


Figura 1.1: Sistema de codificação baseado em quantização vetorial.

Em codificação digital de sinais, quantização vetorial consiste portanto em uma técnica de compressão com perdas, uma vez que o sinal reconstruído é uma versão deteriorada do sinal original. O erro de quantização médio ao se representar o sinal de entrada por sua versão quantizada é chamado *distorção do quantizador* (MADEIRO et al., 2004).

Por sua vez, a taxa de codificação do quantizador vetorial, valor que exprime o número de *bits* por componente do vetor, é dada por $R = \frac{1}{K} \log_2 N$. Em codificação de sinais de voz, R é expressa em *bits* por amostra. Para problemas de codificação de imagens, R é expressa em *bits* por *pixel* (bpp).

Do exposto, pode-se afirmar que algoritmos de quantização vetorial tem por objetivo reduzir a redundância nos dados ao mesmo tempo que maximiza a qualidade da informação reconstruída. Este dois objetivos podem ser resumidos em uma só afirmação: o objetivo das técnicas de quantização vetorial é reduzir (para uma determinada taxa R) a distorção introduzida ao se representar os vetores de entrada \mathbf{x}^μ por suas correspondentes versões quantizadas $Q(\mathbf{x}^\mu)$.

De qualquer modo, existe uma relação muito estreita entre formação de agrupamentos e quantização vetorial. Uma vez que o vetor-protótipo é o elemento representativo de um certo grupo de dados, pode-se encontrar os dados que são representados por aquele protótipo e assim encontrar os agrupamentos. A Figura 1.2 ilustra qualitativamente a ligação entre a área de pesquisa em redes neurais competitivas com algumas áreas de conhecimento de interesse para esta dissertação.

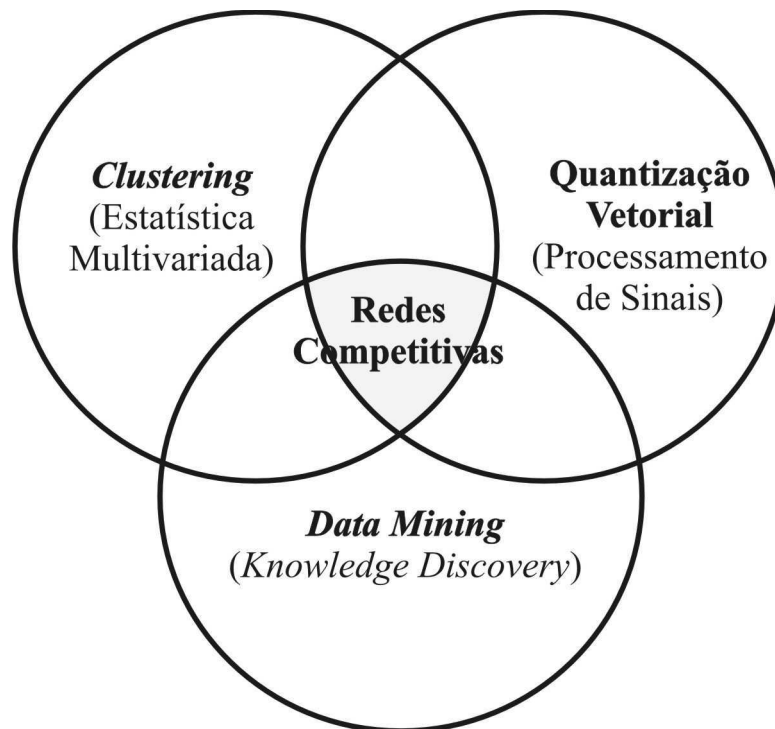


Figura 1.2: Projeção das redes competitivas nas diversas áreas.

Este trabalho não tem meta explorar a relação entre redes neurais competitivas e *Data Mining* (Mineração de Dados).

1.4 Motivações da Dissertação

Segundo Principe et al. (2000), apesar da importância histórica no campo das redes neurais e de serem utilizadas em diversas aplicações práticas, redes neurais competitivas são a classe menos estudada dentre os algoritmos neurais mais populares. Há algumas possíveis explicações para isto, entre elas o fato de o cenário da pesquisa em redes neurais ser amplamente dominado por redes neurais de aprendizado supervisionado, tais como redes Perceptron multicamadas (MLP) e redes de funções de base radial (RBF). Uma outra possível razão, mais plausível do ponto de vista da análise de dados, está no fato de que esta área já está repleta de algoritmos estatísticos equivalentes aos neurais, de tal modo que o pesquisador nesta área mantém-se hesitante (ou até descrentes) em usar algoritmos neurais que foram propostos originalmente para outro fim. Vejamos a Tabela 1.1 que mostra um comparativo entre o número de trabalhos em redes neurais competitivas e não-competitivas no evento ICANN'07 (*International Conference on Artificial Neural Networks 2007*).

REDES NEURAIAS COMPETITIVAS	REDES NEURAIAS NÃO-COMPETITIVAS
25	49

Tabela 1.1: Comparativo em números de trabalhos entre redes competitivas e não-competitivas no ICANN'07.

Esse certo grau de descrença de alguns pesquisadores com relação a algoritmos neurais competitivos, apesar de inúmeras aplicações práticas, dá-se principalmente pela ausência de estudos comparativos sistemáticos entre algoritmos neurais e não-neurais de análise de agrupamentos e de quantização vetorial. Embora existam alguns trabalhos que comparem diversos métodos (HAN et al., 2007; DIMITRIADOU et al., 2004; MAULIK; BANDYOPADHYAY, 2002), alguns pontos importantes ainda permanecem em aberto, tais como os mencionados a seguir.

- (i) Redes neurais competitivas são aplicadas sem distinção tanto em análise de agrupamentos, quanto em quantização vetorial, dando a idéia de que esta classe de redes neurais é uma panacéia em análise de dados. Assim, seguindo este raciocínio, bastaria alguém estudar redes competitivas que elas serviriam para qualquer uma das aplicações supracitadas. É importante notar a arrogância por trás desta abordagem. É comum encontrar

pesquisadores da área de redes neurais afirmarem, sem muito fundamento, que algoritmos de redes neurais competitivas têm desempenho superior quando aplicados em análise de agrupamentos e quantização vetorial. Contudo, estas áreas nasceram e se desenvolveram independentemente da área de redes neurais competitivas. Logo, é justo que critérios e metodologias de avaliação já existentes nas áreas de análise de agrupamentos e quantização vetorial sejam também utilizados para avaliar redes neurais competitivas. Somente depois disto, pode-se fazer um julgamento confiável se determinado algoritmo é realmente superior aos já existentes.

- (ii) A metodologia de comparação é inadequada ou ineficiente, principalmente em análise de agrupamentos. Por exemplo, em muitos artigos, principalmente aqueles oriundos da área de redes neurais, não há validação dos agrupamentos encontrados em conformidade com vários critérios confiáveis e já bem-estabelecidos na comunidade científica de análise de dados, tais como os descritos em (HALKIDI et al., 2001; FACELI et al., 2005).
- (iii) A diversidade de algoritmos é pequena, ou seja, ou se limitam a comparar alguns algoritmos clássicos não-neurais, ou comparam apenas algoritmos neurais. É comum os autores apresentarem um “novo” algoritmo, mas a comparação de desempenho é feita com algoritmos básicos, de fraco desempenho, e não com algoritmos de desempenho comprovadamente superior, o que certamente favorece o algoritmo proposto. Este fato é o principal fator motivador deste projeto de pesquisa. Com base na revisão bibliográfica feita nesta dissertação, o trabalho de Dimitriadou et al. (2004) é um dos mais completos neste sentido ao comparar o desempenho de nove algoritmos¹ em análise de agrupamentos. Em quantização vetorial, o trabalho de Hofmann & Buhmann (1998) é um dos poucos que compara o desempenho de vários algoritmos, dentre eles K -médias, WTA, SOM e Máxima Entropia.
- (iv) Um elemento importante para o sucesso de qualquer estudo comparativo para quantização vetorial está associado à determinação do número ótimo de neurônios para uma determinada aplicação. Em geral, isto é feito por meio de técnicas heurísticas ou por experimentação. Como não há um método que funcione para todas as situações, a alternativa que resta é avaliar a influência que cada tipo de métrica exerce sobre um dado algoritmo. A dificuldade reside no fato de que dependendo da aplicação (quantização vetorial ou análise de agrupamentos) as métricas de avaliação usadas são diferentes. Por exemplo, em tarefas de análise de agrupamentos, a validação dos agrupamentos encontrados é feita

¹A saber: aglomerativo hierárquico, K -médias, WTA, SOM, Neural-Gas, K -médias fuzzy, fuzzy-CL, CLARA, distância maximin.

com base em índices que quantificam os graus de compacidade e separabilidade entre agrupamentos, tais como Índice Dunn e Índice Davies-Bouldin (DB). Já em tarefas de quantização vetorial, determinado algoritmo é avaliado em função da qualidade da informação reconstruída, daí as métricas mais usadas serem o erro quadrático médio (MSE) ou a relação sinal-ruído de pico (PSNR). Um estudo comparativo poderia esclarecer se as métricas usadas em determinada área podem ser úteis em outra, e vice-versa.

- (v) Nenhuma das métricas citadas no item anterior leva em consideração a complexidade do algoritmo, ou seja, o número de parâmetros do modelo, o que no presente trabalho está associado ao número de neurônios (protótipos ou agrupamentos). Por exemplo, em quantização vetorial um número pequeno de neurônios pode implicar numa reconstrução inadequada da informação. Por outro lado, um número elevado pode tornar o método de recuperação excessivamente lento, inviabilizando sua aplicação em tempo real. A análise de agrupamentos também sofre com a questão do número adequado de parâmetros. Poucos agrupamentos podem esconder detalhes relevantes dos dados, enquanto muitos agrupamentos, por sua vez, podem revelar detalhes que são na verdade irrelevantes, tais como ruído.

Todos os cinco itens mencionados acima são elementos motivadores do presente trabalho, de tal modo que os objetivos da dissertação são fortemente balizados pelas observações feitas em cada um deles.

1.5 Objetivos da Dissertação

Esta seção descreve os objetivos desta dissertação, dividindo-os em dois grupos básicos: objetivo geral e objetivos específicos.

1.5.1 Objetivo Geral

De maneira geral, esse trabalho objetiva a avaliação de algoritmos de redes neurais competitivas em tarefas de análise de agrupamentos e quantização vetorial. Busca-se uma fertilização cruzada (*cross-fertilization*) entre estas áreas, ou seja, entender como cada área individualmente pode se beneficiar de conceitos e métodos próprios das outras áreas.

1.5.2 Objetivos Específicos

Em virtude dos vários pontos levantados na Seção 1.4, os objetivos específicos deste trabalho de mestrado são variados, realizando além da simples simulação de vários algoritmos de redes neurais competitivas. A lista dos pontos a serem perseguidos neste trabalho é apresentada a seguir.

1. Simular e comparar o desempenho de seis algoritmos neurais competitivos (WTA, SOM, Neural-Gas, RPCL, FSCL e FuzzyCL), bem como de um algoritmo não-neural (K-médias), em análise de agrupamentos;
2. Simular e comparar o desempenho dos algoritmos anteriores em tarefas de compressão de imagens e quantização vetorial;
3. Implementar técnicas de validação de resultados próprias de cada área de aplicação: índices Dunn e DB para análise de agrupamentos e métricas MSE e PNSR para quantização vetorial;
4. Investigar o efeito de cada método de validação de resultados na determinação do número ótimo de neurônios ou de protótipos, independentemente da aplicação;
5. Investigar a utilidade de critérios de validação que levem em consideração o número de parâmetros do modelo, tais como o critério de informação de Akaike (*Akaike's Information Criterion*, AIC) (AKAIKE, 1974) e o critério do Comprimento Descritivo Mínimo (*Minimum Description Length*, MDL) (RISSANEN, 1978).

1.6 Organização do Restante do Documento

O restante desta dissertação está organizada segundo os capítulos abaixo.

Capítulo 2 - Este capítulo traz uma breve descrição sobre as redes competitivas não-supervisionadas a serem avaliados nesta dissertação, a saber: WTA, SOM, FSCL, RPCL, Neural-Gas, bem como do algoritmo K-Médias e *Fuzzy Competitive Learning*. As principais características e idéias que levaram à proposição de cada algoritmo são também apresentadas.

Capítulo 3 - Este capítulo descreve e conceitua os principais técnicas de validação de agrupamentos e de seleção de modelos em quantização vetorial. Discute-se a viabilidade do uso dos métodos de seleção de modelos AIC e MDL.

Capítulo 4 - Este capítulo apresenta os resultados da aplicação dos índices Dunn/DB em um problema de análise de agrupamentos usando dados artificiais.

Capítulo 5 - Este capítulo apresenta e avalia os resultados da aplicação dos índices Dunn/DB, assim como das métricas MSQE/PNSR e dos critérios AIC/MDL dentre outros, em tarefas de quantização vetorial (compressão) de imagens reais.

Capítulo 6 - Neste capítulo estão as conclusões do estudo levado a cabo nesta dissertação e propostas para trabalhos futuros.

Apêndice A - Neste apêndice são apresentados os gráficos referentes às curvas do erro de quantização em função da época de treinamento do Capítulo 4.

Apêndice B - Neste apêndice é apresentada a aplicação desenvolvida para automatizar as execuções dos algoritmos e gerar os valores dos índices e métricas.

2 *Redes Neurais Competitivas*

2.1 Introdução

Este capítulo tem por objetivo apresentar sucintamente os algoritmos estatísticos e as arquiteturas de redes neurais competitivas avaliadas nesta dissertação, como forma de facilitar a compreensão dos métodos de validação de resultados em análise de agrupamentos e quantização vetorial que serão discutidos nos capítulos seguintes. O material a ser apresentado neste capítulo é, em grande parte, baseado nas seguintes referências: Kosko (1992), Haykin (1999), Principe et al. (2000), Kohonen (2001) e Frota (2005). Referências adicionais serão citadas quando necessárias.

É importante destacar que esta dissertação limita o escopo dos algoritmos de agrupamentos e quantização vetorial estudados nesta dissertação àqueles baseados em protótipos, sejam eles de origem neural ou não, e cujos protótipos são atualizados após a apresentação de um padrão de entrada (aprendizado padrão-a-padrão). As arquiteturas descritas neste capítulo são listadas a seguir.

- **Métodos clássicos:** Algoritmo K-médias, nas versões *batch* e seqüencial, e o algoritmo Linde-Buzo-Gray (LBG).
- **Redes neurais competitivas:** *Winner-Take-All* (WTA), *Frequency-Sensitive Competitive Learning* (FSCL), *Self-Organizing Map* (SOM), *Neural-Gas* (NGA) e *Rival Penalized Competitive Learning* (RPCL).
- **Métodos fuzzy:** K-médias *batch* nebuloso e *Fuzzy Competitive Learning* (FuzzyCL).

Para treinar os algoritmos a serem descritos neste capítulo assume-se que está disponível um conjunto de N exemplos, cada exemplo sendo representado como um vetor de dimensão p , i.e.

$\mathbf{x} \in \mathbb{R}^p$. Este vetor é representado como

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix}, \quad (2.1)$$

em que a j -ésima componente $x_j(t)$ carrega alguma informação relevante para a análise em questão, sendo denominada por isso de **característica** (*feature*) ou **atributo** (*attribute*). Em algumas aplicações, tal como análise de agrupamentos, as componentes de um vetor podem representar diferentes variáveis (massa, força, tensão, largura, etc.), ou podem representar a mesma variável deslocada espacialmente e/ou temporalmente (*pixels* de uma imagem ou amostras de sinais de voz), tal como em quantização vetorial.

Dessa forma, um vetor \mathbf{x} é, normalmente, chamado de **vetor de características** (*feature vector*) ou **vetor de atributos** (*attribute vector*) em análise estatística de dados (WEBB, 2002). Através do mapeamento dessas características é que os algoritmos não-supervisionados a serem descritos a seguir podem construir sua própria representação estatística dos dados de entrada.

2.2 Algoritmo *K*-Médias *Batch*

O primeiro algoritmo baseado em protótipos a ser descrito é o amplamente conhecido algoritmo *K*-médias (MACQUEEN, 1967), também conhecido no campo de quantização vetorial como algoritmo de Linde-Buzo-Gray (LBG) ou algoritmo de Lloyd generalizado (VASUKI; VANATHI, 2006). A aplicação do algoritmo *K*-médias a um conjunto de N vetores visa encontrar um conjunto de K protótipos, $\{\mathbf{w}_i\}_{i=1}^K$, $K \ll N$, que particione os dados de entrada em exatamente K grupos distintos.

A região de influência de determinado protótipo é chamada de partição de Voronoi (ou Dirichlet) daquele protótipo, sendo definida como

$$V_i = \{\mathbf{x} \in \mathbb{R}^p \mid \|\mathbf{x} - \mathbf{w}_i\| < \|\mathbf{x} - \mathbf{w}_j\|, \forall j \neq i\}, \quad (2.2)$$

em que $\|\cdot\|$ denota a norma euclidiana. Assim, com K protótipos o espaço de entrada é particionado em K regiões de Voronoi.

O algoritmo *K*-médias provê um método simples para a obtenção de K protótipos que minimizem a seguinte função-custo

$$D = \sum_{i=1}^K \sum_{\mathbf{x} \in V_i} \|\mathbf{x} - \mathbf{w}_i\|^2, \quad (2.3)$$

também conhecida pelos seguintes nomes: erro de quantização, erro de reconstrução ou ainda distorção. Esta minimização é realizada através da implementação da seguinte seqüência de passos:

Passo 1 - Seleção aleatória de K vetores do conjunto de dados para funcionar como protótipos iniciais.

Passo 2 - Separação do conjunto de dados em K regiões de Voronoi V_i , $i = 1, \dots, K$, de acordo com a Equação (2.2).

Passo 3 - Os novos protótipos são recalculados como as médias aritméticas (centróides) dos dados alocados a cada região de Voronoi V_i , ou seja

$$\mathbf{w}_i = \frac{1}{N_i} \sum_{\mathbf{x} \in V_i} \mathbf{x}, \quad (2.4)$$

em que N_i é o número de vetores pertencentes à célula de Voronoi do i -ésimo protótipo; ou equivalentemente, é o número de vetores de dados para os quais o protótipo \mathbf{w}_i é o mais próximo, segundo a métrica euclidiana.

Os passos 2 e 3 devem ser repetidos até que não haja mudanças substanciais no valor de D (Equação 2.3) ou um determinado número máximo de iterações tenha sido alcançado. Um critério de parada comumente usada verifica se a taxa de variação da distorção D está abaixo de um limiar de distorção $0 < \varepsilon \ll 1$ preestabelecido, ou seja

$$\left| \frac{D(n+1) - D(n)}{D(n+1)} \right| < \varepsilon, \quad (2.5)$$

em que o operador $|u|$ denota o valor absoluto de u e $D(n)$ corresponde ao valor do erro de quantização na n -ésima rodada de ajuste dos protótipos.

O interesse do algoritmo *K*-médias *batch* para esta dissertação é apenas didático. O interesse maior está em algoritmos de aprendizado padrão-a-padrão, também chamado de aprendizado *online* ou seqüencial, devido a sua maior utilização em aplicações de quantização vetorial.

2.3 Algoritmo *K*-Médias Seqüencial

No algoritmo *K*-médias *batch*, os protótipos são atualizados somente após todo o conjunto de dados ter sido apresentado. Daí a razão do termo *batch* no nome do algoritmo. Cada apresentação completa do conjunto de dados constitui uma *época* de treinamento. Assim, o treinamento do algoritmo *K*-médias *batch* se dá época-a-época, até a sua completa convergência.

Uma versão do algoritmo K -médias em que um dos protótipos é atualizado logo após a apresentação de um vetor de entrada é mais apropriada para problemas de quantização vetorial. Tal versão é comumente conhecida como K -médias seqüencial ou adaptativo (DARKEN; MOODY, 1990).

Os protótipos do algoritmo K -médias são iniciados como na versão *batch*. Em seguida, a cada iteração t , determina-se o índice $i^*(t)$ do protótipo mais próximo do vetor de entrada atual $\mathbf{x}(t)$. Por fim, atualiza-se o protótipo selecionado por meio da seguinte regra:

$$\mathbf{w}_{i^*}(t+1) = \mathbf{w}_{i^*}(t) + \frac{1}{C_{i^*}(t)} [\mathbf{x}(t) - \mathbf{w}_{i^*}(t)], \quad (2.6)$$

em que $C_{i^*}(t)$ denota o número de vetores de dados para os quais o protótipo \mathbf{w}_{i^*} foi selecionado até a iteração atual. Pode-se mostrar facilmente que os protótipos do algoritmo K -médias seqüencial convergem para a mesma solução que os protótipos da versão *batch*. Uma maior sensibilidade à iniciação dos protótipos é uma limitação importante da versão seqüencial em relação à versão *batch* do algoritmo K -médias.

2.4 Redes Neurais Não-Supervisionadas Competitivas

Grosso modo, redes neurais não-supervisionadas tentam extrair características estatísticas predominantes nos dados de entrada e constróem, de forma auto-organizada (i.e. sem auxílio de um professor externo e sem conhecimento prévio sobre a distribuição dos dados), uma representação reduzida do espaço de entrada, codificando-a em seus pesos sinápticos.

Redes competitivas constituem uma das principais classes de redes neurais artificiais (RNAs), nas quais um único neurônio ou um pequeno grupo deles, chamados **neurônios vencedores**, são ativados de acordo com o grau de proximidade entre seus vetores de pesos e o vetor de entrada atual, grau este medido segundo alguma métrica (HAYKIN, 1999). Esse tipo de algoritmo é comumente utilizado em tarefas de reconhecimento e classificação de padrões, tais como formação de agrupamentos (*clustering*), quantização vetorial e classificação de padrões. Nestas aplicações, o vetor de pesos associado ao neurônio vencedor é visto como um *protótipo* representativo de um determinado grupo de vetores de entrada.

Os modelos neurais competitivos avaliados neste trabalho são baseados na distância euclidiana como métrica utilizada para a determinação do neurônio vencedor. Neste caso, tem-se que um certo neurônio i é escolhido como o neurônio vencedor, simbolizado como $i^*(t)$, para o

vetor de entrada atual, se a seguinte relação for satisfeita:

$$i^*(t) = \arg \min_{\forall i} \|\mathbf{x}(t) - \mathbf{w}_i(t)\|, \quad (2.7)$$

na qual $i^*(t)$ é o índice do neurônio vencedor na rede, $\mathbf{x}(t)$ é um vetor de entrada da rede na iteração t e $\mathbf{w}_i(t) \in R^p$, é o vetor de pesos associado ao neurônio i . Os algoritmos descritos a seguir têm sua operação baseada na Equação (2.7).

2.5 Rede *Winner-Take-All*

No algoritmo competitivo mais simples, conhecido como *Winner-take-all* (WTA), durante a fase de treinamento apenas o neurônio vencedor tem seu vetor de pesos $\mathbf{w}_{i^*}(t)$ atualizado em resposta a um dado vetor de entrada $\mathbf{x}(t)$. O treinamento da rede WTA é resumido a seguir.

Passo 1 - Atribuição de valores iniciais aos K protótipos.

Passo 2 - Apresentação de um vetor de treinamento $\mathbf{x}(t)$ à rede.

Passo 3 - Determinação do neurônio vencedor, $i^*(t)$, para o vetor de entrada atual usando a Equação (2.7).

Passo 4 - Atualização do vetor de pesos do neurônio vencedor através da seguinte regra de aprendizagem:

$$\mathbf{w}_{i^*}(t+1) = \mathbf{w}_{i^*}(t) + \eta(t)[\mathbf{x}(t) - \mathbf{w}_{i^*}(t)], \quad (2.8)$$

em que $0 < \eta \ll 1$ denota o passo de aprendizagem.

Algumas observações importantes sobre a rede WTA são necessárias.

1. Para uma rede com K neurônios, a iniciação de seus protótipos é comumente feita de duas maneiras: (i) através da seleção aleatória de K vetores de dados; ou (ii) através da atribuição de valores aleatórios uniformemente distribuídos no intervalo $[0, 1]$. Recomenda-se a escolha do primeiro procedimento sempre que possível.
2. A rede WTA é equivalente ao algoritmo K -médias seqüencial. De fato, os dois algoritmos coincidem se o passo de aprendizagem for definido como $\eta = 1/C_{i^*}(t)$.
3. A taxa de aprendizagem pode ser fixa ou variável no tempo. Se for fixa, seu valor deve ser escolhido bem pequeno (e.g. $\eta = 0,01$ ou $\eta = 0,001$), a fim de garantir convergência

do algoritmo. Se for variável, pode-se escolher uma taxa inicial relativamente alta (e.g. $\eta_0 = 0,5$) para acelerar o aprendizado e uma taxa final pequena (e.g. $\eta_T = 0,01$ ou $\eta_T = 0,001$) para garantir a convergência do algoritmo. Entre os valores inicial e final pode-se adotar um decaimento linear ou exponencial. Em todas as redes competitivas simuladas nesta dissertação, utiliza-se um decaimento exponencial:

$$\eta(t) = \eta_0 \left(\frac{\eta_T}{\eta_0} \right)^{(t/T)}, \quad (2.9)$$

tal que η_0 e η_T ($\eta_0 \gg \eta_T$) são os valores inicial e final de η . A velocidade de decaimento é controlada pelo parâmetro T , que simboliza o número máximo de iterações de treinamento.

4. Pode-se mostrar que o vetor de pesos de um determinado neurônio i converge para o centróide (centro de gravidade) do conjunto de vetores de treinamento para o qual o neurônio i foi selecionado vencedor. Para isto basta perceber que os valores esperados de $w_i(t+1)$ e $w_i(t)$ são iguais para $t \rightarrow \infty$. Daí, simbolizado por w_i^o o valor final do vetor de pesos w_i , a seguinte expressão pode ser obtida:

$$E \{ \eta [\mathbf{x} - \mathbf{w}_i^o] \} = 0 \quad \Rightarrow \quad \mathbf{w}_i^o = \frac{\int_{V_i} \mathbf{x} p(\mathbf{x}) d\mathbf{x}}{\int_{V_i} p(\mathbf{x}) d\mathbf{x}}, \quad (2.10)$$

em que V_i é o conjunto de vetores de treinamento para o qual o neurônio i foi selecionado vencedor.

A despeito de sua simplicidade, a rede WTA é afetada por algumas questões que comprometem seriamente seu desempenho:

- **Escolha dos valores iniciais dos pesos da rede:** dependendo dos valores iniciais atribuídos aos pesos, alguns neurônios podem dominar o treinamento, sendo sempre selecionados como vencedores, enquanto outros nunca o são. As unidades não selecionadas são chamadas de unidades mortas (*dead units*).
- **Valorização excessiva da informação contida na entrada $\mathbf{x}(t)$ mais recente:** pela própria natureza do algoritmo, as entradas apresentadas à rede no início do treinamento têm menos influência no valor final dos pesos dos neurônios que aquelas apresentadas por último.

Para minimizar esses problemas, é comum modificar o algoritmo da rede WTA, criando variantes mais eficientes. As principais maneiras de se fazer isso são através da alteração da Equação

(2.7) ou da alteração da Equação (2.8). Algumas destas modificações dão origem aos três algoritmos seguintes.

2.6 Rede *Frequency-Sensitive Competitive Learning*

O primeiro algoritmo, chamado *Frequency-Sensitive Competitive Learning* (FSCL) foi proposto por Ahalt et al. (1990). A rede FSCL altera a Equação (2.7) a fim de penalizar neurônios que são escolhidos vencedores com muita frequência, de modo a permitir vitórias de outros neurônios:

$$i^*(t) = \arg \min_{\forall i} \{f_i(t) \cdot \|\mathbf{x}(t) - \mathbf{w}_i(t)\|\} \quad (2.11)$$

$$f_i(t) = \left[\frac{C_i}{t} \right]^z, \quad (2.12)$$

em que C_i é o número de vezes que o neurônio i foi escolhido vencedor até o instante t , e $z > 0$ é uma constante. O ajuste dos pesos na rede FSCL continua sendo feito de acordo com a Equação (2.8).

Nota-se que a presença do fator $f_i(t)$ como elemento ponderador da distância euclidiana ajuda a minimizar a ocorrência de unidades mortas. Caso alguns poucos neurônios dominem o processo de competição, ou seja, sejam escolhidos vencedores com maior frequência, eles tenderão a ter valores elevados de C_i com o passar do tempo. Isso fará com que outros neurônios, que antes eram selecionados com menor frequência, passem a também ser selecionados. Com o passar do tempo, todos os neurônios terão sido escolhidos em um número aproximadamente equivalente de vezes, tornando a competição mais justa.

2.7 Rede *Rival Penalized Competitive Learning*

A rede *Rival Penalized Competitive Learning* (RPCL) foi proposta por Xu et al. (1993) como um algoritmo para análise de agrupamentos. A idéia por trás da rede RPCL é a mesma que motivou a proposição da rede FSCL, ou seja, evitar neurônios mortos. Contudo, a abordagem é diferente. Na rede FSCL, dificilmente haverá neurônios mortos. Já na rede RPCL poderão existir neurônios mortos. Caso, eles ocorram, é porque eles eram realmente irrelevantes, podendo ser descartados da rede. Os neurônios remanescentes são suficientes para modelar os dados de entrada com precisão.

O principal aspecto deste algoritmo é que, ao mesmo tempo em que o vetor de pesos do

neurônio vencedor é modificado para se aproximar do vetor de entrada, o vetor de pesos de seu *rival* (isto é, o segundo vencedor) é modificado para distanciar-se do vetor de entrada. Em outras palavras, o neurônio vencedor é atraído para o vetor de entrada, enquanto o neurônio rival é repelido do vetor de entrada.

A implementação da rede RPCL é bastante similar à da rede WTA, bastando incluir o processo de determinação do neurônio rival e a regra de penalização de seu protótipo. Assim, primeiramente determina-se o neurônio vencedor através da Eq. (2.7) e atualiza-se o seu vetor de pesos através da Eq. (2.8). Em seguida, determina-se o neurônio rival por meio da seguinte regra:

$$i_2^*(t) = \arg \min_{\forall i \neq i_1^*} \|\mathbf{x}(t) - \mathbf{w}_i(t)\|, \quad (2.13)$$

em que o subscrito "2" enfatiza o fato de este ser o neurônio cujo protótipo é o segundo mais próximo do vetor de entrada atual.

Por fim, o protótipo do neurônio rival é penalizado por meio da seguinte regra de (des)-aprendizagem:

$$\mathbf{w}_{i_2^*}(t+1) = \mathbf{w}_{i_2^*}(t) - \gamma[\mathbf{x}(t) - \mathbf{w}_{i_2^*}(t)], \quad (2.14)$$

em que $0 < \gamma < 1$ é o fator de penalização. Uma das desvantagens da rede RPCL está na delicada relação entre o passo de aprendizagem (η) e o fator de penalização (γ). Geralmente, γ tem seu valor escolhido de maneira empírica entre 0,01 e 0,1 para não penalizar demais os neurônios rivais, o que pode causar uma desestabilização do aprendizado como um todo.

2.8 Rede Auto-Organizável de Kohonen

A rede neural competitiva conhecida comumente como SOM (*Self-Organizing Map*), foi proposta por Kohonen (1990, 2001) como um modelo computacional de mapas topográficos corticais, tais como mapa somatosensório, mapa tonotópico, dentre outros. Mapas corticais são representações sensoriais e motoras existentes no córtex cerebral. Para cada sensação haveria um verdadeiro mapa da superfície sensível (corpo, retina, ouvido) na área cortical primária¹ correspondente (GATTASS, 1993).

Apesar de sua origem como modelo neurobiológico, a rede SOM logo transcendeu as fronteiras de sua área de aplicação original, sendo hoje um dos modelos neurais mais conhecidos e utilizados, juntamente com as redes MLP, RBF e de Hopfield. As principais áreas de aplicação são análise de agrupamentos, quantização vetorial e visualização de dados multidimensionais.

¹Aquela que primeiro recebe a informação sensorial codificada vinda da superfície receptora.

Em termos de algoritmo, a rede SOM difere das redes competitivas anteriormente descritas pelo fato de seus neurônios estarem dispostos em uma grade (*grid*) fixa, geralmente uni- ou bidimensional, de modo que se possa definir uma relação de *vizinhança* espacial entre neurônios desta grade.

Assim, a Equação (2.8) é alterada pela inserção do conceito de *vizinhança*, que é o conjunto de neurônios que estão em torno do neurônio vencedor $i^*(t)$. Durante o treinamento os vetores de pesos dos neurônios na vizinhança do neurônio vencedor também passam a ser ajustados, de acordo com a seguinte regra de aprendizagem:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta(t)h(i^*, i; t)[\mathbf{x}(t) - \mathbf{w}_i(t)], \quad (2.15)$$

em que $h(i^*, i; t)$ é a função de vizinhança, geralmente do tipo gaussiana

$$h(i^*, i; t) = \exp\left(-\frac{\|\mathbf{r}_i(t) - \mathbf{r}_{i^*}(t)\|^2}{\lambda^2(t)}\right), \quad (2.16)$$

em que $\lambda(t)$ define o raio de influência da função de vizinhança, enquanto $\mathbf{r}_i(t)$ e $\mathbf{r}_{i^*}(t)$ são, respectivamente, as posições dos neurônios i e i^* no arranjo geométrico da rede.

A função de vizinhança funciona como uma espécie de janela de ponderação (*weighting window*), fazendo com que os neurônios mais próximos do neurônio vencedor atual tenham seus vetores de pesos atualizados mais intensamente que aqueles neurônios que estão mais distantes do neurônio vencedor. O neurônio vencedor tem seus pesos reajustados com maior intensidade, visto que para ele tem-se $h(i^*, i; t) = 1$. Para todos os outros neurônios, tem-se $h(i^*, i; t) < 1$.

Por questões de convergência e estabilização do aprendizado, a função de vizinhança deve decrescer no tempo, ou seja, o raio de influência $\lambda(t)$ decai com o decorrer do treinamento de modo semelhante à Equação (2.9):

$$\lambda(t) = \lambda_0 \left(\frac{\lambda_T}{\lambda_0}\right)^{(t/T)}, \quad (2.17)$$

na qual λ_0 e λ_T ($\lambda_0 \gg \lambda_T$) são os valores inicial e final de ϑ . Em suma, a Equação (2.17) faz com que a vizinhança diminua com o passar, começando com um grau.

É importante enfatizar que se os neurônios da rede SOM estão dispostos em uma grade unidimensional, tem-se que $\mathbf{r}_i(t) \in \mathbb{N}$, ou seja, a posição de um neurônio i qualquer coincide com seu próprio índice, $\mathbf{r}_i(t) = i$. Neste caso, cada neurônio possui apenas vizinhos à direita e à esquerda. Contudo, se os neurônios da rede SOM estão dispostos em uma grade bidimensional, tem-se que $\mathbf{r}_i(t) \in \mathbb{N}^2$, ou seja, a posição de um neurônio i na grade é dada pelas coordenadas

(x_i, y_i) em relação a uma origem pré-fixada. Neste caso, um neurônio pode ter vizinhos à esquerda, à direita, acima, abaixo e diagonalmente.

Em razão de sua arquitetura peculiar e de seu algoritmo de treinamento, a rede SOM implementa uma projeção não-linear Φ do espaço de entrada contínuo $\chi \subset \mathbb{R}^n$ (espaço dos dados), em um espaço de saída discreto \mathcal{A} , representado pelo espaço das coordenadas dos neurônios na grade, tal que $\dim(\mathcal{A}) \ll n$. Matematicamente, esta projeção pode ser simbolizada por:

$$\Phi : \chi \rightarrow \mathcal{A} \quad (2.18)$$

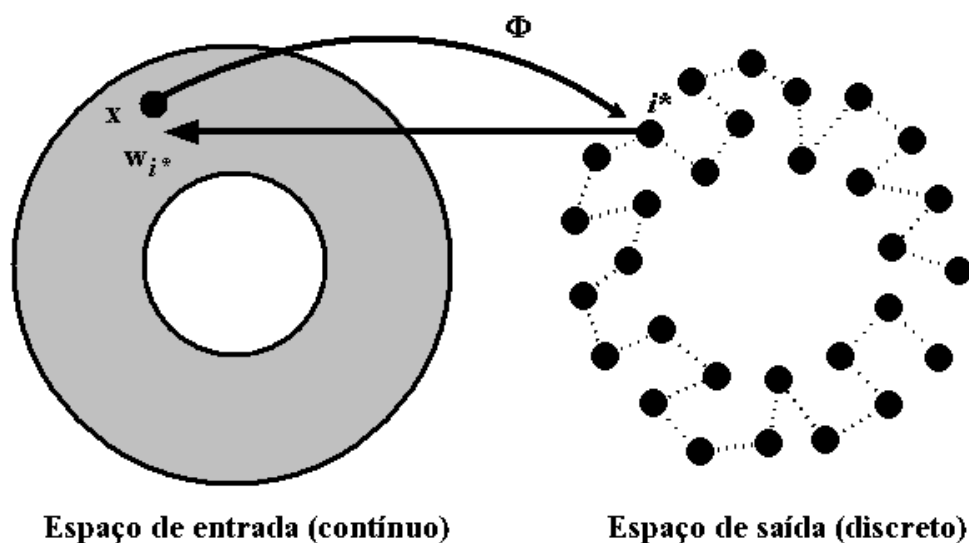


Figura 2.1: Projeção implementada pela rede SOM

A rede SOM tem tido grande utilização em aplicações de mineração de dados e reconhecimento de padrões. Grande parte do seu sucesso se deve à combinação de dois princípios essenciais de **auto-organização de sistemas** (von der Malsburg, 2003): (i) competição entre neurônios por recursos limitados, implementada pela Equação (2.7); e (ii) cooperação, implementada pela função vizinhança. O resultado da atuação destes dois princípios na rede SOM é uma projeção Φ que preserva relações de proximidade espacial entre os dados de entrada, ou seja, o mapeamento preserva a topologia do espaço de entrada no espaço de saída (HAYKIN, 1999), conforme ilustrado na Figura (2.1). Nesta figura, $\dim(\chi) = N = 2$ e $\dim(\mathcal{A}) = 1$, os pontos pretos correspondem às coordenadas dos vetores de pesos do i -ésimo neurônio. Neurônios que são vizinhos na grade unidimensional são conectados por linhas tracejadas.

Pode-se expressar a propriedade preservação de topologia da rede SOM da seguinte forma (HERTZ et al., 1991; PRINCIPE et al., 2000). Sejam \mathbf{x}_1 e \mathbf{x}_2 dois vetores no espaço de entrada χ . Sejam $r_{i_1}^*$ e $r_{i_2}^*$ as coordenadas dos neurônios vencedores para \mathbf{x}_1 e \mathbf{x}_2 , respectivamente. Diz-se que a rede SOM, corretamente treinada, preserva a topologia do espaço de entrada se as seguinte relação for observada

$$\|\mathbf{x}_1 - \mathbf{x}_2\| \rightarrow 0 \quad \Rightarrow \quad \|\mathbf{r}_{i_1}^* - \mathbf{r}_{i_2}^*\| \rightarrow 0, \quad (2.19)$$

ou seja, se quaisquer dois vetores estão fisicamente próximos no espaço de entrada, então eles terão neurônios vencedores espacialmente próximos na rede. As principais conseqüências desta propriedade são listadas a seguir:

- **Aproximação do espaço de entrada:** a rede SOM constrói uma aproximação discreta do espaço de entrada, na qual cada neurônio da rede representa uma determinada região do espaço de entrada que define sua **região de atração** ou **campo receptivo** (*receptive field*). Esta região é conhecida também como **célula de Voronoi** (*Voronoi cell*). Assim, uma das principais aplicações da rede SOM é a categorização de dados não-rotulados em agrupamentos (*clusters*) e sua posterior utilização na classificação de vetores de características que não estavam presentes durante o treinamento.
- **Estimação pontual da função densidade de probabilidade:** o mapeamento da rede SOM reflete variações na estatística do espaço de entrada. Ou seja, regiões no espaço de entrada χ de onde as amostras \mathbf{x} têm uma alta probabilidade de ocorrência são povoadas com um maior número de neurônios, possuindo, conseqüentemente, uma melhor resolução do que regiões em χ de onde amostras \mathbf{x} são retiradas com baixa probabilidade de ocorrência.

2.9 Rede Neural-Gas

O algoritmo Neural-Gas (NGA), proposto por Martinetz & Sculten (1991), foi motivado pelo fato de a disposição dos neurônios da rede SOM em estruturas geométricas fixas pré-definidas forçarem os neurônios a buscar principalmente um estado de ordenamento topológico (ou seja, de manutenção da preservação da vizinhança acima de tudo), em detrimento de um estado que represente bem a estatística dos dados. Por exemplo, a forma como a preservação da topologia é forçada na rede SOM pode fazer com que neurônios sejam alocados a regiões do espaço de entrada em que não há dados.

Em outras palavras, neurônios que poderiam estar representando algum grupo de dados, na verdade não estão representando grupo algum. Note que este é um fenômeno diferente daquele que causa neurônios mortos. Aqui os neurônios não estão mortos, no sentido de que nunca foram selecionados vencedores, apenas foram alocados para uma região em que não há dados a representar. Este fenômeno será estudado em mais detalhes no Capítulo 4.

A fim de evitar esta sub-utilização de neurônios o algoritmo da rede Neural-Gas é uma modificação simultânea das Equações (2.7) e (2.8). Não se busca diretamente um único vencedor, mas sim o ordenamento de todos eles na ordem crescente das distâncias euclidianas dos respectivos seus vetores de pesos à entrada atual $\mathbf{x}(t)$. Desta forma o neurônio vencedor i_1 é o primeiro da lista, o segundo mais próximo i_2 , e assim até o neurônio i_K , cujos pesos estão mais distantes da entrada:

$$\|\mathbf{x}(t) - \mathbf{w}_{i_1}(t)\| < \|\mathbf{x}(t) - \mathbf{w}_{i_2}(t)\| < \dots < \|\mathbf{x}(t) - \mathbf{w}_{i_K}(t)\|. \quad (2.20)$$

Então, o ajuste dos pesos é feito da seguinte forma:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta(t)h_\lambda(k,t)[\mathbf{x}(t) - \mathbf{w}_i(t)], \quad (2.21)$$

na qual $h_\lambda(k,t)$ funciona de modo equivalente à função vizinhança da rede SOM, sendo definida pela seguinte expressão:

$$h_\lambda(k,t) = \exp\left\{-\frac{(k-1)}{\lambda(t)}\right\}, \quad (2.22)$$

com k representando a posição do neurônio na lista ordenada definida pela Expressão (2.20). A variável $\lambda(t)$ decai com o tempo, como na Equação (2.9), equivalendo-se ao conceito de largura da vizinhança da rede SOM.

A rede Neural-Gas também é capaz de preservar a topologia do espaço de entrada no espaço dos pesos dos neurônios. Contudo, esta propriedade é alcançada sem que os seus neurônios sejam dispostos segundo o arranjo geométrico fixo da rede SOM. A maior limitação da rede Neural-Gas é seu custo computacional, pois a ordenação dos protótipos de acordo com sua distância em relação ao vetor de entrada é necessária a cada iteração.

2.10 Agrupamentos Fuzzy

Anteriormente foram descritos algoritmos de aprendizagem competitiva em que o vencedor é o único a ser atualizado a cada iteração, tal como a rede WTA. Este tipo de aprendizagem competitiva é denominada de competição dura (*hard competition*). Em outros algoritmos, como a

rede SOM, não só o neurônio vencedor tem seu protótipo atualizado, mas também os protótipos de seus vizinhos físicos são modificados. Este tipo de aprendizagem competitiva é denominada de competição suave (*soft competition*). Contudo, qualquer que seja o tipo de competição, um vetor de atributos só pode pertencer a um único agrupamento de dados, que é aquele representado pelo protótipo mais próximo.

Já no caso das técnicas de agrupamentos nebulosos (*fuzzy clustering*) (HÖPPNER et al., 1999), os dados podem possuir características que permitam que eles pertençam a diversos grupos com uma intensidade controlada por uma função de pertinência (ZADEH, 1965).

Seja $\mu_i(\mathbf{x})$ uma função que determina o grau de pertinência de um vetor \mathbf{x} ao agrupamento representado pelo protótipo \mathbf{w}_i . As seguintes propriedades devem ser obedecidas por $\mu_i(\mathbf{x})$:

$$\mu_i(\mathbf{x}) \in [0, 1] \quad \text{e} \quad \sum_{i=1}^K \mu_i(\mathbf{x}) = 1. \quad (2.23)$$

A incorporação da função de pertinência ao processo de particionamento dos vetores de entrada retarda a decisão sobre qual conjunto cada vetor pertencerá até o fim do treinamento. Isto pode ser vantajoso quando os *clusters* se sobrepõem, ou seja, não estão bem separados. A seguir são apresentados dois algoritmos de análise de agrupamentos fuzzy, um de treinamento estilo *batch* e o outro de treinamento seqüencial.

2.10.1 Algoritmo *K*-Médias *Batch* Nebuloso

A função custo associada ao algoritmo *K*-médias *batch*, mostrada na Equação (2.3), pode ser estendida para uma versão fuzzy proposta por Dunn (1973):

$$D = \sum_{i=1}^K \sum_{\mathbf{x} \in V_i} \|\mathbf{x} - \mathbf{w}_i\|^2 (\mu_i(\mathbf{x}))^z. \quad (2.24)$$

em que o expoente z representa o grau de nebulosidade da função. Usualmente atribui-se a ele um valor um pouco maior que 1. Se $z = 0$, recai-se no algoritmo *K*-médias *batch* clássico. Note que o somatório é feito a cada época, para todos os protótipos.

A atualização dos pesos usa a função de pertinência fuzzy. Para isso, é preciso calcular o grau de pertinência de um certo vetor a cada um dos agrupamentos, ou seja

$$\mu_i(\mathbf{x}) = \left(\sum_{j=1}^K \left(\frac{\|\mathbf{x} - \mathbf{w}_i\|}{\|\mathbf{x} - \mathbf{w}_j\|} \right)^{1/(z-1)} \right)^{-1}. \quad (2.25)$$

A minimização da Equação (2.24) leva à seguinte regra fuzzy de atualização dos protótipos:

$$\mathbf{w}_i = \frac{\sum_{\mathbf{x} \in V_i} \mathbf{x} \mu_i(\mathbf{x})}{\sum_{\forall \mu_i} \mu_i(\mathbf{x})}. \quad (2.26)$$

É importante notar que a regra da Equação (2.26) pode ser vista simplesmente como uma generalização para o caso de agrupamentos fuzzy daquela mostrada na Equação (2.3). Os critérios de iniciação dos protótipos do algoritmo K -médias *batch* nebuloso e seu critério de parada são os mesmos de sua versão não-nebulosa.

2.10.2 Rede *Fuzzy Competitive Learning*

Conforme mencionado anteriormente, o foco principal desta dissertação está na análise de desempenho de algoritmos seqüenciais em tarefas de clustering e de quantização vetorial. Faz-se necessário, portanto, obter uma versão seqüencial do algoritmo K -médias *batch* nebuloso.

A versão descrita a seguir é, na verdade, uma variante nebulosa do algoritmo WTA (CHUNG; LEE, 1994), com a diferença de que todos os protótipos são ajustados a cada iteração, em vez de um só como na rede WTA. A intensidade do ajuste é regulada justamente pela função de pertinência fuzzy. De forma geral, o ajuste dos pesos é dado pela seguinte expressão:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta(t) \mu_i(\mathbf{x}(t)) [\mathbf{x}(t) - \mathbf{w}_i(t)]. \quad (2.27)$$

É importante destacar que, como os protótipos são atualizados a cada iteração t , os valores de $\mu_i(\mathbf{x}(t))$ devem ser atualizados também a cada apresentação de um novo vetor de entrada \mathbf{x} .

Outro aspecto importante a destacar é a semelhança da regra de aprendizagem da Eq. (2.27) com as regras de aprendizagem das redes SOM e Neural-Gas. É devido a esta semelhança que alguns autores (BARALDI; BLONDA, 1999) afirmam que algoritmos neurais baseados em competição suave são funcionalmente equivalentes a algoritmos de análise de agrupamentos nebulosos.

2.11 Sobre Aplicações de Redes Neurais Competitivas

Uma das vantagens das redes competitivas é que elas são capazes de realizar quantização vetorial (GRAY, 1984), que pode ser entendida como uma estimação pontual da função densidade de probabilidade dos dados de entrada feita pelos vetores de pesos. Vários autores têm

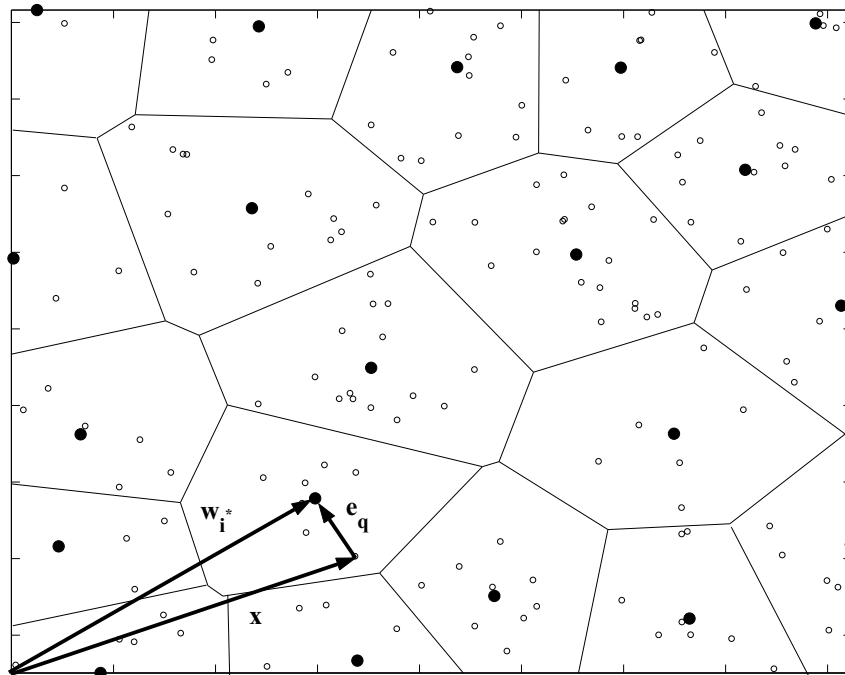


Figura 2.2: Ilustração do vetor erro de quantização \mathbf{e}_q . Os círculos abertos ('o') simbolizam os vetores de dados, enquanto os círculos fechados ('•') simbolizam os vetores de pesos (centróides). Figura extraída de (FROTA, 2005) sob permissão do autor.

explorado esta habilidade computacional das redes neurais competitivas em aplicações práticas de quantização vetorial (ROVETTA; MASULLI, 2007; LEUNG et al., 1997; LEUNG; CHAN, 1997; DONY; HAYKIN, 1995; YAIR et al., 1992).

Quantização vetorial consiste numa compressão de dados na qual, naturalmente, ocorre alguma perda de informação que pode ser avaliada por uma medida denominada **erro de quantização**. O vetor de erros de quantização, $\mathbf{e}_q(t)$, indica a qualidade da estimação, sendo definido como a diferença entre o vetor de entrada atual e o vetor de pesos do neurônio vencedor correspondente, ou seja:

$$\mathbf{e}_q(t) = \mathbf{x}(t) - \mathbf{w}_{i^*}(t), \quad (2.28)$$

sobre o qual opera uma medida de distância, em geral, euclidiana, dando origem a uma grandeza escalar denominada **erro de quantização** associado ao vetor $\mathbf{x}(t)$:

$$e_q(\mathbf{x}(t)) = \|\mathbf{x}(t) - \mathbf{w}_{i^*}(t)\| = \|\mathbf{e}_q(t)\| = \sqrt{\sum_{j=1}^p [x_j(t) - w_{i^*j}(t)]^2}, \quad (2.29)$$

na qual N é a dimensão de $\mathbf{x}(t)$. A Figura 2.2 mostra o *vetor erro de quantização* \mathbf{e}_q , cuja norma corresponde ao erro de quantização definido na Equação (2.29).

Duas importantes vantagens oferecidas pelas redes competitivas para o problema de análise

de agrupamentos e quantização vetorial são apresentadas a seguir:

- **Compressão de dados:** após a estabilização da rede com um limiar de erro de quantização aceitável, os pesos da rede podem ser usados em vez dos próprios dados, a quantidade de vetores de pesos (protótipos) é bastante reduzida em relação à quantidade de dados, portanto, tem-se uma redução considerável de esforço computacional. Além de reduzir o custo computacional, trabalhar com os protótipos aumenta a robustez do algoritmo, visto que os protótipos extraem qualidades estatísticas médias, filtrando flutuações aleatórias que porventura estejam presentes nos dados originais. Isso pode ser verificado através da iteração da Equação (2.8).
- **Simplificação de critérios de qualidade:** todos os testes podem ser feitos utilizando-se como critério o erro de quantização, alguns índices e métricas que serão estudados no Capítulo 3.

2.12 Conclusão

Este capítulo apresentou sucintamente, porém de forma auto-contida, as arquiteturas de redes neurais competitivas e algoritmos estatísticos a serem avaliadas nesta dissertação. As seguintes arquiteturas foram apresentadas neste capítulo:

- **Métodos estatísticos:** Algoritmo K -médias *batch* e seqüencial.
- **Redes competitivas não-supervisionadas:** *Winner-Take-All* (WTA), *Frequency-Sensitive Competitive Learning* (FSCL), *Self-Organizing Map* (SOM), *Neural-Gas* (NGA) e *Rival Penalized Competitive Learning* (RPCL).
- **Métodos fuzzy:** K -médias *batch* nebuloso e *Fuzzy Competitive Learning* (FuzzyCL)

Estes algoritmos descritas cobrem a uma boa parcela das técnicas neurais e não-neurais (estatísticas e fuzzy) utilizadas em problemas de análise de agrupamento e quantização vetorial e, por isso, conhecimentos básicos sobre o funcionamento destas redes será de grande valia na comparação de desempenho entre estes métodos por meio de técnicas de validação de agrupamentos e de seleção de modelos que serão estudadas no próximo capítulo.

3 Seleção de Protótipos em Análise de Agrupamentos e Quantização Vetorial

Este capítulo descreve uma série de metodologias e técnicas comumente utilizadas para avaliar algoritmos de análise de agrupamentos e quantização vetorial. Estes métodos são utilizados, em última instância, para definir o número de protótipos a serem utilizados. Em análise de agrupamentos, o número de protótipos é inferido principalmente a partir de métricas que avaliam o grau de separabilidade e coesão entre os agrupamentos (HALKIDI et al., 2001). Em tarefas de quantização vetorial, por outro lado, o número de protótipos é definido em função de outras métricas; por exemplo, o erro quadrático médio de quantização.

Embora usem os mesmos algoritmos, critérios de validação para uma aplicação não necessariamente é útil em outra. Avaliar o quanto os critérios de análise de agrupamentos podem ser úteis em quantização vetorial, e vice-versa, é um dos objetivos desta dissertação. Além disso, busca-se avaliar também o quanto métricas que levam em consideração a complexidade (número de parâmetros) do algoritmo, os chamados critérios de informação, podem ser úteis para ambas as aplicações. Alguns destes métodos de seleção de modelo são descritos no final deste capítulo.

Este capítulo é, em grande parte, baseado no trabalho de Halkidi et al. (2001) e de Faceli et al. (2005). O trabalho de Aguirre (2000) em critérios de informação, amplamente utilizados na seleção de modelos de um modo geral.

3.1 Objetivos da Validação de Agrupamentos

Descreve-se a seguir as estratégias mais usadas para se validar algoritmos de análise agrupamentos baseados em protótipos. A idéia é utilizar métricas, os chamados índices de validação, que quantifiquem quão bem um determinado algoritmo agrupa ou segrega os dados em K grupos distintos. A validação do resultado de um agrupamento, em geral, é feita com base em índices estatísticos, que julgam, de uma maneira qualitativa, o mérito das estruturas encontradas.

Um índice quantifica alguma informação a respeito da qualidade de um agrupamento. A maneira pela qual um índice é aplicado para validar um agrupamento, é chamada critério de validação. Assim, um critério de validação expressa a estratégia utilizada para validar uma estrutura de agrupamento, enquanto que um índice é uma estatística pela qual a validade é testada.

Existem basicamente três tipos de critérios para investigar a validade de um agrupamento, a saber: critérios internos, externos e relativos. As estruturas que podem ser avaliados na validação de um agrupamento são organizadas em três tipos: hierarquias, partições e agrupamentos individuais.

A avaliação do resultado de um agrupamento deve ser pautada pelo propósito de determinar quantitativamente se os agrupamentos são significativos, ou seja, se a solução é representativa para o conjunto de dados em análise. É importante lembrar que esta é uma tarefa de otimização bastante complexa. O número de combinações de dados em K grupos distintos é muito grande, sendo uma tarefa difícil de ser realizada por um ser humano sem a ajuda de algoritmos computacionais adequados.

Obviamente, há situações triviais que devem ser descartadas, tais como um único agrupamento contendo todos os dados, ou um número de agrupamentos igual ao número de dados. Neste último caso, cada dado corresponderia a um agrupamento com um único elemento. Mesmo eliminando essas soluções triviais, ainda é uma tarefa complexa determinar se os agrupamentos obtidos são válidos.

Uma estrutura de agrupamento é válida se não ocorreu por acaso, ou se é “rara” em algum sentido, já que qualquer algoritmo de agrupamento sempre encontrará agrupamentos, independentemente se existe ou não algum grau de similaridade nos dados. Entretanto, assumindo que há de fato alguma similaridade entre os dados, alguns algoritmos podem encontrar agrupamentos mais adequados que outros.

Algumas técnicas para a validação dos resultados de técnicas de agrupamento têm sido discutidas na literatura, tais como testes de significância nas variáveis utilizadas para criar os agrupamentos, replicação, testes de significância nas variáveis externas e simulação de Monte Carlo. Estudos sobre diversos métodos de validação de agrupamentos podem ser encontrados em (XU; BRERETON, 2005; HALKIDI et al., 2001; GORDON, 1999; JAIN; DUBES, 1988).

As abordagens de validação de agrupamentos tradicionais, descritas na literatura supracitada, obtêm melhores resultados quando os agrupamentos são geralmente compactos. Para aplicações que envolvem agrupamentos de formas arbitrárias, os critérios de validação tradicionais

(variância, densidade, continuidade e separação) são muitas vezes insuficientes. Isto demonstra que a análise de agrupamentos é um campo ainda pouco explorado para o desenvolvimento de novas abordagens e, carente de abordagens comparativas e unificadoras.

3.2 Critérios de Validação de Agrupamentos

Simplificadamente, critérios de validação indicam a maneira pela qual um índice é aplicado para validar um agrupamento. De acordo com Halkidi et al. (2001) e Faceli et al. (2005), existem três tipos de critérios para investigar a validade de um agrupamento, a saber: critérios internos, critérios externos e critérios relativos.

Os critérios internos quantificam a qualidade de um agrupamento com base apenas nos dados originais (matriz de dados, matriz de distância ou matriz de similaridade). Por exemplo, um critério interno pode avaliar o grau em que uma partição obtida por um dado algoritmo de agrupamento é justificado pela matriz de similaridade.

Já os critérios externos quantificam um agrupamento julgando-o a partir de uma estrutura pré-definida, que é imposta ao conjunto de dados e que reflete a intuição do usuário sobre a estrutura presente nos dados. Essa estrutura pré-definida pode ser uma partição que se sabe previamente existir nos dados, ou um agrupamento construído por um especialista da área com base em conhecimento prévio. Um critério externo pode avaliar o grau de correspondência entre o número de agrupamentos obtidos com o agrupamento e os rótulos dos dados conhecidos previamente.

Por fim, os chamados critérios relativos comparam diversos agrupamentos para decidir qual deles é o melhor em algum aspecto, por exemplo, qual é mais o estável ou qual é o mais adequado aos dados. Estes critérios Podem ser utilizados para comparar diversos algoritmos de agrupamento ou para determinar o valor mais apropriado de algum parâmetro do algoritmo aplicado, como o número de agrupamentos. Por exemplo, pode-se medir quantitativamente qual dentre dois algoritmos melhor se ajusta aos dados ou determinar o número de agrupamentos mais apropriado para um agrupamento feito com um determinado algoritmo.

Nos próximos tópicos são detalhados os tipos de critérios, métricas e casos em que são aplicáveis.

3.2.1 Critérios Internos e Externos

Os critérios internos e externos são baseados em testes estatísticos e tendem a ter custo computacional elevado (HALKIDI et al., 2001). Seu objetivo é medir o quanto o resultado obtido confirma uma hipótese pré-especificada. Neste caso, são utilizados testes de hipótese para determinar se uma estrutura obtida é apropriada para os dados. Isto é feito testando se o valor do índice utilizado é muito grande ou muito pequeno. Isto requer o estabelecimento de uma população base ou de referência. O mesmo índice pode ser utilizado em um critério interno e externo, embora as distribuições de referência do índice sejam diferentes (JAIN; DUBES, 1988).

A proposição de um índice para validação é fácil, o difícil é estabelecer limiares de referência com base nos quais se possa afirmar que o valor do índice é grande ou pequeno o suficiente para se considerar o agrupamento atípico (raro) ou válido.

Os índices de validação, ou estatísticas, são funções dos dados que quantificam informações úteis, como o erro quadrático de um agrupamento ou a compactação de seus agrupamentos.

Um índice é, em si, uma variável aleatória, pois depende da distribuição da qual a amostra de dados é coletada, das condições iniciais de um dado algoritmo, tais como os valores dos protótipos iniciais, e dos valores dos parâmetros de treinamento do algoritmo (e.g. número de protótipos).

A distribuição de um índice descreve a frequência relativa com a qual seus valores são gerados sob alguma hipótese. Uma hipótese é uma afirmação sobre a frequência relativa de eventos no espaço amostral que expressa um conceito. Esse conceito pode ser a ausência de estrutura nos dados (FACELI et al., 2005).

No caso de validação de agrupamentos, a hipótese nula, H_0 , é uma afirmação de não estrutura ou aleatoriedade dos dados. A seleção da hipótese nula depende do tipo dos dados e do aspecto dos dados que estão sendo analisados.

Em Jain & Dubes (1988), são resumidos também os principais aspectos relacionados à utilização de um índice de validação, que são listados a seguir:

- Definição do índice: o índice deve não só fazer sentido intuitivamente, mas também ser fundamentado teoricamente. Facilidade na implementação computacional também é um propriedade desejável de um índice.
- Definição de uma distribuição de probabilidades de referência: uma distribuição de referência ou distribuição base é uma distribuição derivada de uma população que não possui

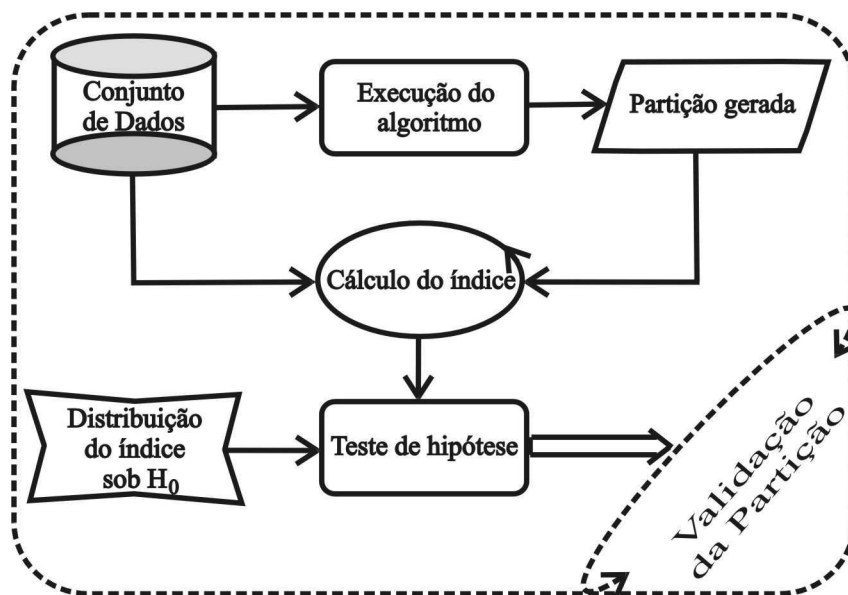


Figura 3.1: Metodologia de aplicação de um critério interno.

estrutura. Uma população referência é definida ou implicada pela distribuição base.

- Verificação da não-aleatoriedade da estrutura: o valor de um índice de validação é comparado com um limiar que estabelece um dado nível de significância. O limiar é definido a partir da distribuição base, que raramente é conhecida na teoria.
- Verificação do tipo de estrutura: a habilidade do índice de validação em indicar uma estrutura conhecida indica seu poder estatístico. A escolha da estrutura depende da aplicação particular.

O cálculo do índice para critérios internos depende somente dos próprios dados, seja na forma de matriz de dados ou matriz de distâncias.

Quanto aos critérios externos, há necessidade da utilização de uma partição dos dados conhecida previamente. Esta partição recebe o nome de “partição real”.

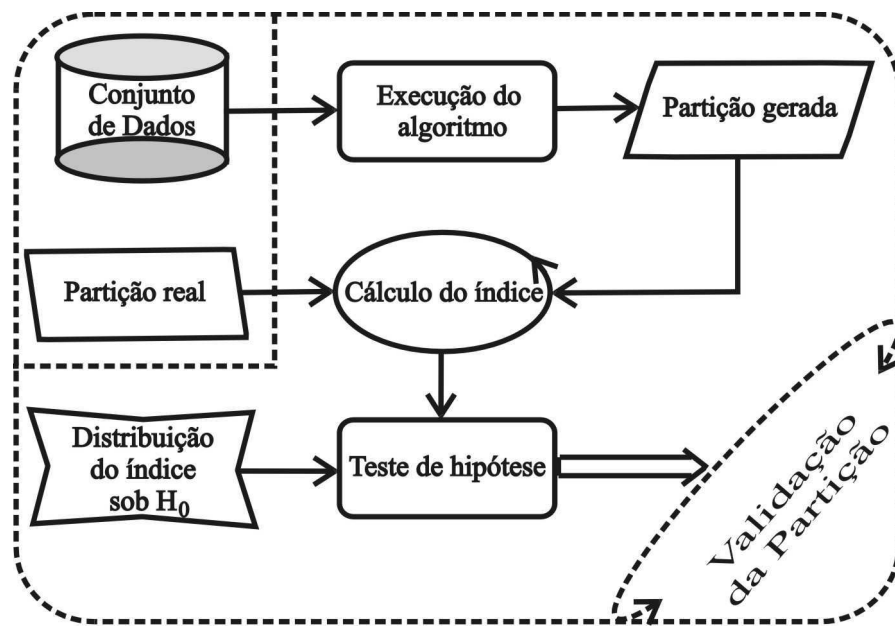


Figura 3.2: Metodologia de Aplicação de Critérios Externos

Uma limitação da validação interna ou externa de agrupamentos está no estabelecimento da distribuição dos índices (estatísticas) sob a hipótese nula (H_0) e conseqüentemente a determinação dos limiares que dizem se uma partição é adequada de acordo com o índice. Os testes reais de validação são geralmente definidos utilizando ferramentas estatísticas, tais como simulação de Monte Carlo e técnicas de reamostragem (e.g. *bootstrapping*).

3.2.2 Simulação de Monte Carlo

Trata-se de um método para estimar parâmetros e taxas de probabilidade por meio de amostragem aleatória por computador. É utilizada quando tais valores são muito difíceis ou impossíveis de serem calculados analiticamente.

Em validação de agrupamentos, uma das formas mais comuns de utilização de simulação de Monte Carlo é no estabelecimento da distribuição referência de um índice sob a hipótese nula. Inicialmente, é gerada uma grande quantidade de conjuntos de dados sintéticos de acordo com a distribuição considerada na hipótese nula H_0 . Cada um desses conjuntos é agrupado e o valor do índice é calculado em cada caso. Com esses valores do índice é traçado um gráfico de dispersão, que é uma aproximação da função de densidade de probabilidade do índice. Dado o valor do índice para o agrupamento que está sendo validado e a distribuição estimada, determina-se a possibilidade da hipótese H_0 ser aceita ou rejeitada.

Uma limitação da análise por simulação de Monte Carlo é a definição do modelo nulo, o que não é uma tarefa trivial. Outra limitação está demanda de grande quantidade de recursos computacionais, uma vez que é necessário um número elevados de replicações para construir a distribuição de referência.

3.2.3 *Bootstrapping*

As técnicas de *bootstrapping* utilizam reamostragem aleatória da amostra de dados originais, com reposição, a fim de criar uma nova amostra aleatória de dados. A nova amostra é utilizada como se fosse uma replicação de um experimento de Monte Carlo (JAIN; DUBES, 1988). Neste caso, as amostras *bootstrap* são utilizadas para construir o modelo sob a hipótese nula. As amostras podem ser obtidas reamostrando-se os padrões ou os atributos do conjunto de dados.

Com *bootstrapping* evita-se os problemas das técnicas de simulação de Monte Carlo relacionados com a escolha do modelo nulo.

3.2.4 Critérios Relativos

Os critérios relativos de validação têm por meta encontrar o melhor agrupamento que um algoritmo pode obter sob certas suposições e valores para seus parâmetros ou determinar o algoritmo mais apropriado para os dados/estruturas analisados.

A forma mais comum de aplicação de um índice com um critério relativo, consiste do cálculo do seu valor para vários agrupamentos obtidos, obtendo-se uma seqüência de valores. O agrupamento mais adequado é determinado pelo valor que se destaca nessa seqüência, como o valor máximo, mínimo ou inflexão na curva do gráfico construído com a seqüência de valores calculados.

A Figura 3.3 resume a metodologia de aplicação de critérios relativos de validação de agrupamentos. Conforme indica a figura, nesse tipo de critério, vários algoritmos, ou um mesmo algoritmo com diferentes valores para seus parâmetros (como no caso do parâmetro z da rede FSCL), são aplicados ao mesmo conjunto de dados. O índice é calculado para cada uma das partições obtidas. Esses valores do índice são avaliados, geralmente com o auxílio de um gráfico, para se determinar o melhor algoritmo ou o melhor valor para um ou mais parâmetros de um dado algoritmo.

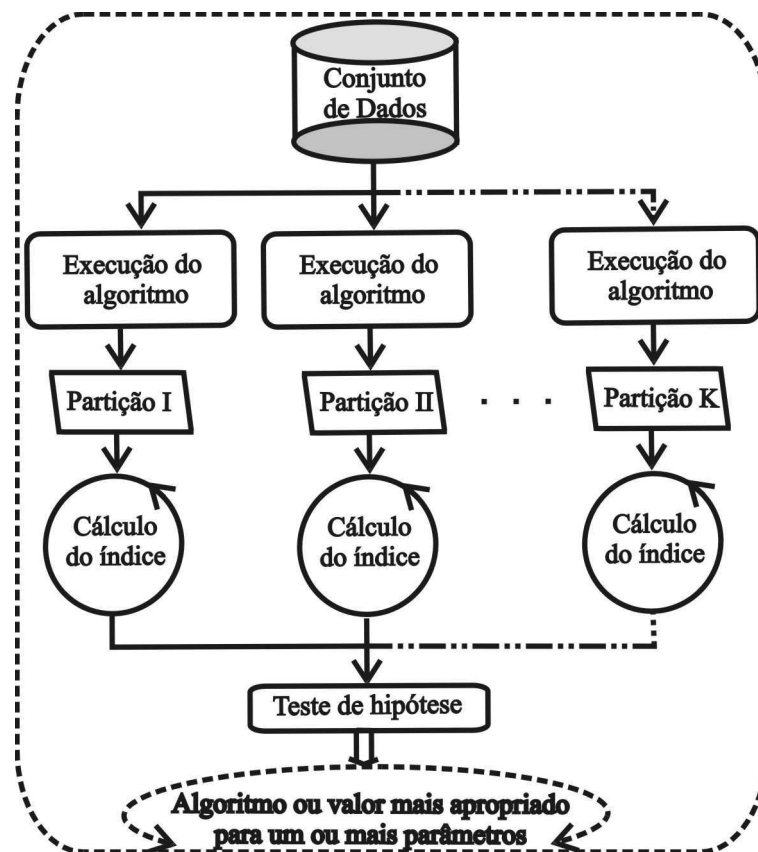


Figura 3.3: Metodologia de Aplicação de Critérios Relativos.

Conforme mencionado anteriormente neste capítulo, três tipos de estruturas podem ser avaliadas: hierarquias, partições e agrupamentos individuais. Os três critérios de validação podem ser empregados na validação de qualquer um destes tipos de estrutura. Esta dissertação tem como foco a avaliação de partições geradas por redes neurais competitivas não-supervisionadas, e os critérios de validação de interesse são do tipo relativo. Na próxima seção são descritos alguns destes critérios.

3.3 Índices Baseados em Critérios Relativos

O objetivo da validação relativa é encontrar um particionamento dos dados em agrupamentos que melhor caracterizem os dados, para um dado algoritmo sob certas suposições e valores para seus parâmetros.

A partir da próxima seção começam a ser apresentados alguns exemplos de índices de validação de agrupamentos, tanto tradicionais quanto propostos recentemente, que são baseados

em critérios relativos. Estes índices serão alvo de estudos desta dissertação. O leitor interessado pode encontrar estes e vários outros índices em Faceli et al. (2005), Xu & Brereton (2005) e Halkidi et al. (2001).

3.3.1 Estatística Γ de Hubert Modificada

A estatística Γ de Hubert modificada (THEODORIDIS; KOUTROUBAS, 1999) é expressa pela equação:

$$\Gamma = (1/N) \sum_{i=1}^{n-1} \sum_{j=i+1}^n M(i, j) \cdot Q(i, j), \quad (3.1)$$

em que $N = n(n-1)/2$, M é a matriz de proximidade do conjunto de dados e Q é uma matriz $n \times n$, na qual cada célula (i, j) é a distância entre os pontos representativos (vs_i, vs_j) dos agrupamentos dos padrões x_i e x_j . A estatística $\hat{\Gamma}$ normalizada é representada pela seguinte equação:

$$\hat{\Gamma} = \frac{[(1/N) \sum_{i=1}^{n-1} \sum_{j=i+1}^n (M(i, j) - \mu_M)(Q(i, j) - \mu_Q)]}{\sigma_M \sigma_Q}, \quad (3.2)$$

em que μ_M e μ_Q são os valores médios dos elementos das matrizes M e Q , respectivamente, e σ_M e σ_Q representam os desvios padrões correspondentes.

É importante destacar que o gráfico desse índice tende a decrescer monotonicamente com o aumento do número de agrupamentos. O valor desse índice é 1 para o agrupamento trivial, em que cada padrão pertence a um agrupamento individual, e não está definido para $K = 1$. A curva tende a ser plana (paralela ao eixo) quando os dados contêm agrupamentos, mas tende a ter uma inclinação positiva para dados aleatórios. Um valor alto de Γ e/ou $\hat{\Gamma}$ sugere a existência de agrupamentos compactos.

Segundo Jain & Dubes (1988), o índice Γ é recomendado para dados dados organizados em agrupamentos hipersféricos, não sendo apropriado para agrupamentos de formas arbitrárias, tais como hiperelipsoidais. Em geral, os índices Γ e $\hat{\Gamma}$ são usados para validação de agrupamentos testando se a associação entre M e Q é muito grande sob a hipótese de rótulos aleatório.

3.3.2 Família de índices Dunn

A família de índices Dunn (DUNN, 1973) é representada genericamente pela seguinte expressão:

$$D(K) = \frac{\min_{i \neq j} \{\delta(S_i, S_j)\}}{\max_{1 \leq l \leq k} \{\Delta(S_l)\}}, \quad (3.3)$$

em que $\delta(S_i, S_j)$ denota uma função de dissimilaridade (e.g. distância euclidiana) entre os agrupamentos S_i e S_j , e $\Delta(S_i)$ é uma medida da dispersão dos dados dentro do agrupamento S_i .

No índice Dunn original, $\delta(S_i, S_j)$ é definido pela equação:

$$\delta(S_i, S_j) = \min_{x \in S_i, y \in S_j} \{d(x, y)\}, \quad (3.4)$$

e $\Delta(S_i)$ é dada pela equação:

$$\Delta(S_i) = \max_{x, y \in S_i} \{d(x, y)\}. \quad (3.5)$$

O índice Dunn é, portanto, a razão da separação entre os agrupamentos (*intercluster*) e dentro dos agrupamentos (*intracluster*) (PAKHIRA et al., 2004).

O gráfico índice $D(K)$ não apresenta nenhuma tendência em relação ao número de agrupamentos. O ponto máximo no gráfico de $D(K)$ contra K pode ser uma indicação do número de agrupamentos que mais se ajusta aos dados. De acordo com Halkidi et al. (2002), o índice Dunn apresenta bons resultados na identificação de agrupamentos compactos e bem separados. Valores altos do índice sugerem a presença desse tipo de agrupamento.

As principais limitações do índice Dunn original são a sua complexidade e sensibilidade a ruído. Assim como a estatística Γ de Hubert, esse índice também não é apropriado para agrupamentos de formas arbitrárias.

3.3.3 Índice Davies-Bouldin (DB)

Seja $R_{i,j}$ uma medida de similaridade entre dois agrupamentos S_i e S_j , dada pela equação:

$$R_{j,k} = \frac{e_{S_j} + e_{S_k}}{d(S_j, S_k)}, \quad (3.6)$$

em que e_{S_j} e e_{S_k} são, respectivamente, os erros médios para os agrupamentos S_j e S_k , e $d(S_j, S_k)$ denota a distância euclidiana entre os protótipos dos dois agrupamentos.

Seja o índice para o k -ésimo agrupamento dado indicado por

$$R_k = \max_{j \neq k} \{R_{j,k}\}. \quad (3.7)$$

O índice de DB é representado pela seguinte expressão:

$$DB(K) = \frac{1}{K} \sum_{k=1}^K R_k. \quad (3.8)$$

O índice DB, proposto por Davies & Bouldin (1979), é uma função da razão da soma da dispersão dentro dos agrupamentos pela dispersão entre os agrupamentos (PAKHIRA et al., 2004).

O melhor agrupamento é dado pelo menor valor do índice DB para uma seqüência de valores calculados para diferentes número de agrupamentos K . É importante notar que o valor do índice DB é nulo para o agrupamento trivial, em que cada padrão pertence a um agrupamento individual, devendo ser computado apenas quando cada agrupamento contiver um número razoável (ou seja, maior do que 1) de padrões.

Jain & Dubes (1988) aponta que o índice DB é mais indicado para dados que se organizam em agrupamentos hiperesféricos, não sendo apropriado para agrupamentos de formas arbitrárias.

3.3.4 Silhuetas

A medida de silhueta, proposta por Rousseeuw (1987), é calculada para cada padrão que faz parte de um agrupamento. As silhuetas medem a validade dos agrupamentos com base na proximidade entre os padrões de um agrupamento e na distância dos padrões de um agrupamento ao agrupamento mais próximo.

Em termos mais formais, denota-se por $a(x_i)$ a dissimilaridade média do padrão x_i em relação a todos os outros padrões do agrupamento S_i e por $d(x_i, S_j)$ a dissimilaridade média do padrão x_i em relação aos padrões do agrupamento S_j .

Seja $b(x_i)$, a menor dissimilaridade média de x_i em relação a todos os demais agrupamentos, quantificada pela seguinte equação:

$$b(x_i) = \min_{C_j \neq C_i} \{d(x_i, C_j)\}. \quad (3.9)$$

Assim, a silhueta de um dado padrão, $\delta(x_i)$, empregando uma medida de dissimilaridade, é dada por

$$\delta(x_i) = \begin{cases} 1 - a(x_i)/b(x_i), & a(x_i) < b(x_i) \\ 0, & a(x_i) = b(x_i) \\ b(x_i)/a(x_i) - 1, & a(x_i) > b(x_i) \end{cases} . \quad (3.10)$$

Silhuetas visam indicar quais padrões estão bem situados dentro dos seus agrupamentos e quais estão fora de um agrupamento apropriado. Elas podem ser calculadas para agrupamentos

realizados utilizando tanto medidas de similaridade (e.g. produto escalar) quanto medidas de dissimilaridade (e.g. distância euclidiana).

Diferentemente dos três critérios relativos descritos nas seções anteriores o método de validação de agrupamentos por silhuetas pode ser aplicado tanto a agrupamentos de formatos hiperesféricos, quanto a agrupamentos de forma arbitrárias. A principal limitação do método de validação de agrupamentos por silhuetas está no elevado custo computacional.

Os critérios de validação de agrupamentos relativos visam quantificar os graus de compactação e separabilidade dos agrupamentos. A princípio, estes métodos não levam em consideração a complexidade do algoritmo, ou seja, o número de agrupamentos pode ser elevado, desde que os graus de compactação e separabilidade sejam considerados adequados pelo usuário. Sabe-se que a estimação de um número adequado de protótipos é de fundamental importância não só para análise de agrupamentos como também para quantização vetorial.

A seguir são descritos alguns critérios comumente utilizados na área de identificação de sistemas e séries temporais para seleção e validação de modelos. Estes são problemas correlatos ao problema de ajuste de curva, em que um modelo matemático (e.g. polinômio de ordem K) tem seus parâmetros estimados a partir dos dados. O objetivo é encontrar o modelo que melhor explique o processo gerador dos dados. De forma simplificada, em seleção de modelos busca-se pelo modelo que melhor *explique* os dados.

Esta dissertação propõe-se a avaliar a plausibilidade do uso de tais critérios em tarefas de análise de agrupamentos e quantização vetorial.

3.4 Critérios de Informação

No âmbito da identificação de sistemas, existem diversos procedimentos que permitem estimar a ordem de modelos dinâmicos a partir de dados medidos. Entre tais procedimentos, destacam-se o critério do erro final de predição (*final prediction error*, FPE), o critério de informação de Akaike (*Akaike's Information criterion*, AIC) (AKAIKE, 1974), o critério de informação bayesiano (*Bayesian Information criterion*, BIC) (KASHYAP et al., 1977; CRUTCHFIELD; MCNAMARA, 1987), e o critério do comprimento mínimo de descrição (*Minimum Description Length*, MDL) (RISSANEN, 1983, 1978).

Fazendo um paralelo com os critérios de validação de agrupamentos previamente apresentados, estas métricas correspondem a critérios do tipo relativo. Esta dissertação propõe avaliar o uso dos critérios de estimação da ordem para selecionar o número ótimo de protótipos. Um

estudo semelhante também foi feito no trabalho de (HU; XU, 2004). Nas próximas seções dá-se início à descrição dos critérios de informação.

3.4.1 Critério do Erro Final de Predição

O critério FPE foi proposto por Akaike (1969) para selecionar a ordem de um processo linear auto-regressivo (AR), de modo a minimizar a variância do erro médio de predição ao mesmo tempo que penaliza o excesso de parâmetros do modelo. Matematicamente, o critério FPE pode ser descrito pela seguinte expressão:

$$FPE(K) = N \ln \left(\frac{RSS(K)}{N} \right) + N \ln \left(\frac{N+K}{N-K} \right), \quad (3.11)$$

em que N é o número de amostras, K é a ordem do modelo e $RSS(K)$ é a soma dos quadrados dos resíduos¹ para o modelo com K parâmetros.

A primeira parte do lado direito da Equação 3.11 representa uma função com uma tendência exponencial decrescente à medida que o valor de K aumenta. Por outro lado, a segunda parte dessa equação deve atuar como um termo de penalização para o excesso de parâmetros e, por isso, exibe uma tendência crescente à medida que K aumenta. Assim, acredita-se que a função $FPE(K)$ é convexa e que o seu ponto de mínimo indica a ordem mais adequado do modelo de ordem K , para aquele conjunto de dados.

Akaike (1976) destaca que, embora o critério FPE funcione perfeitamente para processos AR puros, ele a se tornar uma métrica bastante conservadora quando submetida a sinais reais, normalmente selecionando uma ordem muito baixa.

3.4.2 Critério de Informação de Akaike

O critério AIC, proposto também por Akaike (1974), determina a ordem K do modelo minimizando uma função-custo obtida a partir de conceitos oriundos da teoria da informação. Supondo um processo AR com ruído branco gaussiano, esta função assume a seguinte forma:

$$AIC(K) = N \ln \left(\frac{RSS}{N} \right) + 2K, \quad (3.12)$$

em que o termo $2K$ representa uma função de tendência linear usada para penalização dos coeficientes AR excedentes, os quais não resultam na redução do erro quadrático de predição.

De acordo com Kashyap (1980), em função de N , os critérios AIC e FPE são assintoti-

¹O resíduo é o erro entre o valor real e o valor predito da variável sendo estimada.

camente equivalentes, apresentando o mesmo comportamento para sinais reais, já que a possibilidade de erro na escolha da ordem correta não tende a zero à medida que N aumenta. A tendência, então, é de subestimar a ordem dos dados à medida que aumenta número de amostras.

3.4.3 Critério de Informação Bayesiana

O critério BIC é uma outra estatística para seleção de modelo, também chamado do critério de informação de Schwarz (SIC), pela interpretação bayesiana dada a ele por (SCHWARZ, 1978). Matematicamente, este critério é descrito pela seguinte equação:

$$BIC(K) = N \ln \left(\frac{RSS}{N} \right) + K \ln N. \quad (3.13)$$

De acordo com a Equação (3.13), dados quaisquer dois modelos cujos parâmetros foram estimados, o modelo com o menor valor de BIC é aquele a ser selecionado. De modo semelhante ao critério AIC, o critério BIC é uma função decrescente de RSS , adicionada a uma função crescente de K , contudo o critério BIC penaliza mais os parâmetros excedentes do que o AIC.

3.4.4 Critério do Comprimento Mínimo de Descrição

O critério MDL é obtido a partir de uma variante da função-custo baseada em teoria da informação utilizada pelo critério AIC:

$$MDL(K) = N \ln \left(\frac{RSS}{N} \right) + \frac{K}{2} \ln N, \quad (3.14)$$

É importante destacar a semelhança entre os critérios BIC e MDL, contudo neste último o termo $\frac{K}{2} \ln N$ aumenta mais rápido com relação à N do que com K .

3.4.5 Critérios de Informação em Quantização Vetorial

Para tornar os critérios FPE, AIC, BIC MDL métricas de valia para esta dissertação, o valor RSS deve substituído pelo erro quadrático médio de quantização (EQMQ), definido como

$$MSQE(K) = \frac{1}{N} \sum_{t=1}^N \|\mathbf{x}(t) - \mathbf{w}_{i^*}(t)\|^2, \quad (3.15)$$

em que o parâmetro K passa a indicar o número de protótipos do modelo em vez da ordem do modelo AR.

Assim procedendo, espera-se encontrar o número ótimo de protótipos que resultem no número adequado de protótipos para a tarefa de interesse, principalmente tarefas de quantização vetorial em que o custo computacional do modelo é fator de grande relevância em aplicações práticas.

3.5 Conclusão

Este capítulo descreveu uma série de metodologias e técnicas comumente utilizadas para avaliar algoritmos de análise de agrupamentos e quantização vetorial. Estes métodos são utilizados, em última instância, para definir o número de protótipos a serem utilizados.

Em análise de agrupamentos, o número de protótipos é inferido principalmente a partir de métricas que avaliam o grau de separabilidade e coesão entre os agrupamentos, tais como os índices Dunn e DB (HALKIDI et al., 2001). Em tarefas de quantização vetorial, por outro lado, o número de protótipos é definido em função de outras métricas; por exemplo, o erro quadrático médio de quantização. Por fim, foram descritos critérios que levam em consideração a complexidade (número de parâmetros) do algoritmo, os chamados critérios de informação, com o intuito de utilizá-los para determinar o número ótimo de protótipos em aplicações de quantização vetorial.

No próximo capítulo, as várias redes neurais competitivas descritas no Capítulo 2, assim como o algoritmo K -médias seqüencial, serão avaliados em uma tarefa de análise de agrupamentos pelos vários critérios descritos neste capítulo.

4 Resultados - Análise de Agrupamentos

4.1 Introdução

Este capítulo traz uma série de experimentos computacionais relativos à utilização dos algoritmos descritos no Capítulo 2 em tarefas de análise de agrupamentos. O propósito principal destas simulações é avaliar os agrupamentos gerados usando os vários critérios de validação de agrupamentos estudados no Capítulo 3 e como os resultados são afetados por parâmetros próprios de cada algoritmo.

Os resultados para a validação de agrupamentos apresentados neste trabalho encontram-se divididos em quatro seções. A Seção 4.2 apresenta os agrupamentos resultantes da execução dos algoritmos competitivos utilizando um conjunto de dados gerado artificialmente. Os resultados obtidos da aplicação de alguns índices relativos estão agrupados na Seção 4.4 e os melhores agrupamentos por algoritmo segundo tais índices estão na Seção 4.5. Para evitar uma sobrecarga de informação, gráficos referentes às curvas do erro de quantização em função da época de treinamento são mostrados apenas no Apêndice A.

4.2 Metodologia de Simulação

Um conjunto de dados artificiais¹ foi especialmente construído pelo Prof. Guilherme de Alencar Barreto para testar os vários algoritmos de geração e validação de agrupamentos apresentados nesta dissertação (ver Figura 4.1).

Em uma rápida análise visual é possível perceber que estes dados estão organizados em quatro agrupamentos. Embora artificial, este conjunto de dados possui uma série de características importantes que permite avaliar com fidelidade o poder computacional de tais algoritmos. Dentre as principais características podem ser destacadas as seguintes:

¹Este conjunto de dados está disponível em: www.deti.ufc.br/~guilherme/Codes/dataset1.dat.

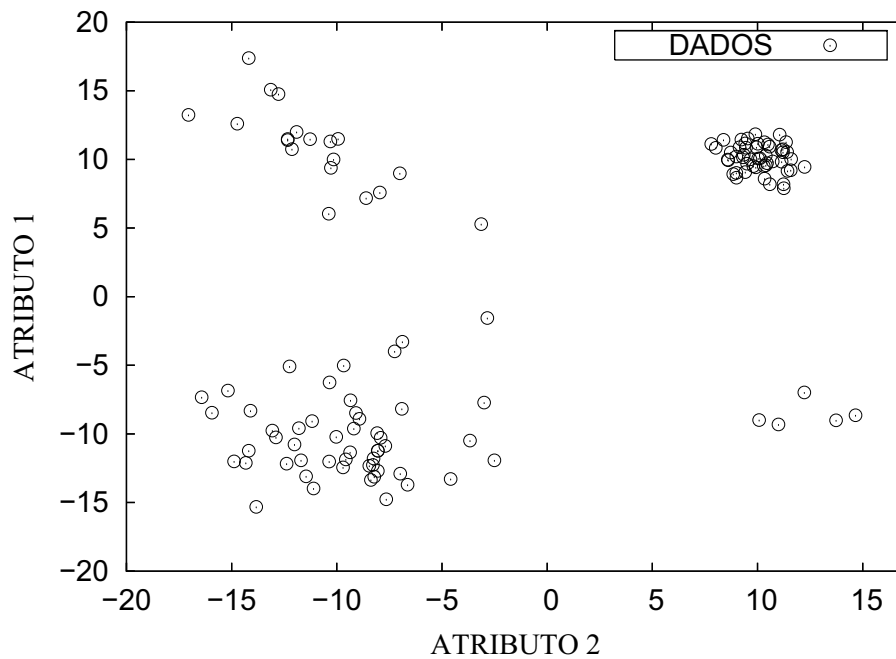


Figura 4.1: Conjunto de dados experimentais.

1. A baixa dimensionalidade dos dados (cada vetor pertence ao \mathbb{R}^2) permite uma imediata avaliação da posição em que cada protótipo de um certo algoritmo foi posicionado após o treinamento.
2. Os grupos possuem densidade bem diversas. É possível notar que alguns grupos possuem vários elementos, enquanto outros possuem muito poucos elementos.
3. A forma dos agrupamentos não é circular. Vários dos algoritmos de validação estudados no capítulo anterior têm melhor desempenho em agrupamentos hiperesféricos.
4. A variabilidade (espalhamento) dos agrupamentos também é bastante diferente dependendo do agrupamento.
5. A correlação entre os atributos é diferente, dependendo do agrupamento.

Em especial, as quatro últimas características do conjunto de dados artificiais permitem uma avaliação mais rápida e clara do desempenho de cada algoritmo. Permite também responder a uma série de perguntas, por exemplo: como determinado algoritmo posiciona seus protótipos em função do número de protótipos disponíveis? Qual fator tem maior influência nesse posicionamento: a densidade dos grupos, a variabilidade ou a forma deles? Como determinado parâmetro afeta o desempenho do algoritmo?

A fim de avaliar os vários algoritmos de agrupamento da maneira mais justa possível foram estabelecidos os seguintes procedimentos:

1. Um conjunto de K vetores é selecionado aleatoriamente a partir do próprio conjunto de dados. Estes K vetores são usados para iniciar os K protótipos de um determinado algoritmo.
2. Os algoritmos são treinados por um número fixo de épocas, salvo indicação contrária. Este valor foi selecionado como suficiente após alguma experimentação com todos os algoritmos de análise de agrupamentos.
3. Cada valor de um índice ou métrica mostrado em gráficos ou tabelas correspondem à média de 20 (vinte) realizações de treinamento, repetidas sob as mesmas condições.
4. A taxa de aprendizagem de todos os algoritmos neurais decai exponencialmente de um valor máximo de 0,5 a um valor mínimo de 0,0001.
5. A cada época a ordem de apresentação dos dados é aleatória, a fim de evitar qualquer tipo de viés durante a aprendizagem.
6. Para simular a rede SOM foram escolhidas duas topologias: uma bidimensional (2D) com vizinhança retangular e unidimensional (1D). Para os dois casos, a função de vizinhança gaussiana.
7. O decaimento com o tempo das funções de vizinhança das rede SOM e Neural-Gas é exponencial.
8. Os parâmetros da função de vizinhança da rede SOM são $\sigma_0 =$ menor inteiro maior que $K/2$ e $\sigma_T = 0,0001$.
9. Os parâmetros da função de vizinhança da rede Neural-Gas são $\lambda_0 = K$ e $\lambda_T = 0,01$.
10. As posições (coordenadas) dos dados nas figuras são denotados por círculos abertos (\circ) e as posições dos protótipos são denotadas por triângulos (\triangle).

A fim de automatizar os procedimentos de simulação foi desenvolvido um aplicativo chamado *Java Neural Network - Competitive Learning*. Veja com maiores detalhes seu funcionamento no Apêndice B. Ele também foi utilizado para simulações de quantização vetorial.

A partir da próxima seção são mostrados os resultados para cada um dos sete algoritmos de análise de agrupamentos estudados nesta dissertação: K -médias, rede WTA, rede SOM,

rede Neural-Gas, rede RPCL, rede FSCL e rede FuzzyCL. Optou-se por mostrar inicialmente os resultados de uma avaliação qualitativa dos resultados do agrupamento. Em seguida, são mostrados os resultados quantitativos frutos da aplicação dos índices Dunn e Davies-Bouldin.

4.2.1 Resultados - Rede WTA

A Figura 4.2 mostra o posicionamento para dois protótipos após treinamento da rede WTA. Uma rápida análise desta figura permite notar que $K = 2$ protótipos é um número insuficiente para representar adequadamente os dados. Como era de se esperar, o principal fator de atração de protótipos é a densidade de dados, ou seja, os protótipos foram posicionados juntos a agrupamentos com maior número de dados (agrupamento na extremidade superior direita e agrupamento na extremidade inferior esquerda).

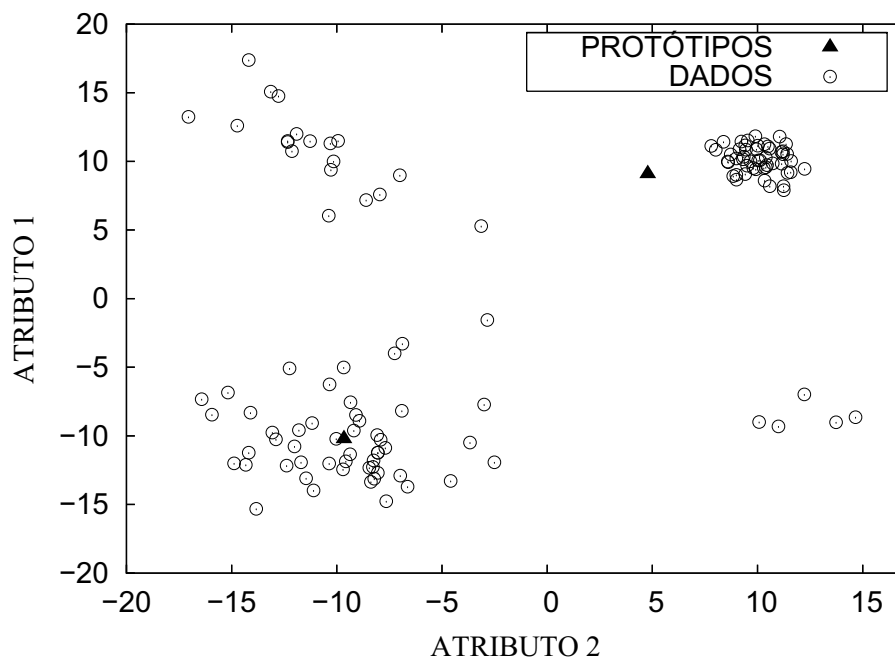


Figura 4.2: Posicionamento de $K = 2$ protótipos da rede WTA após 50 épocas.

A Figura 4.3 mostra o posicionamento para dois protótipos após treinamento da rede WTA. Nota-se que o terceiro protótipo foi posicionado no agrupamento com o terceiro maior número de dados. No entanto, o número de protótipos ainda é insuficiente, pois existem agrupamentos que ainda não foram representados adequadamente, como aquele contendo poucos dados no canto inferior direito.

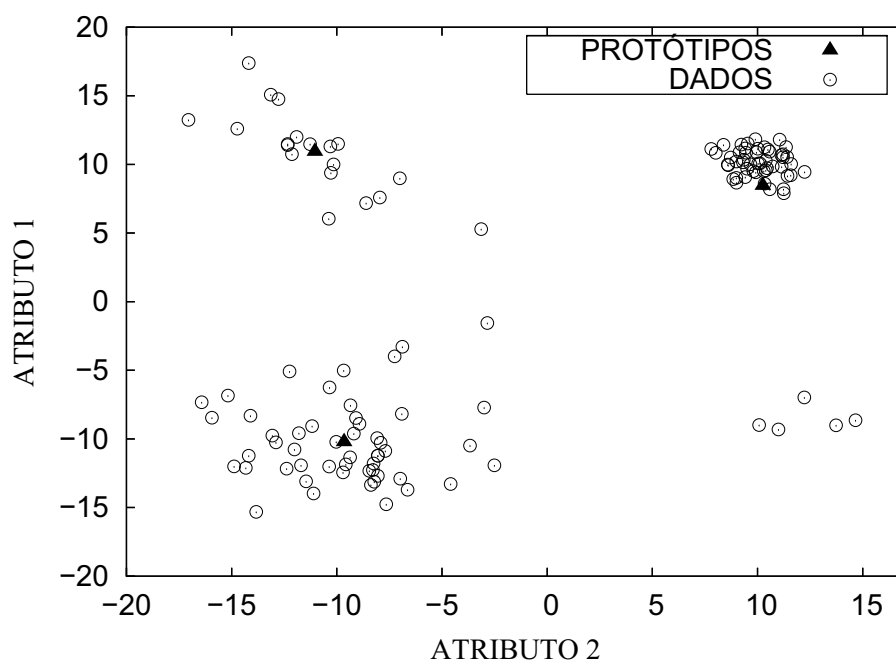


Figura 4.3: Posicionamento de $K = 3$ protótipos da rede WTA após 50 épocas.

A Figura 4.4 mostra o posicionamento para quatro protótipos após treinamento da rede WTA. Nota-se que os quatro principais grupos de dados estão representados adequadamente, com um protótipo cada.

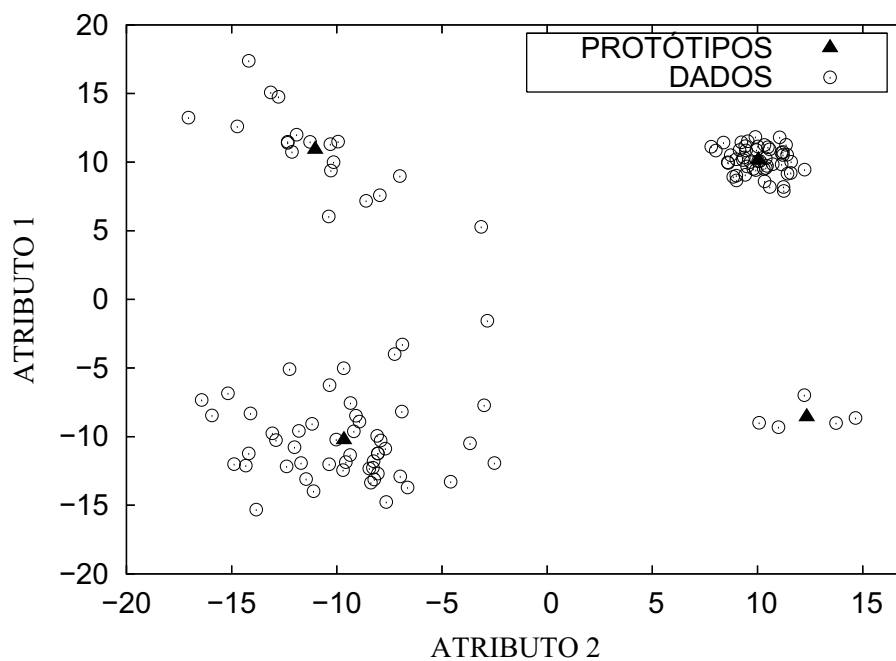


Figura 4.4: Posicionamento de $K = 4$ protótipos da rede WTA após 50 épocas.

A Figura 4.5 mostra o posicionamento para cinco protótipos após treinamento da rede WTA. Aqui acontece um fenômeno típico de algoritmos de análise de agrupamentos. Mesmo que quatro protótipos sejam aparentemente suficientes para representar o conjunto de dados, sempre que for disponibilizado um protótipo adicional, este vai ser alocado para algum subgrupo de dados seguindo um critério específico (e.g. minimização do erro de quantização). Ou seja, o algoritmo passa a subdividir os agrupamentos já existentes. Percebe-se que o protótipo adicional foi alocado para o agrupamento com maior número de dados e de maior dispersão. Em outras palavras, o algoritmo dividiu o maior agrupamento em dois, a fim de representá-lo melhor.

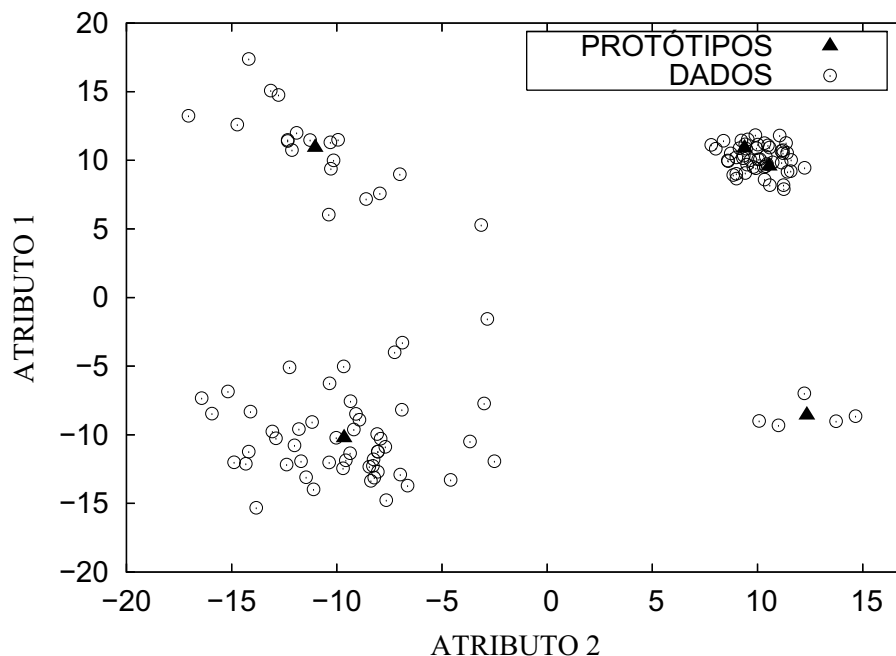


Figura 4.5: Posicionamento de $K = 5$ protótipos da rede WTA após 50 épocas.

4.2.2 Resultados - Rede FSCL

A Figura 4.6 mostra o posicionamento para $K = 4$ protótipos após treinamento da rede FSCL com $z = 0,1$. O resultado obtido é bastante similar ao da rede WTA. Porém, a rede FSCL possui um parâmetro adicional (z) que regula o número de vitórias por protótipo durante o treinamento. O efeito deste parâmetro no desempenho do algoritmo está na tendência dos

protótipos se deslocarem para os agrupamentos de maior densidade, quanto maior o valor de z maior é este efeito.

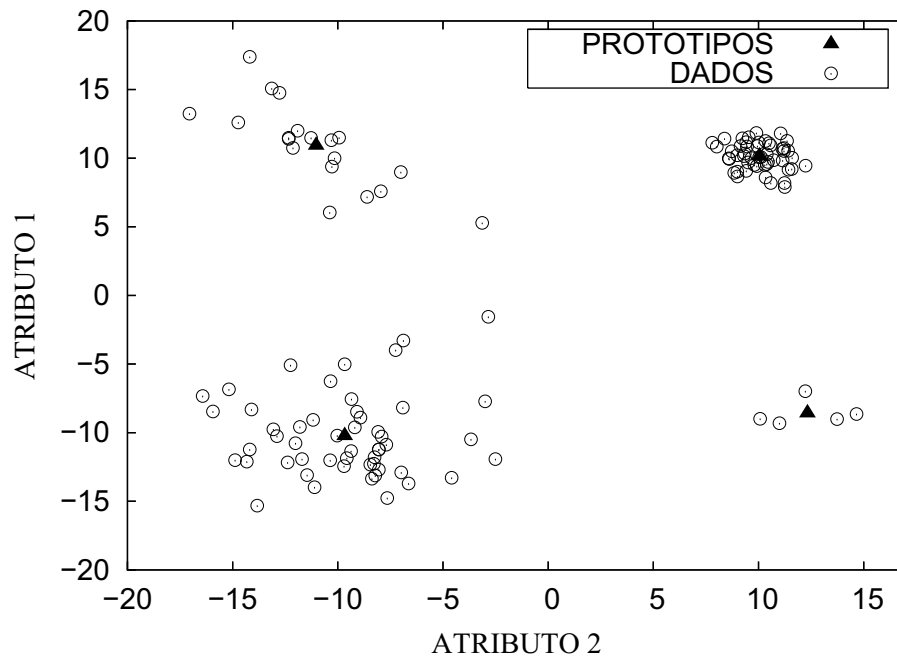


Figura 4.6: Posicionamento de $K = 4$ protótipos da rede FSCL após 50 épocas ($z = 0, 1$).

A Figura 4.7 mostra o posicionamento para $K = 4$ protótipos após treinamento da rede FSCL com $z = 3$. Nota-se que o protótipo que representava o menor grupo (canto inferior direito) foi movido em direção ao protótipo do maior grupo (canto inferior esquerdo). A interpretação deste resultado é bem interessante.

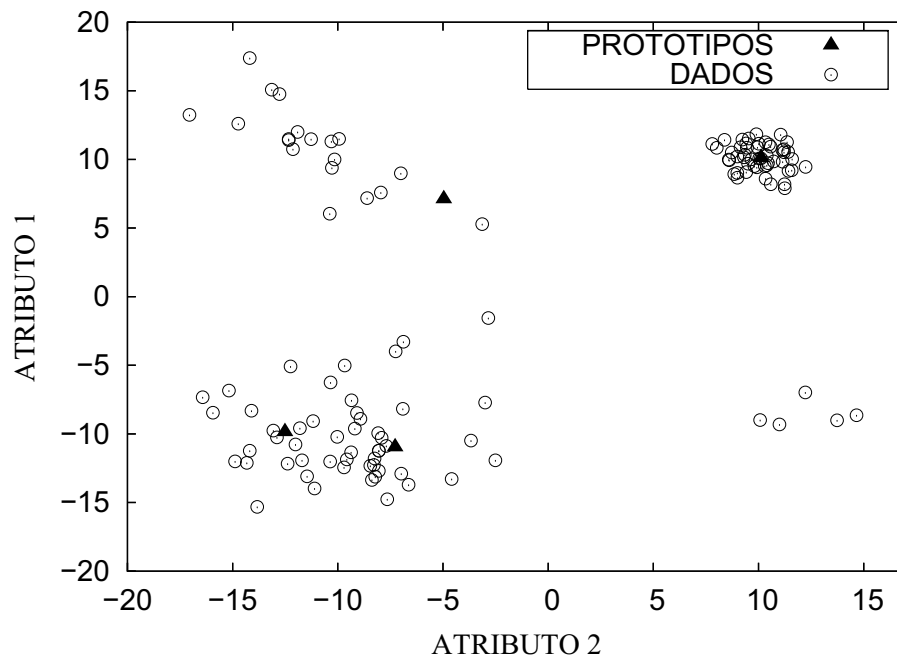


Figura 4.7: Posicionamento de $K = 4$ protótipos da rede FSCL após 50 épocas ($z = 3$).

Como o maior grupo tem mais elementos, o protótipo que for alocado para ele vai obviamente ter uma probabilidade maior de vitórias ao longo do treinamento. Como a rede FSCL visa justamente equalizar o número de vitórias por protótipo, o protótipo inicialmente alocado para o maior grupo vai deixando de ser selecionado vencedor e seus pesos vão deixando de ser ajustados. Em compensação, o segundo protótipo mais próximo do maior grupo, que é o protótipo alocado para o menor grupo, vai começar a ser selecionado vencedor para dados do maior grupo e seus pesos vão sendo atualizados na direção deste grupo. Como o menor grupo não tem um poder de atração muito elevado (possui muito menos dados que o maior grupo) o protótipo do menor grupo vai sendo movido para o maior grupo.

Do exposto no parágrafo anterior, pode-se concluir que um valor de z muito elevado tende a favorecer um maior equilíbrio (uniformização) do número de vitórias por protótipo, em detrimento de um melhor posicionamento dos protótipos. Assim, o ideal é usar um valor de z pequeno para este conjunto de dados.

4.2.3 Resultados - Rede RPCL

A Figura 4.8 mostra o posicionamento de $K = 4$ protótipos após treinamento da rede RPCL usando $\gamma = 0,05$, melhor resultado obtido para esta rede. Mesmo assim, o protótipo alocado

para o menor grupo não está exatamente posicionado sobre o centróide deste grupo.

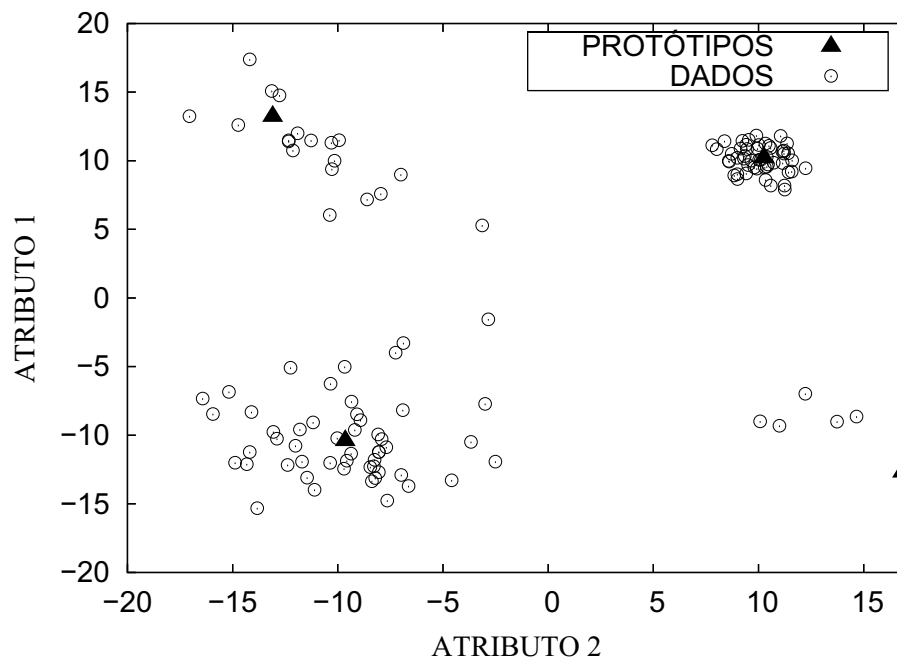
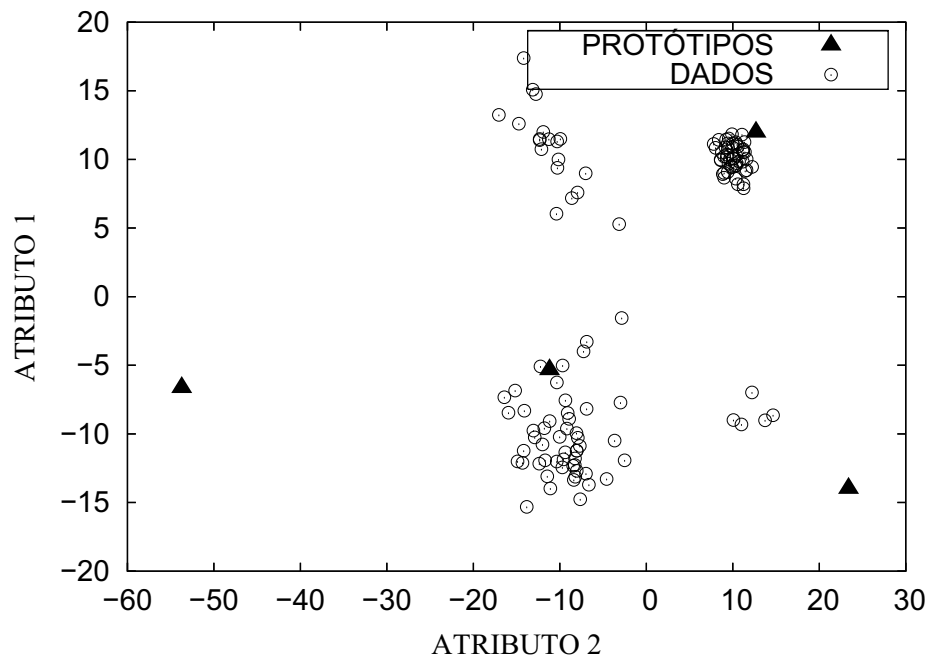
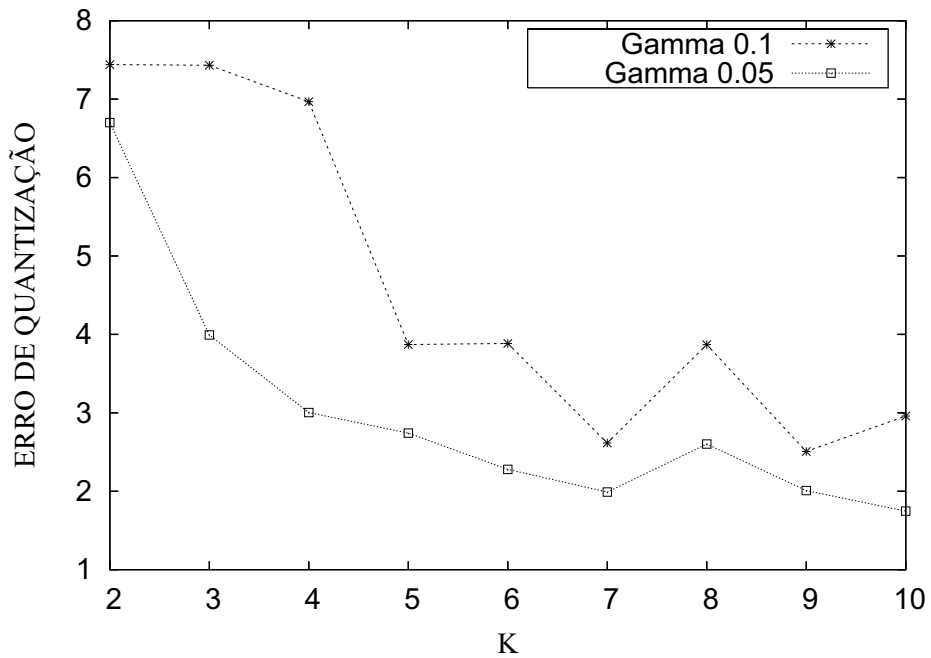


Figura 4.8: Posicionamento de $K = 4$ protótipos da rede RPCL ($\gamma = 0,05$).

A Figura 4.9 mostra o posicionamento de $K = 4$ protótipos após treinamento da rede RPCL usando $\gamma = 0,1$. Nota-se que para um valor de γ maior os protótipos tendem a se afastar mais um do outro. Nestas condições o algoritmo tende a errar muito (alto erro de quantização), pois penaliza protótipos importantes (ver Figura 4.10).

Figura 4.9: Posicionamento de $K = 4$ protótipos da rede RPCL ($\gamma = 0, 1$).Figura 4.10: Erro médio de quantização da rede RPCL em função de K para $\gamma = 0,05$ e $0,1$.

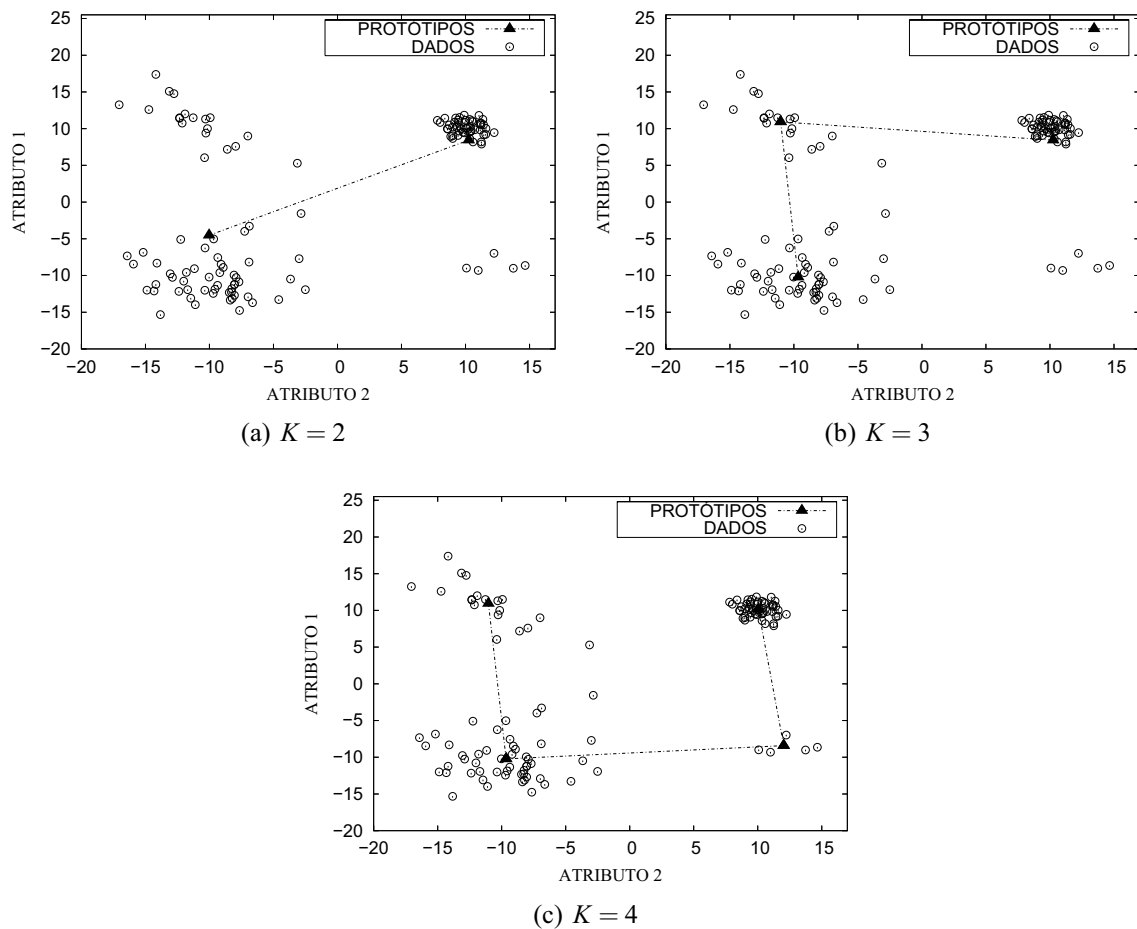


Figura 4.11: Posicionamento de $K = 2, 3$ e 4 protótipos da rede SOM-1D.

4.2.4 Resultados - Rede SOM

Na rede SOM os neurônios estão dispostos em arranjos geométricos fixo, i.e. em linha (SOM-1D) ou em grade (SOM-2D). Assim, além de espalhar os protótipos a fim de obter um erro de quantização mínimo, a rede SOM também tenta fazer com que os protótipos de neurônios que são próximos no arranjo geométrico também estejam próximos no espaço de entrada. A esta propriedade dá-se o nome de preservação da topologia. Assim, duas “forças” regulam o comportamento da rede SOM, uma que tenta minimizar o erro de quantização, enquanto a outra age tentando manter os protótipos ordenados segundo a vizinhança espacial dos neurônios no arranjo da rede. Nas figuras a seguir, os protótipos cujos neurônios são vizinhos imediatos no arranjo geométrico da rede são conectados por uma linha tracejada. O ideal é que não haja cruzamento entre as linhas, indicando uma falha topológica, o que equivale a um mínimo local do erro de quantização (BARRETO, 2003).

SOM-1D - Após treinamento adequado, as posições dos protótipos para $K = 2, 3$ e 4 são mostradas na Figura 4.11, enquanto as posições dos protótipos para $K = 5, 9, 16$ e 25 estão

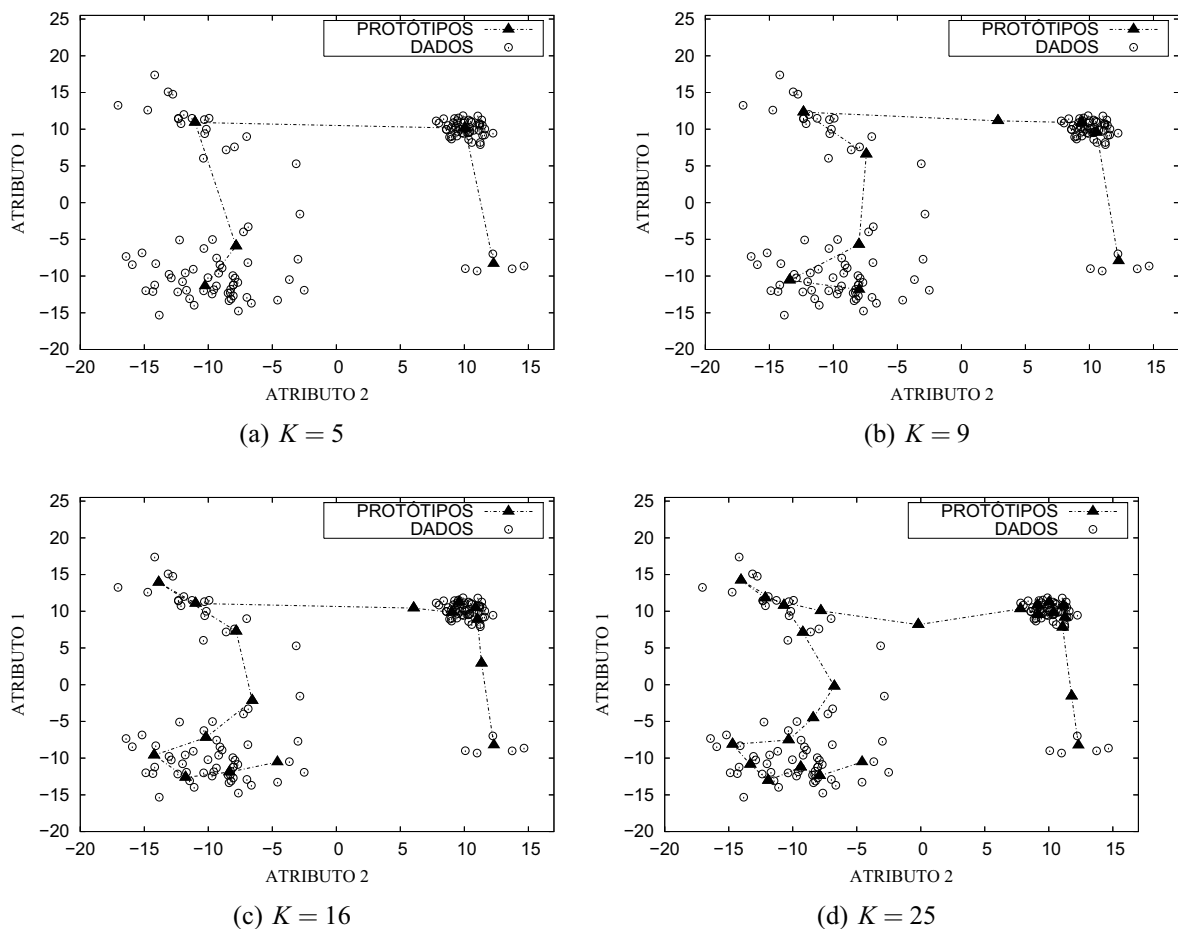


Figura 4.12: Posicionamento de $K = 5, 9, 16$ e 25 protótipos da rede SOM-1D.

mostradas na Figura 4.12. Assim como observado para os outros algoritmos de análise de agrupamentos, os protótipos tendem a ficar mais próximos das regiões de maior concentração de dados.

É importante destacar que para todos os casos mostrados a rede SOM-1D foi capaz de preservar a topologia dos protótipos, contudo para $K = 9, 16$ e 25 alguns protótipos são posicionados onde não há dados. Isto sempre vai ocorrer com a rede SOM para um grande número de protótipos, independente do arranjo geométrico dos neurônios da rede (SOM-1D ou SOM-2D), pois esta é uma característica própria da rede SOM. Esta propriedade é útil, por exemplo, em visualização de dados de alta dimensionalidade, mas não é útil em análise de agrupamentos. É importante lembrar que a rede SOM não foi projetada originalmente como um algoritmo de análise de agrupamentos, mas sim como um modelo da organização dos neurônios em mapas corticais sensorio-motores. Mesmo assim, quando o número de protótipos é baixo e algum critério de validação de agrupamentos for utilizado corretamente a rede SOM pode também ser usada para análise de agrupamentos, conforme será visto nas simulações seguintes.

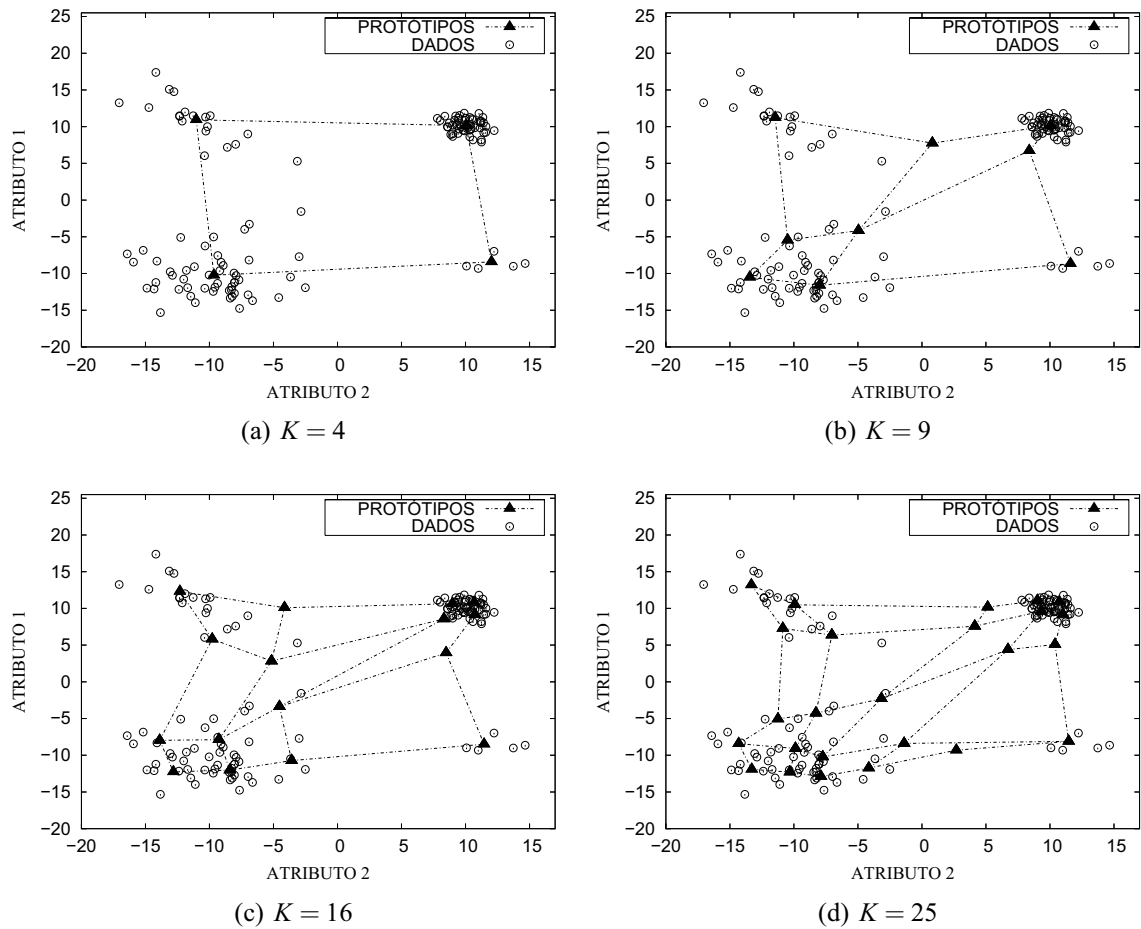
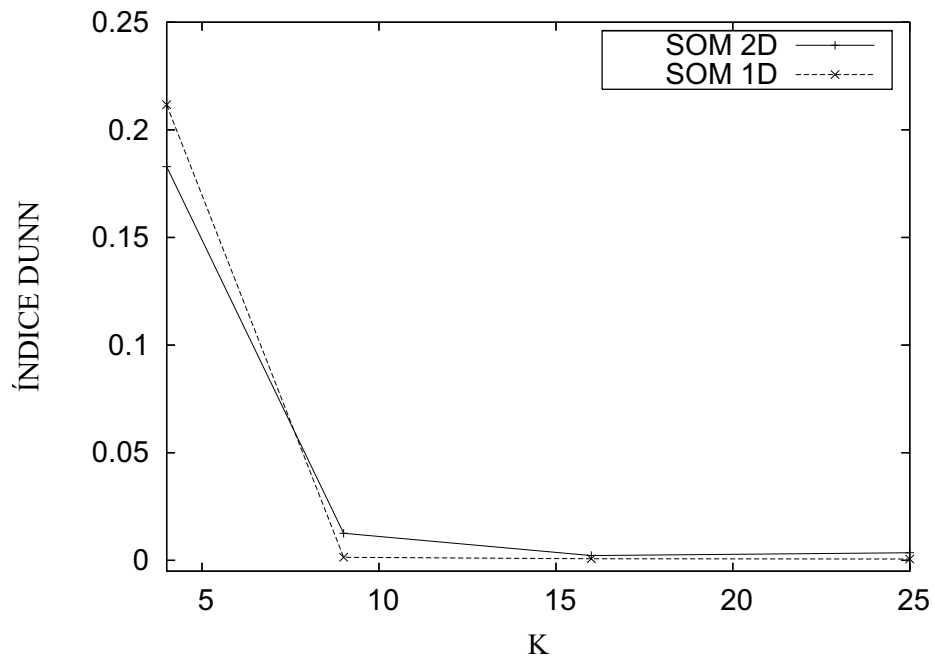


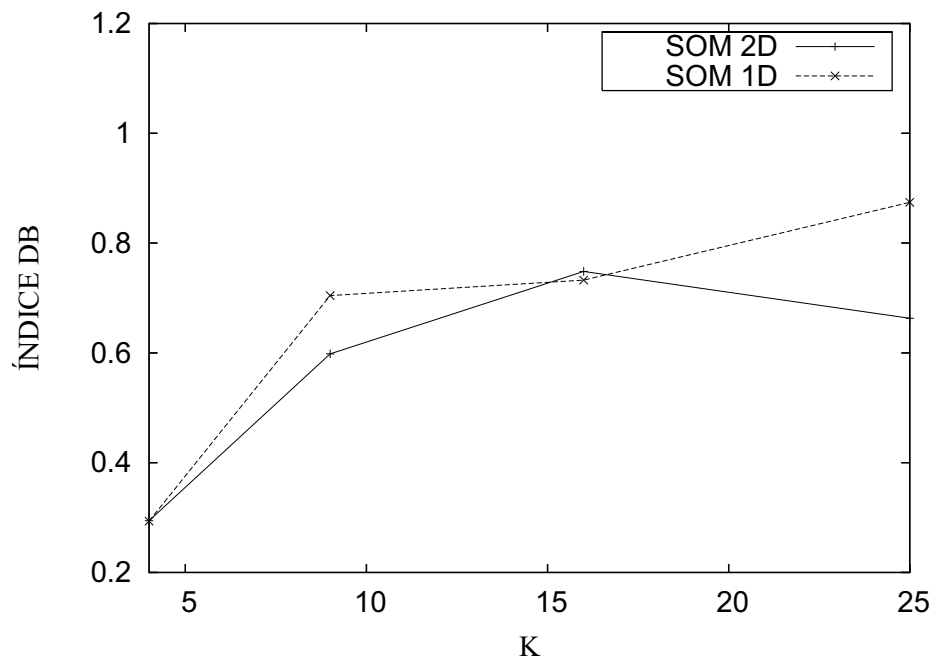
Figura 4.13: Posicionamento de $K = 4, 9, 16$ e 25 protótipos da rede SOM-2D.

SOM-2D - Após treinamento adequado, as posições dos protótipos para $K = 4, 9, 16$ e 25 são mostradas na Figura 4.13. Note que para todos os valores de $K > 4$ a rede SOM-2D posiciona alguns protótipos em regiões onde não há dados com o intuito de preservar a organização topológica dos protótipos. Com $K = 4$, o resultado é equivalente ao da rede SOM-1D.

A Figura 4.14 mostra os gráficos dos índices Dunn e DB para $K = 4, 9, 16$ e 25 protótipos da rede SOM, para as ambas as topologias, uni- e bidimensional. Note que foram simulados apenas os casos em que K é quadrado. Verificou-se que para as duas topologias os índices selecionaram o agrupamento com para $K = 4$ protótipos. Este resultado confirma que a rede SOM pode ser usada para análise de agrupamentos, desde que critérios adequados de validação sejam utilizados.



(a) Índice Dunn



(b) Índice DB

Figura 4.14: Os gráficos dos índices Dunn e DB para $K = 4, 9, 16$ e 25 protótipos das redes SOM-1D e SOM-2D.

É importante avaliar também a influência da topologia no erro de quantização. As curvas dos erros de quantização em função da época de treinamento, mostradas na Figura 4.15 para o

caso em que $K = 4$ protótipos (ou seja, o número de protótipos indicado pelos índices acima para as duas topologias). No início do treinamento, a rede SOM-1D tem uma convergência mais rápida, pois usa curva decaí mais acentuadamente. Porém, ao final do treinamento, as duas curvas estabilizam no mesmo patamar de erro. Daí, é possível concluir que esta métrica sozinha não é adequada para validar agrupamentos da rede SOM em redes SOM com topologias distintas.

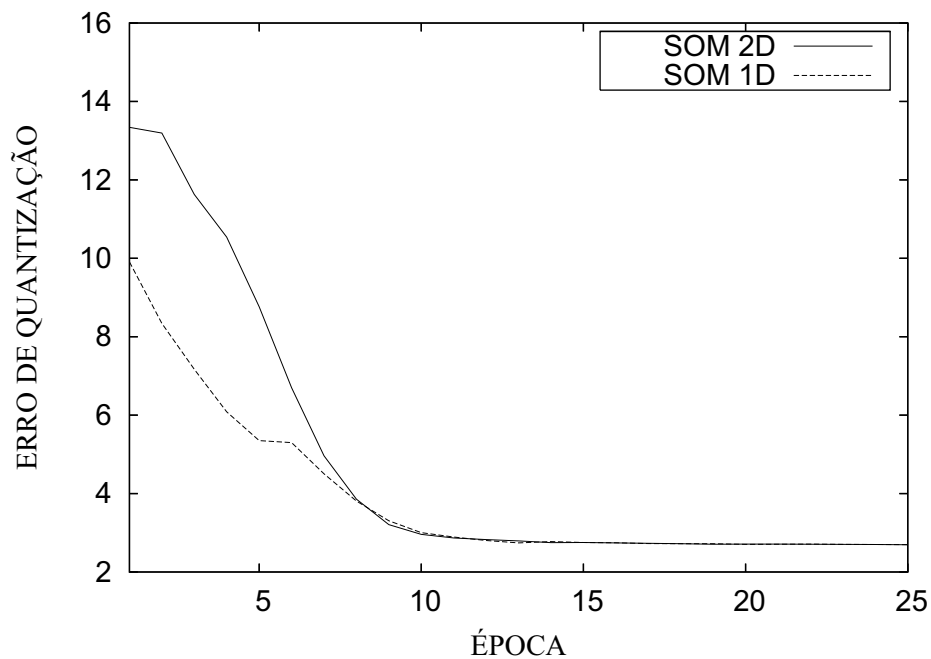


Figura 4.15: Curvas do erro médio de quantização da rede SOM para topogias 1D e 2D com $K = 4$ protótipos.

4.2.5 Resultados - Algoritmo K -médias

A Figura 4.16 mostra o posicionamento de dois protótipos do algoritmo K -médias após o treinamento. O resultado é similar ao da rede WTA, porém com um posicionamento menos adequado do protótipo (i.e. um pouco afastado do centróide do grupo) alocado para o grupo do canto superior direito.

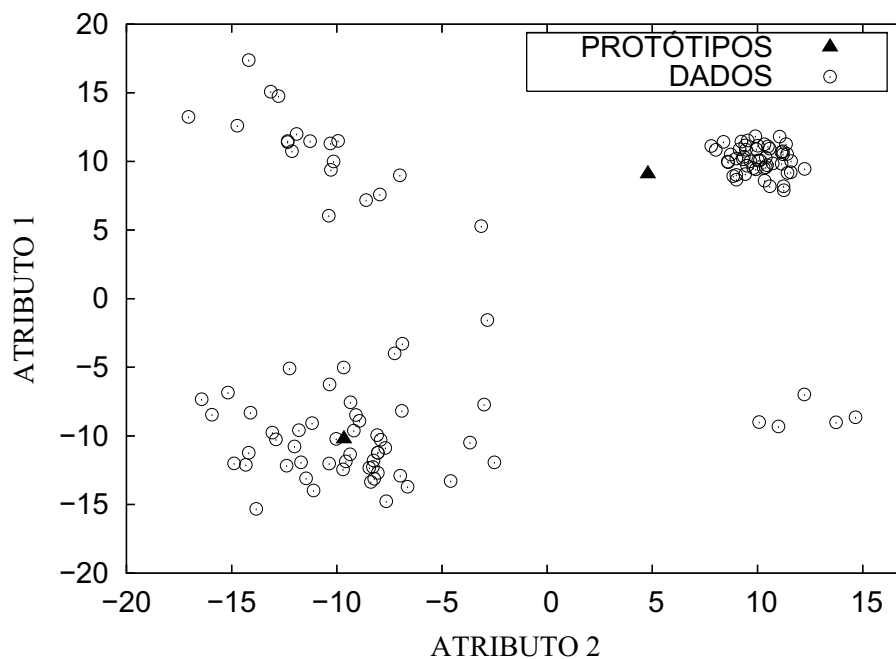


Figura 4.16: Posicionamento de $K = 2$ protótipos do algoritmo K -médias após treinamento.

O posicionamento de três protótipos do algoritmo K -médias após cinquenta épocas de treinamento é mostrado na Figura 4.17.

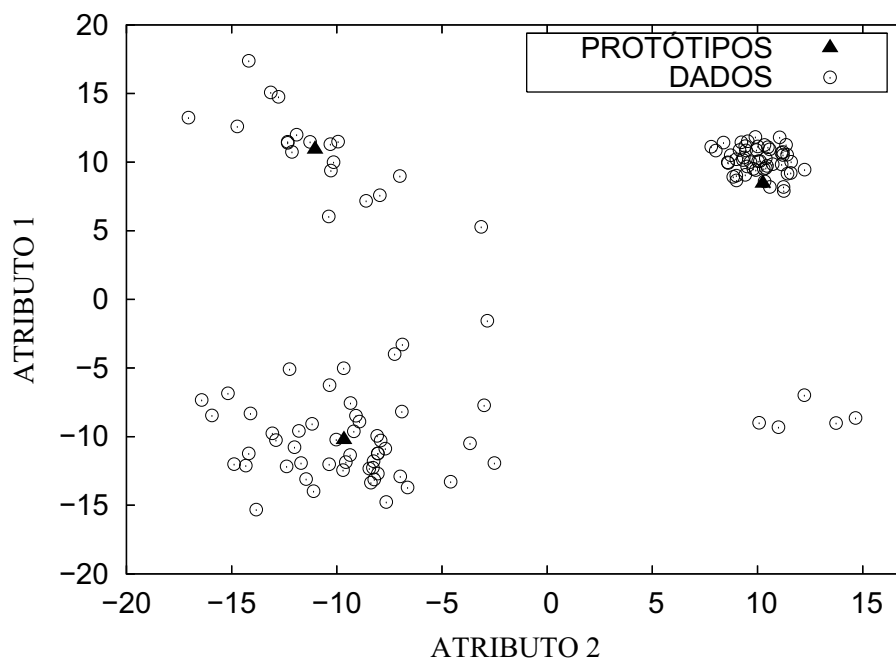


Figura 4.17: Posicionamento de $K = 3$ protótipos do algoritmo K -médias após treinamento.

A Figura 4.18 mostra o posicionamento de quatro protótipos do algoritmo K -médias após cinquenta épocas de treinamento.

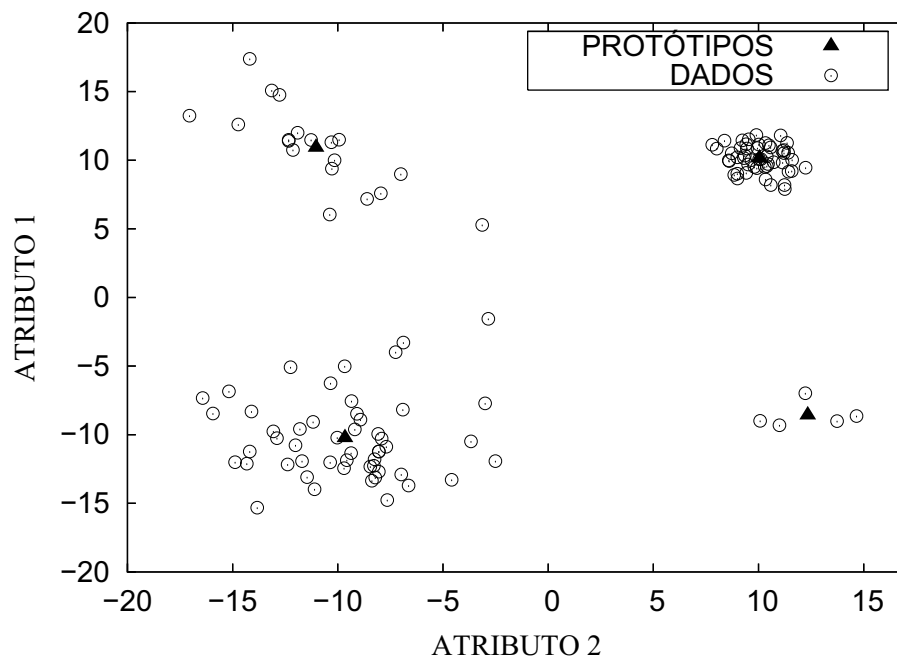


Figura 4.18: Posicionamento de $K = 4$ protótipos do algoritmo K -médias após treinamento.

O posicionamento para cinco protótipos após cinquenta épocas de treinamento usando a rede K -médias é mostrado na Figura 4.19. É importante destacar que esta figura mostra uma solução subótima (mínimo local) em que o quinto protótipo é alocado para o agrupamento no canto superior esquerdo, em vez de ir para o agrupamento que possui mais dados (canto inferior esquerdo). Esta solução ocorre eventualmente em função da iniciação dos pesos. Este comportamento demonstra que mesmo algoritmos cuja a teoria é bem-definida, sofrem com as restrições práticas, tais como iniciação dos pesos e tamanho limitado da amostra de dados.

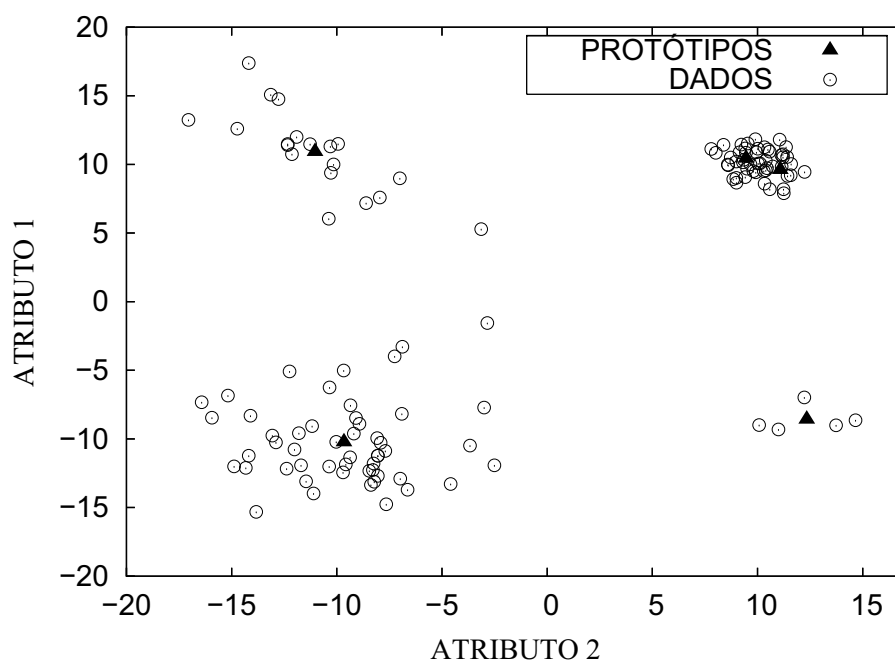


Figura 4.19: Posicionamento de $K = 5$ protótipos do algoritmo K -médias após treinamento.

O comportamento do algoritmo K -médias é muito similar ao da rede WTA em virtude da semelhança entre os dois algoritmos de análise de agrupamentos.

4.2.6 Resultados - Rede FuzzyCL

A Figura 4.20 mostra o posicionamento de quatro protótipos da rede FuzzyCL após cinquenta épocas de treinamento, usando o valor $z = 1, 1$.

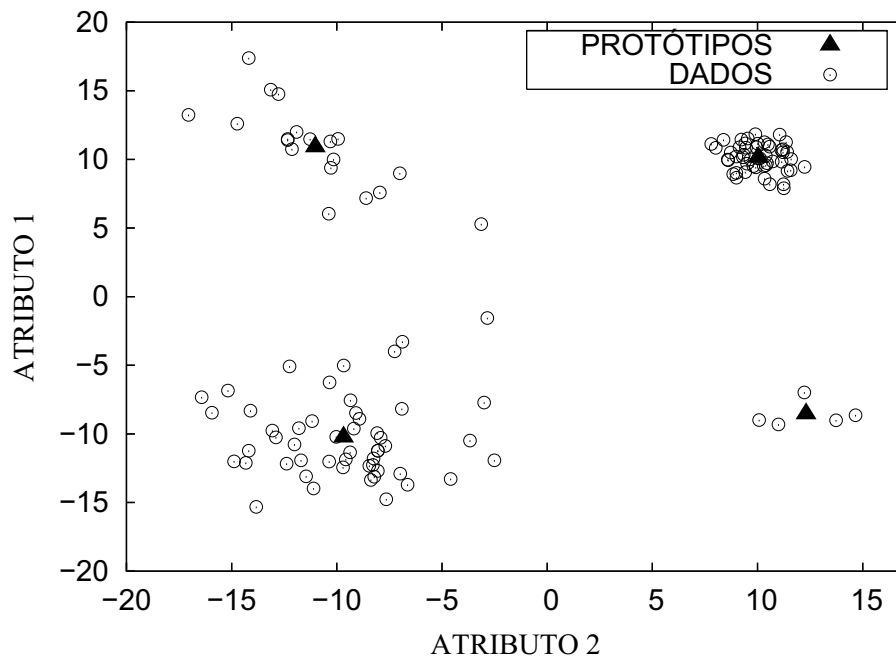


Figura 4.20: Posicionamento de $K = 4$ protótipos da rede FuzzyCL após treinamento ($z = 1, 1$).

O posicionamento para quatro protótipos da rede FuzzyCL após cinquenta épocas de treinamento usando $z = 2$.

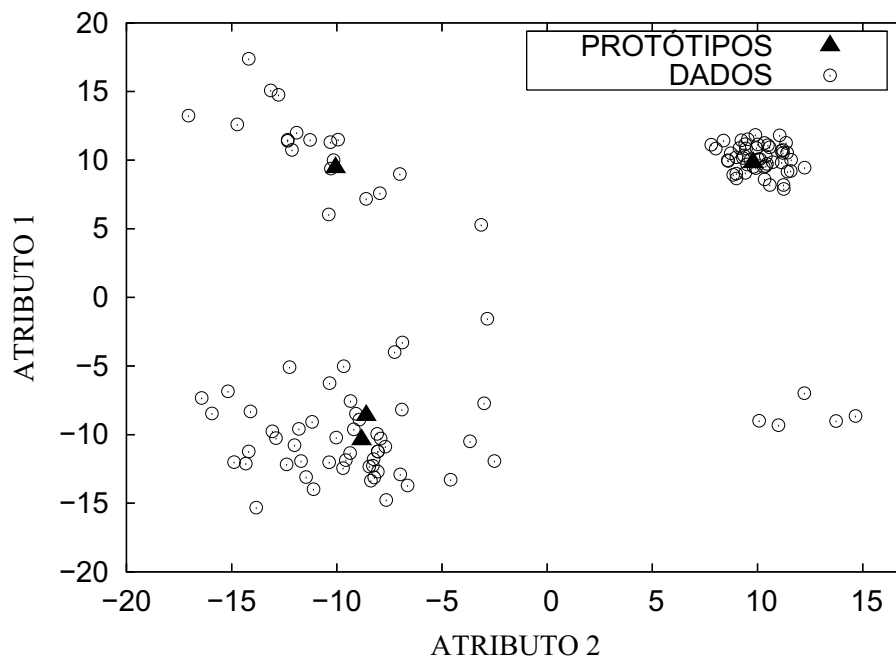


Figura 4.21: Posicionamento de $K = 4$ protótipos da rede FuzzyCL após treinamento ($z = 2$).

Comparando as duas figuras, nota-se que quando se usa um valor para z próximo de 1, os protótipos tendem a se posicionar próximo ao centróide dos agrupamentos. Quando o valor de z cresce, o comportamento se assemelha bastante àquele da rede FSCL, ou seja, o protótipo alocada para o menor dos agrupamentos é movido para o maior agrupamento.

4.2.7 Resultados - Rede Neural-Gas

A Figura 4.22 mostra o posicionamento de dois protótipos da rede Neural-Gas após cinquenta épocas de treinamento. Nota-se que os protótipos são posicionados próximos aos agrupamentos de modo muito semelhante ao algoritmo WTA.

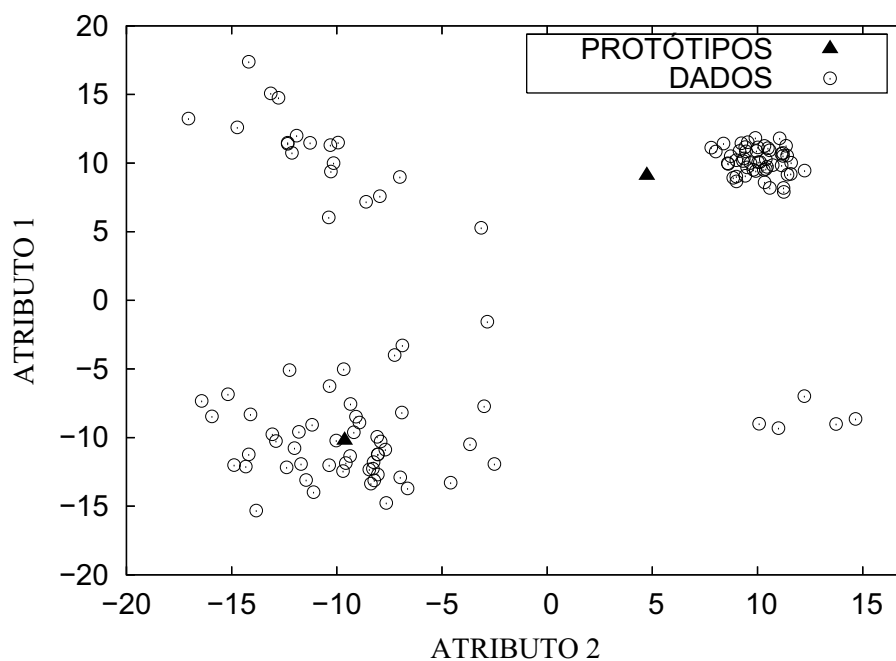


Figura 4.22: Posicionamento de $K = 2$ protótipos da rede Neural-Gas após treinamento.

O posicionamento de três protótipos da rede Neural-Gas após cinquenta épocas de treinamento é mostrado na Figura 4.23. O resultado é muito similar aos das redes WTA e SOM e também ao do algoritmo K -médias.

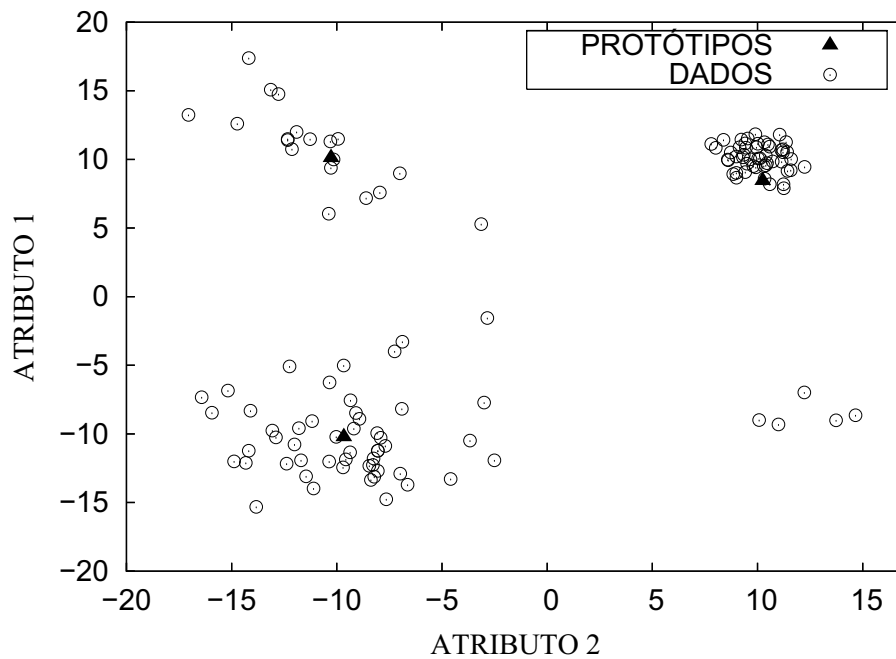


Figura 4.23: Posicionamento de $K = 3$ protótipos da rede Neural-Gas após treinamento.

A Figura 4.24 mostra o posicionamento de quatro protótipos da rede Neural-Gas após o treinamento. A solução obtida é similar àquelas geradas pelas outras redes e algoritmos.

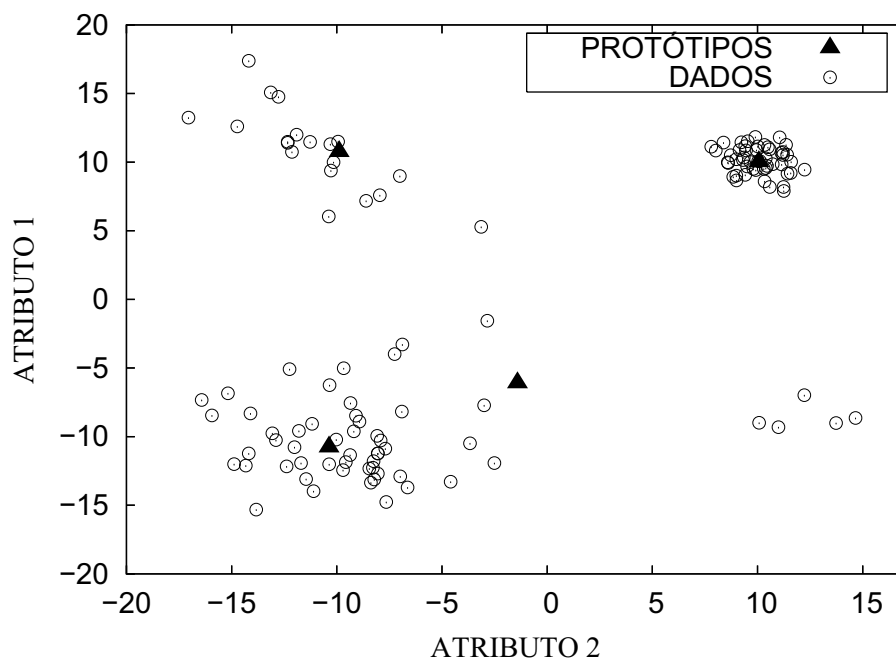


Figura 4.24: Posicionamento de $K = 4$ protótipos da rede Neural-Gas após treinamento.

O posicionamento para cinco protótipos da rede Neural-Gas treinada é mostrado na Figura 4.25. Esta solução é bastante diferente daquelas obtidas para as demais redes; mesmo assim, obedece à lógica de distribuir os protótipos em função da densidade de dados.

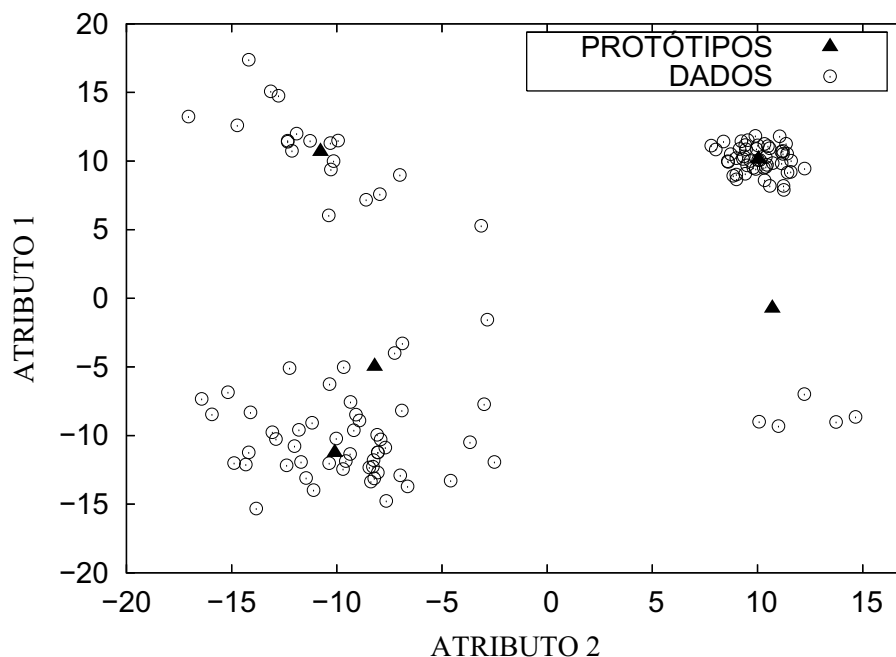


Figura 4.25: Posicionamento de $K = 5$ protótipos da rede Neural-Gas após treinamento.

Como conclusão geral desta seção, pode-se afirmar que para $K = 4$ protótipos, o que corresponde ao número ótimo do problema de análise de agrupamentos em estudo, todos os algoritmos apresentam solução equivalente, com pequenas variações que podem ser explicadas pelas condições iniciais diferentes. Quando o número de protótipos é diferente do ótimo, ou seja $K \neq 4$, as soluções de posicionamento de protótipos entre as redes pode diferir bastante, cada uma seguindo um princípio teórico diferente.

Na próxima seção, a validação dos agrupamentos é feita de forma mais quantitativa através da utilização de índices apropriados para este fim, tais como os índices relativos Dunn e Davies-Bouldin.

4.3 Validação de Agrupamentos via Erro de Quantização

Embora não seja a métrica mais usada para este fim, o erro médio de quantização pode ser usado para uma avaliação inicial do número agrupamentos existentes nos dados. Como este tipo de métrica tem uma tendência de sempre diminuir com o aumento do número de protótipos, o número de protótipos seria àquele correspondente ao “joelho” da curva, ou seja, seu ponto de maior curvatura.

A Figura 4.26 mostra as curvas dos erros de quantização para cada um dos algoritmos estudados neste capítulo. O joelho de quase todas as curvas coincide com o valor $K = 4$, mas o mais correto seria afirmar que o número de agrupamentos está na faixa de $3 \leq K \leq 5$.

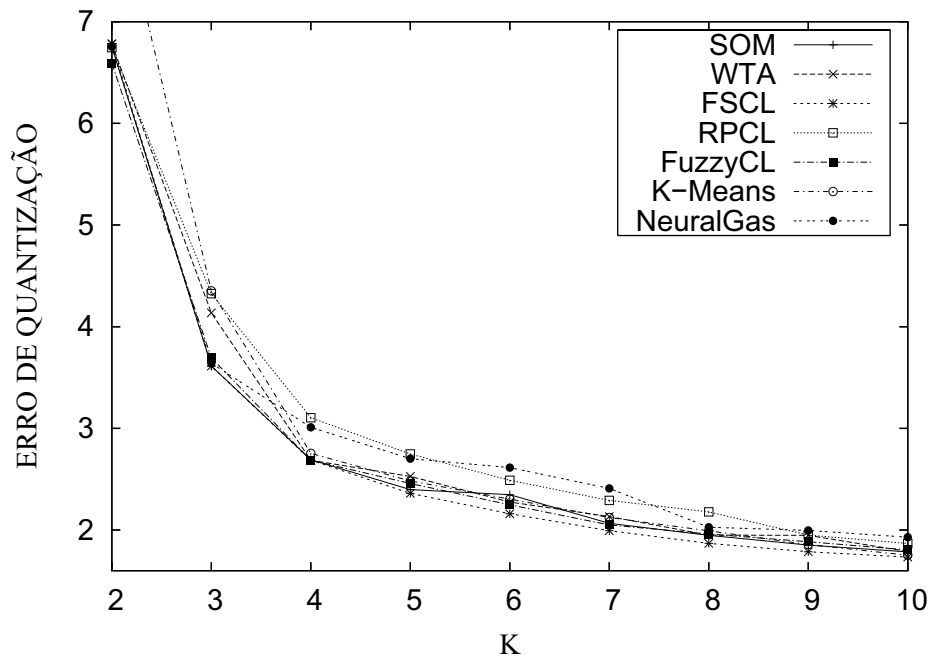


Figura 4.26: Gráfico do Erro de Quantização em função do número de protótipos.

4.4 Validação via Índices Dunn e Davies-Bouldin

Conforme estudado no Capítulo 3, os índices Dunn e DB avaliam explicitamente o grau de compactação e separabilidade dos agrupamentos. A aplicação do índice Dunn a todos os algoritmos de interesse está mostrada na Figura 4.27, enquanto que a aplicação do índice DB está mostrada na Figura 4.28.

No caso do índice Dunn, o ponto máximo nas curvas em função de K representa a melhor formação dos agrupamentos. Para o índice DB, o melhor número de agrupamentos para um determinado conjunto de dados é indicado pelo ponto mínimo no gráfico correspondente.

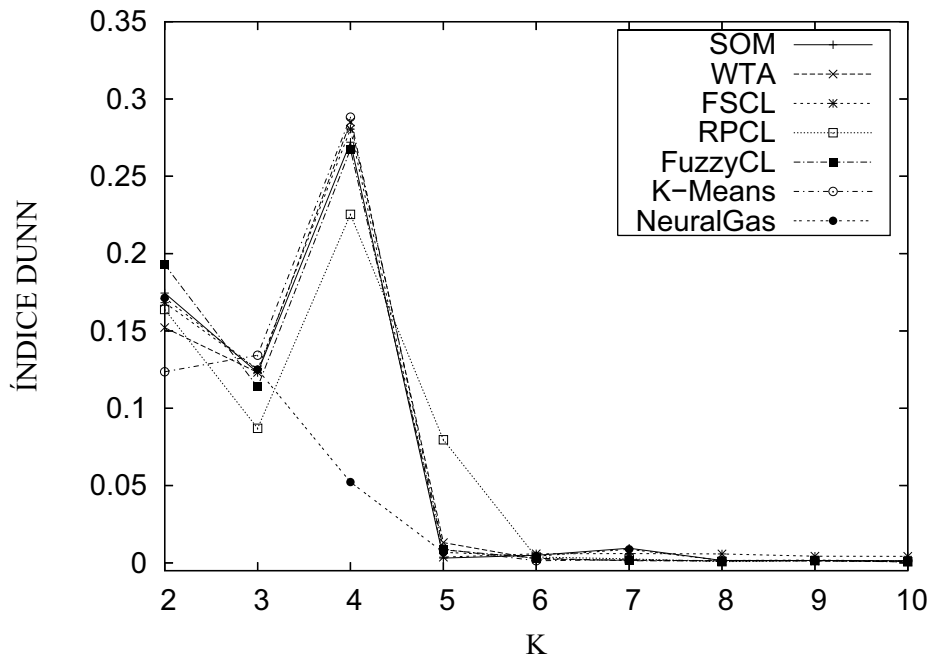


Figura 4.27: Índice Dunn por número de protótipos.

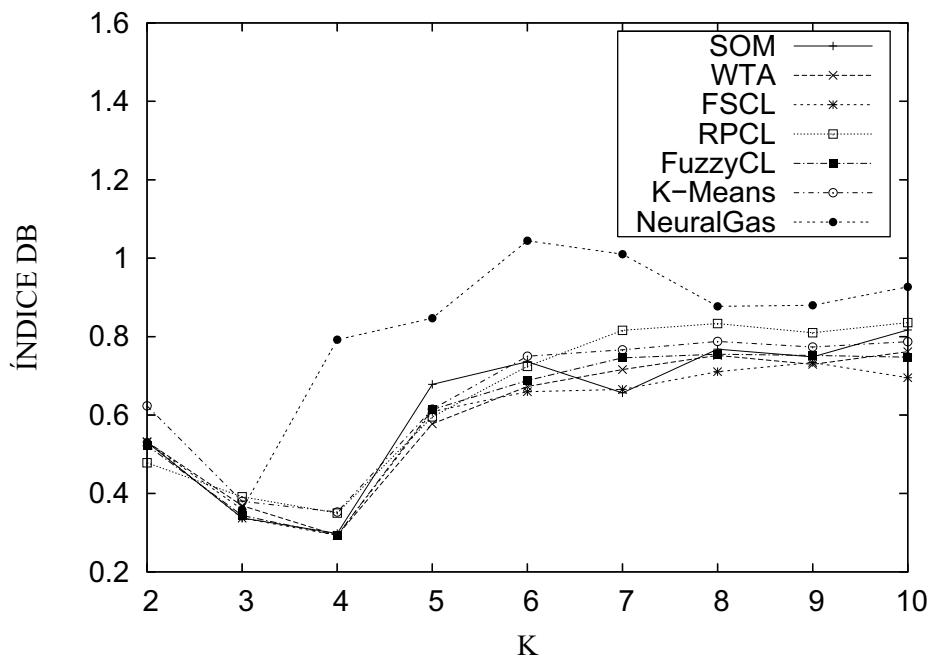


Figura 4.28: Índice Davies-Bouldin por número de protótipos.

Ambas as figuras mostram que o melhor particionamento em agrupamentos é aquele representado por quatro protótipos. A única exceção ocorreu para o índice DB aplicado à rede Neural-Gas, cujo melhor particionamento sugerido corresponde àquele com três protótipos.

4.5 Melhores Agrupamentos segundo os Índices Dunn e DB

As Figuras 4.29 a 4.35 mostram os melhores agrupamentos obtidos, segundo os índices Dunn e/ou DB, para os algoritmos SOM-1D, WTA, FSCL, RPCL, *K*-médias, FuzzyCL e Neural-Gas, respectivamente.

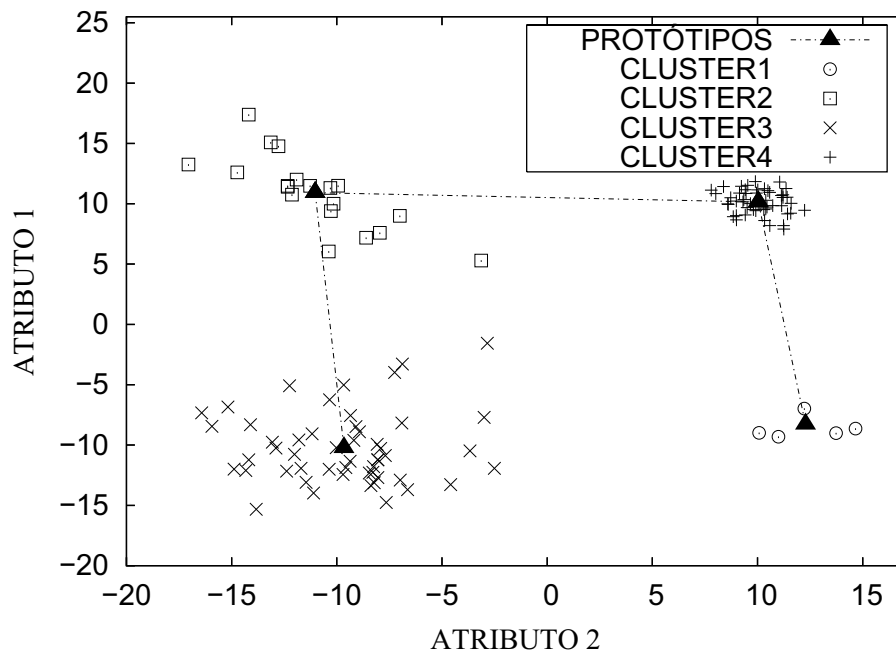


Figura 4.29: Melhor posicionamento dos protótipos da rede SOM-1D.

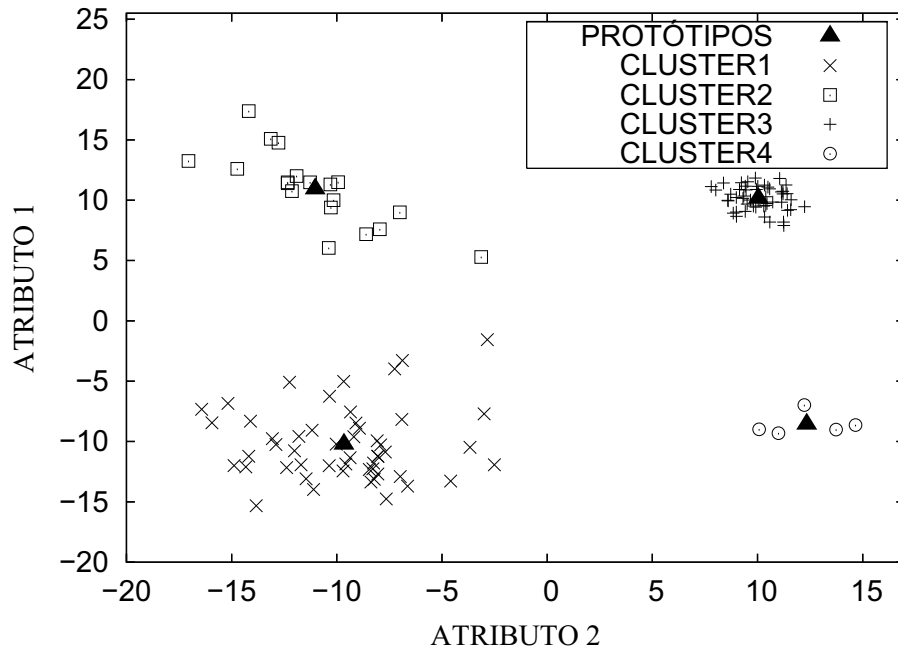


Figura 4.30: Melhor posicionamento dos protótipos da rede WTA segundo índices Dunn e DB.

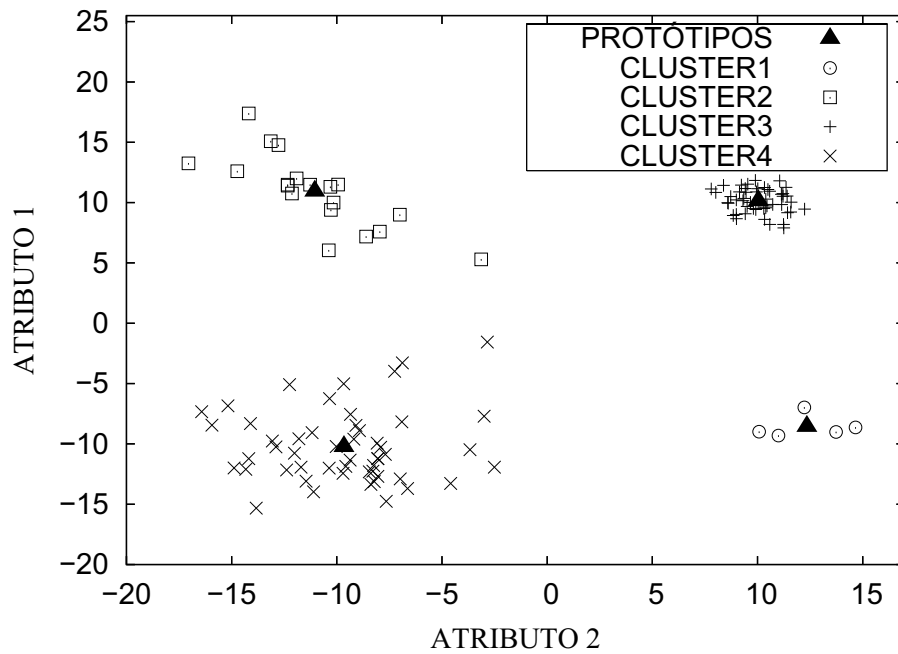


Figura 4.31: Melhor posicionamento dos protótipos da rede FSCL segundo índices Dunn e DB.

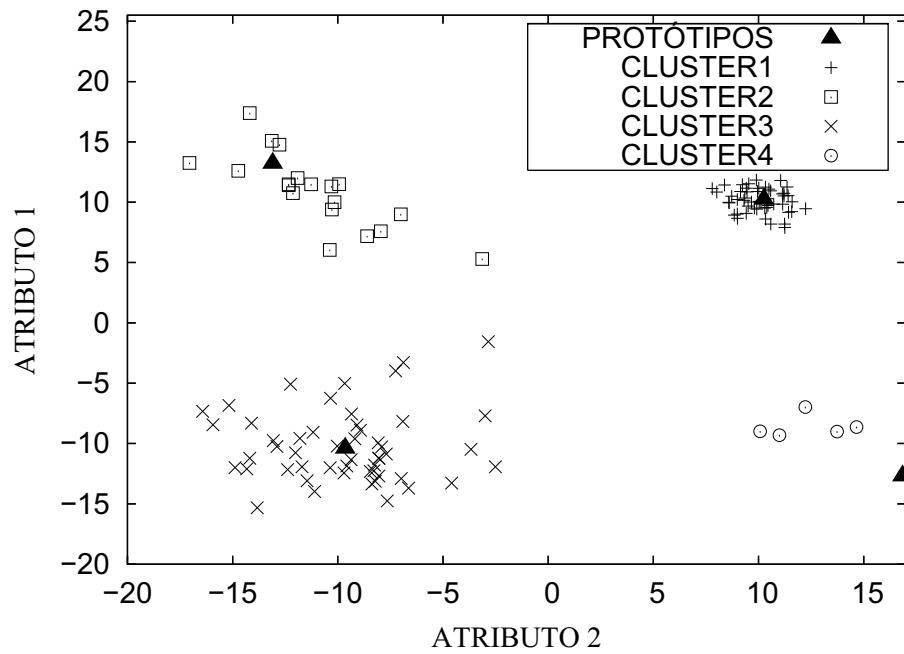


Figura 4.32: Melhor posicionamento dos protótipos da rede RPCL segundo índice Dunn.

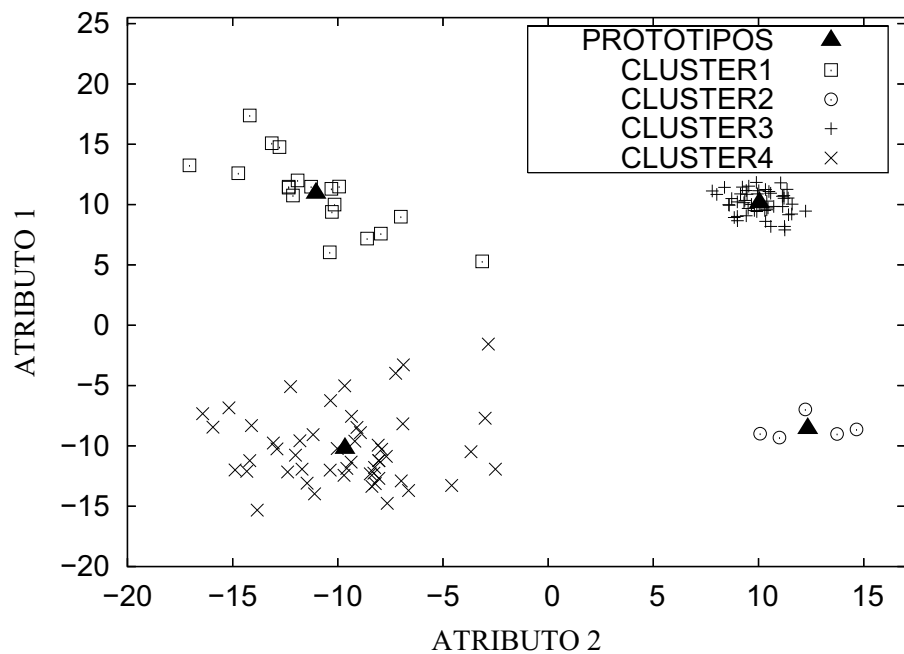


Figura 4.33: Melhor posicionamento dos protótipos do K-médias segundo índices Dunn e DB.

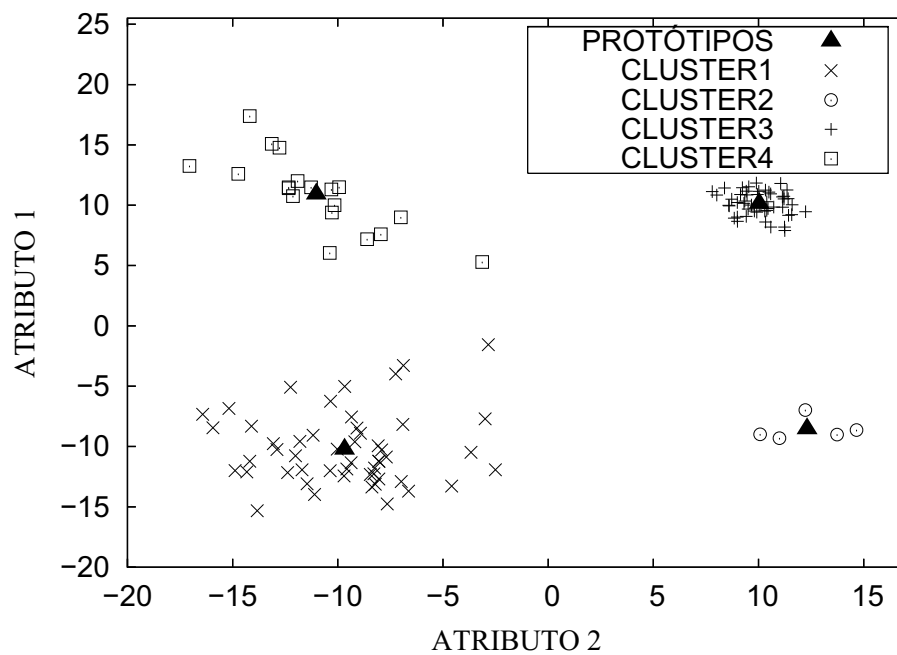


Figura 4.34: Melhor posicionamento dos protótipos da rede FuzzyCL segundo índices Dunn e DB.

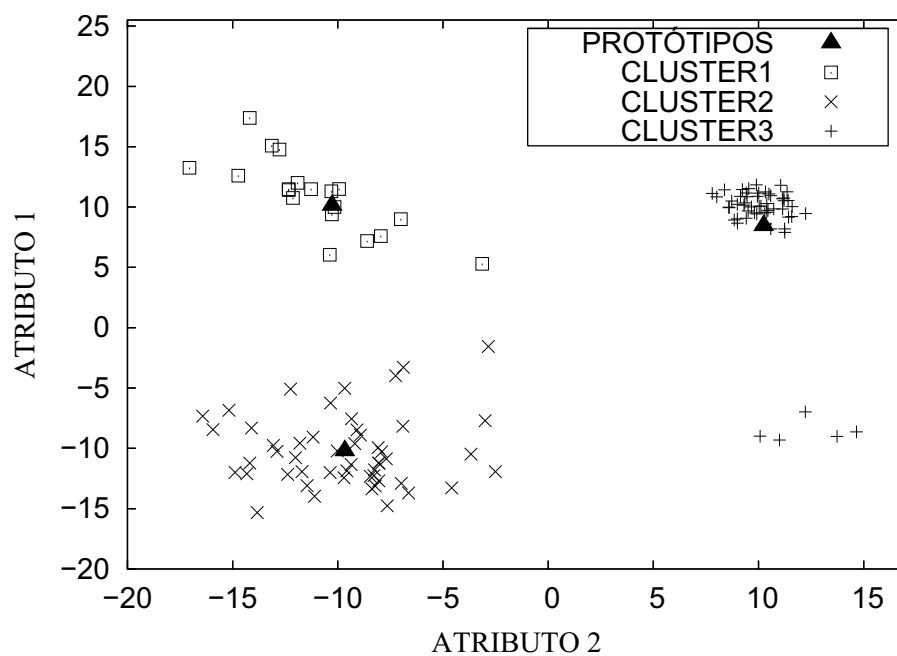


Figura 4.35: Melhor posicionamento dos protótipos da Neural-Gas segundo índice DB.

4.6 Conclusão

Este capítulo avaliou o desempenho de algoritmos neurais e não-neurais em tarefas de análise de agrupamentos. Para isso, foram testados diversas métricas de validação de agrupamentos, tais como métodos visuais, erro de quantização e índices Dunn e DB.

Os resultados obtidos mostram que se os resultados dos agrupamentos gerados pelas redes neurais competitivas forem devidamente avaliados, qualitativa e quantitativamente, as soluções ótimas obtidas se equiparam àquelas obtidas por algoritmo clássicos, tal como o algoritmo K -médias. Os resultados mostraram também que o fator de maior importância para o bom desempenho dos índices não é muito a forma do agrupamento, mas sim a quantidade de dados em cada um.

Os índices Dunn e Davies-Bouldin são das métricas mais usadas para avaliar resultados de análise agrupamentos. Por outro lado, sua utilidade não está clara quando os dados são oriundos, por exemplo, de tarefas de quantização vetorial. Em outras palavras, o número de agrupamentos sugeridos por estes índices pode não ser o mais adequado àquela classe de problemas. Além disso, o erro de quantização, por si só, pode não ser útil também em quantização vetorial, uma vez que esta métrica pode sugerir valores muito altos para K , a fim de minimizar o erro de reconstrução dos dados.

No próximo capítulo são apresentados resultados concernentes à compressão de imagens para quantização vetorial, usando os índices Dunn e DB, bem como de alguns critérios de informação.

5 *Resultados - Quantização Vetorial*

5.1 Introdução

Este capítulo apresenta os resultados referentes à aplicação dos algoritmos de redes neurais competitivas e do algoritmo K -médias em uma tarefa de quantização vetorial. Escolheu-se a tarefa de compressão de imagens por ser bem representativa dessa classe de problemas e de amplo interesse para as áreas de processamento de imagens e comunicações multimídia.

Experimentos realizados com o intuito de selecionar o número adequado de protótipos segundo as várias métricas estudadas nesta dissertação são mostrados na Seção 5.3. Usando apenas os melhores algoritmos da seção anterior, os novos experimentos realizados estão descritos na Seção 5.4. A Seção 5.5 é reservado ao estudo do desempenho de alguns dos algoritmos estudados em tarefas de quantização vetorial com ruído na fonte ou no canal.

5.2 Metodologia de Simulação

A fim de avaliar os vários algoritmos nas tarefas de quantização vetorial que se seguem foram estabelecidos os seguintes procedimentos:

1. A resolução das imagens é 256×256 ou 512×512 :
2. Imagens digitais são codificadas em 256 (8 bits) níveis de cinza.
3. Os vetores de dados são construídos a partir de blocos de dimensão bem menor que a dimensão da imagem (e.g. 2×2 , 4×4 ou 8×8).
4. Nenhum algoritmo de pré-processamento (e.g. Transformada Discreta do Cosseno ou wavelets) é aplicado aos blocos extraídos da imagem original.
5. Assume-se um modelo de canal binário, simétrico e estacionário para todos experimentos de quantização vetorial descritos neste capítulo.

6. O índice do protótipo vencedor é convertido para notação binária convencional.
7. O número de protótipos é sempre feito igual a uma potência de 2, conforme abordagem padrão em quantização vetorial
8. Nos experimentos envolvendo um canal ruidoso, a probabilidade de um *bit* comutar aleatoriamente seu valor é simbolizada por P . Com o objetivo do trabalho ser autocontido, no Apêndice C encontra-se detalhes sobre canais de comunicação incluindo o *BSC* usado nas simulações.
9. Um conjunto de K são usados para iniciar os K protótipos de um determinado algoritmo.
10. Os algoritmos são treinados por um número fixo de 50 épocas, salvo indicação contrária.
11. Cada valor de um índice ou métrica mostrado em gráficos ou tabelas correspondem à média de 5(cinco) realizações de treinamento, repetidas sob as mesmas condições.
12. A taxa de aprendizagem de todos os algoritmos neurais decai exponencialmente de um valor máximo de 0,5 a um valor mínimo de 0,0001.
13. A cada época a ordem de apresentação dos dados é aleatória, a fim de evitar qualquer tipo de viés durante a aprendizagem.
14. Para simular a rede SOM foi escolhida uma topologia unidimensional (1D) com função de vizinhança gaussiana.
15. o decaimento com o tempo das funções de vizinhança das rede SOM e Neural-Gas é exponencial.
16. Os parâmetros da função de vizinhança da rede SOM são $\sigma_0 =$ menor inteiro maior que $K/2$ e $\sigma_T = 0,0001$.
17. Os parâmetros da função de vizinhança da rede Neural-Gas são $\lambda_0 = K$ e $\lambda_T = 0,01$.

A Figura 5.1 ilustra o funcionamento da tarefa de quantização vetorial simuladas neste capítulo.

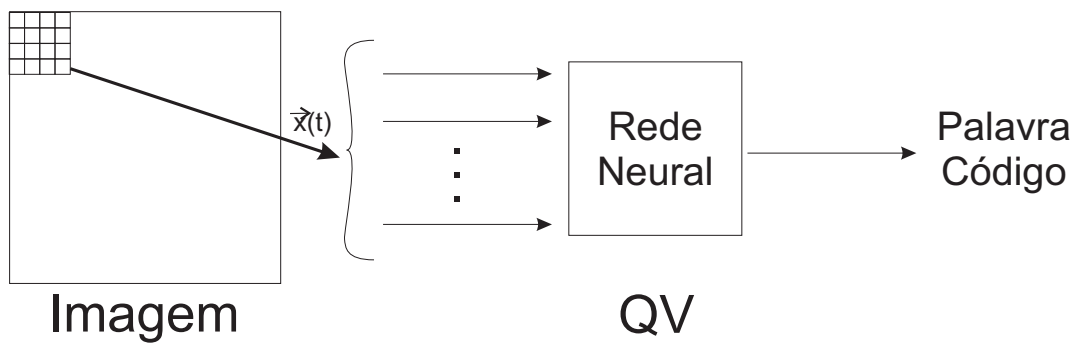


Figura 5.1: Figura ilustrativa mostrando a entrada dos dados (*pixels*) no processo de quantização vetorial usando as redes neurais competitivas.

5.3 Experimentos de Seleção Inicial de Modelos

Para uma seleção inicial de modelos foi usada a imagem do macaco Mandrill não-ruidosa com dimensão 256×256 *pixels*, considerando a quantização vetorial com dimensão $p = 16$, usando blocos de 4×4 *pixels*. Utilizou-se, portanto, um conjunto de treino com 4096 vetores de dimensão 16. Foram testados algoritmos com $K = 4, 8, 16, 32, 64, 128$ e 256 protótipos.

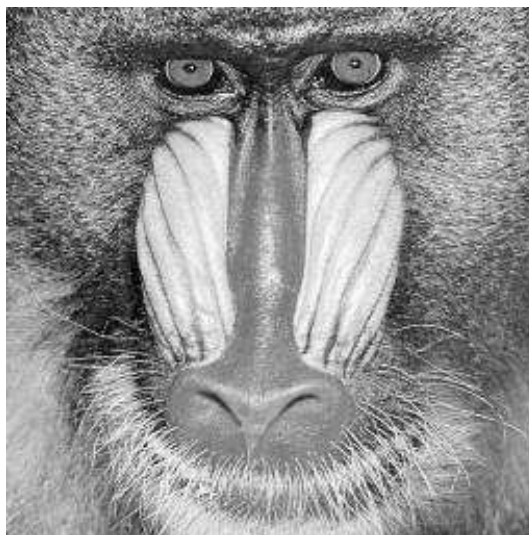


Figura 5.2: Imagem original do macaco Mandrill em 256×256 *pixels* e 8 *bits*.

5.3.1 Seleção via Erro de Quantização

A Figura 5.3 apresenta as curvas de valores do erro médio de quantização após a execução do treinamento. Nota-se claramente que o procedimento de escolher o número de protótipos como sendo àquele valor onde ocorre o joelho da curva, em geral leva a valores pequenos para K . Valores pequeno para este parâmetro produzem erros de reconstrução elevados.

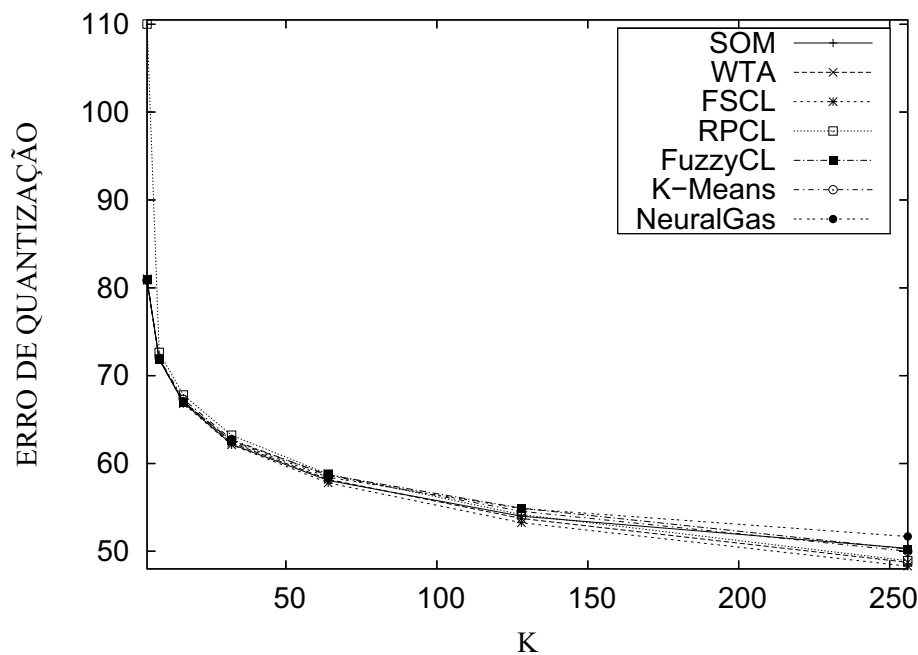


Figura 5.3: Gráfico do erro médio de quantização em função do número de protótipos.

Uma alternativa seria escolher valores para K nas regiões em que a derivada da curva é suficientemente próxima de zero. Neste caso, os valores de K selecionados tenderiam a ser altos. A desvantagem desta alternativa é justamente definir o significado de “suficientemente próximo de zero”.

A Figura 5.4 mostra a resultados da reconstrução da imagem do Mandrill quantizada vetorialmente pela rede RPCL, para dois valores de $K = 4$ (método do joelho) e $K = 256$ (método da derivada nula). Este algoritmo foi escolhido por ter apresentado os piores resultados na Figura 5.3. Nota-se que embora tenha apresentado os piores resultados no geral, as imagens reconstruídas são coerentes com a imagem original.



(a) RPCL/4K

(b) RPCL/256K

Figura 5.4: (a) Imagem do macaco Mandrill reconstruída pela rede RPCL com $K = 4$ protótipos. (b) Imagem do macaco Mandrill reconstruída pela rede RPCL com $K = 256$ protótipos.

5.3.2 Seleção via Razão Sinal-Ruído de Pico

Em quantização vetorial é comum o uso da métrica *Peak Signal-to-Noise Ratio* (PSNR) (HARRANDI; GHARAVI-ALKHANSARI, 2003), definida como

$$PSNR = 10 \log_{10} \left(\frac{N \cdot p}{\sum_{i=1}^K \sum_{\mathbf{x}^\mu \in V_i} \|\mathbf{x} - \mathbf{w}_i\|^2} \right). \quad (5.1)$$

em que V_i é a região de Voronoi do i -ésimo protótipo, K é o número de protótipos e p é a dimensão do vetor de entrada e N é o número total de vetores de treinamento.

A Figura 5.25 apresentam as curvas de valores da métrica PSNR para todos os algoritmos gerada durante a reconstrução das imagens.

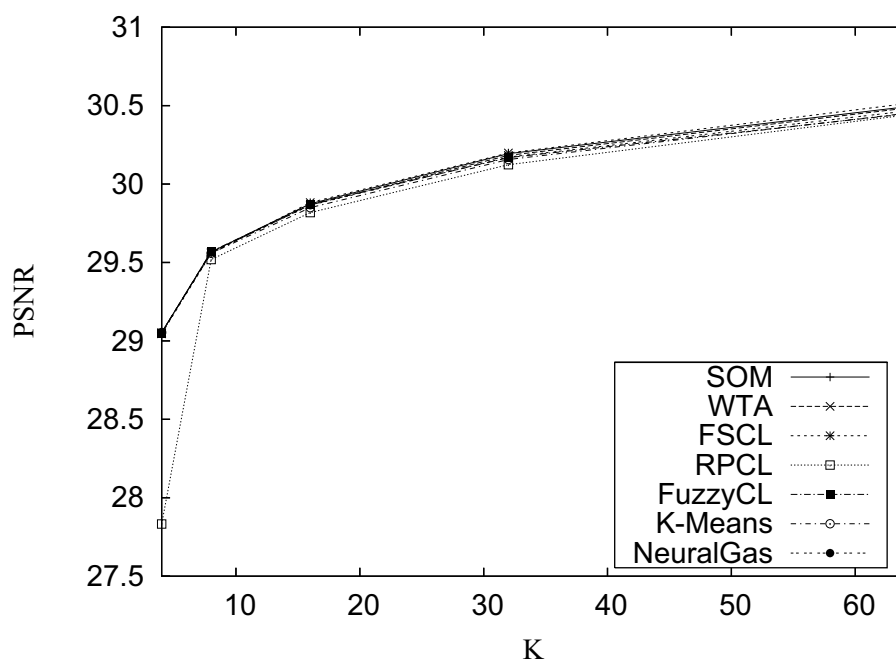


Figura 5.5: Gráficos da métrica PSNR versus número de protótipos.

Na função do PSNR quanto maior for o valor do índice melhor é o resultado da reconstrução. A avaliação via PSNR funciona de maneira inversa a realizada com o erro de quantização. De maneira equivalente ao erro médio de quantização, todos os algoritmos deram obtiveram o maior valor do índice para $K = 256$ protótipos.

5.3.3 Seleção via Índices de Validação de Agrupamentos

A Figura 5.6 apresenta as curvas de valores do índice Dunn após a execução do treinamento. Observa-se um comportamento interessante para este índice. Todas as curvas, após um período de indefinição, têm uma leve tendência de crescimento. Como no índice Dunn o melhor agrupamento é caracterizado pelo maior valor, a tendência mostrada no gráfico é que quanto maior o número de protótipos, melhor para o algoritmo. Assim, este índice tende a favorecer valores altos para K , embora não deixe claro qual valor deve ser escolhido.

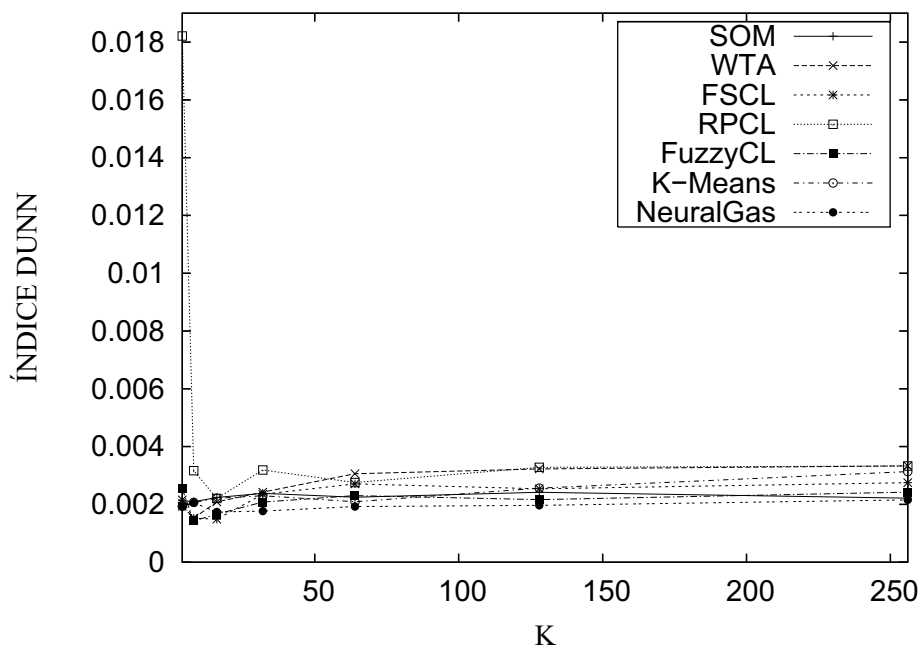


Figura 5.6: Gráfico do Índice Dunn versus número de protótipos.

A título de ilustração, a Figura 5.7 mostra a curva do índice Dunn para a rede WTA, assim como o resultado da reconstrução da imagem do Mandrill quantizada vetorialmente para $K = 256$. Nota-se que a imagem reconstruída é praticamente indistingüível da imagem reconstruída pela rede RPCL.

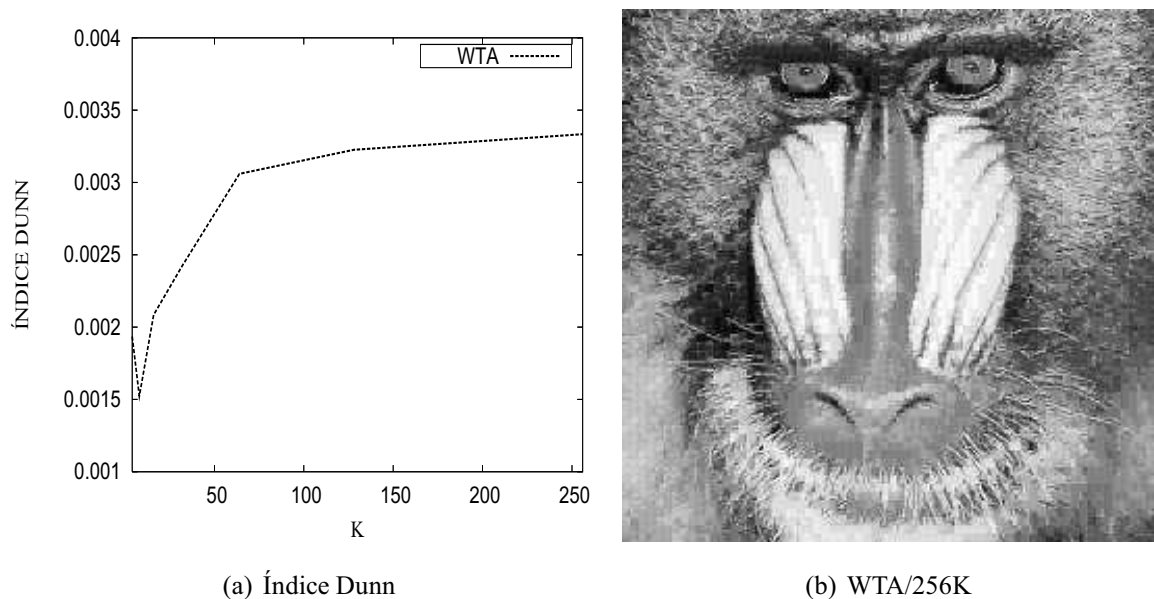


Figura 5.7: (a) Gráfico da evolução do índice Dunn para a rede WTA. (b) Imagem do macaco Mandrill reconstruída pela rede WTA com $K = 256$ protótipos.

A Figura 5.8 apresenta as curvas de valores do índice DB para todos os algoritmos após a execução do treinamento. Observa-se um comportamento interessante para este índice. Todas as curvas têm uma tendência bem acentuada de crescimento. Como no índice DB o melhor agrupamento é caracterizado pelo menor valor de uma seqüência de valores, a tendência mostrada no gráfico é que quanto menor o número de protótipos, melhor para o algoritmo. Assim, este índice tende a favorecer valores baixos para K (próximos ao joelho das curvas), o que claramente não resulta em boas imagens reconstruídas.

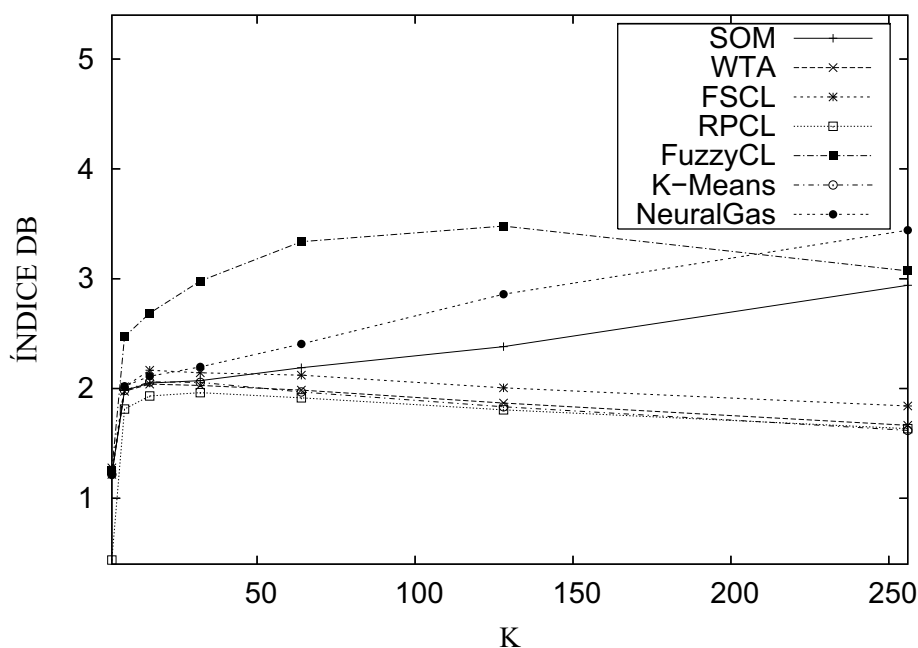
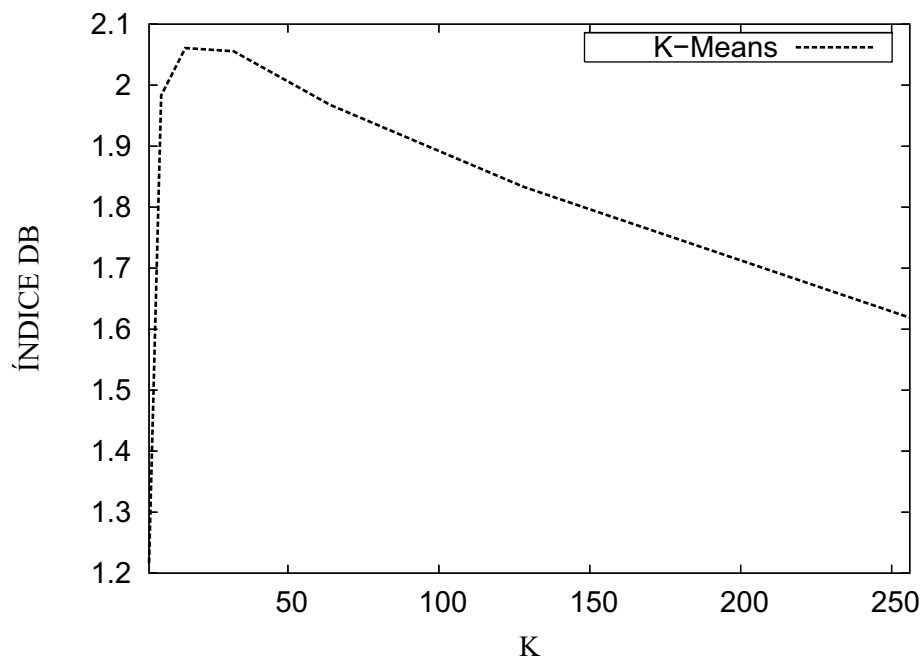


Figura 5.8: Gráfico do Índice Davies-Bouldin versus número de protótipos.

A título de ilustração, a Figura 5.9 mostra a curva do índice Dunn para o algoritmo K -médias, assim como o resultado da reconstrução da imagem do Mandrill quantizada vetorialmente para $K = 4$ e $K = 256$ protótipos. Nota-se que a imagem reconstruída para $K = 256$ é praticamente indistinguível da imagem reconstruída pelas redes RPCL e WTA. Já a imagem reconstruída para $K = 4$ é de qualidade bem inferior.



(a) Índice DB

(b) K -médias/4K(c) K -médias/256K

Figura 5.9: (a) Gráfico da evolução do índice DB para o algoritmo K -médias. (b) Imagem do macaco Mandrill reconstruída pelo algoritmo K -médias com $K = 4$ protótipos. (c) Imagem do macaco Mandrill reconstruída pelo algoritmo K -médias com $K = 256$ protótipos.

As Tabelas 5.1, 5.2 e 5.3 apresentam respectivamente os valores do erro médio de quantização, índice Dunn e índice Davies-Bouldin para as imagens reconstruídas com diferentes valores de K .

K	K-médias	WTA	SOM	FSCL	RPCL	NGA	FuzzyCL
4	80,855	80,856	80,855	81,023	110,033	80,856	80,935
8	71,978	71,875	71,837	71,997	72,640	71,878	71,805
16	67,315	67,053	66,940	66,815	67,804	67,085	66,985
32	62,728	62,349	62,183	62,120	63,210	62,498	62,587
64	58,628	58,181	58,065	57,790	58,782	58,436	58,688
128	54,556	53,744	53,991	53,255	54,183	54,860	54,912
256	50,002	48,751	50,339	48,296	48,931	51,689	50,240

Tabela 5.1: Valores do Erro de Quantização para cada algoritmo por número de protótipos.

Na Tabela 5.1 os algoritmos em destaque são o FSCL e RPCL. O primeiro por apresentar o menor erro de quantização entre os algoritmos para a maioria dos experimentos e o segundo por apresentar o maior erro dentre os algoritmos. Contudo, à medida que o número de protótipos aumenta, a diferença de desempenho entre os algoritmos tende a ser insignificante do ponto de vista estatístico.

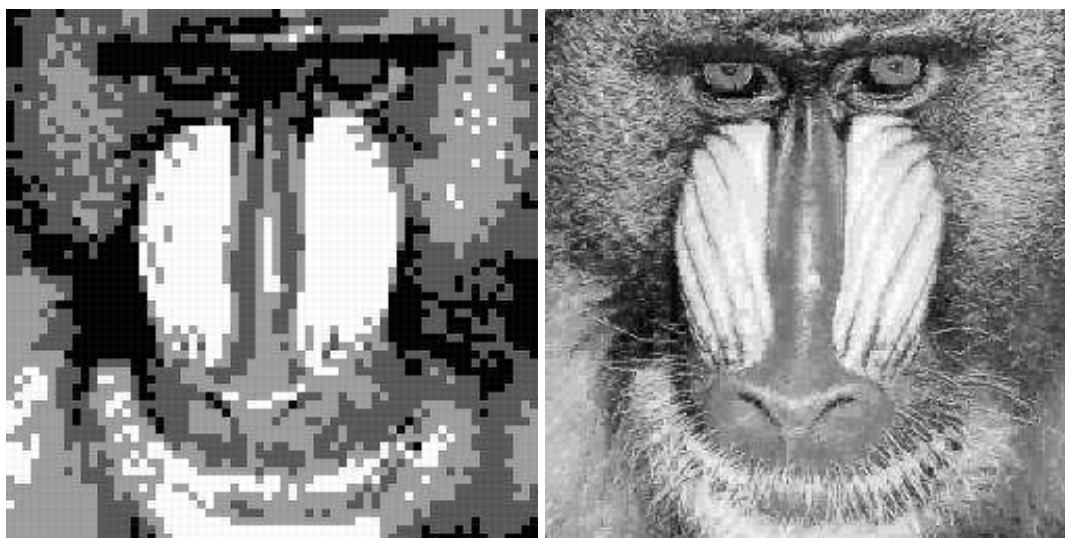
K	K-médias	WTA	SOM	FSCL	RPCL	NGA	FuzzyCL
4	0,0019	0,0019	0,0021	0,0022	0,0182	0,0019	0,0025
8	0,0021	0,0015	0,0021	0,0015	0,0032	0,0021	0,0014
16	0,0022	0,0021	0,0022	0,0015	0,0022	0,0017	0,0017
32	0,0023	0,0024	0,0024	0,0023	0,0032	0,0018	0,0021
64	0,0021	0,0031	0,0022	0,0027	0,0027	0,0019	0,0023
128	0,0026	0,0032	0,0024	0,0025	0,0033	0,0020	0,0022
256	0,0031	0,0033	0,0022	0,0028	0,0033	0,0022	0,0024

Tabela 5.2: Valores do Índice Dunn para cada algoritmo por número de protótipos.

K	K-médias	WTA	SOM	FSCL	RPCL	NGA	FuzzyCL
4	1,2169	1,2177	1,2177	1,2775	0,4371	1,2178	1,2533
8	1,9833	1,9738	1,9811	2,0158	1,8133	2,0204	2,4711
16	2,0608	2,0404	2,0497	2,1661	1,9323	2,1112	2,6866
32	2,0555	2,0283	2,0717	2,1437	1,9618	2,1971	2,9791
64	1,9677	1,9856	2,1894	2,1213	1,9142	2,4054	3,3373
128	1,8335	1,8669	2,3808	2,0057	1,8060	2,8583	3,4789
256	1,6190	1,6642	2,9402	1,8409	1,6336	3,4423	3,0711

Tabela 5.3: Valores do Índice Davies-Bouldin para cada algoritmo por número de protótipos.

Nas Tabelas 5.2 e 5.3 percebe-se claramente a inadequação da avaliação feita pelos índices Dunn e DB para problemas de quantização vetorial. Para a primeira tabela, os melhores resultados ocorrem para $K = 256$ protótipos, enquanto que para a segunda tabela ocorrem para $K = 4$ protótipos. A título de ilustração, a Figura 5.10 mostra imagens reconstruídas pela rede FSCL com $K = 4$ e $K = 256$ protótipos.



(a) $DB = 1,2775, K = 4$

(b) $Dunn = 0,0028, K = 256$

Figura 5.10: (a) Imagem do macaco Mandrill reconstruída pela rede FSCL com $K = 4$ protótipos. (b) Imagem do macaco Mandrill reconstruída pela rede FSCL com $K = 256$ protótipos.

A conclusão geral para esta seção é de que os índices Dunn e DB não são adequados para uso em tarefas de quantização vetorial, pois ou favorecem valores muito altos (índice Dunn) ou muito baixos para K (índice DB). Nem uma situação, nem a outra é desejável em aplicações reais. Valores baixos resultam em reconstrução de baixa qualidade, enquanto valores altos aumentam o custo computacional do processo de reconstrução. Na próxima seção, estudam-se critérios que tentem minimizar ao mesmo tempo o erro de reconstrução e o número de protótipos do algoritmo.

5.3.4 Seleção via Critérios de Informação

Critérios de informação foram estudados no Capítulo 3. Em todas as expressões que os definem, o parâmetro k representa o número de parâmetros ajustáveis de um dado modelo. No contexto da aplicação de redes neurais competitivas em quantização vetorial, o número total de parâmetros ajustáveis de uma rede competitiva é dado pelo produto do número de protótipos K pela dimensão do vetor de entrada p ; logo, $k = K \cdot p$.

A Figura 5.11 apresenta as curvas com os valores do índices AIC obtidas após a reconstrução da imagem para cada valor de K . Para este índice os pontos mínimos são indicativos de configurações ótimas para um dado algoritmo em termos do número de protótipos.

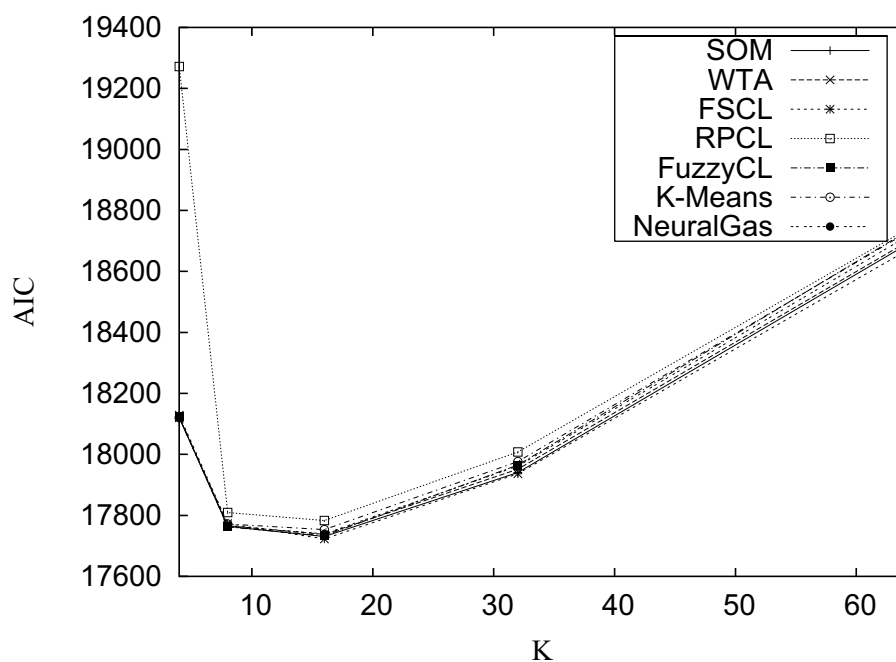


Figura 5.11: Gráfico do Índice AIC por número de protótipos.

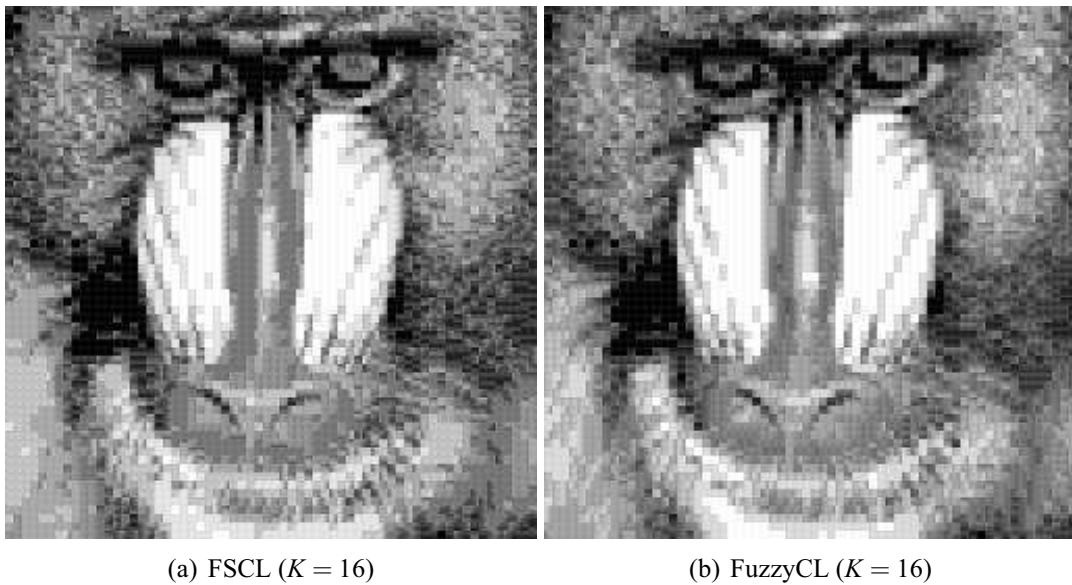


Figura 5.12: (a) Imagem do macaco Mandrill reconstruída pela rede FSCL com $K = 16$ protótipos. (b) Imagem do macaco Mandrill reconstruída pela rede FuzzyCL com $K = 16$ protótipos.

De acordo com o índice AIC, os algoritmos que obtiveram melhores índices foram as redes FSCL e FuzzyCL. A Figura 5.12 traz imagens reconstruídas por estas duas redes para a melhor configuração sugerida pelo índice AIC ($K = 16$ protótipos). As imagens reconstruídas são praticamente indistinguíveis uma da outra.

A Figura 5.19 apresenta as curvas com os valores do índice MDL obtidas após a reconstrução da imagem por todos os algoritmos para um certo valor de K . Para este índice os pontos mínimos também indicam de configurações ótimas para um dado algoritmo, em termos do número de protótipos.

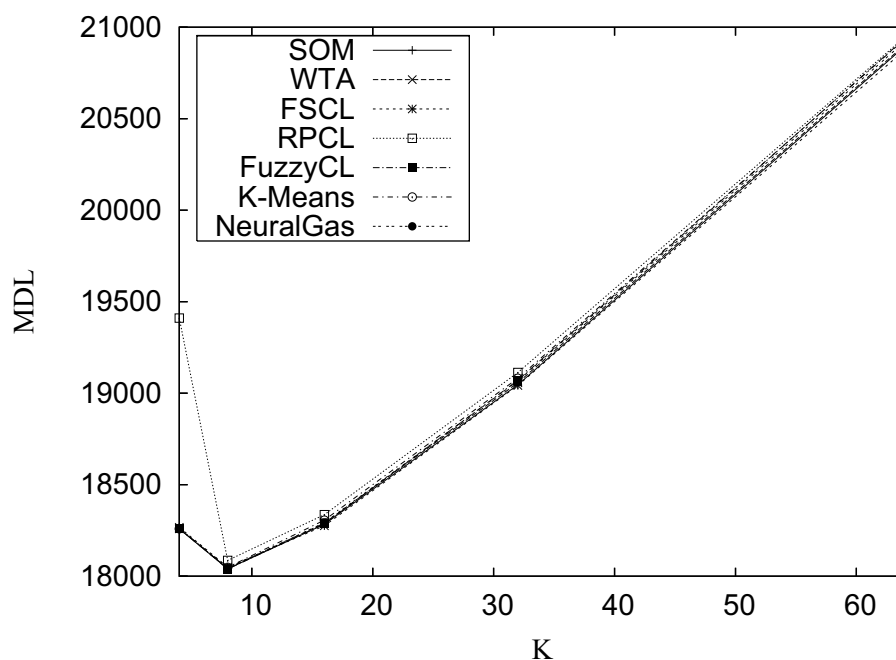


Figura 5.13: Curvas do índice MDL versus número de protótipos.

Segundo o critério MDL, os algoritmos que obtiveram melhores índices foram as redes FuzzyCL e SOM. A Figura 5.19 mostra que os pontos de mínimo são referentes à configurações com $K = 8$ protótipos. A Figura 5.14 traz imagens reconstruídas por estas duas redes para a melhor configuração sugerida pelo índice MDL ($K = 8$ protótipos). As imagens reconstruídas tem qualidade inferior às reconstruídas com $K = 16$ protótipos, porém entre si são praticamente indistinguíveis uma da outra.

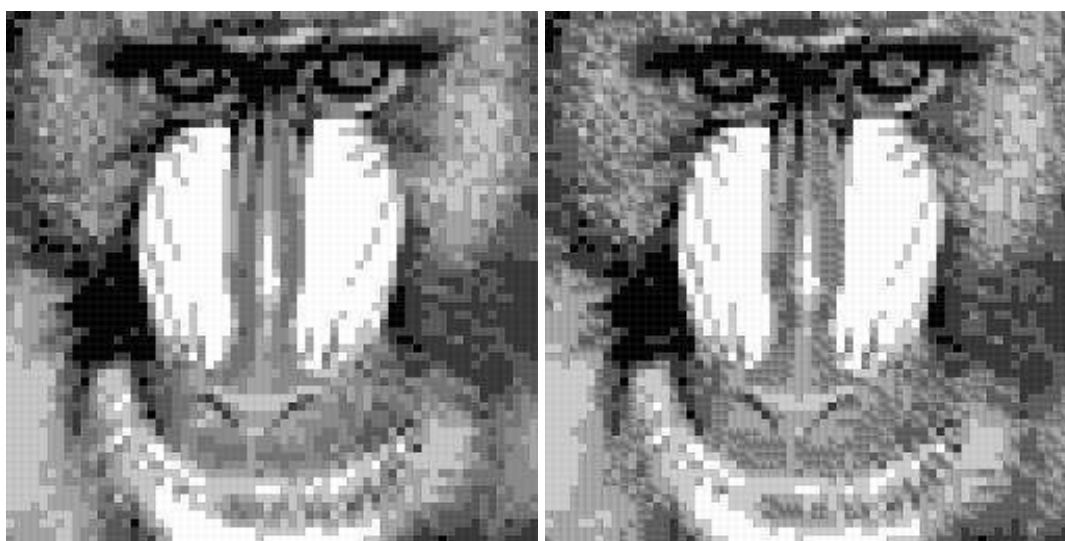
(a) FuzzyCL ($K = 8$)(b) SOM ($K = 8$)

Figura 5.14: (a) Imagem do macaco Mandrill reconstruída pela rede FuzzyCL com $K = 8$ protótipos. (b) Imagem do macaco Mandrill reconstruída pela rede SOM com $K = 8$ protótipos.

Embora os gráficos dos índices AIC e MDL em função de K resultem em curvas nas quais fica fácil distinguir o valor ótimo para K , os resultados obtidos ainda não são bons o suficiente segundo o principal critério de julgamento de qualidade de imagens, a percepção humana. Assim, na próxima seção será proposta uma pequena modificação nestes critérios de informação a fim de adequá-los ao problema de quantização vetorial. Ainda serão estudados outros critérios de informação a fim de gerar conclusões mais confiáveis.

5.4 Critérios de Informação em Quantização Vetorial: Uma Nova Abordagem

O parâmetro k representa o número de parâmetros ajustáveis de um certo modelo, que é usado na expressão que define os vários critérios de informação. Esse parâmetro foi definido como $k = K \cdot p$. Contudo, tomando como base os resultados da seção anterior, percebeu-se que este parâmetro pode estar incorretamente definido no contexto de quantização vetorial. Propõe-se então uma modificação na definição de k , passando a significar agora o número de protótipos: $k = K$.

Uma série de simulações são realizadas a seguir, a fim de verificar o resultado desta nova definição do parâmetro k . Para estas simulações, utilizou-se uma imagem de 512×512 pixels, blocos de 8×8 pixels. Resultando, portanto, em um conjunto de treinamento com $N = 4096$ vetores de dimensão $p = 64$.

A Figura 5.15 apresenta a curva do índice FPE em função de K para algoritmos que obtiveram melhores resultados na seção anterior (redes FSCL e FuzzyCL) segundo os critérios AIC e MDL. Cada valor do índice FPE, para um certo valor de K , corresponde a uma reconstrução da imagem pelo algoritmo sendo avaliado.

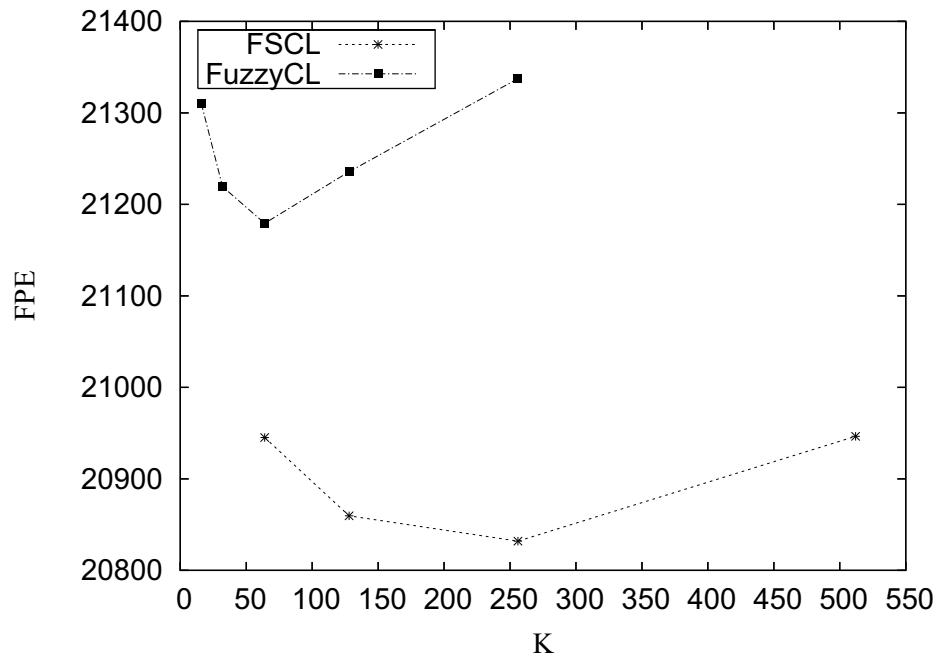


Figura 5.15: Curvas do índice FPE versus número de protótipos para as redes FSCL e FuzzyCL.

A Figura 5.16 traz as imagens reconstruídas pelas redes FSCL e FuzzyCL para o número ótimo de protótipos sugeridos pelo critério FPE.

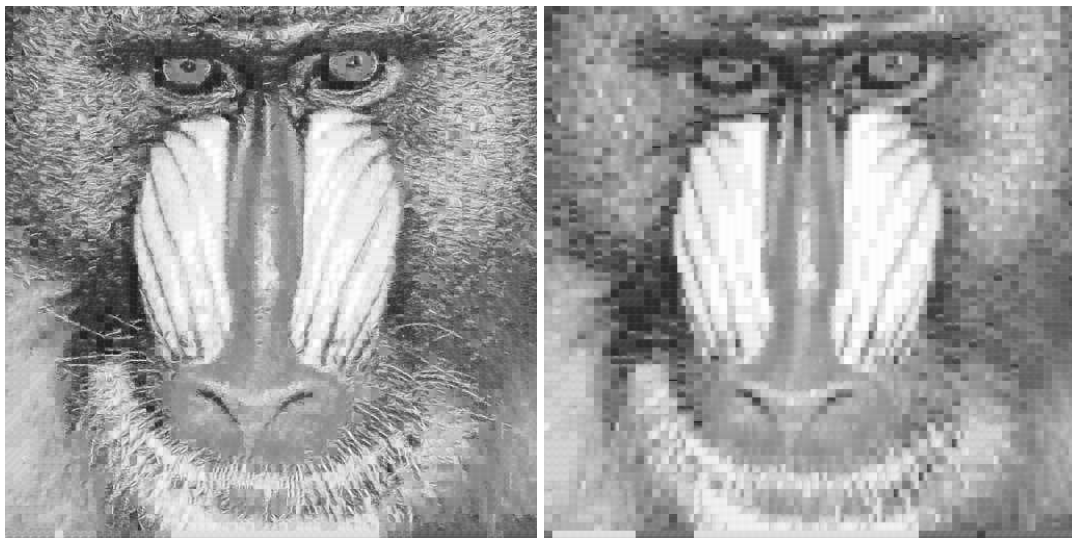
(a) FSCL ($K = 256$)(b) FuzzyCL ($K = 64$)

Figura 5.16: Imagens reconstruídas pelas redes FSCL com $K = 256$ protótipos e FuzzyCL com $K = 64$ protótipos segundo o critério FPE.

A Figura 5.17 apresenta a curva do índice BIC em função de K para as redes FSCL e FuzzyCL. Cada valor do índice BIC, para um certo valor de K , corresponde a uma reconstrução da imagem pelo algoritmo sendo avaliado.

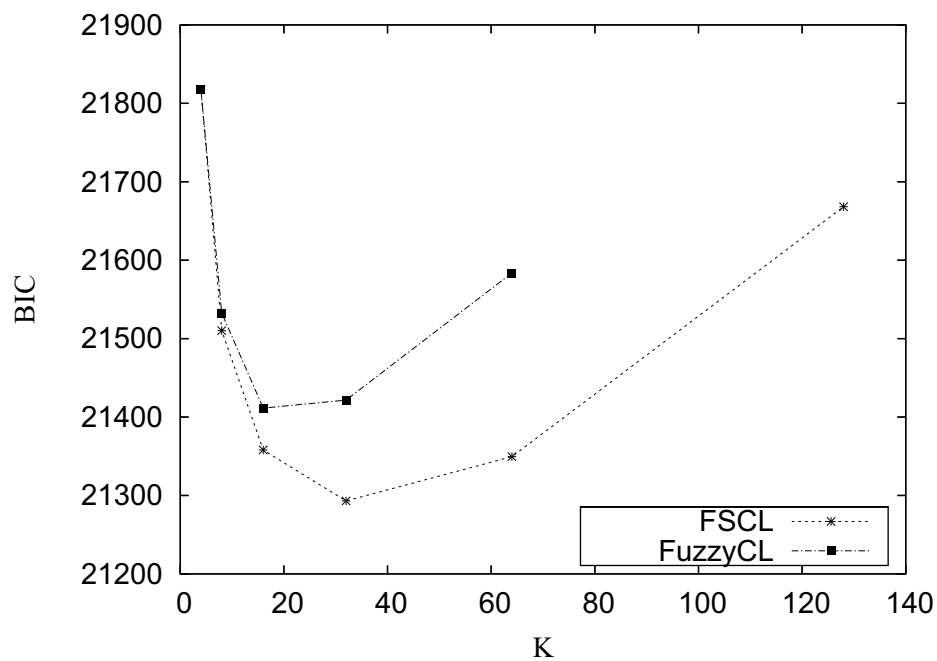


Figura 5.17: Curvas do índice BIC versus número de protótipos (redes FSCL e FuzzyCL).

Pode-se notar claramente que o ponto de mínimo na curva do índice FPE (Figura 5.15) corresponde a um valor para K maior do que na curva para o índice BIC (Figura 5.17). Assim, as imagens reconstruídas com o valor de K sugerido pelo índice BIC possuem uma qualidade visual pior do que as reconstruídas com o valor sugerido pelo índice FPE. Isto pode ser verificado comparando as imagens reconstruídas mostradas na Figura 5.16 com as mostradas na Figura 5.18.

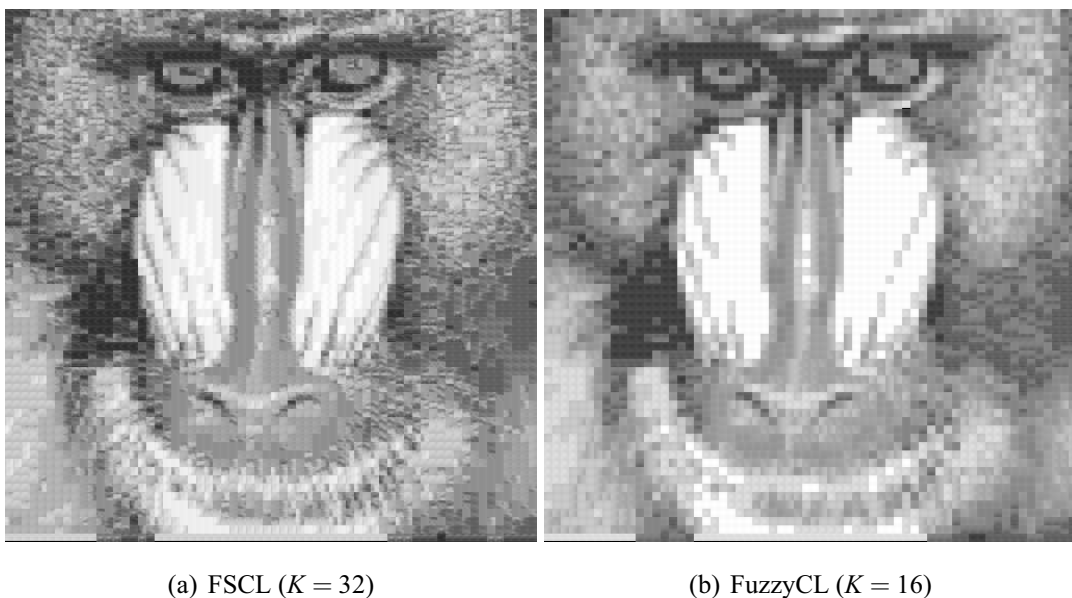


Figura 5.18: Imagens reconstruídas pelas redes FSCL com $K = 32$ protótipos e FuzzyCL com $K = 16$ protótipos segundo o critério BIC.

Para finalizar os testes com os critérios de informação, a Figura 5.19 traz as curvas resultantes da execução das redes FSCL e FuzzyCL para o número ótimo de protótipos sugeridos pelo critério MDL.

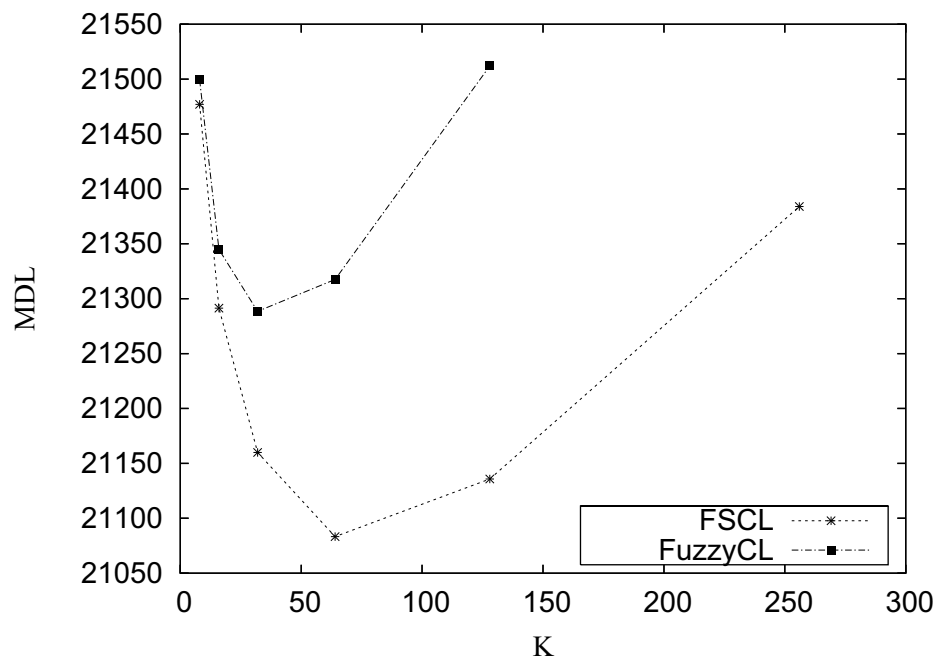


Figura 5.19: Curvas do índice MDL versus número de protótipos para as redes FSCL e FuzzyCL.

A Figura 5.19 mostra que o critério MDL tem valores intermediários para K em comparação com as Figuras 5.16 e 5.18. Isto significa que as imagens reconstruídas possuem uma qualidade visual um pouco inferior que as reconstruídas com o número de protótipos sugerido pelo índice FPE, porém o custo computacional é menor.

A Figura 5.20 traz as imagens reconstruídas pelas redes FSCL e FuzzyCL para o número ótimo de protótipos sugeridos pelo critério MDL.

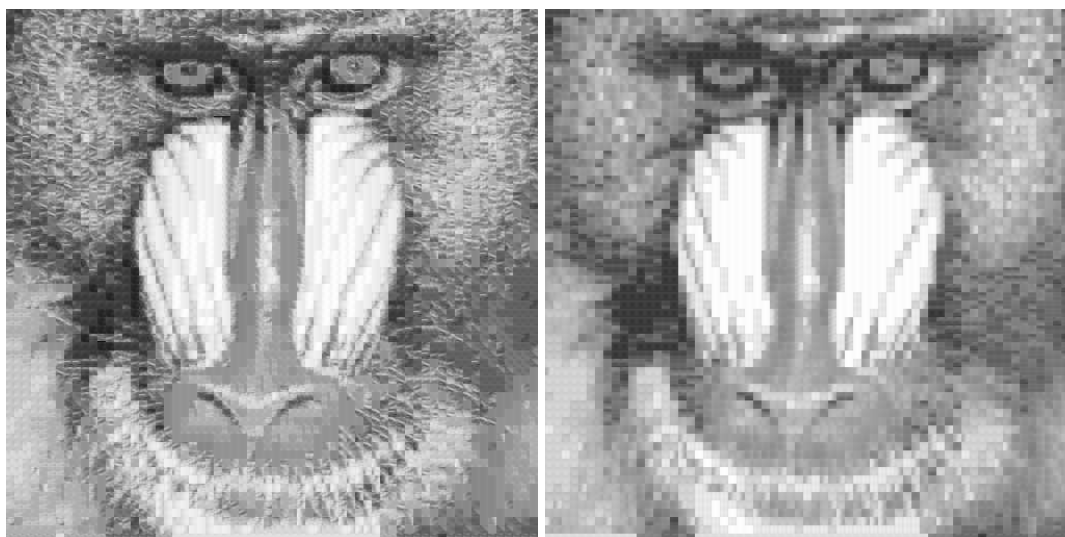
(a) FSCL ($K = 64$)(b) FuzzyCL ($K = 32$)

Figura 5.20: Imagens reconstruídas pelas redes FSCL com $K = 64$ protótipos e FuzzyCL com $K = 32$ protótipos segundo o critério MDL.

A conclusão geral desta seção é que as versões modificadas dos critérios de informação podem ser bastante úteis na escolha do número adequado de protótipos para uma dada tarefa de quantização vetorial, uma vez que as curvas destes critérios em função de K apresentam claramente um ponto ótimo (mínimo). Este mínimo correspondente ao número de protótipos que satisfaz simultaneamente uma condição de melhor reconstrução possível, sem aumentar demais o custo computacional do algoritmo.

5.5 Testes de Robustez ao Ruído

Em muitas aplicações práticas de quantização vetorial, a presença de ruído é uma realidade indesejável. Por isso, faz-se necessário avaliar alguns dos algoritmos neurais em situações que simulam a presença de ruído, seja ruído na imagem ou ruído no canal.

Quando o ruído está presente na imagem, são simulados ruídos do tipo branco gaussiano (aditivo) e sal-e-pimenta (multiplicativo). O objetivo é avaliar se a qualidade da imagem reconstruída fica muito comprometida devido à presença de ruído.

5.5.1 Ruído Gaussiano na Imagem *Peppers*

Ruído branco gaussiano é um tipo de ruído aditivo em que cada pixel de uma imagem tem seu valor alterado para mais ou para menos aleatoriamente seguindo uma distribuição de probabilidade de média zero e variância σ^2 . O resultado é uma imagem desfocada ou ligeiramente borrada. Quando os valores alterados dos *pixels* ultrapassam os limites inferior (zero) ou superior (255), eles são truncados nestes limites.

Para esta simulação é usada a imagem *Peppers* com dimensão 256×256 pixels, codificada em 256 níveis de cinza (8 *bits*). Para a quantização vetorial é adotado um tamanho de bloco de 4×4 *pixels*, o que resulta em um conjunto de treino com 4096 vetores de dimensão 16. O algoritmo escolhido para esta avaliação é a rede SOM com $K = 128$ protótipos.

É importante destacar que o treinamento do algoritmo é feito a partir da imagem sem ruído. O ruído é aplicado somente durante o processo de reconstrução da imagem. Assume-se que o canal é não-ruído. A imagem *Peppers* não-ruído usada para treinamento da rede SOM está mostrada na Figura 5.21.



Figura 5.21: Imagem *Peppers* sem ruído usada no treinamento com resolução 256×256 pixels.

Os resultados da reconstrução da imagem *Peppers* ruidosa pela rede SOM, para $K = 128$ protótipos e dois diferentes níveis de ruído, estão mostrados na Figura 5.22. Os bons resultados obtidos sugerem que a reconstrução não é muito afetada por ruído branco gaussiano. Resultados semelhantes foram obtidos para outras redes competitivas, não sendo mostrados aqui por

economia de espaço. Uma possível explicação para o bom desempenho pode estar no fato de os algoritmos estudados utilizarem a distância euclidiana como medida de dissimilaridade. Sabe-se que esta medida de distância é ótima para agrupamentos com formato circular, o que corresponde a grupos cujo os atributos são não-correlacionados gaussianamente distribuídos (WEBB, 2002).

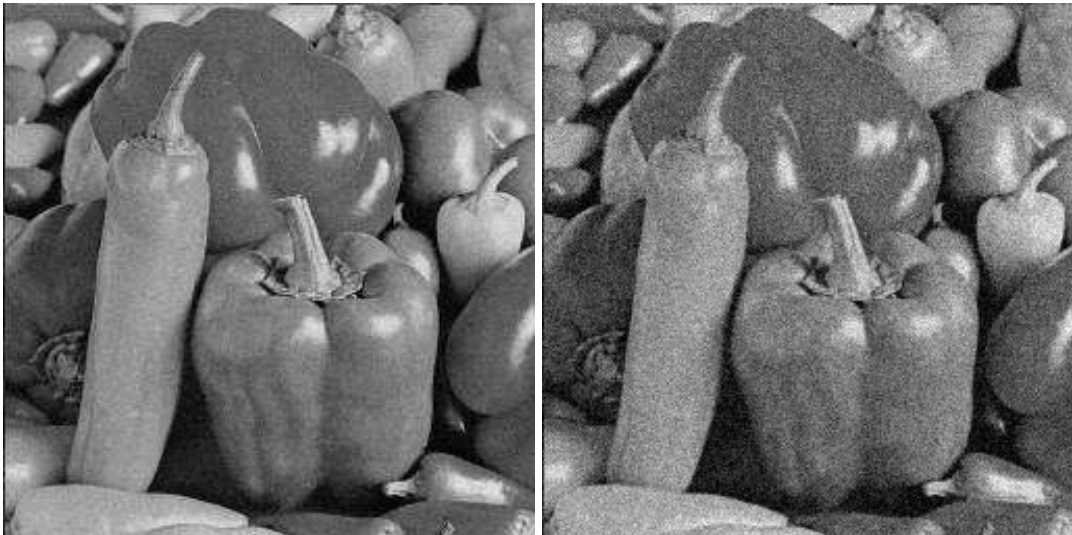
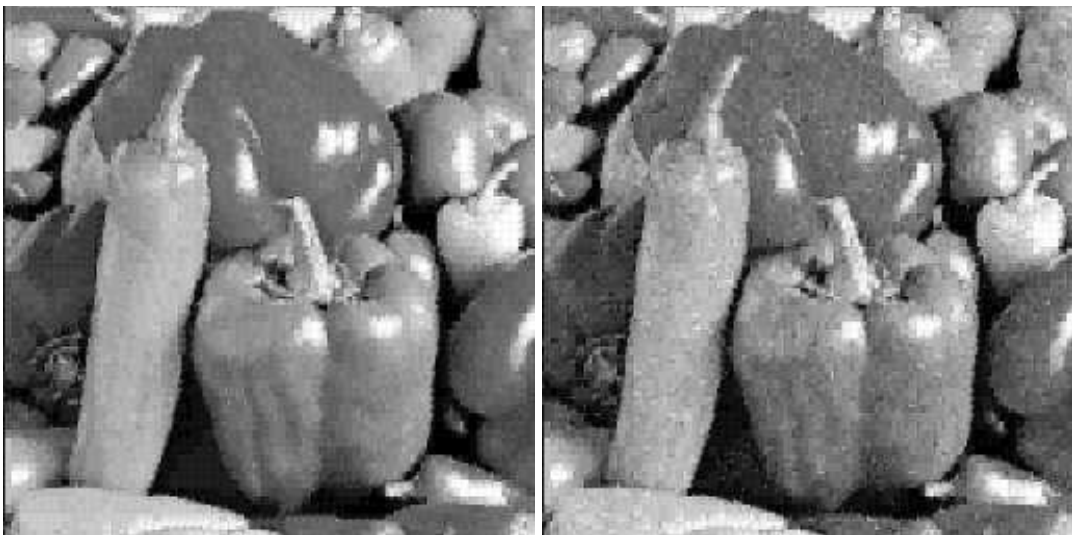
(a) $\sigma = 5$ (b) $\sigma = 15$ (c) $\sigma = 5$ (d) $\sigma = 15$

Figura 5.22: (a) Imagem *Peppers* com ruído gaussiano ($\sigma = 5$). (b) Imagem *Peppers* com ruído gaussiano ($\sigma = 15$). (c) Reconstrução da imagem ruidosa ($\sigma = 5$) pela rede SOM com $K = 128$ protótipos. (d) Reconstrução da imagem ruidosa ($\sigma = 15$) pela rede SOM com $K = 128$ protótipos.

5.5.2 Ruído Sal-e-Pimenta na Imagem *Lena*

Ruído sal-e-pimenta (*salt and pepper*) é um tipo de ruído multiplicativo em que cada pixel de uma imagem tem uma certa probabilidade de ser alterado para totalmente branco ou totalmente preto. O resultado é uma imagem com “chuviscos”

Para esta simulação é usada a já clássica imagem *Lena* com dimensão 256×256 pixels, codificada em 256 níveis de cinza (8 bits). Para a quantização vetorial foi adotado um tamanho de bloco de 4×4 pixels, o que resulta em um conjunto de treino com 4096 vetores de dimensão 16. Os algoritmos escolhidos para avaliação foram o algoritmo *K*-médias e a rede SOM, em configurações com $K = 32, 64$ e 128 protótipos.

É importante destacar que o treinamento do algoritmo é feito a partir da imagem sem ruído. O ruído é aplicado somente durante o processo de reconstrução da imagem. Assume-se que o canal é não-ruído. A imagem *Lena* não-ruído usada para treinamento e a imagem ruidosa usada para teste dos algoritmos estão mostradas na Figura 5.23.



(a) sem ruído

(b) sal-e-pimenta

Figura 5.23: (a) Imagem *Lena* sem ruído usada no treinamento. (b) Imagem *Lena* com ruído sal-e-pimenta usada no teste.

O ruído foi adicionado à imagem em duas execuções, nas quais existiram substituições radômicas de 2,5% dos pixels pretos e 2,5% dos pixels brancos em cada. Os resultados da reconstrução da imagem ruidosa pelo algoritmo *K*-médias e pela rede SOM, para diferentes valores de *K*, estão mostrados na Figura 5.24.

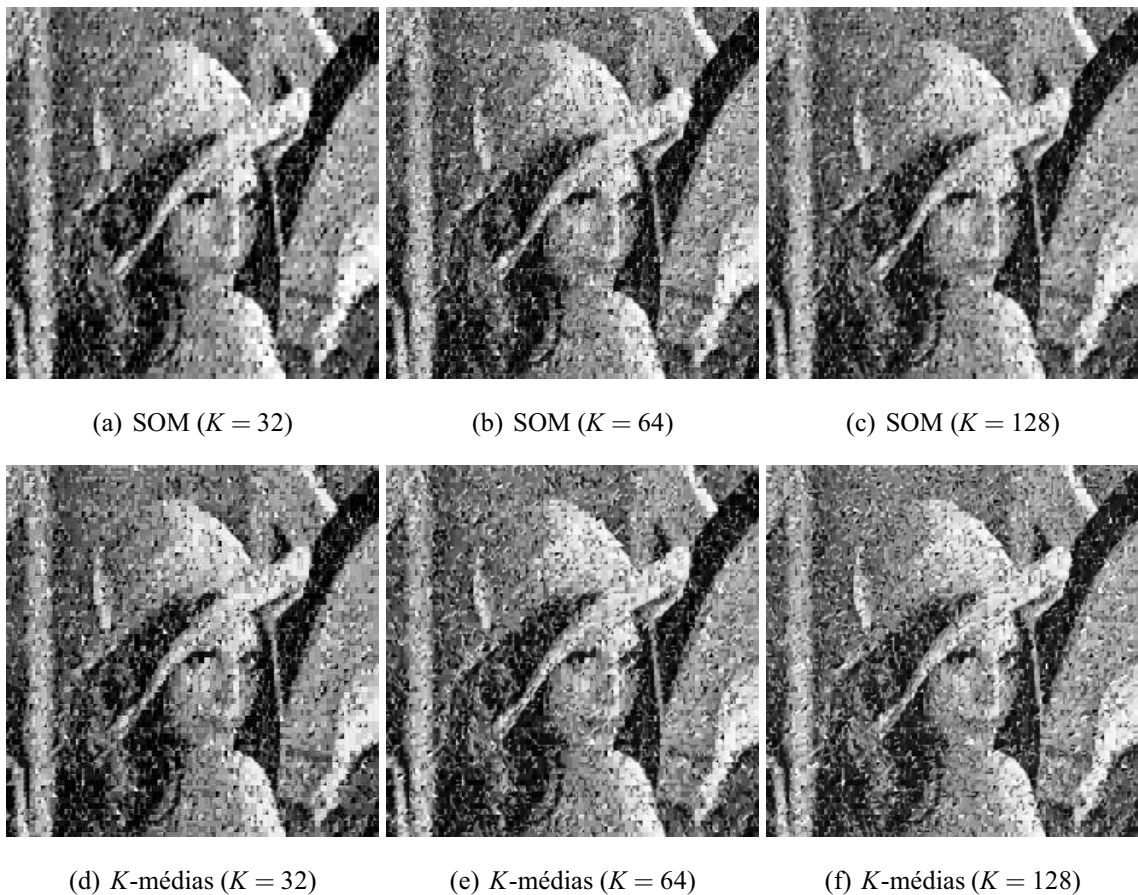


Figura 5.24: (a) Imagem reconstruída pela rede SOM com $K = 32$ protótipos. (b) Imagem reconstruída pela rede SOM com $K = 64$ protótipos. (c) Imagem reconstruída pela rede SOM com $K = 128$ protótipos. (d) Imagem reconstruída pelo algoritmo K -médias $K = 32$ protótipos. (e) Imagem reconstruída pelo algoritmo K -médias $K = 64$ protótipos. (f) Imagem reconstruída pelo algoritmo K -médias $K = 128$ protótipos.

Como era de se esperar, o ruído sal-e-pimenta distorce consideravelmente as imagens reconstruídas. Contudo, vale lembrar que este tipo de ruído é considerado um dos mais difíceis de tratar, devido a sua natureza multiplicativa. Além disso, não há uma etapa intermediária de filtragem entre a imagem e o algoritmo de reconstrução. Tendo isto em mente, pode-se afirmar que os resultados obtidos são considerados muito bons. Com relação ao desempenho, não há diferenças significativas entre o desempenho da rede SOM e do algoritmo K -médias.

5.5.3 Ruído no Canal na Imagem Mandrill

Simulação de ruído no canal corresponde a incluir algum tipo de interferência aleatória no código binário que representa o índice do protótipo vencedor que é transmitido ao longo do canal de comunicação. Como este índice é que é usado no receptor para fins de reconstrução da imagem, a chegada de um código binário alterado implicará na escolha do protótipo incorreto pelo receptor, distorcendo assim a imagem reconstruída. O mecanismo aleatório simulado corresponde à atribuição de uma probabilidade P de comutar cada *bit* da mensagem binária transmitida.

Para esta simulação é usada a imagem do macaco Mandrill com dimensão 256×256 pixels, codificada em 256 níveis de cinza (8 bits). Para a quantização vetorial foi adotado um tamanho de bloco de 4×4 pixels, o que resulta em um conjunto de treino com 4096 vetores de dimensão 16. O algoritmo escolhido para avaliação foi a rede WTA com $K = 16$ protótipos.

É importante destacar que o treinamento do algoritmo é feito sem ruído no canal. O ruído é aplicado somente durante o processo de reconstrução da imagem. Foram testadas as seguintes probabilidades de comutação de *bit*: $P = 0, 1, 0,2, 0,3, 0,4 e 0,5$.

A Figura 5.25 mostra como varia o índice PSNR em função da probabilidade de comutação do *bit*. Como era de se esperar, percebe-se claramente uma gradual queda na qualidade da imagem reconstruída. Contudo, o impacto visual dessa queda na qualidade da imagem reconstruída não é tão intenso, conforme pode ser verificado na Figura 5.26. Resultados semelhantes foram gerados pelos outros algoritmos.

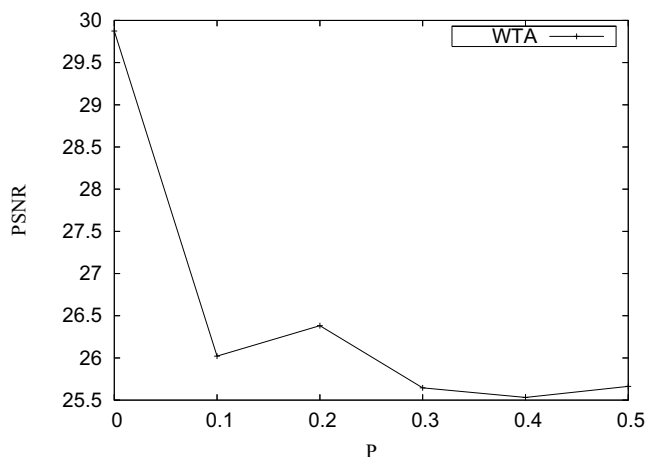


Figura 5.25: Curva do índice PSNR versus a probabilidade de comutação de *bit*.

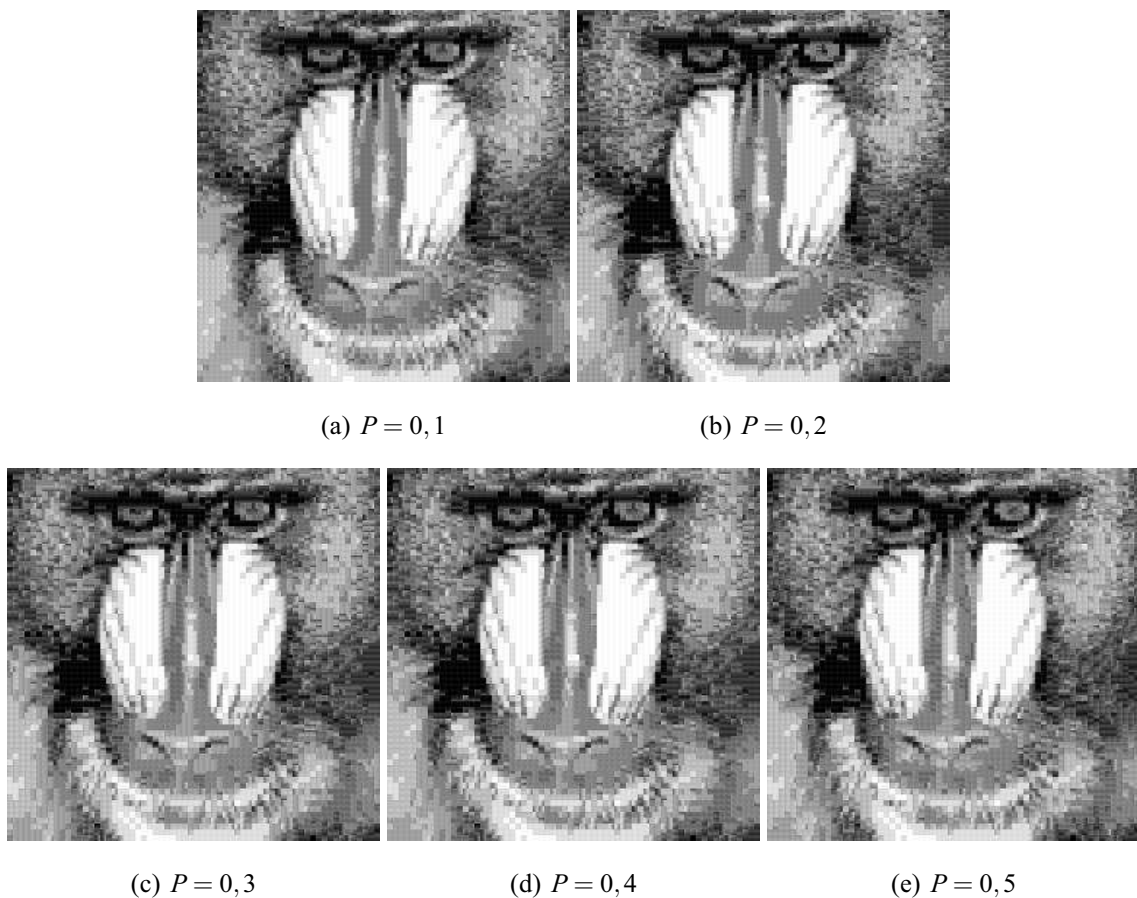


Figura 5.26: (a) Imagem reconstruída pela rede WTA para $P = 0,1$. (b) Imagem reconstruída pela rede WTA para $P = 0,2$. (c) Imagem reconstruída pela rede WTA para $P = 0,3$. (d) Imagem reconstruída pela rede WTA para $P = 0,4$. (e) Imagem reconstruída pela rede WTA para $P = 0,5$.

Desta forma, a conclusão geral desta seção é de que os algoritmos testados, independente dos paradigmas de aprendizado, são robustos a vários tipos de ruído na imagem (aditivo ou multiplicativo) ou ruído no canal. A escolha de um algoritmo em particular fica mais a critério do usuário ou de questões de custo computacional. Os algoritmos WTA, FSCL e K -médias são os que apresentam menor custo computacional.

6 *Conclusões e Perspectivas*

Este trabalho é focado num estudo comparativo do desempenho de algoritmos de redes neurais competitivas não-supervisionadas em problemas de quantização vetorial e aplicações correlatas, tais como análise de agrupamentos (*clustering*) e compressão de imagens. As redes neurais competitivas não-supervisionadas foram escolhidas para este trabalho, pois constituem uma classe de redes neurais artificiais usada para construir uma representação estatística compacta de um conjunto de dados de entrada não-rotulados. Problemas de análise de agrupamentos necessitam de ferramentas capazes separar de uma população de entidades (objetos ou indivíduos), representados numericamente por vetores de características (*feature vectors*), em determinados subgrupos ou categorias, a fim de se identificar e representar a estrutura organizacional subjacente a cada subgrupo. Já em problemas de compressão de imagens que são bastante dinâmicos, necessitamos de ferramentas que possam determinar os modelos mais eficientes, melhor relação custo computacional e qualidade da imagem resultante.

No Capítulo 2 fez-se uma breve descrição sobre as redes competitivas não-supervisionadas a serem avaliados nesta dissertação, a saber: WTA, SOM, FSCL, RPCL, Neural-Gas, bem como do algoritmo K-Médias e *Fuzzy Competitive Learning*. As principais características e idéias que levaram à proposição de cada algoritmo são também apresentadas. Este Capítulo é muito importante, pois ele serve de embasamento teórico dos algoritmos estatísticos e as arquiteturas de redes neurais competitivas avaliadas nesta dissertação, como forma de facilitar a compreensão dos métodos de validação de resultados em análise de agrupamentos e quantização vetorial que serão discutidos nos capítulos seguintes.

No Capítulo 3 são destacados critérios tanto de validação de agrupamentos como também aqueles utilizados para seleção de modelos. Com o objetivo de melhor apresentar os índices de validação utilizados nesta dissertação, primeiro foram estudadas as estratégias usadas para se validar um agrupamento, os critérios de validação de *clusterings*, e estimar a ordem dos modelos. Após a explanação dos critérios de validação e apresentação das estatísticas capazes de validar os agrupamento de forma a quantificar quão bom é um agrupamento, chamadas índices de validação, abordamos também alguns critérios de estimação da ordem dos modelos. Desta forma,

fez-se completa a base teórica que foi fundamental obtermos soluções e resultados apropriadas a cada simulação contidas Capítulos 4 e 5.

Redes neurais aplicadas para formação de agrupamentos e compressão de imagens modelagem nos Capítulos 4 e 5 são assim o objetivo principal deste trabalho, pois realizam tarefas difíceis, como *clustering* e quantização vetorial. São usadas para tal redes neurais competitivas não-supervisionadas, que constituem uma classe de redes neurais artificiais usada para construir uma representação estatística compacta de um conjunto de dados de entrada não-rotulados. O foco ainda maior desta dissertação é certificar a eficiência desta classe de redes neurais, através de critérios e metodologias de avaliação já existentes nas áreas de análise de agrupamentos e quantização vetorial. Assim, juntamente com a capacidade de criar agrupamentos, as redes neurais competitivas são uma excelente ferramenta nas tarefas de compressão de imagens analisadas neste estudo.

De particular interesse é o problema da seleção do número ótimo de neurônios para uma determinada tarefa de quantização vetorial. Como não há um método que funcione para todas as situações, a alternativa que resta é avaliar a influência que cada tipo de métrica exerce sobre um mesmo algoritmo. Por exemplo, os algoritmos de quantização vetorial supracitados são bastante usados em tarefas de *clustering*. Neste tipo de aplicação, a validação dos agrupamentos é feita com base em índices que quantificam os graus de compacidade e separabilidade entre agrupamentos, tais como Índice Dunn e Índice Davies-Bouldin (DB), como mostrado na Seção 4.5. Já em tarefas de compressão de imagens, determinado algoritmo de quantização vetorial é avaliado em função da qualidade da informação reconstruída, daí as métricas mais usadas serem o erro quadrático médio (MSE) ou a relação sinal-ruído de pico (PSNR).

Na Seção 5.3 está uma das contribuições desta dissertação, que foi verificar empiricamente que, enquanto os índices Dunn e DB não mostraram informações precisas favorecendo tanto arquiteturas com poucos protótipos como também o oposto, as métricas MSE e PSNR favoreceram somente as arquiteturas com quantidades bem maiores. A mesma seção esta dissertação propõe o uso do critério de informação de Akaike (AIC) e o critério de comprimento mínimo de descrição (MDL) de Rissanen para selecionar o número ótimo de protótipos. Este tipo de métrica mostrou-se útil na busca do número de protótipos que satisfaça simultaneamente critérios opostos, ou seja, critérios que buscam o menor erro de reconstrução a todo custo (MSE e PSNR) e critérios que buscam clusters mais compactos e coesos (Índices Dunn e DB). Como consequência para tarefas de quantização vetorial, o número de protótipos obtidos pelas métricas AIC e MDL é geralmente um valor intermediário, i.e. nem tão baixo ou impreciso quanto o sugerido pelos índices Dunn e DB, nem tão altos quanto o sugerido pelas métricas MSE e

PSNR.

Outra conclusão importante, baseando-se nos resultados da Seção 5.3 é que não necessariamente os algoritmos mais sofisticados do ponto de vista da modelagem, tais como as redes SOM e Neural-Gas, são os que apresentam melhores desempenhos. Os algoritmos FSCL e FuzzyCL são os que apresentam melhores resultados na qualidade da informação reconstruída, com a rede FSCL apresentando melhor relação custo-benefício, em função do seu menor custo computacional. Na rede neural FSCL, descrita na Seção 2.6, nota-se que a presença do fator $f_i(t)$ como elemento ponderador da distância euclidiana ajuda a minimizar a ocorrência de unidades mortas. Este fator fará com que outros neurônios, que antes eram selecionados com menor frequência, passem a também ser selecionados. Com o passar do tempo, todos os neurônios terão sido escolhidos em um número aproximadamente equivalente de vezes, tornando a competição mais justa. As redes Neural-Gas e RPCL obtiveram desempenhos pífios nos problemas de análise de agrupamentos, principalmente em função da elevada sensibilidade aos pesos iniciais.

Uma perspectiva deste trabalho de mestrado é a atribuição de um **conceito** aos *clusters* encontrados pelos algoritmos de **agrupamento** que, em geral, é uma tarefa complexa que deve ser realizada pelo especialista do domínio da aplicação. Sob esse aspecto, seria interessante que essa tarefa fosse totalmente automática. Mas, as abordagens de *clustering* tradicionais não possibilitam que essa análise seja feita automaticamente, pois não utilizam conhecimento a priori, mas somente os dados para a extração do conhecimento neles embutido. Como alternativa para facilitar essa tarefa de interpretação dos *clusters*, (MARTINS, 2003) propôs uma metodologia para auxiliar a encontrar uma descrição simbólica dos *clusters*. Segundo essa metodologia, inicialmente os dados não rotulados são submetidos à um algoritmo de *clustering* para obter um conjunto de agrupamentos. Esse resultado é utilizado como entrada para uma ferramenta que rotula os exemplos, adicionando um atributo cujo valor é o *cluster* ao qual o exemplo pertence. Esse novo conjunto de dados é, então, utilizado como entrada de algum algoritmo de aprendizado supervisionado, utilizando o novo atributo como atributo classe do conjunto de dados, com o intuito de encontrar uma descrição simbólica para os *clusters* gerados. Finalmente, com a obtenção dessa representação simbólica, a interpretação do conhecimento extraído torna-se mais simples e menos custosa.

Uma outra aplicação de interesse é entender a utilização dos métodos de seleção de modelos propostos para construir um mecanismo de reconstrução de imagens que sofreram ação de ruído tanto de fonte quanto de canal.

Referências Bibliográficas

- ABUT, H.; GRAY, R. M.; REBOLLEDO, G. Vector quantization of speech and speech-like waveforms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 30, n. 3, p. 423–435, 1982.
- AGUIRRE, L. A. *Introdução à Identificação de Sistemas*. [S.l.]: Editora da UFMG, 2000.
- AHALT, S. et al. Competitive learning algorithms for vector quantization. *Neural Networks*, v. 3, n. 3, p. 277–290, 1990.
- AKAIKE, H. Fitting autoregressive models for prediction. *Annals of Institute of Statistical Mathematics*, v. 21, p. 243–247, 1969.
- AKAIKE, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, v. 19, n. 6, p. 716–723, 1974.
- AKAIKE, H. Fitting autoregressive models for prediction. *Annals of Institute of Statistical Mathematics*, v. 21, p. 243–247, 1976.
- AMARI, S.-I. Field theory of self-organizing neural nets. *IEEE Transactions on Systems, Man and Cybernetics*, v. 13, n. 9-10, p. 741–748, 1983.
- BARALDI, A.; BLONDA, P. A survey of fuzzy clustering algorithms for pattern recognition - part i. *IEEE Transactions on Systems, Man and Cybernetics*, B-29, n. 6, p. 778–785, 1999.
- BARRETO, G. A. *Redes Neurais Nao-Supervisionadas Temporais para Identificacao e Controle de Sistemas Dinamicos*. Tese (Doutorado) — Universidade de São Paulo, Departamento de Engenharia Elétrica, São Carlos, 2003.
- CARPENTER, G.; GROSSBERG, S. Adaptive resonance theory. In: ARBIB, M. (Ed.). *The Handbook of Brain Theory and Neural Networks*. 2nd. ed. Cambridge, MA: MIT Press, 2003. p. 87–90.
- CHUNG, F. L.; LEE, T. Fuzzy competitive learning. *Neural Networks*, v. 7, n. 3, p. 539–551, 1994.
- COVER, T. M.; THOMAS, J. A. *Elements of information theory*. 1st. ed. New York: Wiley-Interscience, 1991.
- CRUTCHFIELD, J.; MCNAMARA, B. S. Equations of motion from a data series. *Complex Systems*, v. 1, p. 417–452, 1987.
- DARKEN, C.; MOODY, J. Fast adaptive k-means clustering: Some empirical results. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN'90)*. [S.l.: s.n.], 1990. v. 2, p. 233–238.

- DAVIES, D. L.; BOULDIN, D. W. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 1, n. 2, p. 95–104, 1979.
- DIMITRIADOU, E. et al. A quantitative comparison of functional MRI cluster analysis. *Artificial Intelligence in Medicine*, v. 31, p. 57–71, 2004.
- DONY, R.; HAYKIN, S. Neural network approaches to image compression. *Proceedings of the IEEE*, v. 83, n. 2, p. 288–303, 1995.
- DUNN, J. C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, v. 3, n. 3, p. 32–57, 1973.
- EVERITT, B. S. *Cluster Analysis*. 3rd. ed. [S.l.]: Edward Arnold, 1993.
- FACELI, K.; CARVALHO, A.; SOUTO, M. C. P. Validação de algoritmos de agrupamento. São Carlos, SP, n. 254, 2005. ISSN 0103-2569.
- FLECK, E. M. *Agrupamento e Visualização de Dados Sísmicos Através de Quantização Vetorial*. Tese (Doutorado) — Departamento de Engenharia Elétrica – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, 2004.
- FROTA, R. A. *Avaliação de algoritmos de redes neurais artificiais em tarefas de detecção de novidades: uma abordagem unificadora*. Dissertação (Mestrado) — Departamento de Engenharia de Teleinformática, Universidade de Federal do Ceará, 2005.
- FUKUSHIMA, K. Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, v. 20, n. 3-4, p. 121–136, 1975.
- FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, v. 36, n. 4, p. 193–202, 1980.
- GATTASS, R. Os mapas da visão. *Ciência Hoje*, v. 16, n. 94, 1993.
- GERSHO, A.; CUPERMAN, V. Vector quantization: A pattern-matching technique for speech coding. *IEEE Communications Magazine*, p. 15–20, 1983.
- GERSHO, A.; GRAY, R. M. Boston, MA: Kluwer Academic Publishers, 1992.
- GORDON, A. *Classification*. [S.l.]: Chapman and Hall/CRC, 1999.
- GRAY, R. M. Vector quantization. *IEEE Acoustics, Speech, and Signal Processing Magazine*, v. 1, n. 2, p. 4–29, 1984.
- GROSSBERG, S. Adaptive pattern classification and pattern recoding, I: Parallel development and coding of neural feature detectors. *Biological Cybernetics*, v. 23, n. 3, p. 121–134, 1976.
- GROSSBERG, S. Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, v. 23, p. 121–134, 1976.
- GROSSBERG, S. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, v. 11, p. 23–63, 1987.

- HAIR, J. F. et al. *Análise Multivariada de Dados*. 5a. ed. New York: Bookman, 2005.
- HAKEN, H. *Synergetics: Introduction and Advanced Topics*. 1st. ed. [S.l.]: Springer Verlag, 2004.
- HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. On clustering validation techniques. *Intelligent Information Systems*, v. 17, n. 2–3, p. 107–145, 2001.
- HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. Cluster validity methods: Part ii. *ACM SIGMOD Record*, v. 31, n. 3, p. 19–27, 2002.
- HAN, C.-C. et al. A novel approach for vector quantization using a neural network, mean shift, and principal component analysis-based seed re-initialization. *Signal Processing*, v. 87, p. 799–810, 2007.
- HARANDI, M. T.; GHARAVI-ALKHANSARI, M. Low bitrate image compression using self-organized Kohonen maps. In: *Proceedings of the 2003 International Conference on Image Processing*. [S.l.]: IEEE Press, 2003. v. 3, p. 267–270.
- HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2nd. ed. [S.l.]: Prentice-Hall, 1999.
- HERTZ, J.; KROGH, A.; PALMER, R. G. *Introduction to the theory of neural computation*. Redwood City, CA: Addison-Wesley, 1991.
- HOFMANN, T.; BUHMANN, J. Competitive learning algorithms for robust vector quantization. *IEEE Transactions on Signal Processing*, v. 46, n. 6, p. 1665–1675, 1998.
- HÖPPNER, F. et al. *Fuzzy Cluster Analysis*. 1st. ed. [S.l.]: Wiley, 1999.
- HU, X.; XU, L. Investigation on several model selection criteria for determining the number of cluster. *Neural Information Processing - Letters and Reviews*, v. 4, n. 1, p. 1–10, 2004.
- JAIN, A. K.; DUBES, R. C. *Algorithms for Clustering Data*. [S.l.]: Prentice-Hall, 1988.
- JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. *ACM Computing Surveys*, v. 31, n. 2, p. 264–323, 1999.
- KASHYAP, R. L. Inconsistency of the AIC rule for estimating the order of autoregressive models. *IEEE Transactions on Automatic Control*, v. 25, n. 5, p. 996–998, 1980.
- KASHYAP, R. L.; SUBAS, S. K. C.; YAO, S. B. Analysis of the multiple-attribute-tree data-base organization. *IEEE Transactions on Software Engineering*, v. 3, n. 6, p. 451–467, 1977.
- KELSO, J. S. *Dynamic Patterns: the self-organization of brain and behavior*. [S.l.]: MIT Press, 1995.
- KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, v. 43, n. 1, p. 59–69, 1982.
- KOHONEN, T. The self-organizing map. *Proceedings of the IEEE*, v. 78, n. 9, p. 1464–1480, 1990.

- KOHONEN, T. *Self-Organizing Maps*. 3rd. ed. [S.l.]: Springer-Verlag, 2001.
- KOSKO, B. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. [S.l.]: Prentice-Hall, 1992.
- LEUNG, C.-S.; CHAN, L.-W. Transmission of vector quantized data over noisy channel. *IEEE Transactions on Neural Networks*, v. 8, n. 3, p. 582–589, 1997.
- LEUNG, C.-S.; CHAN, L.-W.; MOW, W.-H. Soft-decoding SOM for VQ over wireless channels. *Neural Processing Letters*, v. 24, p. 179–192, 1997.
- MACKAY, D. J. C. *Information Theory, Inference, and Learning Algorithms*. Cambridge: Cambridge University Press, 2003.
- MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: Le Cam, L. M.; NEYMAN, J. (Ed.). *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, California. University of California Press: [s.n.], 1967. v. 1, p. 281–297.
- MADEIRO, F. et al. Complexidade computacional de um algoritmo competitivo aplicado ao projeto de quantizadores vetoriais. *Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais (SBRN)*, v. 1, n. 3, p. 180–194, 2004.
- MARTINETZ, T. M.; SCULTEN, K. J. A 'neural-gas' network learns topologies. *Artificial Neural Networks*, Amsterdam, p. 397–402, 1991.
- MARTINS, C. A. *Uma Abordagem para Pré-processamento de Dados Textuais em Algoritmos de Aprendizado*. Tese (Doutorado) — ICMC-USP, 2003.
- MAULIK, U.; BANDYOPADHYAY, S. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 12, p. 636–649, 2002.
- MUSZKAT, M. Como o cérebro aprende. In: _____. [S.l.]: Revista Viver Mente & Cérebro - Scientific American, 2006. v. 8, cap. Dinâmica do Conhecimento, p. 41–47.
- NASRABADI, N. M.; KING, R. A. Image coding using vector quantization: A review. *IEEE Transactions on Communications*, v. 36, n. 8, p. 957–971, 1988.
- PAKHIRA, M. K.; BANDYOPADHYAY, S.; MAULIK, U. Validity index for crisp and fuzzy clusters. *Pattern Recognition*, v. 37, n. 3, p. 487–501, 2004.
- PRINCIPE, J. C.; EULIANO, N. R.; LEFEBVRE, W. C. *Neural and Adaptive Systems: Fundamentals through Simulations*. [S.l.]: John Wiley & Sons, 2000.
- RAMAMURTHI, B.; GERSHO, A. Classified vector quantization of images. *IEEE Transactions on Communications*, v. 34, n. 11, p. 1105–1115, 1986.
- RISSANEN, J. Modeling by shortest data description. *Automatica*, v. 14, p. 465–471, 1978.
- RISSANEN, J. A universal prior for the integers and estimation by minimum description length. *Annals of Statistics*, v. 11, n. 4, p. 417–431, 1983.

- ROSENBLATT, F. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, p. 386–408, 1958.
- ROSENBLATT, F. Two theorems of statistical separability in the perceptron. In: *Proceedings of the Symposium on the Mechanization of Thought Processes*. [S.l.]: Her Majesty's Stationary Office, 1959. v. 1, p. 421–456.
- ROSENBLATT, F. *Principles of Neurodynamics*. New York: Spartan Books, 1962.
- ROUSSEEUW, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, v. 20, n. 1, p. 53–65, 1987.
- ROVETTA, S.; MASULLI, R. Vector quantization and fuzzy ranks for image reconstruction. *Image and Vision Computing*, v. 25, n. 2, p. 204–213, 2007.
- RUMELHART, D. E.; ZIPSER, D. Feature discovery by competitive learning. *Cognitive Science*, v. 9, n. 1, p. 75–112, 1985.
- SCHWARZ, G. Estimating the dimension of a model. *The Annals of Statistics*, v. 6, n. 2, p. 461–464, 1978.
- STARK, L.; OKAJIMA, M.; WHIPPLE, G. H. Computer pattern recognition techniques: electrocardiographic diagnosis. *Communications of the ACM*, v. 5, n. 10, p. 527–531, 1962.
- THEODORIDIS, S.; KOUTROUBAS, K. *Pattern recognition*. [S.l.]: Academic Press, 1999.
- VASUKI, A.; VANATHI, P. T. A review of vector quantization techniques. *IEEE Potentials*, v. 25, n. 4, p. 39–47, 2006.
- von der Malsburg, C. Self-organization of orientation sensitive cells in the striate cortex. *Biological Cybernetics*, v. 14, n. 2, p. 85–100, 1976.
- von der Malsburg, C. Self-organization in the brain. In: ARBIB, M. (Ed.). *The Handbook of Brain Theory and Neural Networks*. 2nd. ed. Cambridge, MA: MIT Press, 2003. p. 1002–1005.
- WEBB, A. *Statistical Pattern Recognition*. 2nd. ed. [S.l.]: John Wiley & Sons, 2002.
- XU, L.; KRZYZAK, A.; OJA, E. Rival penalized competitive learning for clustering analysis, RBF net and curve detection. *IEEE Transactions on Neural Networks*, v. 4, n. 4, p. 636–649, 1993.
- XU, Y.; BRERETON, R. G. A comparative study of cluster validation indices applied to genotyping data. *Chemometrics and Intelligent Laboratory Systems*, v. 78, n. 1-2, p. 30–40, 2005.
- YAIR, E.; ZEGER, K.; GERSHO, A. Competitive learning and soft competition for vector quantizer design. *IEEE Transactions on Signal Processing*, v. 40, n. 2, p. 294–309, 1992.
- ZADEH, L. A. Fuzzy sets. *Information and Control*, n. 8, p. 338–353, 1965.

APÊNDICE A – Avaliação do Erro de Quantização Durante o Treinamento

Este apêndice traz os resultados da avaliação de todos os algoritmos de análise de agrupamento em função do erro de quantização gerado ao longo das épocas de treinamento. Todas as figuras abaixo mostram a evolução do erro ao longo de cinquenta épocas.

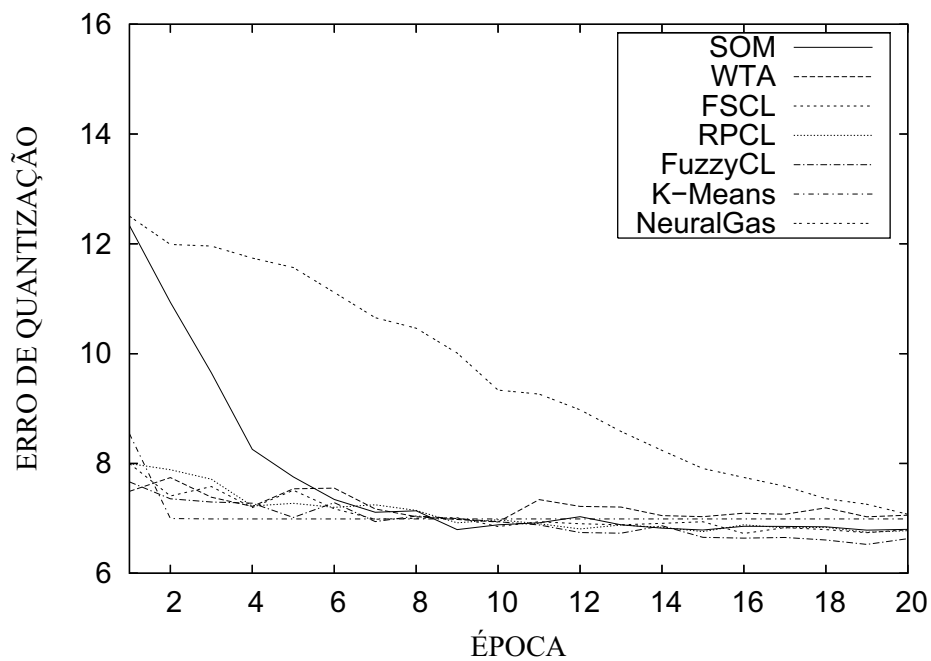


Figura A.1: Erro de Quantização para $K = 2$ protótipos.

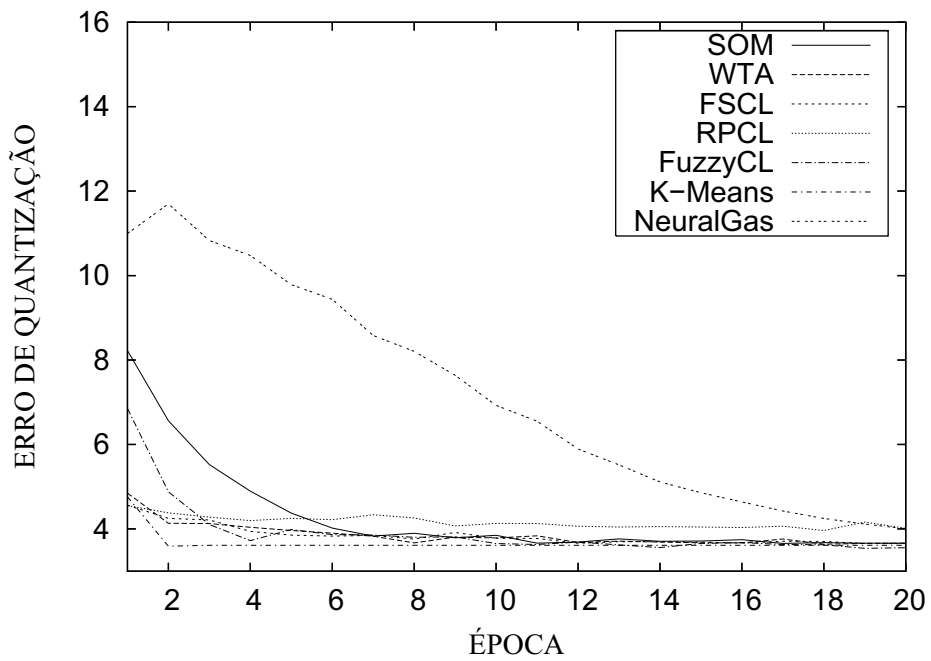


Figura A.2: Erro de quantização com $K = 3$ protótipos.

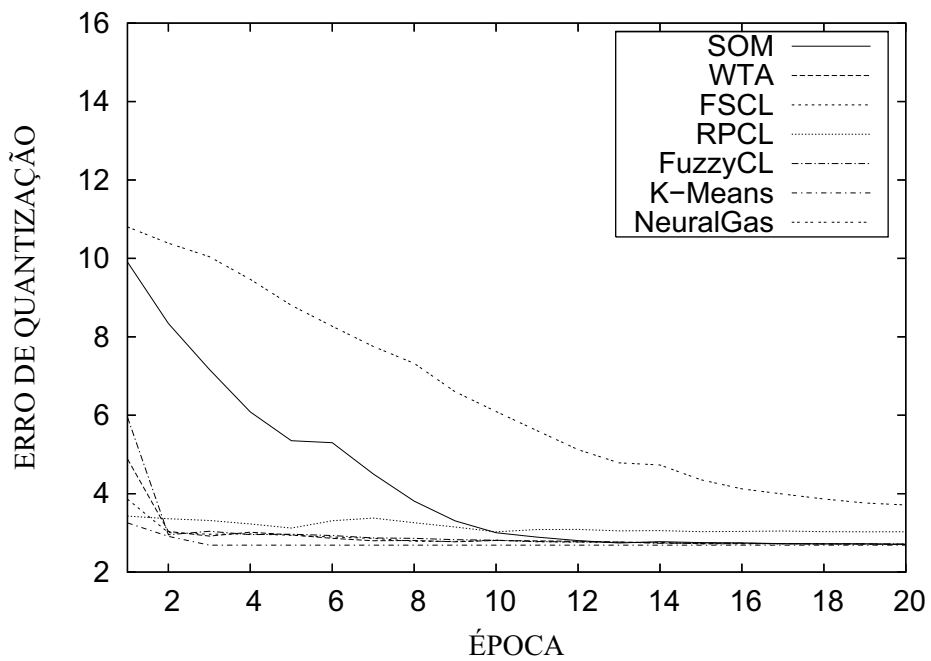


Figura A.3: Erro de quantização para $K = 4$ protótipos.

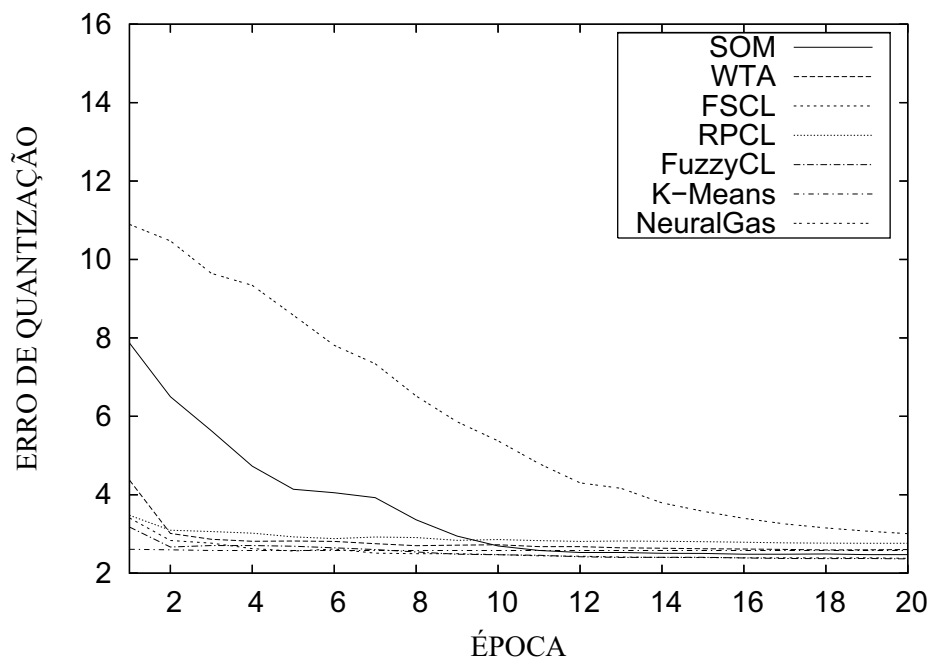


Figura A.4: Erro de quantização para $K = 5$ protótipos.

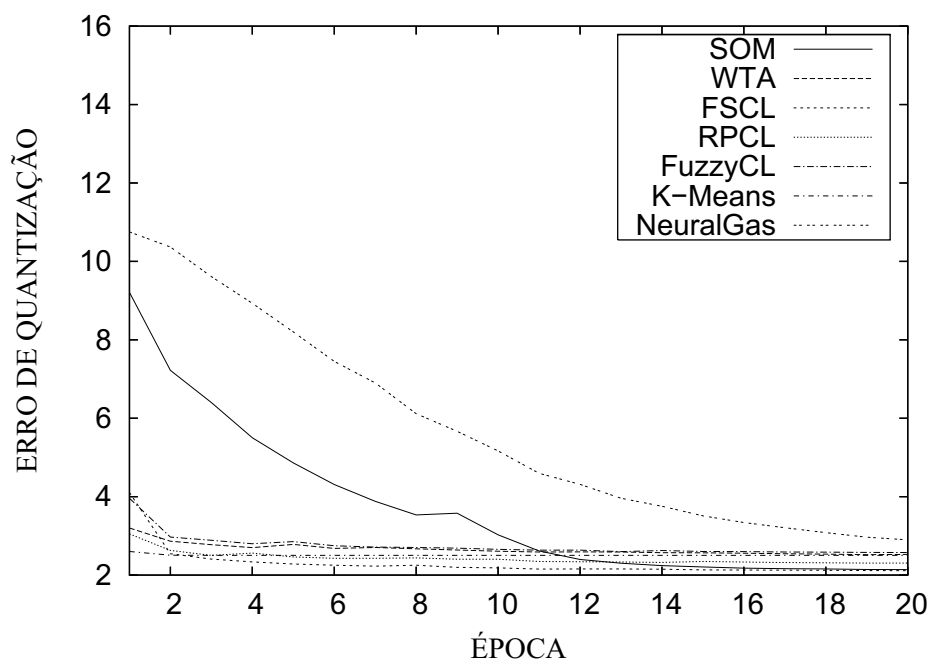


Figura A.5: Erro de quantização para $K = 6$ protótipos.

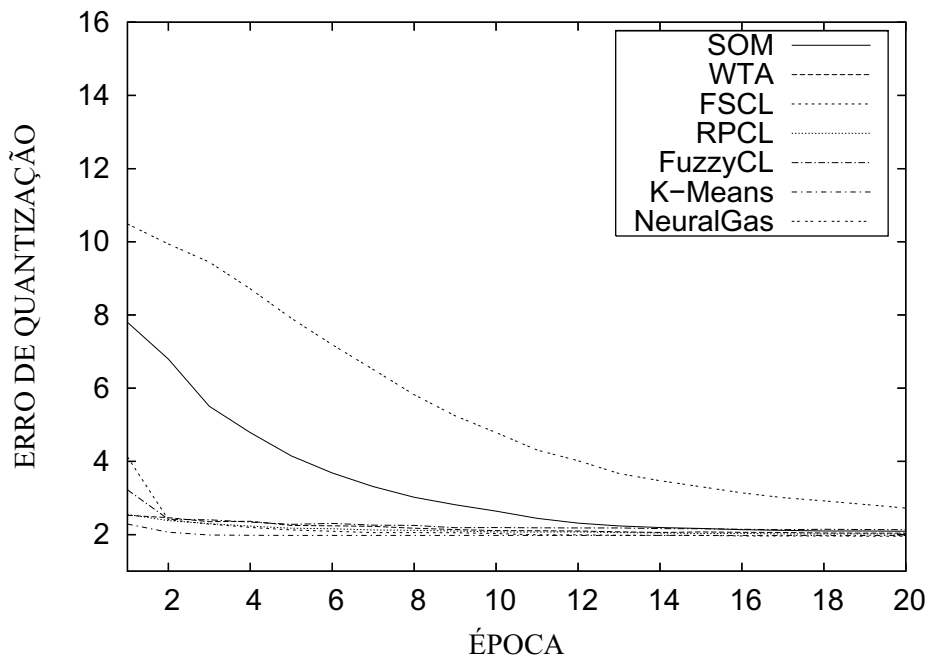


Figura A.6: Erro de quantização para $K = 7$ protótipos.

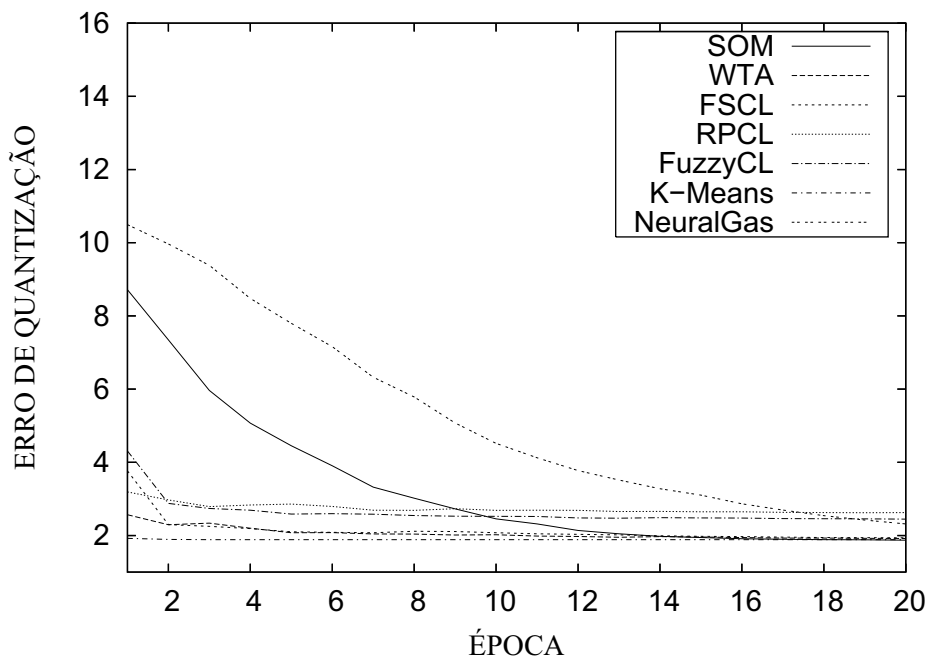


Figura A.7: Erro de quantização para $K = 8$ protótipos.

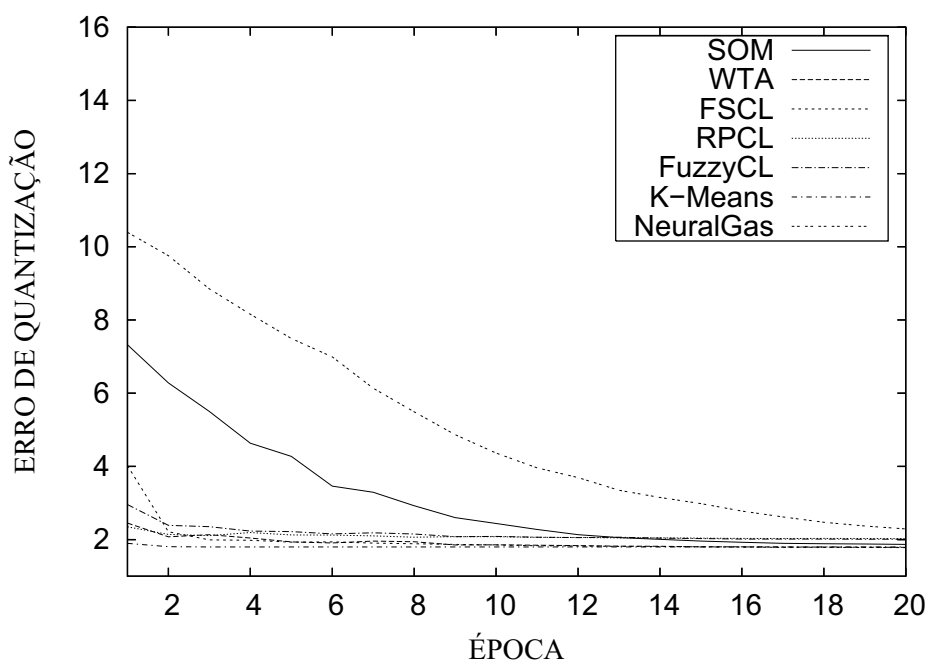


Figura A.8: Erro de quantização para $K = 9$ protótipos.

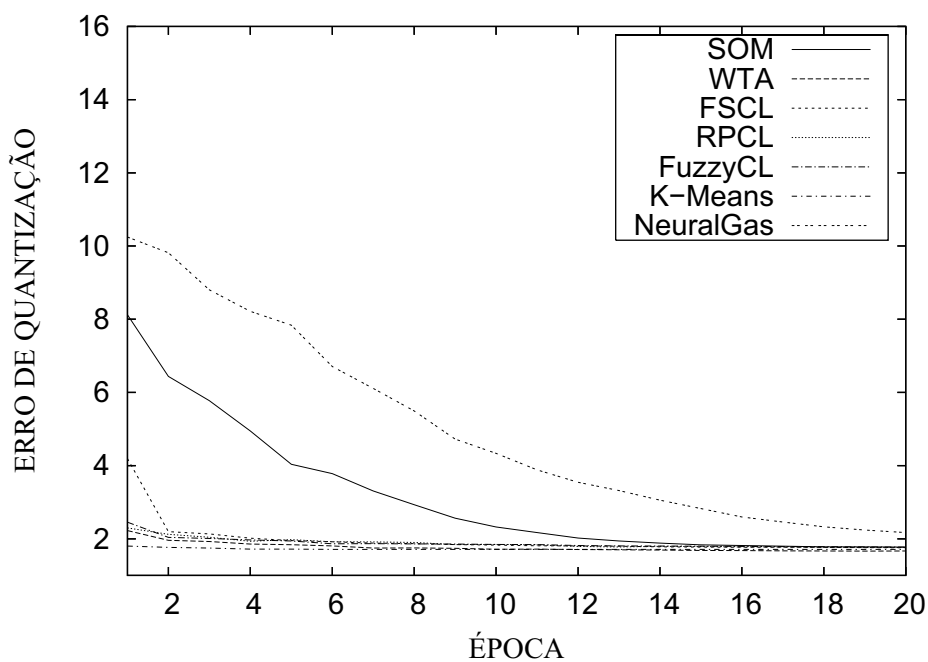


Figura A.9: Erro de quantização para $K = 10$ protótipos.

O padrão de comportamento das curvas é muito parecido, ou seja, todas têm uma tendência de decrescer com o aumento do número de épocas. O valor final de cada curva também não é muito significativo estatisticamente. Isto leva à conclusão de que o erro de quantização utilizado

isoladamente não é muito útil para se determinar o número de agrupamentos, pois o valor desta métrica tende a ser sempre menor à medida que $K \rightarrow N$.

APPENDIX B – Aplicação Java

A fim de automatizar os procedimentos de simulação, um aplicativo chamado *Java Neural Network - Competitive Learning* foi desenvolvido para executar os algoritmos e gerar os valores dos índices de validação. Utilizou-se para desenvolvimento um programa chamado *Eclipse IDE* e a linguagem JAVA que têm como principais características serem multiplataforma e *opensource* (código aberto); ou seja, pode ser executado nos principais sistemas operacionais sem custo financeiro.

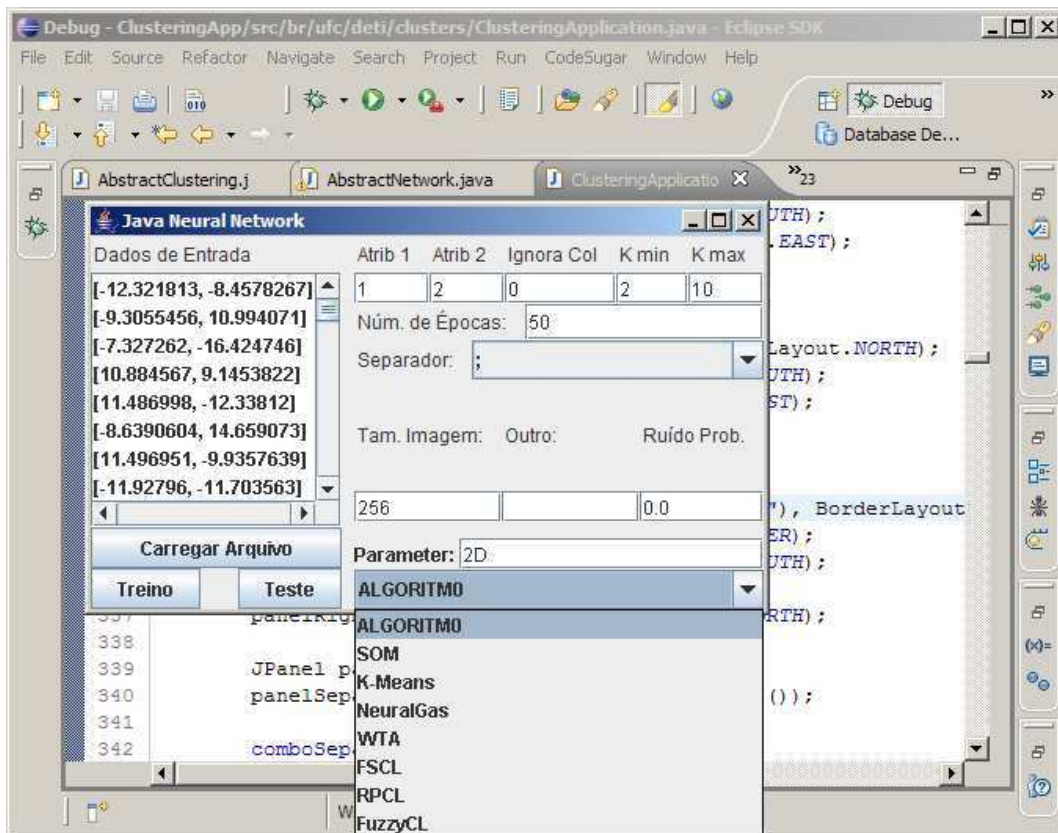


Figura B.1: Aplicação Java de Redes Neurais com *Eclipse IDE* de *background* (fundo).

Um total de sete algoritmos competitivos estão implementados na aplicação JAVA, conforme pode ser visto na Figura B.1. Esta aplicação possui vários campos que são preenchidos com valores padrões (*default*) ou com valores fornecidos pelo usuário. Botões para carregar dados, treinar/testar o algoritmo e área para visualização dos dados também estão disponíveis. Por fim, vale ressaltar que a mesma aplicação foi utilizada para simulações de análise de agrupamentos e de quantização vetorial.

APÊNDICE C – Canais de Comunicação

Canal, em comunicações, também chamado canal de comunicações, refere-se ao meio usado ao envio de informações do remetente (ou transmissor) para o destinatário (ou receptor).

Um canal pode ter muitas formas. Exemplos de canais de comunicações:

1. Uma conexão entre nós iniciais e finais de um circuito;
2. Um *buffer* pelo qual mensagens podem ser extraídas ou recuperadas.
3. Um simples ligação provida por um meio de transmissão:
 - (a) separação física, por um cabo ou;
 - (b) separação elétrica, por frequência ou divisão do tempo (multiplexação).
4. Um meio pelo qual são enviados sinais elétricos ou eletromagnéticos, usualmente separados um do outro de forma paralela;
5. Uma partição de um disco, como uma trilha ou uma banda, que está acessível para leitura ou escrita;
6. Em um sistema de comunicação, a parte que conecta um *data source* a um *data sink*;
7. Uma específica frequência de rádio ou banda de frequências, usualmente associada a uma determinada letra, número, ou palavra-código, que é definida por um padrão internacional. Exemplos:
 - (a) *Wi-Fi* consiste de canais não-licenciados: 1-13 de 2412MHz a 2484MHz em 5MHz de graus;
 - (b) Canais de televisão como os da TV Norte Americana: Canal 2 = 55,25MHz, Canal 13 = 211,25MHz.

8. Uma sala como na rede *Internet Relay Chat* (IRC), na qual os participantes podem se comunicar uns com os outros.

Todos esses canais de comunicação tem o objetivo de transferir informações. A informação transferida pelo canal é chamada de sinal.

C.1 Modelos de canal

Um canal pode ser modelado fisicamente, tentando-se simular os processos físicos que modificam o sinal transmitido. Por exemplo, em comunicações sem fio o canal pode ser modelado calculando-se a interferência externa de cada objeto no ambiente. Uma seqüência de números aleatórios poderia também ser adicionada para simular uma interferência externa e/ou ruído eletrônico no receptor.

Estatisticamente, um canal de comunicação normalmente é modelado como uma tríade composta de um alfabeto de entrada, um de saída e, para cada par de elementos (i, o) de entrada e saída, uma probabilidade de transição $p(o|i)$. Semanticamente, esta probabilidade de transmissão é a probabilidade de que o símbolo o ser recebido uma vez que i foi transmitido ao longo do canal.

A modelagem estatística e física podem ser combinadas. Por exemplo nas comunicações sem fios o canal é freqüentemente modelado por uma atenuação aleatória (conhecido como desbotamento) do sinal transmitido, seguido pelo ruído aditivo. O termo atenuação é uma simplificação dos processos físicos subjacentes e representa a alteração da energia do sinal, no decurso da transmissão. O ruído representa interferências externas e/ou eletrônicas no receptor. A atenuação também descreve o tempo relativo que leva para receber um sinal através do canal (tecnicamente chamada de fase turno). As estatísticas da atenuação aleatória são decididas por anteriores medições físicas ou simulações.

Os modelos de canal podem ser contínuos, nos quais não há limites para determinar precisamente os valores para o sinal.

Os canais de comunicação também são estudados em uma configuração discreta. Isto corresponde a abstração de um sistema de comunicação do mundo real no qual as conversões de blocos analógicos em digitais e digitais em analógicos estão fora do controle do projetista. O modelo matemático é constituído por uma probabilidade de transição que indica uma saída para cada possível seqüência de entradas do canal. Na teoria da informação, é comum começar com canais mais simples em que a distribuição de probabilidade de saída só depende do atual canal

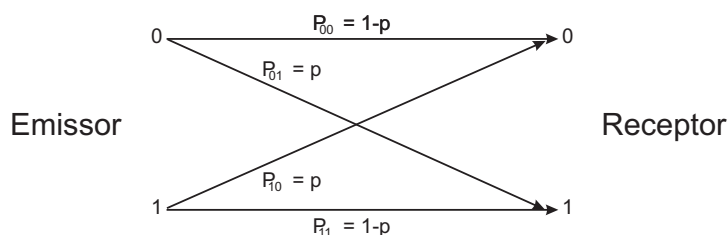


Figura C.1: Representação de um Canal Binário Simétrico.

de entrada.

Neste trabalho usamos especificamente um tipo de canal chamado canal binário simétrico.

C.2 Canal Binário Simétrico

Um canal binário simétrico (ou *binary symmetric channel-BSC*) é um modelo de canal de comunicação muito utilizado na teoria da codificação e teoria da informação (MACKAY, 2003; COVER; THOMAS, 1991). Neste modelo, um transmissor deseja enviar um *bit* (zero ou um), bem como o receptor deseja recebê-lo corretamente. Assume-se que o *bit* é normalmente transmitido corretamente, mas que ele terá uma pequena probabilidade de ser invertido, chamada **probabilidade de cruzamento**. Este canal é utilizado com frequência em teoria da informação, porque é um dos canais mais simples para analisar.

O *BSC* é um canal binário, ou seja, ele pode transmitir apenas dois tipos de símbolos, usualmente 0 e 1. Um canal não-binário seria capaz de transmitir mais de dois símbolos, ou até um número infinito de opções. A transmissão não é perfeita e, ocasionalmente chega ao receptor um *bit* errado.

Este canal é frequentemente utilizado porque ele é um dos mais simples canais ruidosos para se analisar teoricamente. Muitos problemas na teoria da comunicação pode ser reduzidos a um *BSC*. Por outro lado, ele é capaz de transmitir eficazmente sobre o *BSC* pode dar origem a soluções para mais complicada canais.

Um canal binário simétrico com probabilidade p para inversão do *bit* é um canal com entrada e saída binária e probabilidade de erro p (ver Figura C.1). Isto é, se X é o conjunto de informações de entrada a serem transmitidas e Y é o conjunto de informações recebidas. A seguir, o canal é caracterizado pelas seguintes probabilidades condicionais:

$$P_{00} = P(Y = 0|X = 0) = 1 - p \quad (\text{C.1})$$

$$P_{01} = P(Y = 1|X = 0) = p \quad (\text{C.2})$$

$$P_{10} = P(Y = 0|X = 1) = p \quad (\text{C.3})$$

$$P_{11} = P(Y = 1|X = 1) = 1 - p \quad (\text{C.4})$$

Assume-se que $0 \leq p \leq 1/2$. Se $p > 1/2$ o receptor inverte a saída (interpreta 1 quando é enviado 0, e vice-versa) e obtém-se um resultado equivalente quando $1 - p \leq 1/2$.