

# **Investigação de Técnicas de Extração e Seleção de Características e Classificadores Aplicados ao Problema de Classificação de Dígitos Manuscritos de Imagens de Documentos Históricos**

**Trabalho de Conclusão de Curso**

**Engenharia da Computação**

**Elias Rodrigues da Silva Júnior**  
**Orientador: Prof. Adriano Lorena Inácio de Oliveira**  
**Co-orientador: Prof. Wellington Pinheiro dos Santos**

**Recife, novembro de 2007**

# **Investigação de Técnicas de Extração e Seleção de Características e Classificadores Aplicados ao Problema de Classificação de Dígitos Manuscritos de Imagens de Documentos Históricos**

**Trabalho de Conclusão de Curso**

**Engenharia da Computação**

Este Projeto é apresentado como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

**Elias Rodrigues da Silva Júnior**  
**Orientador: Prof. Adriano Lorena Inácio de Oliveira**  
**Co-orientador: Prof. Wellington Pinheiro dos Santos**

**Recife, novembro de 2007**

Elias Rodrigues da Silva Júnior

**Investigação de Técnicas de  
Extração e Seleção de  
Características e Classificadores  
Aplicados ao Problema de  
Classificação de Dígitos Manuscritos  
de Imagens de Documentos  
Históricos**

## Resumo

Este trabalho apresenta um estudo de técnicas de extração e seleção de características, no intuito de possibilitar aumento do desempenho, ou, aumento da precisão de classificação de dígitos manuscritos de imagens de documentos históricos, por classificadores frequentemente encontrados na literatura. Este trabalho contribui para a preservação e difusão de documentos históricos e nosso objetivo é reconhecer datas manuscritas nos documentos para fins de indexação. Estudaremos a aplicação de métodos de extração de características, a saber, (1) análise dos componentes principais – *PCA*, (2) sub-amostragem, (3) morfologia matemática, no auxílio à tarefa de classificação realizada pelos classificadores (1) *k*-vizinhos mais próximos – *kNN*, (2) rede neural com função de base radial – *RBFN*, (3) máquina de vetor de suporte – *SVM* com função de núcleo polinomial e função de base radial – *RBF*. Adicionalmente, estudaremos um método que integra um algoritmo genético a uma *SVM* com núcleo *RBF*, com o objetivo de selecionar as características dos padrões, que são dígitos manuscritos de imagens de documentos históricos, e otimizar a configuração dos parâmetros da *SVM*, a fim de obter o melhor desempenho de classificação possível.

## Abstract

This work presents a study about feature extraction and selection techniques, in order to enable increased performance, or, increased accuracy of classification handwritten digits of historical document images, by often classifiers found at literature. This work contributes to historical documents preservation and diffusion and our goal is to recognize handwritten dates in the documents for indexing purposes. We will study the application of features extraction methods, namely, (1) principal components analysis – *PCA*, (2) undersampled bitmaps, (3) mathematical morphology, in aid to classification's task held by classifiers (1) k-nearest neighbors – *kNN*, (2) radial basis function networks – *RBFN*, (3) support vector machine – *SVM* with polynomial and radial basis function – *RBF* kernel functions. Additionally, we will study a method which integrates a genetic algorithm to *SVM* with *RBF* kernel, in goal to pattern features selection, which are handwritten digits of historical document images, and optimizing the configuration of *SVM* parameters, in order to obtain the best possible classification performance.

# Sumário

<b>Índice de Figuras</b>	<b>iv</b>
<b>Índice de Tabelas</b>	<b>vi</b>
<b>Tabela de Símbolos e Siglas</b>	<b>vii</b>
<b>1 Introdução</b>	<b>9</b>
1.1 Motivação	10
1.2 Objetivos do Trabalho	11
1.3 Estrutura do Trabalho	12
<b>2 Técnicas de Extração de Características</b>	<b>13</b>
2.1 Análise dos Componentes Principais	13
2.2 Sub-amostragem	14
2.3 Técnicas de Extração de Características Invariantes a Rotação, Translação e Escala	14
2.3.1 Morfologia Matemática	15
<b>3 Classificadores</b>	<b>18</b>
3.1 K-Vizinhos Mais Próximos	19
3.2 Redes Neurais com Funções de Bases Radiais	20
3.3 Máquina de Vetor de Suporte	21
<b>4 Abordagem Baseada em Algoritmos Genéticos para a otimização aplicada à Classificação de Padrões Manuscritos</b>	<b>24</b>
4.1 Breve introdução a Algoritmos Genéticos	24
4.2 Algoritmo Genético para Seleção de Características e Otimização dos Parâmetros de uma Máquina de Vetor de Suporte	26
<b>5 Experimentos</b>	<b>30</b>
5.1 Estudo de Extrações de Características e Classificadores na Classificação de Dígitos Manuscritos de Documentos Históricos	31
5.1.1 Configurações	32
5.1.2 Resultados	33
5.2 GA Integrado a SVM na Classificação de Dígitos Manuscritos de Documentos Históricos	35
5.2.1 Configurações	37
5.2.2 Resultados	39
<b>6 Conclusão e Trabalhos Futuros</b>	<b>40</b>
6.1 Trabalhos Futuros	41

# Índice de Figuras

<b>Figura 1.1</b> – atividades frequentemente envolvidas em sistemas de reconhecimento de manuscritos.	10
<b>Figura 1.2</b> - exemplos da escrita a mão livre do número 1896 efetuados por uma mesma pessoa.	11
<b>Figura 2.1</b> – processo de extração de características por sub-amostragem. Onde (a) é a representação matricial de um padrão sem extração de características, (b) divisão do padrão em blocos não sobrepostos e (c) nova representação do padrão onde cada célula guarda a soma de pixels ativos em cada bloco, oriundos da representação inicial. Para esta imagem estamos considerando os pixels pretos como os pixels ativos.	14
<b>Figura 2.2</b> - (a) dígito manuscrito retirado de documento histórico, (b), (c) e (d) são transformações de rotação, translação e escala na imagem (a) respectivamente.	15
<b>Figura 2.3</b> – (a) exemplo de padrão de dígito manuscrito, (b) rotação de 90° sobre (a), (c) translação de (a) e (d) redimensionamento de (a) por fator 1,2. Todas com seus espectros de padrões logo abaixo, obtidos com elemento estruturante cruz 3x3.	17
<b>Figura 2.4</b> – (a) exemplo de padrão de dígito manuscrito, (b) rotação de 90° sobre (a), (c) translação de (a) e (d) redimensionamento de (a) por fator 1,2. Todas com seus espectros de padrões logo abaixo, obtidos com elemento estruturante cruz 3x3.	17
<b>Figura 3.1</b> – ilustração as fases de treinamento e teste do processo de classificação no paradigma de aprendizado supervisionado.	19
<b>Figura 3.2</b> – ilustra o processo de classificação de padrões de novidades.	19
<b>Figura 3.3</b> - processo de classificação do 3NN.	20
<b>Figura 3.4</b> – arquitetura de uma RBFN completamente conectada.	20
<b>Figura 3.5</b> – ilustração de um conjunto de treinamento de um problema linearmente separável, hiperplano de separação encontrado pelo SVM e sua margem de separação $\rho$ .	22
<b>Figura 3.6</b> – ilustração de um conjunto de treinamento de um problema não linearmente separável, hiperplano de separação encontrado pelo SVM e sua margem de separação.	22
<b>Figura 4.1</b> – representação binária das variáveis do problema para criação do cromossomo.	25
<b>Figura 4.2</b> – ciclo evolucionário do GA.	25
<b>Figura 4.3</b> - operação genética de cruzamento com um ponto de corte.	26
<b>Figura 4.4</b> - operação genética de mutação.	26
<b>Figura 4.5</b> – modelo do cromossomo, onde $n_C$ e $n_\gamma$ são os números de bits usados na representação dos parâmetros $C$ e $\gamma$ respectivamente; e $n_A$ é a quantidade de atributos do padrão.	27

<b>Figura 4.6</b> – diagrama esquemático das atividades envolvidas na execução do método de integração de um algoritmo genético a uma máquina de vetor de suporte na tarefa de classificação de padrões.	29
<b>Figura 5.1</b> – método de avaliação por validação cruzada estratificado em 10 grupos.	31
<b>Figura 5.2</b> – elementos estruturantes usados neste trabalho. Sendo (a) elemento estruturante vertical, (b) elemento estruturante inclinado a direita, (c) elemento estruturante horizontal, (d) elemento estruturante inclinado a esquerda.	32
<b>Figura 5.3</b> – operação genética de cruzamento com dois pontos de corte aleatórios.	36
<b>Figura 5.4</b> – esquema do GA integrado a SVM.	37

# Índice de Tabelas

<b>Tabela 5.1</b> - métodos de extração de características usados neste trabalho.	32
<b>Tabela 5.2</b> - classificadores usados neste trabalho.	32
<b>Tabela 5.3</b> – bases referenciadas neste trabalho.	33
<b>Tabela 5.4</b> – classificadores e suas configurações.	33
<b>Tabela 5.5</b> – desempenho de classificação obtido pelos classificadores em suas configurações propostas, sobre as bases de dados com extração de características abordadas. A tabela relata os erros obtidos pela validação cruzada estratificada em 10 grupos.	34
<b>Tabela 5.6</b> – menor erro percentual de classificação obtido em cada base de dados e com qual classificador foi obtido.	34
<b>Tabela 5.7</b> – quantidade de bits usados na representação binária dos parâmetros $C$ e $\gamma$ , dos atributos dos padrões e do cromossomo.	38
<b>Tabela 5.8</b> – configurações dos parâmetros $C$ e $\gamma$ , e a precisão obtida.	38
<b>Tabela 5.9</b> – configuração do experimento.	38
<b>Tabela 5.10</b> – resultados obtidos após a execução das dez instâncias do GA.	39
<b>Tabela 5.11</b> – média de desvio padrão do percentual de acerto e quantidade de características selecionadas.	39

# Tabela de Símbolos e Siglas

(Dispostos por ordem de aparição no texto)

OCR – Optical Character Recognition (reconhecimento ótico de caracteres)  
ICR – Intelligent Character Recognition (reconhecimento inteligente de caracteres)  
RGB – Red Green Blue (vermelho verde azul)  
DPI – dots per inch (pontos por polegada)  
SBRN – Simpósio Brasileiro de Redes Neurais  
PCA – Principal Componentes Analysis (análise dos componentes principais)  
kNN – k Nearest Neighbors (k vizinhos mais próximos)  
NN – Nearest Neighbor (vizinho mais próximo)  
RBFN – Radial Basis Functions Network (rede neural com função de base radial)  
RBF – Radial Basis Function (função de base radial)  
SVM – Support Vector Machine (máquina de vetor de suporte)  
SRM – Structural Risk Minimization (minimização do risco estrutural)  
GA – Genetic Algorithm (algoritmo genético)

# Agradecimentos

Agradeço primeiramente a Deus, por toda a sorte de provisão que me tem favorecido, possibilitando meu crescimento em todos os aspectos da vida. Obrigado.

Obrigado pai e mãe, por todo carinho, apoio, incentivo, conselhos e tudo o mais que não se expressa facilmente em palavras relativo a vossa participação em minha vida. Obrigado.

Ao meu irmão e irmãs, que sempre me ensinaram, que sempre são influências positivas em minha vida, contribuindo diretamente para a conclusão desta etapa e de igual modo para as etapas que virão. Obrigado.

Aos amigos, pela companhia, pela diversão, pelo apoio no sofrimento, pela fuleiragem, pelo “– tá bom, vamos estudar?” “– agora não... deu banzo!!”. Valeu galera.

Aos professores Adriano e Wellington pela orientação, estímulo e paciência. Obrigado.

# Capítulo 1

## Introdução

Reconhecimento de padrões, especificamente de caracteres manuscritos, é tema vastamente abordado na literatura. Devido a isto podemos ver o uso de sistemas, até em nível comercial, que englobam esta área de conhecimento, a saber, sistemas de reconhecimento ótico de caracteres – *OCR – Optical Character Recognition* [1], sistemas de reconhecimento inteligente de caracteres – *ICR – Intelligent Character Recognition* [2].

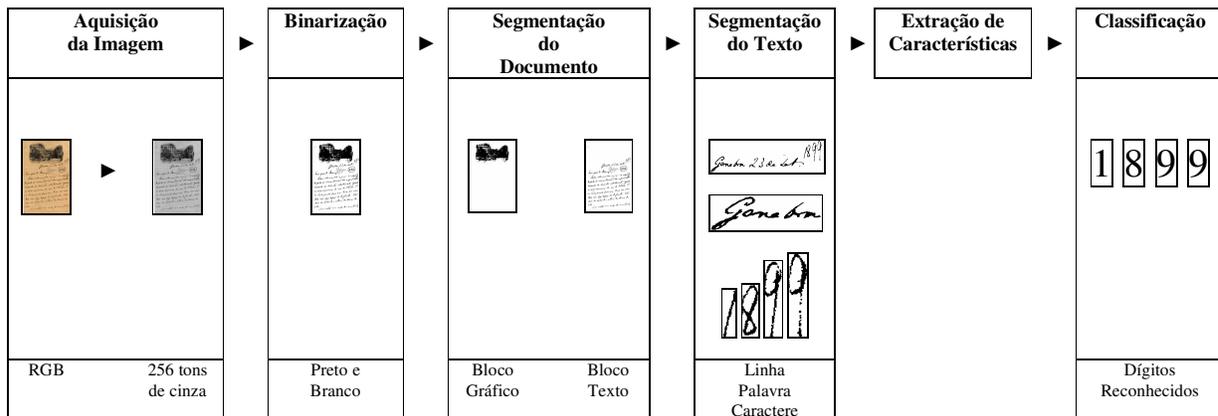
A Figura 1.1 ilustra, de modo geral, as atividades envolvidas nos sistemas de reconhecimento de dígitos manuscritos, onde a primeira preocupação é com a aquisição da imagem, pois o sistema de representação das cores, o formato de compressão e a resolução usada influenciam diretamente na qualidade da imagem. A tabela de cores usualmente encontrada nas digitalizadoras – *scanners* – é a tabela de cores reais – *True-Color* – também conhecida com – *RGB – Red Green Blue* – que consegue representar 16777216 variações de cores. O formato de compressão das imagens de documentos históricos deve ser ponderado a fim de equilibrar duas grandezas inversamente proporcionais que são, (1) a força da compressão e (2) a quantidade de perda de informação devido à compressão. As imagens que utilizamos para formar nossa base de dados foram armazenadas em JPEG com 1% de perda e a resolução usada foi de 200 *dpi – dots per inch – pontos por polegada* [3]. Para fins de reconhecimento é comum converter a imagem de RGB para tons de cinza com 256 níveis de intensidade, pois a maioria dos algoritmos de binarização são desenvolvidos para receber este tipo de representação na imagem de entrada.

O processo de binarização descarta o fundo da imagem, uma vez que ele é irrelevante no reconhecimento dos caracteres. Em seguida temos a fase de segmentação do documento, onde são identificados na imagem as regiões correspondentes a texto e demais elementos não textuais. Após isso se dá a segmentação das regiões identificadas como texto, onde os algoritmos são classificados quanto à granularidade do texto que se destinam a segmentar, ou seja, (1) algoritmos de segmentação de linhas, (2) palavras ou (3) caracteres.

A fase de extração de características opera sobre as imagens oriundas da fase de segmentação de texto, buscando gerar uma representação vetorial de cada imagem, de modo que cada célula do vetor de características expresse uma determinada relação entre os *pixels* da imagem, buscando desse modo gerar uma representação mais significativa e eficiente do que a simples expressão dos *pixels* ativos ou não.

Concluído isto, os vetores de características são passados ao classificador que é um modelo matemático que procura de maneira eficiente aprender a agrupar os vetores de características em conjuntos de semelhança, chamados de classes, de tal modo que cada elemento de um mesmo conjunto seja a representação de uma imagem de mesmo tipo, ou seja, no caso de

reconhecimento de dígitos, haverá dez conjuntos representado cada dígito possível (0 - 9) e espera-se que cada vetor de características seja agrupado na classe referente a imagem, dígito, da qual ele foi criado. Assim, a resposta do classificador a uma dada representação da imagem, chamada de padrão, é o rótulo da classe a qual o padrão pertence.



**Figura 1.1** – atividades frequentemente envolvidas em sistemas de reconhecimento de manuscritos.

Neste trabalho daremos continuidade à investigação sobre métodos de extração de características e classificadores no auxílio a tarefa de classificação de dígitos manuscritos que realizamos em nosso período de iniciação científica, cujos resultados estão relatados num artigo que foi apresentado no Simpósio Brasileiro de Redes Neurais – SBRN'2006. Neste trabalho anterior, [3], nós estudamos os métodos de extração de características (1) análise dos componentes principais e (2) sub-amostragem, e sua influência na precisão de classificação dos classificadores (1) k-vizinhos mais próximos, (2) redes neurais com funções de bases radiais, (3) máquina de vetor de suporte com núcleo polinomial e com núcleo radial.

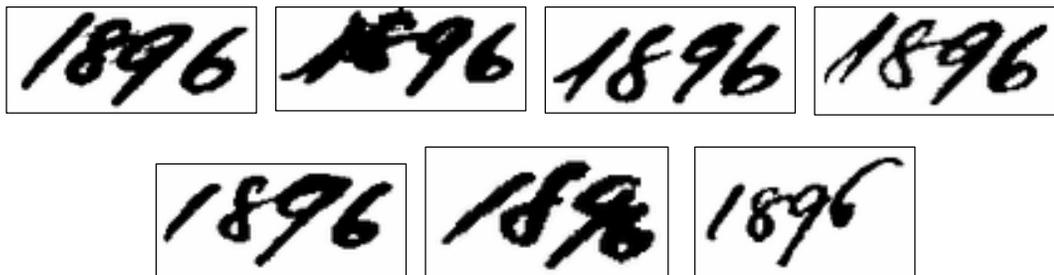
A base de dados usada em [3] e que também usaremos aqui, foi extraída de documentos históricos do final do século XIX e início do século XX. Todos os dígitos extraídos foram segmentados manualmente e a base formada possui um total de duzentas imagens de dígitos, divididos de maneira igualitária em dez conjuntos de acordo com o cardinal do dígito (0 - 9).

Adicionalmente ao que foi feito em [3], nós vamos acrescentar ao estudo técnica de extração de características baseada em morfologia matemática; que são invariantes a fenômenos de translação rotação e escala, que são fenômenos comuns a escrita livre e que influenciam a classificação. Além disso, investigaremos a aplicação de um algoritmo genético na tarefa de (1) selecionar os *pixels* das imagens dos dígitos que melhor contribuem para a classificação das mesmas e (2) otimizar a configuração dos parâmetros de uma máquina de vetor de suporte com núcleo radial para a classificação segundo esta seleção de *pixels* [4].

## 1.1 Motivação

O problema do reconhecimento de dígitos manuscritos tem sido largamente estudado e podemos encontrar na literatura diversos modelos propostos para a resolução dele [5], [6] e [7]. Todavia, devido à complexidade do problema, nenhuma das soluções existentes se mostra absoluta na resolução de todas as situações ou variações que o problema pode assumir, o que motiva novas pesquisas.

No universo do reconhecimento de caracteres manuscritos, as imagens possuem alto grau de complexidade, variando quanto (1) à forma da letra, que pode variar de pessoa para pessoa ou para a mesma pessoa, (2) quanto ao estilo da letra, podendo ser de forma, extenso, itálico e (3) quanto à espessura; além de que são comuns efeitos de rotação, translação e escala devido a fenômenos da escrita. Isto faz com que sistemas de reconhecimento de caracteres manuscritos, ou mesmo de dígitos manuscritos, sejam de igual modo complexos a fim de tratar essas situações. A Figura 1.2 mostra alguns exemplos da escrita a mão livre do número 1896 efetuados por uma mesma pessoa, onde podemos ver diferentes formas de grafia dos numerais, níveis de inclinação, espessuras, tamanhos.



**Figura 1.2** - exemplos da escrita a mão livre do número 1896 efetuados por uma mesma pessoa.

A dimensionalidade dos vetores de características tem efeito direto sobre o esforço computacional no cálculo da solução, já que quanto maior for a dimensionalidade maior será o processamento necessário sobre eles. A dimensionalidade dos vetores de características também influi sobre a capacidade de generalização do sistema, uma vez que, um espaço de características de dimensões reduzidas pode não estar considerando informações relevantes à distinção entre as classes, porém um espaço de características de alta dimensionalidade pode considerar informações irrelevantes a distinção entre elas, o que não resultaria em um aumento da precisão de classificação do sistema, podendo até diminuí-la [3]. Por outro lado, como problemas de reconhecimento são, em geral, não-linearmente separáveis, a abordagem feita sobre eles deve ser mais complexa a fim de obter melhor classificação.

Com isso, a fim de encontrar representação para os padrões de modo a aumentar o grau de separação linear entre as classes com o intuito de obter maior precisão na classificação dos padrões com abordagens mais simples, podemos modelar o espaço de características segundo o teorema da cobertura – *Cover's theorem* – que diz que a probabilidade de que classes sejam linearmente separáveis aumenta quando as características são mapeadas em um espaço de maior dimensionalidade [8].

## 1.2 Objetivos do Trabalho

Primeiramente, nós objetivamos estender o trabalho realizado em [3] acrescentando a este uma abordagem de extração de características invariante a fenômenos de rotação, translação e escala nos dígitos. Com isso pretendemos verificar se essa nova abordagem é mais eficiente na distinção dos dígitos em suas classes possibilitando melhor classificação.

Em segundo lugar, objetivamos utilizar um algoritmo genético para selecionar as características (*pixels*) dos padrões (imagens de dígitos manuscritos) e otimizar os parâmetros de uma máquina de vetor de suporte a fim de obter o melhor desempenho de classificação possível.

O método de integração de um algoritmo genético a uma máquina de vetor de suporte na tarefa de classificação de padrões pode ser visto em [4] aplicado a diversos problemas de classificação, obtendo excelentes resultados. Todavia em [4] o método não foi utilizado para a classificação de dígitos manuscritos de imagens de documentos históricos, assim acreditamos que empregá-lo aqui pode gerar melhores resultados do que os vistos em [3].

## 1.3 Estrutura do Trabalho

Nosso trabalho esta disposto da seguinte forma:

- **Capítulo 2** – Neste capítulo apresentamos o conceito e finalidades das técnicas de extração de características e mostramos aquelas que foram usadas aqui.
- **Capítulo 3** – Apresentamos o conceito de classificadores e descrevemos aqueles que compõem este trabalho.
- **Capítulo 4** – Elucidamos uma técnica que integra algoritmo genético com máquina de vetor de suporte aplicado à tarefa de seleção de características dos padrões e otimização dos parâmetros da máquina de vetor de suporte, com o fim de melhorar a precisão na classificação dos padrões.
- **Capítulo 5** – Trata da realização dos experimentos para este trabalho, mostrando as configurações utilizadas, os resultados e a avaliação deles.
- **Capítulo 6** – Neste capítulo apresentaremos as conclusões do nosso trabalho, suas contribuições e os trabalhos futuros que serão realizados em decorrência deste.

Adicionalmente este trabalho inclui:

- **Apêndice A** – Apresenta os conceitos básicos de morfologia matemática que dão base as operações morfológicas definidas no capítulo 2.

## Capítulo 2

# Técnicas de Extração de Características

Extração de características é toda a técnica na qual se obtém informação não explícita de um conjunto de dados. Por exemplo, para o caso de imagens binárias, uma característica não explícita seria a contagem da quantidade de *pixels* ativos, que exprime o valor da área da imagem.

No universo de visão computacional, o conjunto de dados é formado por imagens que, a princípio, possuem considerável quantidade de informação irrelevante, o que tornaria qualquer computação sobre elas mais custosa. Neste contexto, as técnicas de extração de características visam diminuir a dimensionalidade das características a serem computadas [9].

Essas técnicas promovem uma série de transformações nas imagens podendo resultar em outra imagem ou um valor, ambos representam perda de informação em relação à imagem original. Deste modo, o processo de extrair características deve ser dirigido afim de que as características remanescentes, ou geradas, sejam de tal modo que possibilitem ao classificador generalizar o problema eficientemente e obter taxa de acerto elevada.

### 2.1 Análise dos Componentes Principais

Análise dos componentes principais *PCA – Principal Components Analysis* é um método clássico de extração de características não-supervisionado que deriva novas variáveis, em ordem decrescente de importância, que são combinações lineares das variáveis originais e são não-correlatas [10]. A redução da dimensão é realizada pela escolha de autovetores suficientemente significativos para alguma porcentagem da variância no dado original. Na maioria das aplicações esta porcentagem é 95%. Esta é também a porcentagem considerada aqui.

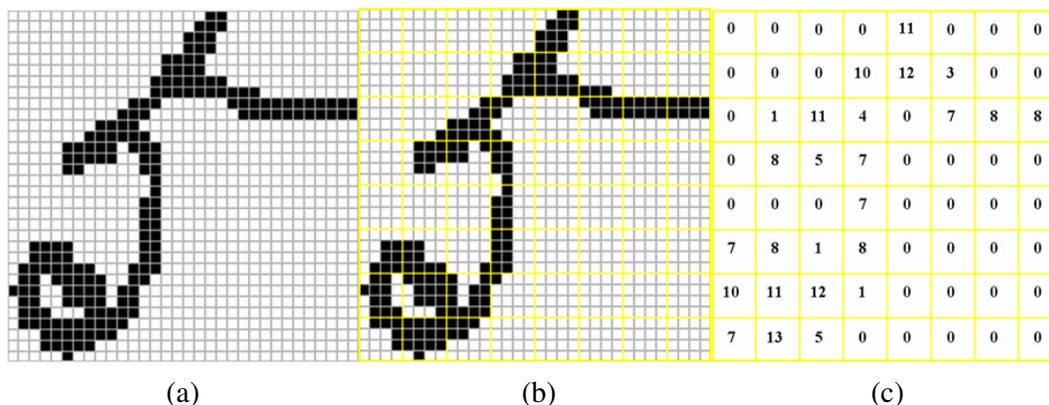
Ou seja, o método de extração de características pela análise dos componentes principais, consiste em encontrar um conjunto de eixos canônicos (ortogonais entre si) de tal modo que este novo conjunto de eixos expresse um limiar (95%) da variância dos padrões. Em [10], [11] e [12] é possível encontrar mais detalhes sobre este método.

## 2.2 Sub-amostragem

Sub-amostragem – *Undersampled Bitmaps* – é uma técnica de extração de características simples de programar e de baixo custo computacional, que tem sido usada em auxílio à tarefa de classificação de dígitos manuscritos, tendo obtido resultados melhores do que técnicas mais complexas [3].

Ela consiste em dividir cada imagem em blocos não sobrepostos de mesmas dimensões. Subseqüentemente, o número de pixels ativos é contado em cada bloco. Isto gera uma matriz de entrada de dimensões reduzidas, em relação à matriz original, com cada elemento da nova matriz sendo um inteiro variando de zero ao número de elementos do bloco.

A Figura 2.1 ilustra o processo de extração de características por sub-amostragem de acordo como foi feito neste trabalho, onde (a) é um padrão com dimensões de 32x32 pixels, (b) divisão da representação inicial em blocos não sobrepostos de 4x4 pixels e (c) é a soma dos pixels ativos em cada bloco, resultando numa matriz de dimensões 8x8 onde seus valores variam de 0 a 16.



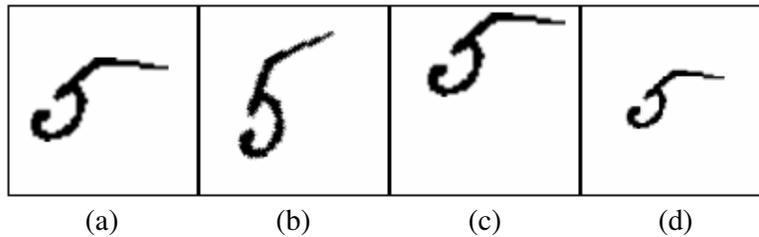
**Figura 2.1** – processo de extração de características por sub-amostragem. Onde (a) é a representação matricial de um padrão sem extração de características, (b) divisão do padrão em blocos não sobrepostos e (c) nova representação do padrão onde cada célula guarda a soma de pixels ativos em cada bloco, oriundos da representação inicial. Para esta imagem estamos considerando os pixels pretos como os pixels ativos.

## 2.3 Técnicas de Extração de Características Invariantes a Rotação, Translação e Escala

Quando padrões de uma mesma classe são morfologicamente semelhantes a distância entre eles é menor, fazendo com que seu agrupamento seja mais coeso do que quando a morfologia entre eles é mais distinta, fazendo com que sua distribuição no hiperespaço seja mais dispersa. A coesão ou dispersão dos padrões em seus agrupamentos influencia diretamente na sofisticação necessária ao classificador para a generalização. Em problemas de reconhecimento de dígitos manuscritos, a escrita livre gera freqüentes fenômenos de rotação, translação e escala, que aumentam a distinção dos padrões, aumentando o espalhamento dos padrões de mesma classe, requerendo assim, como foi dito, classificadores mais sofisticados.

A Figura 2.2 mostra (a) um dígito manuscrito e seus fenômenos de rotação, translação e escala, respectivamente (b), (c) e (d), podemos entender desta figura que mesmo se tratando do mesmo dígito, a ocorrência desses fenômenos gera imagens cujas representações vetoriais diferem da representação vetorial de imagem original (a). Logo serão tratadas como padrões diferentes e estarão sujeitas a classificações diferentes.

A proposta das técnicas de extração de características invariantes aos fenômenos de rotação, translação e escala é gerar um conjunto de informações para cada padrão que o identifique univocamente sem sofrer a influência dos fenômenos acima citados, e deste modo, além de reduzir a dimensionalidade da representação original dos padrões, fazer com que no novo hiperespaço os padrões de mesma classe estejam mais coesos, de forma que, mesmo aplicados a classificadores menos sofisticados a classificação obtenha elevada precisão.



**Figura 2.2** - (a) dígito manuscrito retirado de documento histórico, (b), (c) e (d) são transformações de rotação, translação e escala na imagem (a) respectivamente.

### 2.3.1 Morfologia Matemática

A utilização de morfologia matemática para extração de características que usamos aqui se apóia na técnica denominada granulometria. A granulometria é a aplicação de sucessivas transformações de abertura numa imagem [13], denotada por  $\Psi$  tal que:

$$\Psi_k(I) = \gamma_g^k(I), \quad k \geq 0 \tag{Equação 2.1}$$

$$\gamma_g^k(I) = \underbrace{\delta_g \delta_g \dots \delta_g}_k \left( \overbrace{\varepsilon_g \varepsilon_g \dots \varepsilon_g}^k (I) \right) \tag{Equação 2.2}$$

$$\gamma_g^0(I) = I \tag{Equação 2.3}$$

onde  $\gamma_g$  significa a aplicação abertura com elemento estruturante  $g$ . A abertura e suas operações fundamentais de morfologia matemática,  $\delta$  (dilatação) e  $\varepsilon$  (erosão), estão definidas no apêndice A deste trabalho.

Depois de calculadas as  $k$ -ésimas granulometrias para o padrão (ou imagem) montamos um conjunto das intensidades das imagens resultantes das granulometrias, que no caso de imagens binárias trata-se de calcular a área de pixels ativos. Denotaremos a operação para o cálculo da área como  $A$  tal que:

$$\gamma: I_{m \times n} \times I_{r \times s} \rightarrow I_{m \times n} \quad \text{Então}$$

$$A_k(I) = \sum_{i,j} \left[ \left( \Psi_k(I) \right) (i, j) \right], \quad k \geq 0 \tag{Equação 2.4}$$

$\forall i, j$  com  $1 \leq i \leq m$  e  $1 \leq j \leq n$  sendo,

$$A_0(I) = \sum_{i,j} [I(i,j)] \quad \text{Equação 2.5}$$

ou seja, a área da imagem original.

Observemos que os resultados contidos no conjunto das áreas já guardam invariância quanto à translação e a rotação, pois na ocorrência de algum desses dois fenômenos, ou mesmo ambos, o valor da área resultante é sempre o mesmo.

Com o objetivo de alcançar a invariância quanto à escala definimos uma grandeza relativa que representa uma função de distribuição acumulada discreta da imagem [13] denotada por  $\Xi$  tal que:

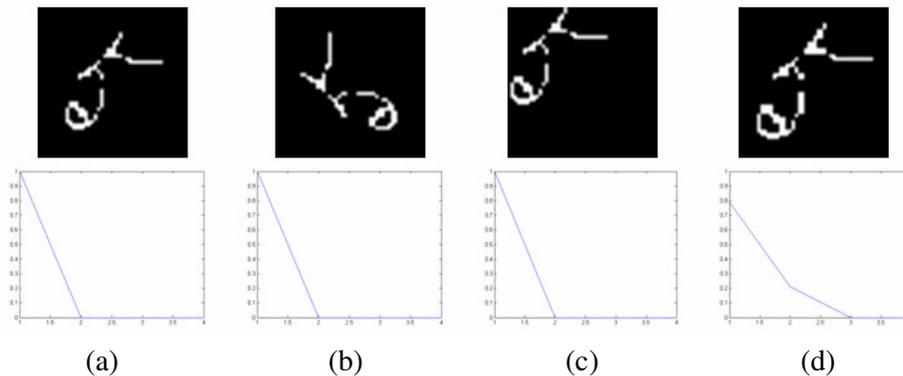
$$\Xi(k) = 1 - \frac{A_k(I)}{A_0(I)}, \quad k \geq 0 \quad \text{Equação 2.6}$$

Em seguida, após calcularmos todas as  $k$  distribuições, calculamos o espectro de padrões do padrão, denotado  $\xi$  por tal que:

$$\xi(k) = \Xi(k+1) - \Xi(k), \quad k \geq 0 \quad \text{Equação 2.7}$$

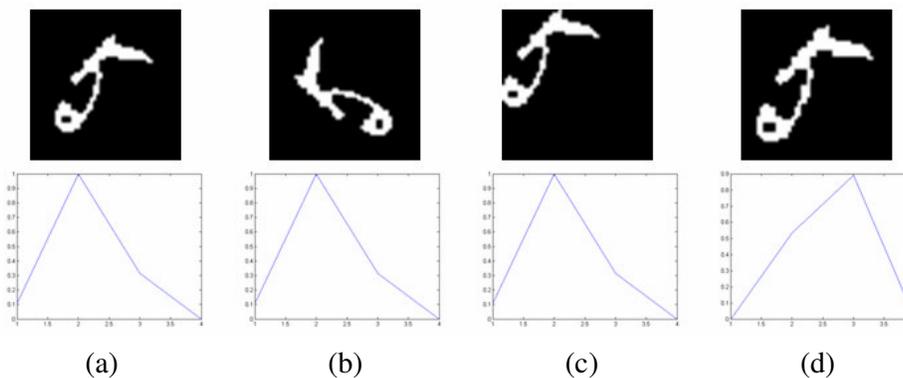
Sendo também, o espectro de padrões, único para a imagem e o elemento estruturante usado, o que nos permite usá-lo como novo conjunto de características do padrão. Deste modo, o elemento estruturante utilizado influencia diretamente nos resultados obtidos, fazendo necessário um estudo da influência de vários elementos estruturantes ao problema abordado.

A Figura 2.3 ilustra a invariância aos fenômenos de rotação, translação e escala obtida pelo uso da técnica de extração de características baseada em morfologia matemática, onde a Figura 2.3 (a) mostra um padrão e seu espectro de padrões, logo abaixo, único para esta imagem e elemento estruturante usado, sendo este a cruz  $3 \times 3$ . As demais imagens são resultados de transformações sobre a Figura 2.3 (a), sendo a Figura 2.3 (b) uma rotação de  $90^\circ$ , a Figura 2.3 (c) uma translação e a Figura 2.3 (d) um redimensionamento de fator 1,2. Podemos perceber que para Figura 2.3 (b) e Figura 2.3 (c), o espectro de padrões foi idêntico ao da Figura 2.3 (a), e que para a Figura 2.3 (d), mesmo sendo o espectro de padrões diferente, ele se assemelha. Cada um dos espectros de padrões da Figura 2.3 e Figura 2.4 tem eixo das abscissas representando o valor do parâmetro “ $k$ ” usado na operação de abertura e o eixo das ordenadas expressa a resposta do espectro de padrões para um dado “ $k$ ”, ou seja,  $\xi(k)$ .



**Figura 2.3** – (a) exemplo de padrão de dígito manuscrito, (b) rotação de  $90^\circ$  sobre (a), (c) translação de (a) e (d) redimensionamento de (a) por fator 1,2. Todas com seus espectros de padrões logo abaixo, obtidos com elemento estruturante cruz  $3 \times 3$ .

A Figura 2.4 é uma ilustração similar a Figura 2.3, e à Figura 2.4 (a) aplicamos as mesmas transformações aplicadas na Figura 2.3 (a), obtendo resultados de igual modo similares. Todavia, pela observação de ambas, podemos notar a singularidade de seus espectros de padrões, indicando a unicidade conferida por esta abordagem a representação dos padrões.



**Figura 2.4** – (a) exemplo de padrão de dígito manuscrito, (b) rotação de  $90^\circ$  sobre (a), (c) translação de (a) e (d) redimensionamento de (a) por fator 1,2. Todas com seus espectros de padrões logo abaixo, obtidos com elemento estruturante cruz  $3 \times 3$ .

## Capítulo 3

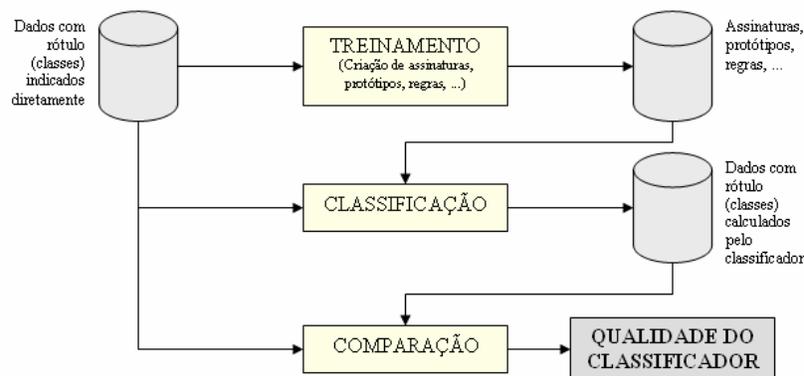
# Classificadores

Técnicas de classificação (classificadores) são um tipo de modelo de inferência que implementam uma estratégia eficiente para a computação da generalização das relações entre atributos de um conjunto de instâncias (padrões), com a finalidade de agrupar novas instâncias em subconjuntos (classes) onde as relações entre os atributos dos seus integrantes se assemelham. Deste modo um classificador avalia a pertinência de uma nova instância aos subconjuntos do universo das instâncias.

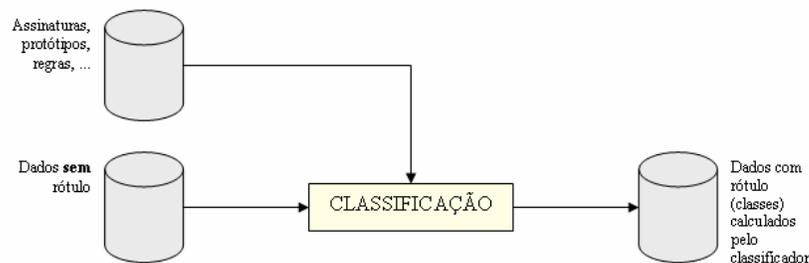
O processo de classificação pode ser estruturado segundo os paradigmas de aprendizagem supervisionado ou não-supervisionado. Sendo que os classificadores usados neste trabalho foram estruturados sobre o paradigma supervisionado, que possibilita avaliar a qualidade do classificador.

O paradigma supervisionado requer uma fase inicial denominada treinamento, de modo que, nesta fase, são apresentados padrões de treinamento, que são instâncias com o rótulo da classe as quais pertencem. O resultado da fase de treinamento é um conjunto de regras que exprimem os relacionamentos entre os atributos dos padrões de treinamento de modo a permitir a classificação de novos padrões nas classes existentes. Após o treinamento dá-se a fase de teste. Nela são apresentados ao classificador o conjunto de regras, protótipos ou assinaturas, obtidos na fase anterior, e outros padrões, diferentes dos padrões usados para o treinamento, que também possuam rótulo das classes as quais pertencem. A finalidade disto é avaliar a consistência das regras, protótipos ou assinaturas vindas da fase de treinamento, assim, esses novos padrões, chamados de padrões de teste, são classificados sem que a informação do rótulo de classe que carregam seja levada em consideração. Depois disso, os padrões de teste terão dois rótulos de classe, (1) o rótulo inicial que informa precisamente a classe a qual eles pertencem e (2) o rótulo calculado pelo classificador. Isto nos possibilita contar os erros e acertos do classificador, nos permitindo aferir sua precisão. A Figura 3.1 ilustra as fases de treinamento e teste, salientando as entradas requeridas em cada fase e seus resultados;

Após a fase de teste, se a precisão do classificador for satisfatória para o problema a qual ele será aplicado, o classificador estará apto classificar novos padrões, que são padrões diferentes dos usados no treinamento e teste e que não possuem rótulo de classe, como mostra a Figura 3.2.



**Figura 3.1** – ilustração as fases de treinamento e teste do processo de classificação no paradigma de aprendizado supervisionado.



**Figura 3.2** – ilustra o processo de classificação de padrões de novidades.

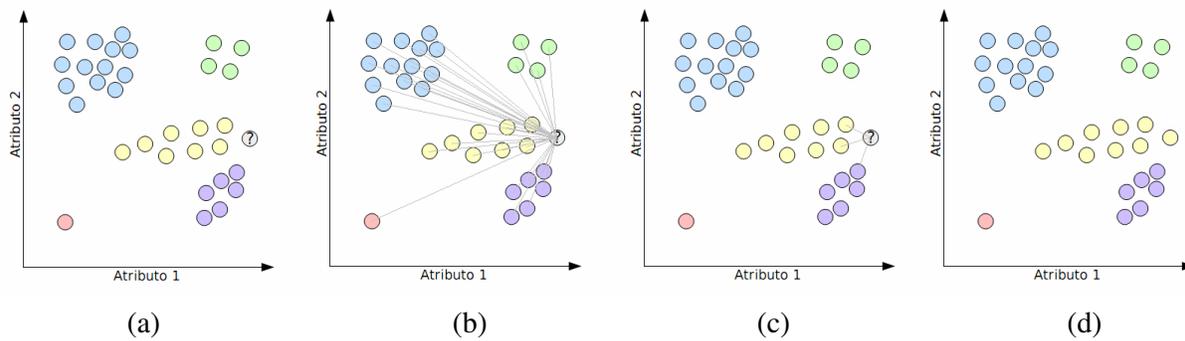
### 3.1 K-Vizinhos Mais Próximos

k-vizinhos mais próximos, *kNN* – *k Nearest Neighbors*, é um classificador clássico baseado em memória, que é frequentemente usado em aplicações do mundo real devido a sua simplicidade [3]. Apesar de simples, ele tem conseguido considerável exatidão na classificação e diversas aplicações e é conseqüentemente muito usado como uma base para comparação com novos classificadores.

A fase de treinamento do *kNN* consiste simplesmente em armazenar todos os padrões de treinamento. O *kNN* tem um parâmetro *k* que é o número de vizinhos a ser considerados para a classificação. Para  $k = 1$ , *kNN* é também chamado de método do vizinho mais próximo, *NN* - *Nearest Neighbor*. *NN* classifica um dado padrão como pertencente a mesma classe do padrão mais próximo a ele no conjunto de treinamento. Há um número de medidas de distâncias usadas neste processo, entretanto a distância euclidiana é a mais frequentemente usada [14]. Nós usamos esta distância neste trabalho.

Quando  $k > 1$ , *kNN* primeiramente computamos as distâncias do novo padrão a ser classificado com todos os padrões do conjunto de treinamento. Subseqüentemente, o algoritmo considera os *k* padrões do conjunto treinamento com as menores distâncias ao padrão a ser classificado. Finalmente, o novo padrão é classificado como pertencente a classe da maioria dos *k* padrões mais próximos do conjunto de treinamento.

A Figura 3.3 ilustra o processo de classificação do 3NN, onde em (a) temos os padrões de treinamento e o padrão a ser classificado, em (b) foi realizado o cálculo da distância da novidade para todos os padrões de treinamento, em (c) são tomados os 3 vizinhos mais próximos do padrão de teste e em (d) o padrão de teste é classificado como pertencente a classe da maioria dos 3 vizinhos mais próximos.



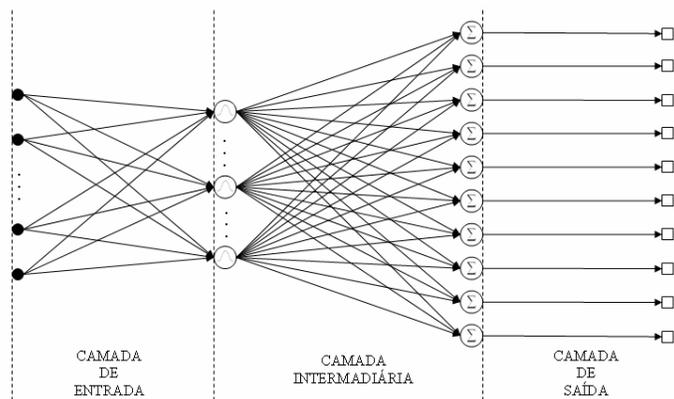
**Figura 3.3** - processo de classificação do 3NN.

Mesmo sendo uma técnica muito simples, o custo computacional do kNN é elevado. Do ponto de vista do custo da memória, o kNN armazena todos os padrões de treinamento e quanto ao desempenho, para cada padrão a ser classificado o kNN deve calcular sua distância para todos os padrões de treinamento.

Lembramos de observar que é usual escolher o k ímpar, para reduzir as ocorrências de impossibilidade de classificação devido a empate na quantidade de vizinhos mais próximos.

### 3.2 Redes Neurais com Funções de Bases Radiais

Neste trabalho, nós usamos redes neurais com funções de bases radiais – *RBFN* – *Radial Basis Functions Network* – com unidades gaussianas escondidas. RBFN é uma das mais populares arquiteturas de redes neurais, que tem sido aplicada com sucesso a muitos problemas de classificação [14]. Uma RBFN é uma rede neural *feedforward* que usualmente tem uma única camada intermediária (ou escondida) com unidades de funções de bases radiais – *RBF* – *Radial Basis Function*, como mostra a Figura 3.4.



**Figura 3.4** – arquitetura de uma RBFN completamente conectada.

O treinamento para RBFN que usamos aqui é não-constructivo. É um método de treinamento mais tradicional, que usa o algoritmo de agrupamento k-médias para prover os centros das gaussianas e suas respectivas larguras. O k-médias é um método de classificação não supervisionado que visa encontrar “k” agrupamentos dos padrões de treinamento, podendo acontecer desses agrupamentos serem mistos, ou seja, formados por indivíduos de diferentes classes, uma vez que o critério de agrupamento é distância dos elementos sem levar em consideração a classe.

Após determinados os k centróides dos agrupamentos de padrões, uma gaussiana é determinada tendo centro em cada um dos centróides. Cada uma dessas gaussianas será a função de ativação dos neurônios da RBFN e seus resultados, que são transmitidos a camada de saída, são ponderados a fim de que, após o treinamento, os estímulos oriundos dos k neurônios sirvam como assinatura do padrão de novidade de tal modo a permitir sua classificação, que é determinada pelo nodo da camada de saída com maior excitação. Observe que, neste algoritmo, o número de unidades RBF escondidas deve ser especificado pelo usuário de antemão.

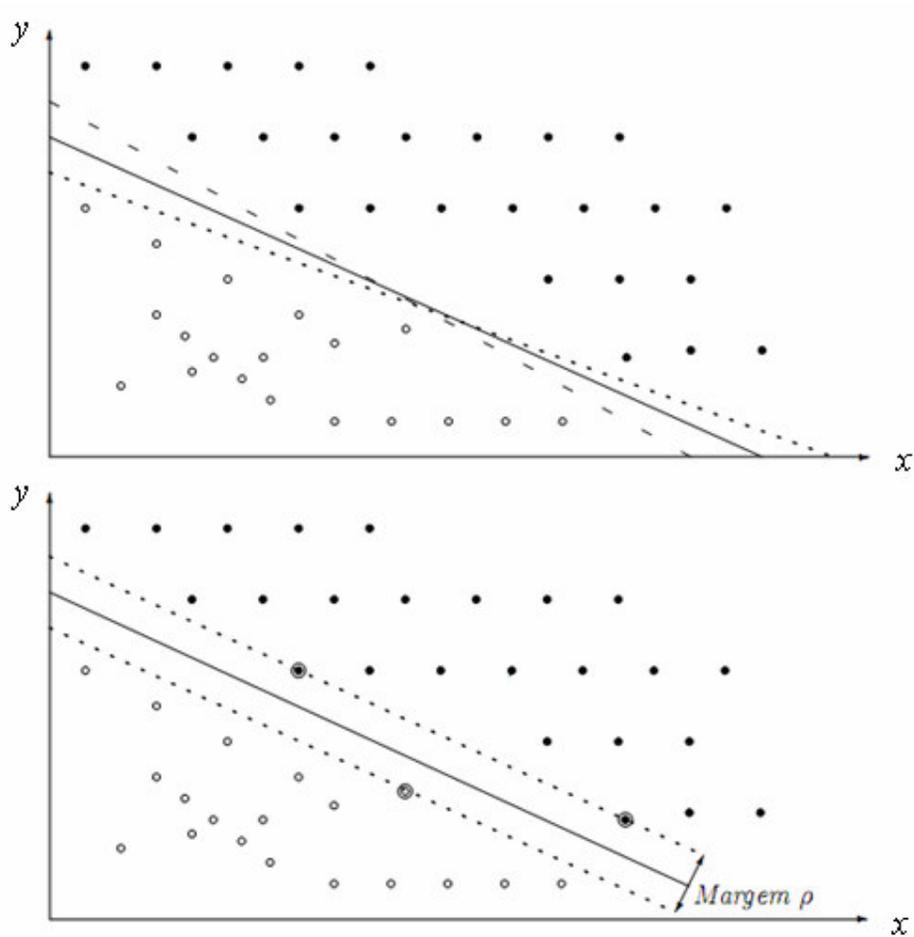
### 3.3 Máquina de Vetor de Suporte

Máquina de vetor de suporte – *SVM – Support Vector Machine* – é uma recente técnica para classificação e regressão que tem conseguido notável exatidão em um grande número de problemas importantes [14], [15]. SVM é baseada no princípio da minimização do risco estrutural – *SRM – Structural Risk Minimization*, que versa que, a fim de conseguir um bom desempenho de generalização, um algoritmo de aprendizado de máquina deve tentar minimizar o risco estrutural ao invés do risco empírico [15]. O risco empírico considera o erro no conjunto de treinamento, enquanto que o risco estrutural considera o erro no conjunto de treinamento e a complexidade das funções de classe usadas para ajustar o dado.

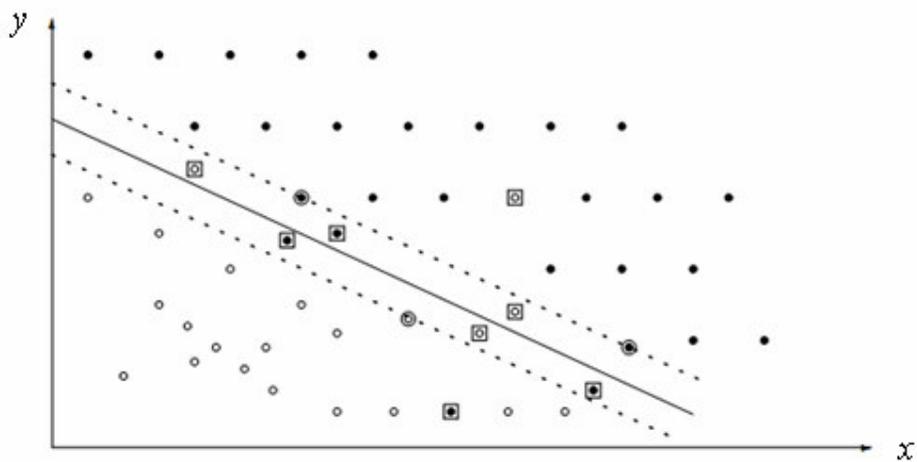
A idéia principal do SVM consiste na construção de hiperplanos ótimos, isto é, hiperplanos que (1) maximizam a margem de separação das classes, com a finalidade de separar padrões de treinamento de diferentes classes; (2) minimizam o numero de erros no conjunto de treinamento. Sendo um hiperplano, uma seção de ordem n-1 em um espaço de n dimensões. Onde para um espaço tri-dimencional um hiperplano é um plano bi-dimencional neste espaço.

Os vetores de treinamento  $x_i$  são mapeados em um espaço dimensional mais elevado pela função  $\phi$ . Então o SVM encontra um hiperplano de separação linear com a margem máxima neste espaço dimensional mais elevado.

A Figura 3.5 mostra os padrões de um conjunto de treinamento de um problema linearmente separável. No gráfico superior são mostrados os hiperplanos de separação no espaço de representação dos padrões, enquanto que no gráfico inferior é mostrada a margem  $\rho$  de separação ótima, máxima, da distribuição dos padrões. Os pontos sobre as linhas tracejadas são os vetores de suporte para o conjunto de treinamento e como o problema é linearmente separável implica na não existência de padrões internos a margem de separação. Na Figura 3.6 nós temos a ilustração de um problema não linearmente separável, o hiperplano de separação (linha sólida) e a margem ótima encontrada de modo a ser a maior margem com menor erro no conjunto de treinamento.



**Figura 3.5** – ilustração de um conjunto de treinamento de um problema linearmente separável, hiperplano de separação encontrado pelo SVM e sua margem de separação  $\rho$ .



**Figura 3.6** – ilustração de um conjunto de treinamento de um problema não linearmente separável, hiperplano de separação encontrado pelo SVM e sua margem de separação.

Um núcleo  $K(x,y)$  é um produto interno de algum espaço de características, tal que:

$$K(x, y) = \varphi^T(x)\varphi(y) \quad \text{Equação 3.1}$$

onde lembramos que a função  $\varphi$  é responsável pelo mapeamento dos vetores de treinamento  $x_i$  em um espaço dimensional mais elevado.

Neste trabalho, nós consideramos os núcleos de funções de bases radiais (RBF) e o núcleo polinomial. Em um núcleo RBF função de núcleo  $K(x_i, x_j)$  dada por:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad \text{Equação 3.2}$$

onde  $\|x_i - x_j\|$  é a distância euclidiana entre os vetores de treinamento  $x_i$  e  $x_j$ . no núcleo polinomial a função de núcleo é dada por:

$$K(x_i, x_j) = (x_i^T \cdot x_j + 1)^d \quad \text{Equação 3.3}$$

SVMs com núcleo RBF ou polinomial tem dois parâmetros, a saber,  $C$ , o parâmetro penalidade do termo erro ( $C > 0$ ) e o parâmetro do núcleo  $\gamma$ , que é a largura da RBF para núcleos RBF ou  $d$ , o expoente do núcleo polinomial. Estes parâmetros podem ter influência significativa no desempenho e conseqüentemente seus valores devem ser selecionados com cuidado para um dado problema.

## Capítulo 4

# Abordagem Baseada em Algoritmos Genéticos para a otimização aplicada à Classificação de Padrões

## Manuscritos

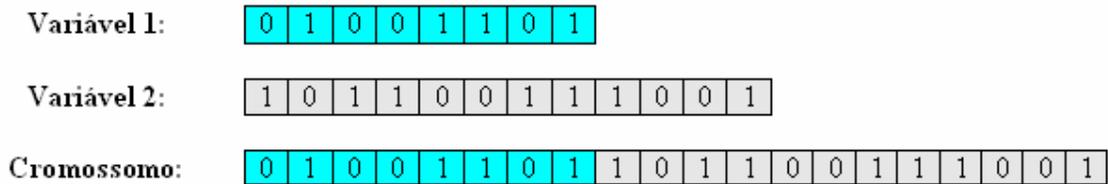
Neste capítulo apresentaremos uma técnica baseada em algoritmos genéticos que visa selecionar as características dos padrões que melhor os distingue em suas classes e, adicionalmente, busca a configuração ótima dos parâmetros de uma máquina de vetor de suporte para a tarefa de classificação dos padrões mediante a seleção de características realizada. Inicialmente introduziremos os conceitos básicos relativos a algoritmos genéticos na seção 4.1 e, na seção 4.2, trataremos da técnica proposta.

### 4.1 Breve introdução a Algoritmos Genéticos

Algoritmos genéticos – *GA* – *Genetic Algorithm* – são ferramentas computacionais de busca e otimização inspirados em sistemas biológicos e evolucionistas. Esta heurística trabalha com um conjunto de soluções candidatas, chamado população, onde cada um de seus membros, ditos cromossomos, são uma possível solução do problema para o qual o GA foi modelado, de tal modo que ao final da sua execução se espera obter o cromossomo mais apto a resolução do problema, ou seja, a solução ótima. Para tanto é necessário definir uma função de custo sobre o domínio das soluções do problema de modo que, para cada uma delas a função de custo retorne um escalar que exprime o grau de eficiência da solução, ou seja, quão apto o cromossomo é na resolução do problema [16].

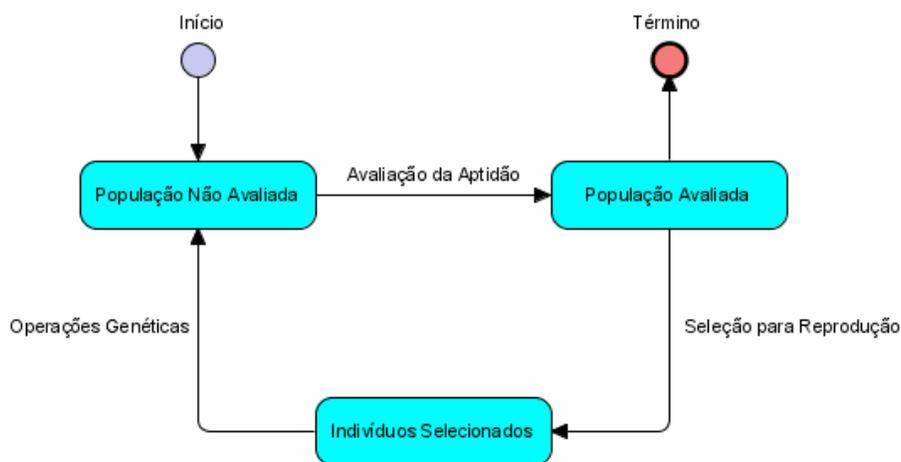
Inicialmente é necessário que as variáveis de interesse do problema sejam modeladas de tal forma a possibilitar a aplicação de algoritmos genéticos. Isto se dá pela representação binária dos valores possíveis a cada variável, sendo o cromossomo expresso pela concatenação dessas representações. Essa representação das variáveis se inicia pela escolha de um número fixo de *bits* para cada variável, conseqüentemente, todos os cromossomos terão um número fixo de *bits* dado pela soma dos números de *bits* usados para representar cada variável. A Figura 4.1 ilustra essa

situação, onde temos duas variáveis hipotéticas com suas respectivas representações binárias e o cromossomo formado pela concatenação delas.



**Figura 4.1** – representação binária das variáveis do problema para criação do cromossomo.

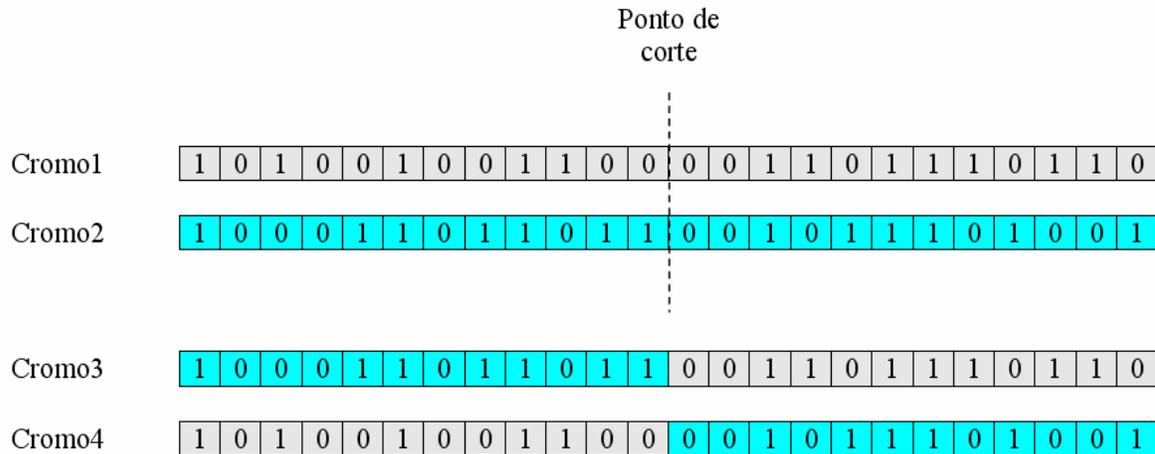
Uma vez definida a forma como representaremos as soluções do problema em cromossomos podemos aplicar ao GA. A Figura 4.2 mostra o ciclo evolucionário de um GA, onde inicialmente é montada uma população de cromossomos que, em seguida, é avaliada, segundo a função de custo, conferindo a cada cromossomo a informação de quão apto ele é na resolução do problema, após isto, os cromossomos são selecionados para a fase de reprodução, onde os mais aptos terão maiores chances de serem selecionados. Depois de selecionar os cromossomos para a reprodução, são aplicados a eles os operadores genéticos de cruzamento e mutação, que tem a finalidade de criar novos cromossomos (1) com a possibilidade destes serem mais aptos do que os cromossomos operandos e (2) de tal modo que todos os cromossomos do domínio do problema possam ser alcançáveis. Esses operadores genéticos são efetuados sobre os cromossomos segundo uma probabilidade pré-determinada, ou seja, nem todos os cromossomos selecionados na fase de reprodução sofreram cruzamento ou mutação. Após essas fases, temos uma nova população, comumente chamada na literatura de população *offspring*, que dará início a um novo ciclo evolucionário do GA. Este processo se repetirá até que, segundo algum critério de parada, que em geral é estabelecido pela quantidade de gerações atingidas pelo GA e ou quantidade de gerações que não apresentaram solução melhor do que a melhor solução até então encontrada. Uma vez que o término seja alcançado o GA retornará o cromossomo mais apto global, dos cromossomos por ele avaliados.



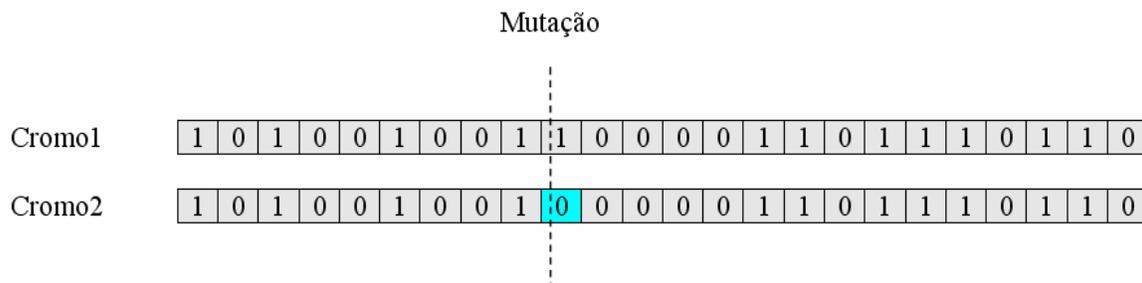
**Figura 4.2** – ciclo evolucionário do GA.

A Figura 4.3 ilustra a operação genética de cruzamento com um ponto de corte, aplicada aos dois cromossomos superiores, Cromo1 e Cromo2, e resultando nos dois cromossomos inferiores, Cromo3 e Cromo4, sendo o ponto de corte escolhido aleatoriamente.

A Figura 4.4 ilustra a mutação, que ocorre sobre probabilidade menor do que o cruzamento e que opera sobre *bit* do cromossomo mudando seu valor pela negação booleana.



**Figura 4.3** - operação genética de cruzamento com um ponto de corte.



**Figura 4.4** - operação genética de mutação.

## 4.2 Algoritmo Genético para Seleção de Características e Otimização dos Parâmetros de uma Máquina de Vetor de Suporte

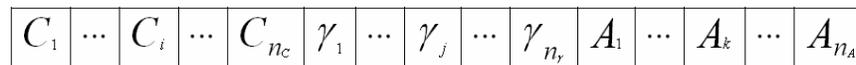
O método de integração de um algoritmo genético a uma máquina de vetor de suporte na tarefa de classificação de padrões pode ser visto em [4] aplicado a várias áreas e obtendo excelentes resultados. Todavia em [4] o método não foi utilizado para a classificação de dígitos manuscritos de imagens de documentos históricos, o que faremos aqui. Este método integra, ao algoritmo genético, a SVM como função de custo para avaliar a aptidão de seus cromossomos.

Algoritmo genético é uma ferramenta de busca e otimização consagrada na literatura devido a sua eficiência em obter soluções ótimas para os problemas para os quais é modelada. Assim, este método tem por objetivo encontrar a solução ótima para, (1) quais características melhor contribuem para a distinção dos padrões em suas respectivas classes e (2) qual a melhor configuração para uma máquina de vetor de suporte na tarefa de classificação dos padrões mediante a seleção de características realizada. Onde entenderemos aqui por solução ótima a

melhor solução dentre as soluções vistas pelo GA em sua execução e não necessariamente a melhor solução global para o problema.

O projeto deste modelo se inicia pelo projeto do cromossomo. Este deve conter as informações dos parâmetros usados na configuração da SVM e quais atributos dos padrões devem ser selecionados. Em [4] foi adotado o núcleo RBF para a SVM, devido ao fato de que com o núcleo RBF é possível analisar dados de alta dimensionalidade requerendo apenas dois parâmetros,  $C$  e  $\gamma$ , que são o valor de penalidade da SVM e a largura da função de núcleo RBF respectivamente. Manteremos esta mesma função de núcleo para a SVM usada aqui e adicionamos a justificativa do uso dela o fato de que em [3] a SVM com núcleo RBF obteve melhores resultados do que a SVM com núcleo polinomial.

Uma vez definido os parâmetros do classificador que serão observados, nos resta definir a quantidade de bits a serem usados em suas representações binárias. Na Figura 4.5 podemos ver o modelo do cromossomo que é formado por três blocos binários, (1) o primeiro bloco, de valores  $C_i$  com  $1 \leq i \leq n_C$ , é a representação binária do parâmetro  $C$  em  $n_C$  bits; (2) o segundo bloco, de valores  $\gamma_j$  com  $1 \leq j \leq n_\gamma$ , é a representação binária do parâmetro  $\gamma$  em  $n_\gamma$  bits; e (3) o terceiro bloco, de valores  $A_k$  com  $1 \leq k \leq n_A$ , é a representação da seleção realizada para os  $n_A$  atributos, onde  $A_k = 1$  representa que o atributo foi selecionado e  $A_k = 0$  representa que o atributo não foi selecionado.



**Figura 4.5** – modelo do cromossomo, onde  $n_C$  e  $n_\gamma$  são os números de bits usados na representação dos parâmetros  $C$  e  $\gamma$  respectivamente; e  $n_A$  é a quantidade de atributos do padrão.

Como os parâmetros da SVM são valores numéricos, se faz necessário que a cada avaliação de aptidão de cromossomo, os valores dos parâmetros  $C$  e  $\gamma$ , representados na forma binária, sejam convertidos para a forma decimal, a fim de possibilitar a configuração do classificador. Este processo é chamado de conversão do genótipo (representação binária) para o fenótipo (representação decimal) e será realizado aqui pela Equação 4.1, que distribui os valores de  $\min_p$  à  $\max_p$  nos  $2^l$  valores que o genótipo do parâmetro pode assumir, permitindo assim, (1) representar valores decimais com dízima e (2) garantir fechamento na representação binária, ou seja, para todos os  $2^l$  valores que o genótipo do parâmetro pode assumir, existe um fenótipo compreendido de  $\min_p$  à  $\max_p$ .

$$p = \min_p + \frac{\max_p - \min_p}{2^l - 1} \times d \tag{Equação 4.1}$$

“ $p$ ” é o fenótipo da representação binária.

“ $\min_p$ ” é o menor valor que o parâmetro pode assumir.

“ $\max_p$ ” é o maior valor que o parâmetro pode assumir.

“ $d$ ” é o valor decimal da representação binária do parâmetro.

“ $l$ ” é a quantidade de bits usadas na representação binária do parâmetro.

Observemos também que, para a Equação 4.1, devido a distribuição que ela realiza sobre a distância entre os valores de  $\min_p$  à  $\max_p$ , que é uma grandeza contínua, na quantidade de genótipos do parâmetro, que é uma grandeza discreta, se faz necessário definir um passo de descontinuidade que possibilite esta distribuição garantindo o fechamento na representação binária do genótipo. Isto é feito na Equação 4.1, pelo termo expresso na Equação 4.2, o qual

definimos como a precisão da configuração do parâmetro. Onde se entenda por configuração do parâmetro a determinação de  $\min_p$ ,  $\max_p$  e  $l$ .

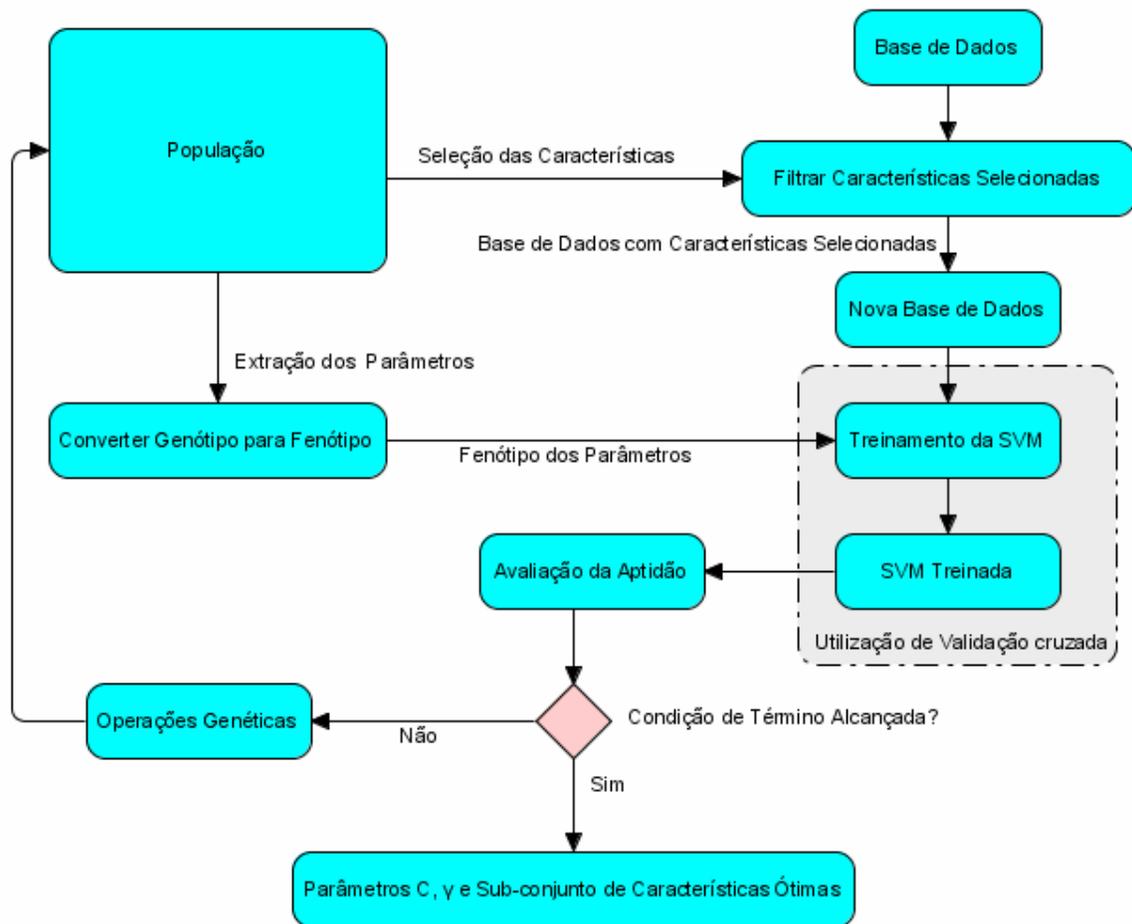
$$P_r = \frac{\max_p - \min_p}{2^l - 1}$$

**Equação 4.2**

“ $p_r$ ” é a precisão da configuração do parâmetro.

A Figura 4.6, ilustra as atividades associadas a execução deste método. Onde, depois de gerada a população inicial, cada cromossomo será avaliado, para tanto, são fragmentados em três blocos, um relativo à representação binária do parâmetro  $C$ , outro relativo à representação binária do parâmetro  $\gamma$  e o último relativo à seleção dos atributos. Os dois primeiros blocos de cada cromossomo são convertidos em fenótipos, que expressam os valores dos parâmetros em suas representações decimais. O último bloco de cada cromossomo é utilizado para fazer a filtragem dos atributos selecionados pelo algoritmo genético, gerando uma nova base de dados.

Depois de gerada a nova base de dados, esta é passada para a SVM que, com os valores dos fenótipos de cada parâmetro, efetua seu treinamento e obtém a avaliação da aptidão de cada cromossomo. Se após isso a condição de término estipulada não for atingida, a população sofrerá operações genéticas e o processo descrito se repetirá até que a condição de término seja atingida e então. O cromossomo mais apto, ou seja, aquele que obteve maior taxa de acerto na classificação pelo classificador SVM é eleito como a solução ótima para o problema.



**Figura 4.6** – diagrama esquemático das atividades envolvidas na execução do método de integração de um algoritmo genético a uma máquina de vetor de suporte na tarefa de classificação de padrões.

## Capítulo 5

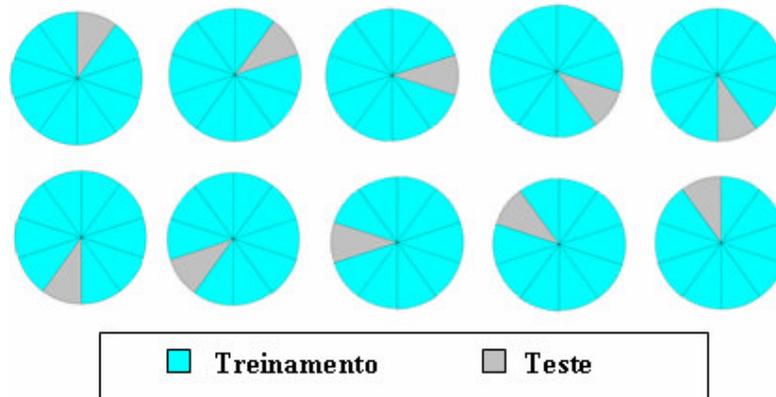
# Experimentos

Este capítulo relata os experimentos que foram realizados neste trabalho, os quais dividimos em duas etapas distintas, sendo a primeira a extensão do trabalho realizado em [3], que se encontra descrita na seção 5.1; a segunda etapa é a integração de um GA com um SVM na tarefa de classificação de dígitos manuscritos de imagens de documentos históricos, que se encontra descrita na seção 5.2.

Lembramos que a base de imagens que utilizamos é a mesma usada em [3] e que esta foi formada a partir de imagens de documentos históricos do final do século XIX e início do século XX. As imagens que compõem esta base estão binarizadas e foram extraídas dos documentos históricos por processo de segmentação manual, sendo extraído um total de 200 imagens de dígitos, 20 imagens de dígitos para cada classe. Em seguida, todas as imagens foram normalizadas para  $32 \times 32$  pixels o que possibilitou a confecção da base de dados inicial, sem qualquer aplicação de extração de características, formada pela representação vetorial de cada imagem normalizada, o que resulta em vetores de 1024 dimensões.

Nós usamos a validação cruzada estratificada em 10 grupos – *10-fold crossvalidation* – para avaliar o desempenho de generalização assim como para comparar as extrações de características e os classificadores considerados neste trabalho. Neste método, um dado conjunto de dados é dividido em dez grupos, tendo aproximadamente (1) tamanhos iguais e (2) mesma frequência de distribuição dos padrões do conjunto de dados original. Um classificador é treinado usando um subconjunto formado pela junção de nove destes grupos e testado usando o grupo remanescente. Isto é feito dez vezes, e cada vez um grupo diferente é usado como conjunto de teste, e computa-se o erro do conjunto de teste,  $E_i$ . Finalmente, o erro da validação cruzada é computado como a média sobre os dez erros  $E_i$ ,  $1 \leq i \leq 10$ . A Figura 5.1 ilustra esse processo.

A vantagem de se usar a validação cruzada é que nela o treinamento é feito com todos os dados, e por isso gera um resultado mais representativo de todos os padrões, diferentemente de abordagens mais tradicionais, onde o conjunto de dados é particionado em um conjunto de treinamento e um de teste, o que gera questões como (1) Quais dados devem ser colocados para o treinamento e quais para o teste? (2) Quantos dados deve ter o conjunto de treinamento e o conjunto de teste?. Adicionalmente, como o resultado obtido pela validação cruzada é a média de 10 treinamentos e testes sobre bases diferentes, esse valor é mais preciso do que a solução apresentada por um avaliação segundo um método tradicional de particionamento da base de dados.



**Figura 5.1** – método de avaliação por validação cruzada estratificado em 10 grupos.

É importante enfatizar que todas as simulações relatadas aqui usaram validação cruzada estratificada em 10 grupos, por meio da qual os subconjuntos são dados forma usando a mesma frequência de distribuição dos padrões do conjunto de dados original.

## 5.1 Estudo de Extrações de Características e Classificadores na Classificação de Dígitos Manuscritos de Documentos Históricos

Apresentaremos nesta seção os experimentos realizados para a composição do estudo da aplicação de extrações de características no auxílio a tarefa de classificação de dígitos manuscritos de documentos históricos, que visa estender o trabalho realizado em [3] pelo acréscimo de uma abordagem de extração de características invariante a fenômenos de rotação, translação e escala nos dígitos. Estudaremos também o comportamento de quatro classificadores na tarefa de classificação realizada sobre bases de dados geradas após a aplicação de cada método de extração de características avaliados aqui.

Os métodos de extração de características usados aqui são mostrados na Tabela 5.1, e os classificadores são apresentados em Tabela 5.2.

Com exceção do PCA, implementamos todos os algoritmos que executaram as extrações de características sobre a base de dados inicial. A plataforma usada para a implementação foi o *MatLab* [17], que é uma ferramenta proprietária voltada ao desenvolvimento de projetos tecnológicos e simulação, e que é largamente usada em pesquisas acadêmicas, devido a sua estabilidade e por possuir variada gama de caixas de ferramentas – *toolboxes* – desenvolvidas para aplicações diversas.

Para a execução do PCA, usamos uma implementação existente contida no *Weka* [18], que é uma ferramenta de mineração de dados de código aberto – *open source* – que agrega várias implementações de técnicas de aprendizado de máquina e foi desenvolvida por pesquisadores da universidade de Waikato, Nova Zelândia.

Todos os classificadores usados aqui foram executados sobre a ferramenta *Weka*, assim como foi realizado em [3].

**Tabela 5.1** - métodos de extração de características usados neste trabalho.

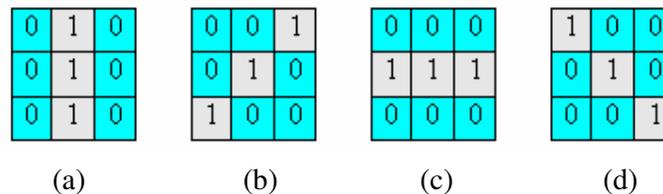
Métodos de extração de características
PCA
Sub-amostragem
Morfologia matemática

**Tabela 5.2** - classificadores usados neste trabalho.

Classificadores
kNN
RBFN
SVM

### 5.1.1 Configurações

Para as extrações de características temos que, (1) o PCA foi executado usando a configuração padrão do Weka, que esta fixada para extrair os autovetores, das combinações lineares dos atributos dos padrões, que expressem 95% da variância, obtendo uma representação dos padrões de 130 dimensões; (2) a sub-amostragem foi modelada para extrair blocos, não sobrepostos, de 4x4 dos padrões, obtendo representação de 64 dimensões; (3) na abordagem por morfologia matemática investigamos a influência de quatro elementos estruturantes de dimensões 3x3 mostrados em Figura 5.2, onde Figura 5.2 (a) é o elemento estruturante vertical, que obteve representação de 15 dimensões; Figura 5.2 (b) é o elemento estruturante inclinado a direita, que obteve representação de 15 dimensões; Figura 5.2 (c) é o elemento estruturante horizontal, que obteve representação de 16 dimensões; e Figura 5.2 (d) é o elemento estruturante inclinado a esquerda, que obteve representação de 10 dimensões.



**Figura 5.2** – elementos estruturantes usados neste trabalho. Sendo (a) elemento estruturante vertical, (b) elemento estruturante inclinado a direita, (c) elemento estruturante horizontal, (d) elemento estruturante inclinado a esquerda.

A Tabela 5.3 apresenta as bases de dados aqui referenciadas, com os rótulos pelos quais as identificaremos, as extrações de características que as geraram e suas respectivas dimensionalidades. Sendo B-0 a base de dados original, formada pela representação matricial das imagens de dígitos e a B-7 é a concatenação da representação de cada padrão obtida em B-3, B-4, B-5 e B-6, formando uma nova base de dados de 56 dimensões.

Quanto aos classificadores nós variamos os seguintes parâmetros, (1) no kNN variamos o parâmetro “k”, (2) para a RBFN variamos o número de agrupamentos – *clusters* – que referenciaremos por “n”, (3) para a SVM nós variamos o parâmetro penalidade “C”, e o tipo de núcleo usado, a saber, um núcleo polinomial e um núcleo RBF, onde para cada tipo de núcleo há um parâmetro específico a ser configurado. No caso da SVM com núcleo polinomial variamos o parâmetro expoente do núcleo, “d” e para a SVM com núcleo RBF variamos o parâmetro largura da RBF, “ $\gamma$ ”.

A Tabela 5.4 mostra os classificadores usados com os parâmetros que foram observados em cada um deles e os valores propostos para cada parâmetro.

**Tabela 5.3** – bases referenciadas neste trabalho.

Rótulo da base	Extração de características	Dimensionalidade
B-0	Nenhuma	1024
B-1	PCA	130
B-2	Sub-amostragem	64
B-3	Morfologia matemática com elemento estruturante vertical	15
B-4	Morfologia matemática com elemento estruturante inclinado a direita	15
B-5	Morfologia matemática com elemento estruturante horizontal	16
B-6	Morfologia matemática com elemento estruturante inclinado a esquerda	10
B-7	Concatenação dos atributos das bases B-3, B-4, B-5 e B-6	56

**Tabela 5.4** – classificadores e suas configurações.

kNN	RBFN	SVM		SVM	
		polinomial		RBF	
k	n	C	d	C	$\gamma$
1	2	1	1	1	0,1
3	10	10	3	10	0,01
5	20	100	9	100	0,001
7	50				
9	70				

Os parâmetros de cada classificador que não são observados aqui, assumiram o valor padrão adotado na ferramenta Weka.

### 5.1.2 Resultados

A Tabela 5.5 mostra os resultados obtidos por cada classificador, com as configurações propostas para os seus parâmetros, na classificação dos padrões a pelas suas representações contidas nas bases B-1 à B-7. Os resultados contidos na Tabela 5.5, são o percentual de erro de classificação dos classificadores e foram obtidos com o uso de validação cruzada estratificada em 10 grupos.

Os campos em azul da Tabela 5.5, representam os melhores valores obtidos por cada classificador e os campos em cinza, os melhores valores globais. Com isso, podemos constatar que, (1) para a maioria dos experimentos, a base B-2, formada pelo método de extração de características sub-amostragem, obteve os melhores resultados e que (2) a melhor combinação classificador X extração de características, foi obtida pelo classificador SVM com núcleo RBF e a técnica de sub-amostragem, obtendo 5,50% de erro na classificação.

A Tabela 5.6 mostra os menores erros percentuais de classificação obtidos em cada base de dados e com qual classificador foram obtidos. Onde, com exceção das bases B-1 e B-2, todas

as bases obtiveram percentual de erro superior a 50%. E que para a maioria das bases, o SVM foi o classificador que obteve menores erros, dividindo estes resultados com ambos os núcleos.

**Tabela 5.5** – desempenho de classificação obtido pelos classificadores em suas configurações propostas, sobre as bases de dados com extração de características abordadas. A tabela relata os erros obtidos pela validação cruzada estratificada em 10 grupos.

		B-1	B-2	B-3	B-4	B-5	B-6	B-7
kNN (k)	1	78%	9%	80%	83%	78%	82%	68%
	3	87,50%	11%	81%	81%	80,50%	82%	71,50%
	5	88%	13%	80,50%	77%	78%	79%	66%
	7	90%	15%	79%	74,50%	78,50%	77,50%	67%
	9	89,50%	18,50%	84%	73%	76,50%	77,50%	69,50%
RBFN (n)	2	36,50%	17%	81,50%	69,50%	80%	76%	62%
	10	50%	18%	80,50%	76%	84,50%	83,50%	68%
	20	31,50%	14%	84,50%	75,50%	81,50%	81,50%	61%
	50	31,50%	14%	84,50%	75,50%	81,50%	81,50%	61%
	70	31,50%	14%	84,50%	75,50%	81,50%	81,50%	61%
SVM-polinomial (C; d)	(1; 1)	24%	6%	86%	69,50%	76%	78,50%	58,50%
	(1; 3)	24%	17%	83,50%	69%	75,50%	77,50%	58,50%
	(1; 9)	32,50%	41,50%	86%	78%	84%	81%	71%
	(10; 1)	24%	6%	81,50%	72%	71%	77%	63,50%
	(10; 3)	24%	16,50%	77,50%	76%	76,50%	72,50%	58,50%
	(10; 9)	32,50%	41%	82,50%	78,50%	81,50%	80,50%	71%
	(100; 1)	24%	6%	78%	76,50%	76%	79%	66,50%
	(100; 3)	24%	16,50%	79%	78,50%	77,50%	70,50%	58,50%
SVM-RBF (C; $\gamma$ )	(100; 9)	32,50%	41%	80,50%	80,50%	81,50%	77,50%	71%
	(1; 0,1)	25,50%	7,50%	82%	69,50%	80%	77,50%	64,50%
	(1; 0,01)	46,50%	24,50%	81%	74,50%	80,50%	80%	79%
	(1; 0,001)	46,50%	54%	85%	77%	79%	78%	73,50%
	(10; 0,1)	20%	5,50%	84,50%	70,50%	72%	73,5%	58,50%
	(10; 0,01)	21,50%	7,50%	83,50%	74,50%	75,50%	81%	63%
	(10; 0,001)	48,50%	23,50%	82%	76%	79%	81%	75%
	(100; 0,1)	20%	5,50%	78%	73%	73,50%	75%	60%
	(100; 0,01)	22,50%	5,50%	87,50%	70,50%	71,50%	75%	58%
(100; 0,001)	21%	7,50%	86%	75,50%	77,50%	80%	65%	

**Tabela 5.6** – menor erro percentual de classificação obtido em cada base de dados e com qual classificador foi obtido.

	B-1	B-2	B-3	B-4	B-5	B-6	B-7
<b>Menor erro</b>	20%	5,5%	77,50%	69%	71%	70,50%	58%
<b>Classificador</b>	SVM RBF	SVM RBF	SVM Polinomial	SVM Polinomial	SVM Polinomial	SVM Polinomial	SVM RBF

## 5.2 GA Integrado a SVM na Classificação de Dígitos Manuscritos de Documentos Históricos

Nesta seção veremos os experimentos realizados para a composição do estudo da integração de um GA a um classificador SVM na tarefa de classificação de dígitos manuscritos de documentos históricos, que visa selecionar as características (atributos) dos padrões e otimizar os parâmetros de uma máquina de vetor de suporte a fim de obter o melhor desempenho de classificação possível.

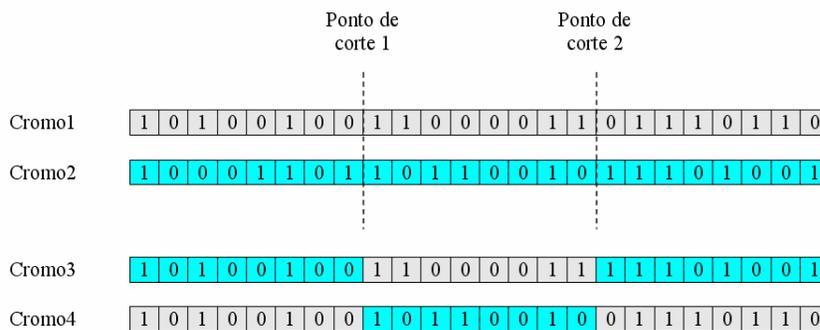
Não adotamos as mesmas políticas usadas em [4], procuramos utilizar políticas que possibilitassem o sistema convergir mais rápido. Por que o problema que abordamos aqui é naturalmente mais computacionalmente custoso do que os vistos em [4], onde a maior base de dados usada possui 60 dimensões, enquanto que a nossa possui 1024. Além disto, o fato da função de custos ser uma SVM que usa validação cruzada estratificada em 10 grupos, significa que a cada avaliação de cromossomo rodamos dez vezes uma SVM para poder calcular sua aptidão.

No intuito de aumentar a convergência, montamos a população inicial pela seleção do cromossomo mais apto de cada  $x$  populações geradas aleatoriamente, onde  $x$  é o tamanho da população. Esta política é inicialmente mais custosa para a criação da população inicial, uma vez que são necessárias  $x^2$  avaliações de custo. Todavia, iniciamos o ciclo evolucionário do GA com uma população que representa  $x$  melhores soluções locais no espaço de soluções.

Utilizamos também o elitismo, que ocorre durante o ciclo evolucionário, fazendo com que, após cada avaliação dos cromossomos da população, o cromossomo mais apto seja copiado diretamente para a população *offspring*, antes mesmo de haver seleção dos cromossomos da população “pai” para a reprodução. Isto garante que a melhor solução não seja perdida por não ser selecionada para reproduzir ou por ser alterada nas operações genéticas.

Para selecionar os cromossomos da população para a reprodução usamos a técnica de torneio com pressão de seleção igual a dois. Ou seja, escolhemos dois cromossomos aleatórios na população “pai” e selecionamos o que tiver maior aptidão. Este cromossomo selecionado não é retirado da população “pai”, o que permite que ele seja selecionado outras vezes. Este processo se repete até que a quantidade de indivíduos necessários para a reprodução seja atingida e então se dá a reprodução. Adicionalmente, esta política tem a vantagem de que o cromossomo menos apto da população nunca consegue se perpetuar.

Assim como em [4], adotamos a política de dois pontos de corte para a realização da operação genética de cruzamento, onde os pontos de corte são escolhidos aleatoriamente sobre os cromossomos operandos e é feita a troca das extremidades, como podemos ver na Figura 5.3. Isto faz com que a quantidade de informação média trocada seja de dois terços do cromossomo, aumentando a aleatoriedade do modelo, o que diminui a probabilidade do algoritmo ficar preso a um mínimo local do espaço de soluções.



**Figura 5.3** – operação genética de cruzamento com dois pontos de corte aleatórios.

A implementação deste método está esquematizada na Figura 5.4, onde podemos ver as políticas adotadas para a criação da população inicial e seleção dos indivíduos para perpetuação genética.

Como já dissemos neste texto, inicialmente geramos uma *população aleatória*, a avaliamos e guardamos seu membro mais apto. Repetimos este processo até que obtenhamos uma população formada pelos membros mais aptos das populações aleatórias, a qual chamamos de *população inicial*. Com isto damos início ao ciclo evolucionário, onde iniciamos as variáveis que são responsáveis pela parada da execução do algoritmo, que são o *limite sem melhora* e o *número de gerações*.

Uma vez que a *população inicial* é formada por membros já avaliados, copiamos dela o membro mais apto para o repositório *+apto global*. Em seguida montamos a população *offspring*, onde por elitismo preservamos o membro mais apto da população pai, que na primeira geração é a *população inicial*, e por torneio selecionamos os indivíduos para reprodução.

Uma vez completa a população *offspring*, que é representada pelo bloco *nova população*, o contador do *número de gerações* incrementado e em seguida a condição de término é verificada, caso o número de gerações ou limite sem melhora sejam alcançados o algoritmo é finalizado e a solução contida em *+apto global* é tida como a melhor solução.

Caso a condição de término não seja alcançada a *nova população* é avaliada por um classificador SVM e seu membro mais é comparado com a solução armazenada em *+apto global*. Caso o membro mais apto da *nova população* for melhor do que a solução armazenada em *+apto global*, trocamos as soluções e zeramos o contador do *limite sem melhora*. Caso contrário, incrementamos o *limite sem melhora* e em seguida verificamos se a condição de término foi alcançada.

O ciclo evolucionário perdurará até que a condição de término seja alcançada e assim o GA apresentará a melhor dentre as soluções percorridas por ele.

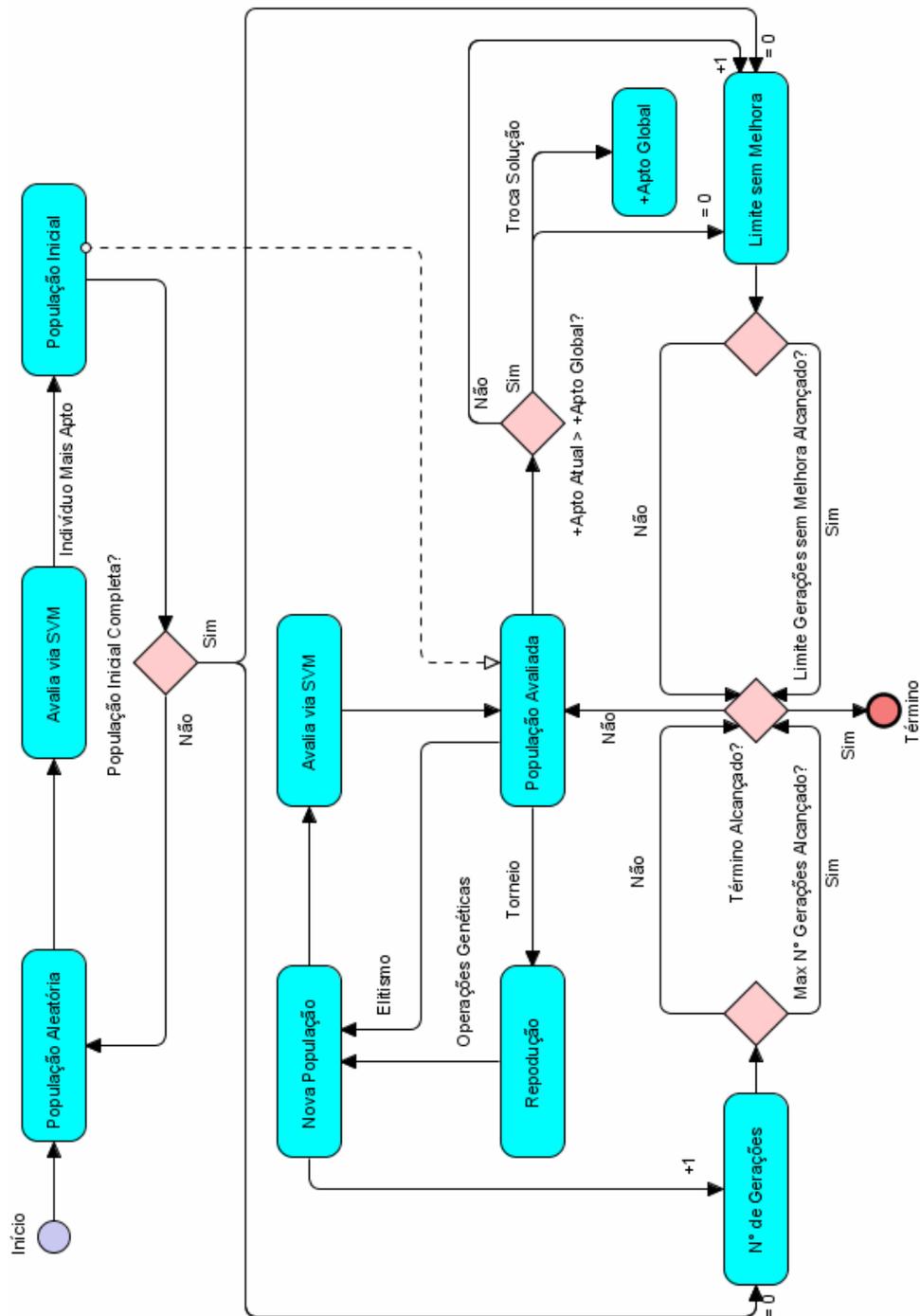


Figura 5.4 – esquema do GA integrado a SVM.

## 5.2.1 Configurações

Diferentemente do que foi feito no estudo da seção 5.1, trabalharemos aqui com uma única base de dados, a B-0 da Tabela 5.3, onde cada um de seus padrões correspondem a

representação vetorial, de 1024 dimensões, de uma imagem de dígito manuscrito extraída de documento histórico.

Como o classificador usado é uma SVM com núcleo RBF, devemos representar de forma binária os valores dos parâmetros  $C$  e  $\gamma$ . A Tabela 5.7, mostra a quantidade de bits usados na representação binária dos parâmetros  $C$  e  $\gamma$ , dos atributos dos padrões e do cromossomo; nela podemos ver que um cromossomo, para esta configuração, possui 1064 genes (bits), o que significa que, para esta quantidade de bits usados na representação binária do cromossomo, o problema possui  $2^{1064} - 1$  soluções, que é um número na ordem de  $1 \times 10^{320}$ .

A Tabela 5.8, relaciona as configurações dos parâmetros  $C$  e  $\gamma$ , e a precisão obtida por elas. Lembramos que o valor precisão representa a distância entre dois fenótipos consecutivos.

Para a realização deste estudo, executaremos um experimento com a configuração vista na Tabela 5.9, que será executado dez vezes, a fim de que possamos obter um comportamento médio do algoritmo nesta configuração.

Ambos os experimentos possuem o mesmo critério de parada, que ocorre quando o GA atinge um máximo de 5000 gerações ou quando a geração corrente dista 500 gerações da geração onde foi encontrada a melhor solução até então avaliada pelo algoritmo. Estes valores foram determinados de maneira empírica.

**Tabela 5.7** – quantidade de bits usados na representação binária dos parâmetros  $C$  e  $\gamma$ , dos atributos dos padrões e do cromossomo.

	<b>C</b>	<b><math>\gamma</math></b>	<b>Atributos</b>	<b>Cromossomo</b>
<b>Qtd. bits</b>	20	20	1024	1064

**Tabela 5.8** – configurações dos parâmetros  $C$  e  $\gamma$ , e a precisão obtida.

	<b>C</b>	<b><math>\gamma</math></b>
<b>Qtd. bits</b>	20	20
<b>Valor min.</b>	0,01	0,00001
<b>Valor max.</b>	1000	0,5
<b>Precisão</b>	~0,000953665689	~0,000000476828

**Tabela 5.9** – configuração do experimento.

<b>Tam. da população</b>	20
<b>Nº max. de gerações</b>	5000
<b>Limite de Gerações sem melhora</b>	500
<b>Prob. de cruzamento</b>	0,7
<b>Prob. de mutação</b>	0,05
<b>Qtd. bits do C</b>	20
<b>Valor min. do C</b>	0,01
<b>Valor max. do C</b>	1000
<b>Qtd. bits do <math>\gamma</math></b>	20
<b>Valor min. do <math>\gamma</math></b>	0,00001
<b>Valor max. do <math>\gamma</math></b>	0,5
<b>Qtd. de atributos</b>	1024

## 5.2.2 Resultados

A Tabela 5.10 mostra os resultados obtidos após a execução das dez instâncias do GA com a configuração vista na Tabela 5.9. Nela relacionamos os percentuais de acerto e quantidade de características selecionadas com a configuração ótima dos parâmetros da SVM para a seleção de características feita. Informamos também em qual geração se encontrou a melhor solução para cada instância do GA. A linha destacada em azul corresponde a melhor solução dentre as dez melhores, onde o critério para sua escolha foi maior percentual de acerto com menor quantidade de características selecionadas.

Em média, o GA obtém percentual de acerto de 95,10%, que é superior ao acerto obtido na seção 5.1.2 pelo uso da técnica de sub-amostragem, que em seu melhor caso obteve 94,50% de acerto. Na Tabela 5.11 temos a média e o desvio padrão do percentual de acerto e quantidade de características selecionadas, calculados segundo as equações Equação 5.1 e Equação 5.2 respectivamente.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad \text{Equação 5.1}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \quad \text{Equação 5.2}$$

**Tabela 5.10** – resultados obtidos após a execução das dez instâncias do GA.

Percentual acerto	Qtd. características	C	$\gamma$	Nascido na população
94,50%	515	45,67437419354839	0,13889809789476196	52
95%	497	531,0368046920821	0,40270609202012253	62
95%	498	819,6770768328446	0,06460351534225019	103
95%	503	169,37435073313782	0,30758747372386336	151
95%	513	966,1315167155425	0,1337826862933028	155
95%	514	751,9095929618768	0,4503784094223112	179
95%	514	138,16087272727273	0,2230181997854231	87
95,50%	504	556,8906815249267	0,49991512460243664	13
95,50%	525	603,3818838709677	0,009414480146865984	0
95,50%	533	855,1591624633431	0,0715289663209594	33

**Tabela 5.11** – média de desvio padrão do percentual de acerto e quantidade de características selecionadas.

	Percentual acerto	Qtd. características
Média	95,100%	511,6000
Desvio padrão	0,316%	11,5104

O tempo de execução de cada uma das *threads* que executavam os GAs não pode ser medido com precisão, uma vez que as dez execuções, ou seja, as dez *threads* foram executadas em paralelo, compartilhando os recursos computacionais da máquina. Todavia o tempo de execução das dez *threads* foi superior a 20 dias, sendo usada uma máquina com processador *Intel core 2 Quad Q6600*, com 2,40GHz e memória principal de 2048MB. Isto se deve (1) a dimensionalidade do objeto em estudo, pois nossa base de dados é de 1024 dimensões enquanto, como já dissemos, a maior base de dados vista em [4] é de 60 dimensões; (2) a função de custo do GA é uma SVM com validação cruzada estratificada em 10 grupos.

## Capítulo 6

# Conclusão e Trabalhos Futuros

Este trabalho investigou métodos para o reconhecimento de dígitos manuscritos de imagens de documentos históricos. Os documentos analisados pertencem a uma coleção de interesse histórico do século final do século 18 e início do século 19.

Em trabalho anterior, nós estudamos combinações de técnicas de extração de características e de classificação populares para o reconhecimento de dígitos manuscritos de imagens de documentos históricos [3]. Neste trabalho, nós acrescentamos, ao estudo realizado, técnica de extração de características invariante a fenômenos de rotação, translação e escala, frequentes na escrita livre. Onde nosso objetivo era verificar se a nova extração de características é mais eficiente no auxílio a distinção dos dígitos em suas classes possibilitando melhor classificação.

Nossos experimentos mostraram que a nova abordagem para extração de características proposta, invariante a fenômenos de rotação, translação e escala, foi bastante ineficiente no auxílio a distinção dos padrões, apresentando percentuais de erro superiores a 50%, enquanto que uma abordagem menos sofisticada, a sub-amostragem, apresentou percentual de erro igual a 5,50%.

Também estudamos aqui, uma abordagem que integra um GA a uma SVM, sendo esta usada como função de custo para o GA, com a finalidade dupla de (1) verificar quais as características dos padrões, da base de dados sem extração de características, melhor os distinguem em suas respectivas classes e (2) otimizar os parâmetros da SVM para a classificação dos padrões mediante a seleção de características feita. Objetivando obter o melhor desempenho de classificação possível.

Nossos experimentos mostraram que a solução encontrada pelo GA tem, em média, 95,10% de acerto no desempenho de classificação e obtendo média de quantidade de características selecionadas igual a 511,6. Tendo o melhor resultado do GA obtido desempenho de classificação igual a 4,50% de erro com 504 características selecionadas.

Deste modo a técnica do GA integrado a SVM representa uma redução de 1% no erro de classificação, para o problema de reconhecimento de dígitos manuscritos de imagens de documentos históricos que abordamos, em relação a melhor configuração classificador X extração de características vista em [3].

## **6.1 Trabalhos Futuros**

Para o futuro planejamos estudar a influência de outros elementos estruturantes na execução de extração de características baseadas em morfologia matemática, bem como, estudar concatenações delas a fim de verificar quais elementos estruturante, em conjunto, contribuem para melhores desempenhos de classificação.

Aplicar técnica similar de integração de um GA a SVM com núcleo RBF, com fim de otimizar os parâmetros do classificador, com o objetivo de aumentar o desempenho de classificação sobre a base de dados que usamos aqui, sendo esta com sub-amostragem como extração de características, e assim poder comparar o desempenho encontrado neste trabalho com o desempenho da sub-amostragem em uma configuração ótima.

Investigar forma eficiente de explorar as características que melhor representam os dígitos manuscritos do problema que abordamos aqui, a fim de desenvolver técnica de extração ou seleção de características de baixo custo computacional e elevado desempenho na distinção dos dígitos.

## Bibliografia

- [1] Mori, S.; Suen, C. Y.; Yamamoto, K. "Historical Review of OCR Research and Development". Proceedings of the IEEE, 80(7), 1992, pp. 1029-1058.
- [2] Cecotti, H.; Belayd, A.. "Hybrid OCR Combination Approach Complemented by a Specialized ICR Applied on Ancient Documents" ICDAR'05, In: Eighth International Conference on Document Analysis and Recognition, 2005, pp. 1045-1049.
- [3] Oliveira, A. L. I.; Mello, C. A. B. ; Silva Júnior, E. R. ; Alves, V. M. O.. Optical Digit Recognition for Images of Handwritten Historical Documents. In: Simpósio Brasileiro de Redes Neurais, 2006, Ribeirão Preto - SP. Proc. of Brazilian Symposium on Neural Networks (SBRN'2006), 2006. p. 29.
- [4] Cheng-Lung, H.; Chieh-Jen, W. A GA-Based Feature Selection and Parameters Optimization for Support Vector Machines. Expert Systems with Applications, Vol. 31, Issue 2, pp 231-240, 2006.
- [5] Wierer, J.; Boston, N.. A Handwritten Digit Recognition Algorithm using Two-Dimensional Hidden Markov Models for Feature Extraction. In: International Conference on Acoustics, Speech and Signal Processing, 2007, Honolulu – HI. Vol. 2. pp 505-508, 2007.
- [6] Perez, C. A.; Holzmann, C. A.. Improvements on Handwritten Digit Recognition by Genetic Selection of Neural Network Topology and by Augmented Training. In: International Conference on Systems, Man and Cybernetics, 1997, Santiago – Chile. Vol. 2, pp 1487-1491, 1997.
- [7] Pawlicki, T. F. et al.. Neural Network Models and their Application to Handwritten Digit Recognition. In: International Conference on Neural Networks, 1988, Buffalo – NY. Vol.2, pp 63-70, 1988.
- [8] Reusch, B.. Computational Intelligence, Theory and Applications. Springer, 2005.
- [9] Benediktsson, J.A.; Arnason, K. et al. Classification and Feature Extraction with Enhanced Statistics. IGARSS '96, Remote sensing for a sustainable future, Vol. 1, pp 414-416, 1996.
- [10] Witten, I. H.; Frank E.. Data Mining Practical Machine Learning Tools and Techniques. Elsevier, 2005.
- [11] Fukunaga, K.. Introduction to Statistical Pattern Recognition. Morgan Kaufmann Publishers, 1990.
- [12] Larose, D. T.. Data Mining Methods and Models. John Wiley & Sons, 2006.
- [13] Santos, W. P.. Análise de Imagens Digitais em Patologia Utilizando Morfologia Matemática e Lógica Nebulosa. Dissertação de Mestrado, Departamento de Engenharia Elétrica/Eletrônica, Universidade Federal de Pernambuco, 2003.
- [14] Webb, A.. Statistical Pattern Recognition. John Wiley & Sons, 2002.

- [15] Lima, A. R. G.. Máquinas de Vetores Suporte na Classificação de Impressões Digitais. Dissertação de Mestrado, Departamento de Computação, Universidade Federal do Ceará, 2002.
- [16] Eiben, A. E.; Smith, J.E.. Introduction to Evolutionary Computing. Springer, 2003.
- [17] MatLab. 2007. <http://www.mathworks.com/>
- [18] Weka. 2007. <http://www.cs.waikato.ac.nz/ml/weka/>
- [19] Gonzales, R.; Woods, R.. Digital Image Processing. Addison-Wesley Publishing Company, 1992.

## Apêndice A

# Conceitos Básicos de Morfologia Matemática

Morfologia matemática é uma teoria de processamento não linear, baseada em transformações de forma, pelo emprego de operadores não lineares dos quais os mais fundamentais são dilatação e erosão [13], [19].

Enunciaremos aqui os conceitos de morfologia matemática, aplicados a imagens binárias, que foram necessários a este trabalho. Inicialmente definiremos alguns conceitos preliminares que são usados em morfologia matemática e em seguida traremos os conceitos de morfologia matemática.

Definimos imagem binária como:

$$I = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad \text{Equação A.6.1}$$

onde  $a_{ij} \in \{0,1\} \forall i, j$  com  $1 \leq i \leq m$  e  $1 \leq j \leq n$

Operador negativo ( $\bar{\phantom{x}}$ ) é a aplicação  $I_{m \times n} \rightarrow I_{m \times n}$  tal que:

$$\bar{I}(i, j) = 1 - I(i, j) \quad \text{Equação A.6.2}$$

$\forall i, j$  com  $1 \leq i \leq m$  e  $1 \leq j \leq n$

Operador união ( $\vee$ ) é a aplicação  $I_{m \times n} \rightarrow I_{m \times n}$  tal que:

$$[I_1 \vee I_2](i, j) = \max[I_1(i, j), I_2(i, j)] \quad \text{Equação A.6.3}$$

$\forall i, j$  com  $1 \leq i \leq m$  e  $1 \leq j \leq n$

Operador interseção ( $\wedge$ ) é a aplicação  $I_{m \times n} \rightarrow I_{m \times n}$  tal que:

$$[I_1 \wedge I_2](i, j) = \min[I_1(i, j), I_2(i, j)]$$

$$\forall i, j \text{ com } 1 \leq i \leq m \text{ e } 1 \leq j \leq n$$

Equação A.6.4

Operador diferença ou subtração ( $\sim$ ) é a aplicação  $I_{m \times n} \rightarrow I_{m \times n}$  tal que:

$$[I_1 \sim I_2](i, j) = [I_1 \wedge \bar{I}_2](i, j)$$

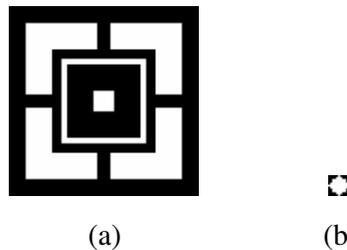
$$\forall i, j \text{ com } 1 \leq i \leq m \text{ e } 1 \leq j \leq n$$

Equação A.6.5

A partir destas operações construiremos os operadores morfológicos utilizados aqui.

A teoria da morfologia matemática é dita “construtiva” pelo fato de que operadores mais complexos podem ser construídos por operadores mais simples, onde os mais fundamentais, como já dissemos, são dilatação e erosão.

A Figura A.1 mostra (a) uma imagem binária e (b) uma outra imagem binária que será usada como elemento estruturante nas operações de morfologia matemática que exemplificaremos aqui. Essas operações possuem grande semelhança com a operação de convolução de teoria dos filtros lineares, uma vez que, como nos filtros lineares, o elemento estruturante, que se assemelha a máscara do filtro, possui um centro que é deslocado por toda a imagem, de modo que a cada passo do deslocamento relacionar-se com a sub-região da imagem justaposta a ele gerando um valor que representa a relação de inclusão do elemento estruturante na sub-região corrente.



**Figura A.1** - imagens binárias utilizadas como operandos de operações de morfologia matemática. (a) imagem a ser transformada e (b) elemento estruturante.

Operação dilatação (ou soma de Minkowsky) denotada por  $\delta_g(I)$  é uma operação de  $I_{m \times n} \times I_{r \times s} \rightarrow I_{m \times n}$  tal que:

$$\delta_g(I)(i, j) = \bigvee_{i,j} [R_{r,s}^{i,j}(I)(k, l) \wedge g(i-k, j-l)]$$

Equação  
A.6.6

$$\forall i, j \text{ com } 1 \leq i \leq m \text{ e } 1 \leq j \leq n$$

$$\forall k, l \text{ com } 1 \leq k \leq r \text{ e } 1 \leq l \leq s$$

$R_{r,s}^{i,j}(I)$  é uma região de  $I$  com Dimensões  $r \times s$  e centro no ponto  $(i, j)$  de  $I$

A Figura A.2 mostra o resultado da dilatação sobre a imagem Figura A.1 (a) tendo como elemento estruturante a Figura A.1 (b). Geometricamente, a cada deslocamento do elemento estruturante sobre a imagem, é observado se há interseção dos pixels ativos da imagem com os pixels ativos do elemento estruturante, se houver ao menos um pixel de interseção a imagem

resultante terá seu pixel, que corresponde ao pixel da imagem original que coincide com o centro do elemento estruturante, ativo.



**Figura A.2** – resultado da aplicação da operação dilatação sobre os operandos vindos da Figura A.1.

Operação erosão (ou subtração de Minkowsky) denotada por  $\mathcal{E}_g(I)$  é uma operação de  $I_{m \times n} \times I_{r \times s} \rightarrow I_{m \times n}$  tal que:

$$\mathcal{E}_g(I)(i, j) = \bigwedge_{i,j} [R_{r,s}^{i,j}(I)(k, l) \vee \bar{g}(i-k, j-l)] \quad \text{Equação A.6.7}$$

$\forall i, j$  com  $1 \leq i \leq m$  e  $1 \leq j \leq n$

$\forall k, l$  com  $1 \leq k \leq r$  e  $1 \leq l \leq s$

$R_{r,s}^{i,j}(I)$  é uma região de  $I$  com Dimensões  $r \times s$  e centro no ponto  $(i, j)$  de  $I$

A Figura A.3 mostra o resultado da erosão sobre a imagem Figura A.1 (a) tendo como elemento estruturante a Figura A.1 (b). Geometricamente, a cada deslocamento do elemento estruturante sobre a imagem, é observado a interseção dos pixels ativos da imagem com os pixels ativos do elemento estruturante, se a interseção for o próprio conjunto de pixels ativos do elemento estruturante, ou seja, o elemento estruturante está totalmente contido numa região clara da imagem, então a imagem resultante terá seu pixel, que corresponde ao pixel da imagem original que coincide com o centro do elemento estruturante, ativo.



**Figura A.3** - resultado da aplicação da operação erosão sobre os operandos vindos da Figura A.1.

Operação gradiente morfológico denotada por é  $\Psi_g(I)$  uma operação de  $I_{m \times n} \times I_{r \times s} \rightarrow I_{m \times n}$  tal que:

$$\Psi_g(I) = (\mathcal{D}_g(I) \sim \mathcal{E}_g(I)) \quad \text{Equação A.6.8}$$

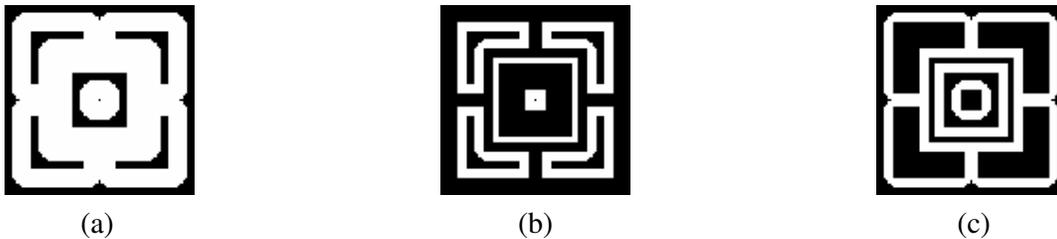
No caso de imagens binárias, o gradiente morfológico é usado para encontrar bordas da imagem, como podemos ver na Figura A.4 (a). Deste processo derivam duas outras técnicas de extração de bordas de imagens que são visualmente mais intuitivas. As técnicas são chamadas

borda interna, Figura A.4 (b), e borda externa, Figura A.4 (c), que são expressos, respectivamente, pela Equação A.6.9 e Equação A.6.10.

Operações de borda interna e externa denotadas respectivamente por  $\psi_g^{in}(I)$  e  $\psi_g^{ex}(I)$  operações de  $I_{mxn} \times I_{rxs} \rightarrow I_{mxn}$  tais que:

$$\psi_g^{in}(I) = [I \sim \epsilon_g(I)] \tag{Equação A.6.9}$$

$$\psi_g^{ex}(I) = [\delta_g(I) \sim I] \tag{Equação A.6.10}$$



**Figura A.4** - resultados das aplicações das operações (a) gradiente morfológico, (b) borda interna e (c) borda externa sobre os operandos vindos da Figura A.1.

Por fim temos a operação abertura, denotada por  $\gamma_g(I)$  é uma operação de  $I_{mxn} \times I_{rxs} \rightarrow I_{mxn}$  tal que:

$$\gamma_g(I) = \delta_g[\epsilon_g(I)] \tag{Equação A.6.11}$$

Esta operação retira da imagem original regiões ativas que não conseguem conter a totalidade dos pixels ativos do elemento estruturante, causando a erosão dos extremos da imagem original bem como das regiões ativas que são menos espessas do que o elemento estruturante, como podemos ver na Figura A.5.



**Figura A.5** - resultado da aplicação da operação abertura sobre os operandos vindos da Figura A.1.

