



**Sandra Maria Oliveira
Ferrão Lopes**

**Técnicas Geométricas de Condensação para o
Classificador k-NN**



**Sandra Maria Oliveira
Ferrão Lopes**

**Técnicas Geométricas de Condensação para o
Classificador k-NN**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Matemática, área de especialização Ciências da Computação, realizada sob a orientação científica da Professora Doutora Gladys Castillo Jordán e do Professor Doutor António Leslie Bajuelos Domínguez, Professores Auxiliares do Departamento de Matemática da Universidade de Aveiro.

To my "José".

o júri

presidente

Professor Doutor Helmuth Robert Malonek
professor catedrático do Departamento de Matemática da Universidade de Aveiro

Professor Doutor João Manuel Portela da Gama
professor associado do Departamento de Economia da Universidade do Porto

Professor Doutor António Leslie Bajuelos Domínguez
professor auxiliar do Departamento de Matemática da Universidade de Aveiro

Professora Doutora Gladys Castillo Jordán
professora auxiliar do Departamento de Matemática da Universidade de Aveiro

agradecimentos

Agradeço aos meus Professores, Doutora Gladys Castillo Jordán e Doutor António Leslie Bajuelos Domínguez, a oportunidade que me concederam com a realização deste trabalho. Sem a sua admirável orientação, a elaboração desta tese não seria possível.

A disponibilidade e amizade, manifestadas ao longo deste tempo, foram determinantes na concretização deste projecto pessoal.

À minha família, de um modo especial aos meus filhos, um profundo agradecimento por me terem concedido o tempo e o espaço necessários à produção deste trabalho.

palavras-chave

Reconhecimento de padrões, classificador k-vizinhos mais próximos, selecção de protótipos, diagrama de voronoi, grafo de gabriel, grafo de vizinhança relativa, técnicas geométricas de condensação.

resumo

Na aproximação ao reconhecimento de padrões por aprendizagem baseada em instâncias, um conjunto de treino formado por padrões (objectos, exemplos, instâncias) são recolhidos e usados para a implementação do classificador (regra de decisão). Entre estes, o mais conhecido, é o classificador k-vizinhos mais próximos (k-NN). Esta regra de decisão, classifica um novo objecto na classe mais votada entre os seus k-vizinhos mais próximos, no conjunto de treino. Quando este (o conjunto de treino) é grande, a taxa de erro obtida para o classificador k-NN é pequena e aproxima-se do erro óptimo de Bayes. Contudo, ao considerarmos um conjunto de treino grande o custo computacional eleva-se, limitando a eficiência do classificador. De facto, para determinar os k-vizinhos mais próximos de um novo objecto (não classificado) é necessário calcular todas as distâncias entre este e o conjunto de elementos do conjunto de treino. As técnicas de condensação, identificam e eliminam os objectos do conjunto de treino que não contribuem para a melhoria do desempenho do classificador, por isso, constituem uma parte fundamental na implementação do classificador k-NN.

A presente dissertação, analisa como as estruturas geométricas de proximidade e vizinhança, tais como o diagrama de Voronoi, grafo de Gabriel e grafo de Vizinhança Relativa, mostram ser boas estratégias para a condensação do conjunto de treino, sem que se degrade significativamente o desempenho do classificador k-NN.

O trabalho inicia-se com uma breve introdução ao reconhecimento de padrões, seguido de um estudo sobre as principais características do classificador k-NN. Depois, é produzida uma análise detalhada das propriedades e dos métodos de construção associados a cada estrutura geométrica. Em seguida, são descritas as técnicas geométricas de condensação para o classificador k-NN, por aplicação de estruturas geométricas de proximidade e vizinhança. Finalmente, são referidos os resultados de estudos experimentais mais recentes, relativos à comparação de diferentes técnicas geométricas de condensação para a implementação da regra k-NN.

keywords

Pattern recognition, k-nearest neighbour classifier, prototype selection, voronoi diagram, gabriel graph, relative neighbourhood graph, geometric condensing techniques

abstract

In instance based learning approach to pattern recognition, a training set of patterns (examples, instances, objects) are collected and used to implement a classifier (decision rule). One of the most well known of such rules is the k-nearest neighbour (k-NN) classifier. This decision rule classifies an unknown object in the majority class among the k-nearest neighbours in the training set. When this one is large the k-NN decision rule gives low error rates approaching the Bayes optimal error. However, for large training datasets its efficiency is limited by a high computationally cost; indeed, to compute the nearest neighbours for each new unclassified object we need to compute all the distances between it and the whole training set. Condensing techniques, identifying and removing objects from the training set that do not improve classification performance, play an important part in the implementation of a k-NN classifier.

The main goal of this thesis is to study how geometric proximity and neighbourhood structures such as Voronoi diagram, Gabriel graph and Relative Neighbourhood graph provide good condensed training sets for the k-NN classifier while maintaining similar accuracy levels.

At first, a brief introduction to pattern recognition is given followed by a study about the principal characteristics of the k-NN classifier. Next, a detailed study of the main characteristics and construction methods of each geometric structure is carried out. Afterwards, the geometric condensing techniques to k-NN classifier based on the application of the geometric proximity and neighbourhood structures are described. Finally, results of recent experimental studies aimed at comparing different geometric condensing techniques for the k-NN decision rule are provided.

Conteúdo

1. Introdução ao Reconhecimento de Padrões	1
1.1. Introdução	1
1.2. Aprendizagem Supervisada. Conceitos Básicos	3
1.3. O Erro de Classificação	4
1.4. O Erro Ótimo de Bayes	8
1.5. Regiões e Fronteiras de Decisão	8
1.5.1. Regiões de Decisão	9
1.5.2. Funções Discriminantes	10
1.5.3. Fronteiras de Decisão	11
1.6. Abordagem Não Paramétrica	12
1.7. Contribuições da Tese	14
1.8. Organização da Tese	14
2. Classificador k-Vizinhos mais Próximos	15
2.1. Introdução	15
2.2. Regra do Vizinho mais Próximo	17
2.2.1. Comportamento Assintótico do Classificador 1–NN	19
2.3. Regra dos k–Vizinhos mais Próximos	19
2.4. Custo Computacional da Regra k–NN	21
2.4.1. Pesagem dos Vizinhos e dos Atributos	22
2.4.2. Redução do Cálculo de Distâncias	24
2.4.3. Seleção de Protótipos	25
2.4.3.1. Técnicas de Edição	26

2.4.3.2. Técnicas de Condensação	28
2.5. Algoritmos Eficientes para a procura do Vizinho mais Próximo	31
3. Estruturas de Proximidade e Vizinhança	35
3.1. Introdução	35
3.2. Diagrama de Voronoi	37
3.2.1. Preliminares, Definições e Propriedades Elementares .	37
3.2.2. Construção do Diagrama de Voronoi	48
A. Construção do DV por Intersecção de Semi-planos . . .	48
B. Construção do DV pelo Método Incremental	49
C. Construção do DV pelo Método de Divisão e Conquista	50
D. Construção do DV pelo Método Linha de Varrimento .	55
3.3. Grafos de Vizinhança	60
3.3.1. Grafo de Gabriel	61
3.3.2. Grafo de Vizinhança Relativa	66
3.4. Relações entre os Grafos	68
4. Aplicação de Estruturas de Proximidade nos Algoritmos de Condensação da Regra k - NN	71
4.1. Introdução	71
4.2. Condensação por Estruturas de Proximidade	72
4.2.1. Condensação de Voronoi	74
4.2.2. Condensação de Gabriel	77
4.2.3. Condensação de Vizinhança Relativa	80
4.3. Resultados Experimentais	82
4.4. Estudos Recentes	84
5. Considerações Finais	89
Bibliografia	93

Capítulo 1

Introdução ao Reconhecimento de Padrões

1.1. Introdução

A Aprendizagem é uma das tarefas que caracteriza o Ser Humano, bem como a capacidade de decisão (classificação) perante um conjunto de informação. Hoje, a Sociedade global dispõe de grandes quantidades de informação que é preciso processar em tempo útil, isto é, extrair, de uma forma criteriosa, informação relevante que possa ser usada na tomada de futuras e importantes decisões assim como descobrir padrões de interesse, comportamentos, tendências, etc. Dada a incapacidade do Ser Humano em realizar tal tarefa, este, pela sua inteligência e engenho, tem vindo a desenvolver técnicas que o possam coadjuvar nesta função.

Aprendizagem Automático (*Machine Learning*) é uma disciplina da área da Inteligência Artificial que visa o estudo e desenvolvimento de programas computacionais que automaticamente melhoram o seu desempenho com a experiência [Mitchell, 1997]. O Reconhecimento de Padrões (RP) é um subtópico de Aprendizagem Automático que se dedica, fundamentalmente, a desenvolver técnicas automatizadas capazes de aprender a descrever, classificar ou reconhecer regularidades significativas perante um conjunto de dados. Segundo Schalkoff, RP consiste em descobrir e classificar padrões de interesse num

determinado meio [Schalkoff, 1992]. Em termos gerais, *padrão* é uma entidade vagamente definida, pela qual é possível interpretar o mundo e à qual pode ser atribuído um nome [Fu, 1982]. Um padrão pode assumir as mais variadas formas: uma sequência de DNA, uma impressão digital, uma imagem, um som, etc. Nos últimos anos, as diferentes técnicas de Reconhecimento de Padrões têm sido amplamente difundidas e eficientemente aplicadas na resolução de problemas da vida real, uma vez que muitas ocorrências da vida humana aparecem sob a forma de padrões. Entre as áreas de aplicação, podemos mencionar:

- Bioinformática (análise da sequência de DNA);
- Reconhecimento Biométrico (impressão digital, reconhecimento da face);
- Diagnóstico médico (TAC, ecografia);
- Classificação de documentos de Internet;
- Inspeção visual em automação industrial;
- Mineração de dados (Data Mining).

Um dos objectivos principais nos problemas de RP é a classificação e descrição de padrões. Para este fim, um conjunto de características principais (atributos) são extraídas dos padrões (objectos).

Na abordagem de um problema de classificação, podemos distinguir dois tipos de aprendizagem:

- **Aprendizagem Não Supervisada** (também conhecida como *Clustering*) – a partir de um conjunto de objectos descritos por um conjunto de atributos, a tarefa de classificação consiste em descobrir grupos (*clusters*) de objectos que pelos seus atributos apresentam alguma similaridade.
- **Aprendizagem Supervisada** – a partir de um conjunto de objectos descritos pelos seus atributos e categorizados numa classe pré-definida, o objectivo fundamental é aprender um classificador (hipótese) que atribui com elevada fiabilidade uma das classes a cada novo objecto (não classificado).

O trabalho desenvolvido, nesta dissertação, enquadra-se no domínio dos problemas de aprendizagem supervisionada, nomeadamente no estudo do classificador do vizinho mais próximo.

1.2. Aprendizagem Supervisada. Conceitos Básicos

Seja $\mathbf{x} = (x_1, x_2, \dots, x_d)$ um objecto (padrão) descrito por um conjunto de atributos (vector de atributos). Ao conjunto de todas as representações de um objecto chama-se *espaço de atributos* (geralmente um subconjunto do espaço Euclidiano d -dimensional, \mathfrak{R}^d). Consideremos $\mathbf{F} \subset \mathfrak{R}^d$ o espaço de atributos.

A tarefa de classificação consiste em atribuir a um objecto $\mathbf{x} \in \mathbf{F}$ uma das classes do conjunto $C = \{c_1, c_2, \dots, c_m\}$. Mais formalmente, podemos definir um classificador como segue:

Definição 1.1. (Classificador) Um classificador é uma função $f: \mathbf{F} \rightarrow C$ que mapeia cada objecto $\mathbf{x} \in \mathbf{F}$ a uma classe $c \in C$.

A partir da definição 1.1, podemos facilmente inferir que o problema de classificação consiste em estabelecer uma relação funcional de dependência entre o conjunto de atributos observados em cada objecto e a variável classe. O resultado da classificação para o objecto \mathbf{x} é $f(\mathbf{x})$. A função f é também chamada *regra de decisão*.

Seja $\mathcal{D} = \{ \langle \mathbf{x}^{(1)}, c^{(1)} \rangle, \langle \mathbf{x}^{(2)}, c^{(2)} \rangle, \dots, \langle \mathbf{x}^{(n)}, c^{(n)} \rangle \}$ um conjunto de dados (objectos, exemplos, instâncias) onde cada *tuple* $\langle \mathbf{x}, c \rangle$ representa um objecto classificado numa das classes. Consideremos $f: \mathbf{F} \rightarrow C$ a *função alvo* (classificador) que devemos aprender a partir do *conjunto de treino* \mathcal{D} .

Definição 1.2. (Aprendizagem Supervisada) A Aprendizagem supervisionada consiste na tarefa de aprender um classificador, uma hipótese $h: \mathbf{F} \rightarrow C$, que melhor aproxime a função alvo a partir do conjunto de treino \mathcal{D} .

Consequentemente, o classificador induzido a partir do conjunto \mathcal{D} não é mais do que uma função, que deve atribuir com alta fiabilidade uma classe a cada novo objecto (obviamente não classificado). A aprendizagem de um classificador decorre numa fase conhecida como *fase de treino* ou de *aprendizagem*, pelo que o conjunto de dados usado é chamado *conjunto de treino*. Numa fase posterior, o classificador induzido pode ser usado para prever as classes de futuros objectos. Esta fase constitui a *fase de classificação* ou de *implementação*.

No contexto da área de Reconhecimento de Padrões, é usual distinguir duas abordagens diferentes ao problema de aprendizagem supervisionada: a abordagem *paramétrica* e a abordagem *não paramétrica*. Na abordagem paramétrica é conhecida uma expressão analítica para a distribuição de probabilidade das classes ou regra de decisão que depende de um conjunto de parâmetros desconhecidos [Duda & Hart, 1973], contrariamente, na abordagem não paramétrica, não existe nenhum conhecimento sobre a estrutura da distribuição de probabilidade ou regra de decisão.

1.3. O Erro de Classificação

A classificação, é em si, um problema de natureza estatística. Objectos que possam ser descritos pelo mesmo vector de atributos podem pertencer a classes diferentes. Consequentemente, não podemos esperar que um classificador seja assertivo, isto é, que atribua a classe correcta a todos os objectos. Por este motivo, torna-se importante estudar o erro do classificador definido como sendo a probabilidade com que um dado objecto é mal classificado. Se assumirmos que a cada exemplo (objecto) $\mathbf{x} \in \mathbf{F}$ é associada a classe verdadeira $c = f(\mathbf{x})$, então,

um classificador h , induzido a partir de um conjunto de dados, pode *falhar* ou *acertar* a verdadeira classe c de um novo exemplo \mathbf{x} .

Assim, dois conceitos importantes sobre o erro de classificação são o *erro de classificação verdadeiro* e a *taxa de erro*.

Definição 1.3. (Erro de classificação verdadeiro) O *erro de classificação verdadeiro* é a probabilidade com que o modelo construído irá falhar na classificação de novos exemplos, ou seja, $P(h(\mathbf{x}) \neq f(\mathbf{x}))$ para $\mathbf{x} \notin \mathcal{D}$.

Se dispuséssemos de um número muito elevado de exemplos no conjunto de treino, poderíamos com exactidão determinar o valor do *erro verdadeiro* de cada possível hipótese e escolher o classificador com menor erro. Mas, na prática, dispomos de poucos dados, sendo a medida natural do desempenho de um classificador a sua *taxa de erro*.

Definição 1.4. (Taxa de erro) A taxa de erro de um classificador h com relação a um conjunto de dados \mathcal{D} é definida como a proporção dos exemplos incorrectamente classificados:

$$Erro(h(\mathbf{x}), \mathcal{D}) = \frac{1}{n} \sum_{\langle \mathbf{x}^{(k)}, c^{(k)} \rangle \in \mathcal{D}} \delta(\mathbf{x}^{(k)}, c^{(k)}, h(\mathbf{x}^{(k)})) \quad , \quad 1 \leq k \leq n \quad (1.1)$$

onde

$$\delta(\mathbf{x}, f(\mathbf{x}), h(\mathbf{x})) = \begin{cases} 1 & \text{se } f(\mathbf{x}) \neq h(\mathbf{x}) \\ 0 & \text{se } f(\mathbf{x}) = h(\mathbf{x}) \end{cases} \quad (1.2)$$

é chamada a *função de perda zero-um*.

Inversamente, podemos usar a *taxa de acertos (accuracy)* definida como a proporção dos exemplos correctamente classificados.

O método trivial, para estimar a taxa de erro, utiliza o conjunto de dados completo, para induzir o classificador e em seguida o mesmo conjunto para estimar a sua taxa de erro. Esta taxa de erro é conhecida como *erro de re-substituição ou erro*

aparente e constitui um estimador optimista do erro verdadeiro (também conhecido por erro de generalização). O classificador induzido, em regra, *overfit* (sobre-ajusta) o conjunto de treino, o que significa que o modelo fica melhor adaptado às instâncias usadas para a sua indução do que àquelas que futuramente vão ser classificadas.

Assim, um algoritmo de aprendizagem deve ser avaliado tendo em conta o seu desempenho (capacidade de generalização) naqueles exemplos que não foram usados para construir o classificador. A ideia básica consiste em dividir o conjunto de dados disponível em dois conjuntos: *um conjunto de treino* usado pelo algoritmo de aprendizagem para induzir o classificador e um *conjunto de teste*, com os restantes exemplos, que serão usados para estimar a taxa de erro. Existem vários métodos para estimar a taxa de erro baseados em diferentes partições do conjunto de dados:

- Método *Holdout* – é o método mais simples de validação; consiste em dividir o conjunto de dados em dois subconjuntos: um conjunto de treino (2/3 partes do conjunto inicial) e outro conjunto de teste (1/3 parte do conjunto inicial). O conjunto de treino é usado para induzir o classificador, enquanto que o conjunto de teste é usado para avaliar o desempenho do classificador induzido, obtendo, assim, uma estimativa da taxa de erro verdadeira.
- Método de *Validação Cruzada k-fold* – esta técnica resume-se à aplicação sucessiva de partições ao conjunto de dados, sendo por isso uma generalização do método Holdout [Stone, 1974]. A validação cruzada consiste em efectuar k partições de igual tamanho, do conjunto de dados. O processo é repetido k vezes. Em cada iteração é escolhida a actual k partição como conjunto de teste e as restantes $k-1$ partições como conjunto de treino. A estimação da taxa de erro verdadeira é obtida como a média das taxas de erro obtidas em cada iteração.

- Método *Leave-one-out* – é um caso particular do método de validação cruzada [Lunts, 1967]. Neste caso, assume-se que k deverá ser igual ao número de exemplos (objectos) de conjunto inicial. Assim, os sucessivos subconjuntos de teste são formados por um único exemplo e o conjunto de treino é constituído por todos os exemplos, à excepção daquele que foi 'retirado' para o conjunto de teste.
- Método *Bootstrapping* – este método baseia-se na técnica estatística de re-amostragem com reposição. A partir do conjunto de dados com n exemplos é gerada uma amostra aleatória com reposição do mesmo tamanho. Esta amostra é usada como conjunto de treino, enquanto que os restantes exemplos do conjunto, que não foram seleccionados inicialmente, formam o conjunto de teste. Sabendo que um exemplo tem uma probabilidade de $(1-1/n)^n \approx e^{-1} = 0.368$ não ser seleccionado, ou seja, de ficar no conjunto de teste, o número esperado de instâncias distintas na amostra gerada é aproximadamente $0.632n$. A taxa de erro obtida a partir do conjunto de teste é um estimador muito pessimista da taxa de erro verdadeira (usa apenas 37% dos exemplos). Por este motivo, a estimativa usada por este método também inclui a taxa de erro no conjunto de treino e é calculada como segue:

$$Erro_{bootstrap} = 0.632Erro_{teste} + 0.368Erro_{treino} \quad (1.3)$$

onde $Erro_{teste}$ e $Erro_{treino}$ representam as taxas de erro no conjunto de teste e de treino, respectivamente. O processo é repetido Y vezes. Como resultado, a estimativa da taxa de erro verdadeiro é a média das taxas de erros obtidas em cada Y experiência.

1.4. O Erro Ótimo de Bayes

Nesta secção, introduzimos a Regra de Bayes, considerada ótima, para um dado problema de classificação. Basicamente, esta regra selecciona a classe mais provável para um objecto observado \mathbf{x} . Pelo Teorema de Bayes [Bolstad, 2004], podemos calcular a probabilidade *a posteriori* de cada classe c_j dado \mathbf{x} ,

$$P(c_j | \mathbf{x}) = \frac{P(\mathbf{x} | c_j)P(c_j)}{P(\mathbf{x})} \quad (1.4)$$

A regra de decisão, que minimiza a probabilidade de erro da classificação de \mathbf{x} , consiste na escolha da classe que maximiza a probabilidade *a posteriori*, ou seja, selecciona a classe c_i se:

$$P(\mathbf{x} | c_i)P(c_i) > P(\mathbf{x} | c_j)P(c_j) \quad \text{para todo } j \neq i \quad (1.5)$$

A regra de classificação (1.4) é conhecida como regra de decisão de Bayes e o seu erro resultante, o erro ótimo de Bayes, é o mais pequeno erro de classificação para um dado problema de classificação. O erro ótimo de Bayes, denotado por P_{erro} , é definido como se segue:

$$P_{\text{erro}} = 1 - \int_{\mathbf{F}} \max_i \{P(\mathbf{x} | c_i)P(c_i)\} d\mathbf{x} \quad (1.6)$$

1.5. Regiões e Fronteiras de Decisão

Os conceitos abordados nesta secção provêm, fundamentalmente, da área do *Reconhecimento Estatístico de Padrões* [Devijver, 1982], [Fukunaga, 1990] e [Devroye et al., 1996] que são fundamentais para o trabalho que se desenvolve nesta tese. Considerando que o espaço de atributos possui uma estrutura de espaço vectorial e/ou métrico, na abordagem geométrica de cada objecto passa a ser visto como um ponto no espaço dos atributos (um espaço d -dimensional) cujas coordenadas representam o vector de atributos.

Admitamos que um classificador (regra de decisão) $h: \mathbf{F} \rightarrow \{c_1, \dots, c_m\}$ é aprendido do conjunto de treino \mathcal{D} .

1.5.1. Regiões de Decisão

A partição gerada pelo classificador no espaço de atributos conduz-nos a uma outra interpretação da tarefa de classificação. A cada objecto é atribuída uma única classe, o que significa que o classificador define quais os objectos descritos pelos seus atributos (pontos no espaço de atributos) que pertencem a uma determinada classe. As porções do espaço de atributos que correspondem às classes c_1, c_2, \dots, c_m são denotadas por R_1, R_2, \dots, R_m , respectivamente. Assim,

$$R_i = \{\mathbf{x} : h(\mathbf{x}) = c_i\} \quad (1.7)$$

representa o conjunto das *regiões de decisão* que formam a partição do espaço de atributos onde $R_i \cap R_j = \emptyset$ para $i \neq j$ e $R_1 \cup \dots \cup R_m = \mathbf{F}$.

Conforme a regra de decisão h define a região de decisão, também as regiões de decisão ajudam a definir uma regra de decisão. Em particular, para espaços de atributos pequenos, a representação das regiões de decisão mostram, facilmente, como o classificador trabalha (ver Figura 1.1).

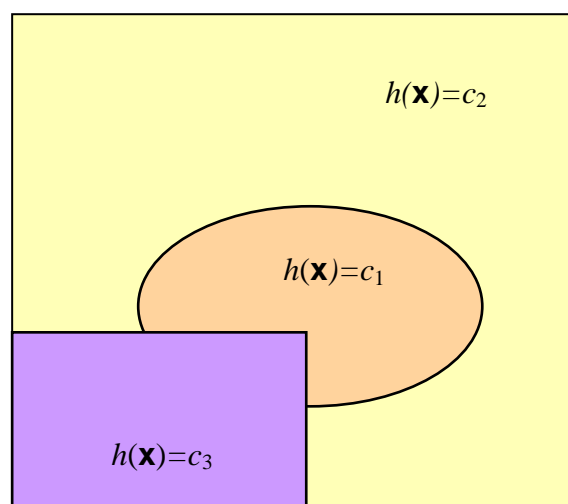


Figura 1.1. Regiões de Decisão

1.5.2. Funções Discriminantes

Em geral, um classificador pode ser definido através de um conjunto de funções discriminantes. Suponhamos que para cada classe c_i é definida a sua própria função discriminante g_i que toma um objecto $\mathbf{x} \in \mathbf{F}$ como entrada. A classe atribuída a este objecto é a classe com o valor máximo para $g_i(\mathbf{x})$, ou seja, o classificador atribui a um objecto $\mathbf{x} \in \mathbf{F}$ a classe c_i se:

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \text{ para todo } i \neq j \quad (1.8)$$

Por exemplo, para a regra de decisão de Bayes, observamos que o mesmo classificador pode ser representado de duas formas diferentes, o que significa que podemos usar dois tipos equivalentes de funções discriminantes para a sua representação:

$$g_i = P(c_i | \mathbf{x}), \quad i = 1, \dots, m \quad (1.9)$$

ou

$$g_i = P(\mathbf{x} | c_i)P(c_i), \quad i = 1, \dots, m \quad (1.10)$$

Deste modo, as funções discriminantes definidas para um certo classificador não são únicas¹. Porém, um conjunto de funções discriminantes define um único classificador. Em geral, pretende-se que a representação escolhida seja a mais simples possível.

Dentro do conjunto de funções discriminantes podemos distinguir funções discriminantes lineares e não lineares. No âmbito deste trabalho, interessa referir o primeiro caso que se define da seguinte forma:

$$g_i(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \quad (1.11)$$

onde \mathbf{w} é um vector com dimensionalidade igual à dimensionalidade do vector de atributos \mathbf{x} e b é um escalar.

¹ Isto significa que as funções discriminantes podem ser multiplicadas por constantes positivas, adicionadas constantes ou outras operações sem que alterem o classificador subjacente.

O interesse em classificadores lineares deve-se ao facto de estes poderem ser representados unicamente por funções discriminantes lineares, que são as mais simples.

1.5.3. Fronteiras de Decisão

Consideremos um problema de classificação que tem duas classes e que o classificador pode ser representado pela função discriminante,

$$g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x}) \quad (1.12)$$

Se $g(\mathbf{x}) > 0$, então \mathbf{x} é classificado em c_1 , caso contrário \mathbf{x} é classificado em c_2 .

O classificador divide o espaço de atributos em duas regiões correspondentes a cada uma das classes. As regiões de decisão (1.7) podem ser definidas à custa de cada função discriminante, como se segue:

$$R_i = \{\mathbf{x} \mid g_i(\mathbf{x}) > g_j(\mathbf{x}), i \neq j\} \quad (1.13)$$

As fronteiras entre as regiões de decisão,

$$R_{ij} = \{\mathbf{x} : g_i(\mathbf{x}) = g_j(\mathbf{x}), g_k(\mathbf{x}) < g_i(\mathbf{x}), k \neq i, j\} \quad (1.14)$$

são chamadas *fronteiras de decisão*.

Resumindo, é fácil entender o aspecto geométrico de um classificador baseado nas fronteiras de decisão que directamente delimitam as regiões de decisão, ou seja, formam a partição do espaço de atributos.

Assim, em aprendizagem supervisionada o objectivo de todo classificador durante a etapa de aprendizagem consistirá, basicamente, em estabelecer um conjunto de fronteiras de decisão no espaço dos atributos, que separam os objectos que pertencem a classes diferentes. Deste modo, a futura classificação de um novo objecto, ou seja, a regra de decisão sobre que classe atribuir a um objecto, virá determinada pelo lado da fronteira de decisão onde este se encontre.

Os classificadores ditos lineares (ver Figura 1.2 (a)), por exemplo, aprendem uma função discriminante que produz uma fronteira de decisão linear definida pelos hiperplanos que separam as classes. Os classificadores ditos não lineares (ver

Figura 1.2 (b)), pelo contrário, aprendem funções discriminantes não lineares, por isso as fronteiras de decisão são também não lineares.

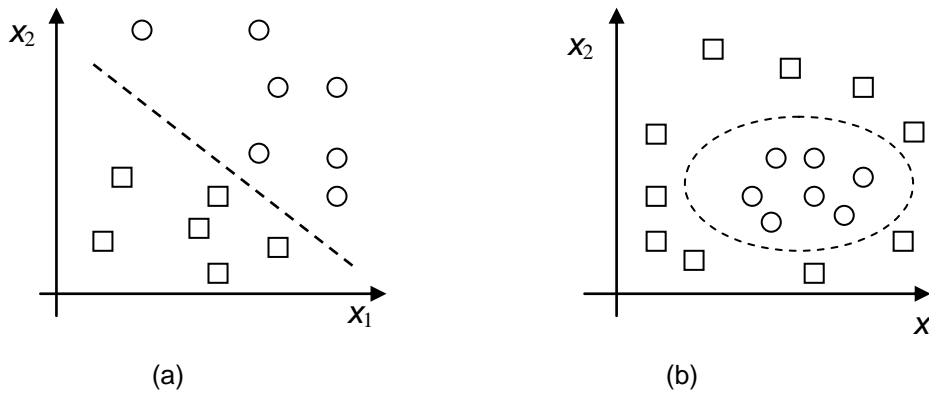


Figura 1.2. (a) Classificador linear; (b) Classificador não linear

1.6. Abordagem Não Paramétrica

Na *abordagem paramétrica* à Aprendizagem Supervisada, assume-se um conhecimento *a priori* da forma estrutural das distribuições de probabilidades de cada classe sobre o espaço de atributos, as quais se apresentam determinadas por um conjunto finito e, normalmente, fixo de parâmetros. Contrariamente, na *abordagem não paramétrica*, o único conhecimento *a priori* será o correspondente à informação induzida a partir do conjunto (de treino) de objectos previamente classificados. Consequentemente, no primeiro caso, as fronteiras de decisão estarão definidas pelas referidas distribuições das classes, enquanto que, no segundo caso, as fronteiras de decisão estarão determinadas pelos objectos do conjunto de treino.

Uma das mais conhecidas técnicas de abordagem não paramétrica, relacionadas com os problemas de RP, são os algoritmos de *aprendizagem baseados em instâncias* (*IBL – Instance Based Learning*). Neste contexto, a classificação de uma nova instância (objecto) faz-se com base no conhecimento de um conjunto

de instâncias previamente rotuladas numa classe. Um dos classificadores mais conhecidos nesta família é a *regra de decisão do vizinho mais próximo* (*nearest neighbour rule*) [Cover & Hart, 1967].

A regra de decisão do vizinho mais próximo é muito simples e, também, muito consistente: “Classifica um novo objecto (com classe desconhecida), com a mesma classe do objecto mais próximo no conjunto de treino” [Cover & Hart, 1967].

Por outro lado, a regra de decisão do vizinho mais próximo pode ser estendida aos k-vizinhos mais próximos (k-NN), segundo a qual, a classificação de um novo objecto de faz de acordo com a classe mais comum entre os k-vizinhos mais próximos.

Uma vez que o classificador actua mediante a chegada de um novo objecto não classificado, é fácil entender, que neste caso, não existe uma fase preliminar de aprendizagem. Por isso, este tipo de classificador é conhecido como *classificador preguiçoso* (*lazy classifier*). Assim, a fase de classificação envolverá um maior custo computacional, pois sempre que chega um novo objecto para ser classificado é necessário calcular todas as distâncias entre o novo objecto e cada elemento do conjunto de treino para, assim, poder escolher a classe do(s) vizinho(s) mais próximo(s). Posto isto, um dos objectivos fundamentais dos algoritmos de aprendizagem, baseados em instâncias, é reduzir o conjunto de treino de tal forma que não afecte a capacidade de generalização da regra de decisão [Aha et al., 1991]. Ao reduzirmos o número de objectos do conjunto de treino, o número de operações a realizar para procurar o vizinho mais próximo será menor, conseqüentemente o custo da implementação do classificador será inferior.

O principal objectivo da presente tese é estudar as técnicas de condensação (redução) do conjunto de treino para a regra de decisão do k-vizinhos mais próximos, pela aplicação de estruturas de proximidade e vizinhança inerentes ao domínio da Geometria Computacional.

1.7. Contribuições da Tese

As contribuições desta tese são:

- Compilar os aspectos mais significativos do classificador k-vizinhos mais próximos;
- Resumir as principais propriedades e métodos de construção do Diagrama de Voronoi, grafo de Gabriel e grafo de Vizinhança Relativa;
- Evidenciar as propriedades naturais das estruturas geométricas de vizinhança e proximidade com as fronteiras de decisão induzidas pelo classificador k-NN;
- Estender as técnicas de Geometria Computacional à condensação do conjunto de treino para a implementação do classificador k-NN.

1.8. Organização da Tese

Os capítulos subsequentes encontram-se organizados da seguinte forma: no Capítulo 2 são apresentadas as propriedades do classificador vizinho mais próximo e analisadas as principais técnicas de selecção de protótipos (objectos): edição e condensação. O Capítulo 3 é dedicado às estruturas de proximidade e vizinhança, no qual se faz um estudo detalhado sobre as principais propriedades e algoritmos de construção do Diagrama de Voronoi, grafo de Gabriel e grafo de Vizinhança Relativa. Em seguida, no Capítulo 4, são analisadas as técnicas geométricas de condensação (do conjunto de treino) baseadas nas estruturas geométricas apresentadas no capítulo anterior. Finalmente, no Capítulo 5, são apresentadas algumas conclusões e considerações finais sobre o trabalho realizado.

Capítulo 2

Classificador k–Vizinhos mais Próximos

No início deste capítulo é feita uma descrição do classificador vizinho mais próximo, assim como a sua extensão: o classificador k–vizinhos mais próximos. São apresentadas as principais características e propriedades de cada um. Em seguida, faz-se uma análise do custo da implementação associado a este classificador e são descritas algumas técnicas que visam reduzir o tamanho do conjunto de treino de uma forma criteriosa, denominadas por técnicas de selecção de protótipos. Por fim, são referidas as duas grandes famílias de algoritmos que procuram de modo eficiente o vizinho mais próximo.

2.1. Introdução

No domínio das técnicas da *aprendizagem baseada em instâncias*, a mais conhecida e referida na literatura [Toussaint et al.,1992], [Toussaint, 2002], [Bhattacharya et al., 2005], etc., é aquela que se baseia em critérios de vizinhança. Como o nome sugere, esta técnica faz a classificação de um novo objecto, com base no conhecimento das classes dos objectos que se encontram mais próximos deste, no espaço de atributos. Os métodos de classificação assentes nesta ideia exigem a definição de uma métrica, que ajude a comparar as distâncias entre os diferentes objectos. O tipo de métrica que mais se utiliza na

classificação por vizinhança é a distância Euclidiana, mas isto não significa que seja a única. Porém, toda a família de *distâncias Minkowski*:

$$\|d(x_i, y_i)\| = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}, \quad x, y \in \mathfrak{R}^n \quad (2.1)$$

para $p=1$: *distância Hamming*;

para $p=2$: *distância Euclidiana*;

para $p=\infty$: *distância Tschebyshev*

podem ser usadas como critério de avaliação dos vizinhos mais próximos.

É importante destacar que, independentemente da métrica que se adopte, qualquer problema de classificação assente em critérios de vizinhança pode ser caracterizado do seguinte modo:

- I. *Dado um conjunto de treino \mathcal{D} com n instâncias (exemplos, objectos) previamente classificados;*
- II. *Pretende-se classificar uma nova instância \mathbf{z} , que não pertence ao conjunto de treino \mathcal{D} ;*
- III. *Entre as diferentes instâncias do espaço de atributos está definida uma métrica;*
- IV. *Nenhuma outra informação se conhece sobre a distribuição dos parâmetros estatísticos associados ao conjunto de treino \mathcal{D} .*

Da caracterização apresentada, podemos entender que a classificação de um novo objecto depende (unicamente) do conjunto de treino \mathcal{D} , isto é, dos objectos correctamente classificados. Todavia, esta característica nem sempre é verdadeira, pois, em certos casos, os objectos estão mal classificados. Assim, uma das primeiras dificuldades a enfrentar consiste em corrigir, de alguma forma, as instâncias erradamente classificadas e eliminar as instâncias que não contribuem para a classificação (assunto tratado na secção 2.4.3 – selecção de protótipos).

Outros aspectos que merecem a nossa atenção no conjunto de questões relacionadas com este classificador são: o número de vizinhos que se tem em conta para a classificação de uma nova instância, o peso de cada atributo e o peso de cada vizinho na decisão da classe, e como procurar de modo eficiente o(s) vizinho(s) mais próximo(s).

2.2. Regra do Vizinho mais Próximo

Fix e Hohges, em 1951 [Fix & Hohges, 1951], conceberam uma das mais simples e interessantes ideias sobre classificação conhecida por *regra do vizinho mais próximo* (*nearest neighbour rule, 1-NN-rule*). A sua concepção surgiu da análise de que objectos de uma mesma classe estariam mais próximos no espaço de atributos F . Assim, a classificação de um novo objecto passaria pela pesquisa da classe dos objectos mais próximos deste.

Seja $\mathcal{D} = \{ \langle \mathbf{x}^{(1)}, c^{(1)} \rangle, \langle \mathbf{x}^{(2)}, c^{(2)} \rangle, \dots, \langle \mathbf{x}^{(n)}, c^{(n)} \rangle \}$ um conjunto de treino com exemplos previamente classificados e \mathbf{z} um novo objecto que se pretende classificar.

Definição 2.1. A regra de decisão 1-NN classifica o novo objecto \mathbf{z} na classe c_j do objecto do conjunto de treino, $\langle \mathbf{x}_j, c_j \rangle \in \mathcal{D}$, mais próximo de \mathbf{z} .

A Figura 2.1 ilustra um exemplo da regra de decisão 1-NN.

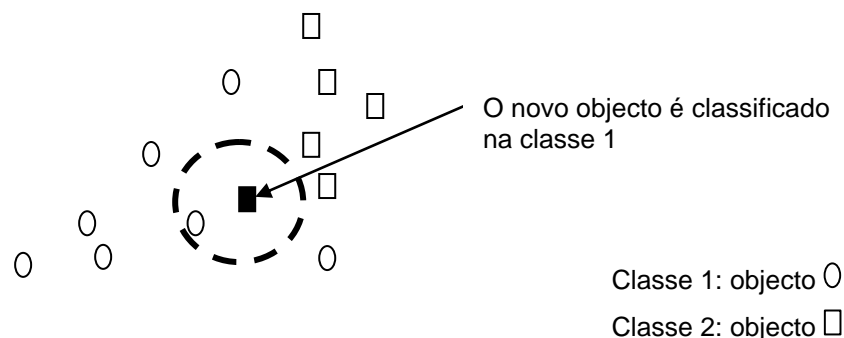


Figura 2.1. Regra do vizinho mais próximo

Outra forma de entender o desempenho do classificador 1-NN é através das fronteiras de decisão induzidas a partir do conjunto de treino. Como foi referido no Capítulo 1 (secção 1.5), o problema da classificação de um novo objecto pode ser abordado através da partição do espaço de atributos em regiões de decisão. Pela sua definição, cada região contém todos os objectos que pertencem a uma mesma classe. As fronteiras de decisão delimitam as regiões de decisão. Como mostra a Figura 2.2, podemos determinar uma fronteira de decisão que separa os objectos da classe 1 da classe 2, situados em duas regiões de decisão. Assim, a classificação de um novo objecto z resume-se à verificação de que lado da fronteira (região de decisão) se encontra.

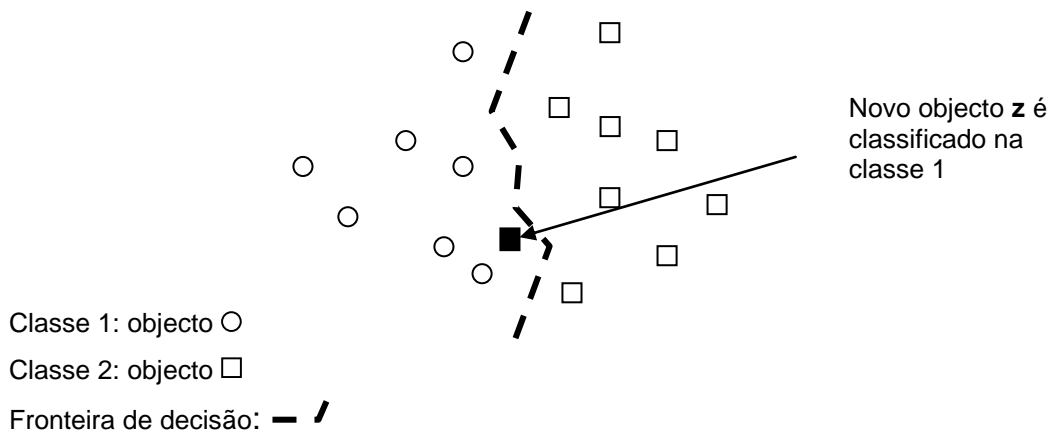


Figura 2.2. Partição do espaço de atributos em duas regiões distintas

A procura do vizinho mais próximo realiza-se através do cálculo de todas as distâncias entre o novo objecto (não classificado) e os restantes n objectos do conjunto de treino (um espaço de atributos de dimensão d). Isto significa que na classificação de um novo objecto intervêm todos os objectos do conjunto de treino, o que implica, por sua vez, um custo de implementação elevado. Por outro lado, este classificador assume que todos os atributos são independentes da classe, hipótese que raramente se verifica em contextos reais. À partida, estas desvantagens desencorajaram a aplicação desta regra em problemas de classificação. Mas Cover e Hart, em 1967, demonstraram um resultado importante sobre o comportamento assintótico deste classificador [Cover & Hart, 1967].

2.2.1. Comportamento Assimptótico do Classificador 1-NN

Seja \mathbf{z} um novo objecto a ser classificado, e seja \mathbf{x}_i o objecto do conjunto de treino \mathcal{D} mais próximo de \mathbf{z} , então, a regra do vizinho mais próximo classifica o objecto \mathbf{z} na classe c_i .

Seja $P_{erro}(1-NN) = P\{c \neq c_i\}$ o resultado da probabilidade de erro, quando c é a verdadeira classe de \mathbf{z} .

Os autores anteriormente referidos demonstraram que, quando n tende para infinito, o erro assimptótico de 1-NN é dado por:

$$P_{erro} \leq P_{erro}(1-NN) \leq P_{erro} \left[2 - m \left(\frac{P_{erro}}{m-1} \right) \right], \text{ para } m \text{ classes} \quad (2.2)$$

onde P_{erro} é a probabilidade de erro de Bayes (óptima) e $P_{erro}(1-NN)$ denota o erro assimptótico do classificador 1-NN. Assim, a probabilidade de erro do classificador 1-NN está limitado superiormente por duas vezes o erro de Bayes, o que significa que esta regra apresenta pelo menos metade da informação que seria dada por um classificador óptimo. Quando se considera um conjunto de treino muito grande ($n \rightarrow \infty$), a probabilidade de erro de 1-NN está próxima do óptimo.

Através de uma conveniente mudança da regra 1-NN, podemos obter uma probabilidade de erro tão próxima do óptimo quanto se deseje. Tal modificação é a chamada regra dos k -vizinhos mais próximos.

2.3. Regra dos k -Vizinhos mais Próximos

A regra dos k -vizinhos mais próximos (k -NN) acaba por ser uma extensão da regra do vizinho mais próximo. Neste caso, tem-se em conta não apenas a classe de um único vizinho, o vizinho mais próximo, mas as classes dos k (número a nomear) vizinhos mais próximos. Assim, a classe atribuída ao novo objecto \mathbf{z} será

a classe mais votada entre os k -vizinhos mais próximos do conjunto de treino. A principal vantagem da regra k -NN em relação à regra 1-NN radica no facto de que, ao se considerarem vários vizinhos na classificação de um novo objecto, aproveita-se de forma mais eficiente a informação que se pode extrair do conjunto de treino [Sierra, 2000].

Tomando o mesmo exemplo apresentado na Figura 2.1, aplicou-se a regra k -NN para $k = 3$ vizinhos (ver Figura 2.3).

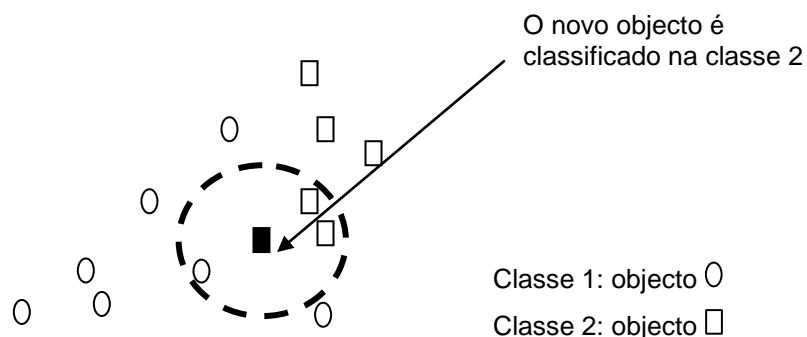


Figura 2.3. Regra do k -vizinho mais próximo ($k = 3$)

Da análise comparativa entre as duas regras, podemos pensar que a regra k -NN é melhor que a regra 1-NN, isto é, que quanto maior for o número de k , melhor será o desempenho do classificador. De facto isto não é verdade, pois como é ilustrado na Figura 2.4², alguns estudos empíricos mostram, que de um modo geral, os melhores resultados são obtidos para $k = 3$ ou $k = 5$.

Salientamos ainda que, por vezes, ao aplicar-se esta regra pode surgir um problema de empate no voto da classe entre os k -vizinhos mais próximos do novo objecto a classificar. Uma forma clássica de resolver este impasse é escolher um número ímpar de vizinhos.

² A Figura 2.4. foi extraída da página 99 de [Sierra, 2000]

Percentagem de objectos bem classificados

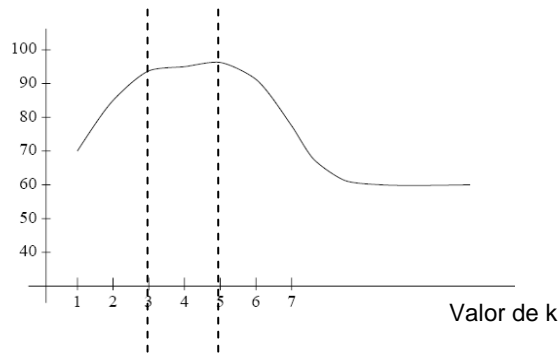


Figura 2.4. Comportamento esperado do algoritmo k-NN em função do valor de k

Em todo caso, existem outros métodos para resolver o empate de voto, como, por exemplo: atribuir a classe do vizinho mais próximo entre as classes empatadas ou atribuir a classe cujo valor da média das distâncias dos vizinhos de cada classe ao objecto seja a menor (para mais informações consultar, por exemplo, [Sierra, 2000]).

2.4. Custo Computacional da Regra k-NN

A aplicação prática da regra k-NN ficou marcada, desde muito cedo, por duas críticas: uma dirigida à complexidade temporal $T(n)$, outra dirigida à complexidade espacial $S(n,d)$.

Uma vez que na classificação de um novo objecto \mathbf{z} é necessário calcular todas as distâncias entre ele e os elementos do conjunto treino, a procura do vizinho mais próximo entre os n objectos requiere o cálculo de n distâncias, o que significa um custo operacional de ordem $O(n)$. Para além disso ainda é preciso avaliar a procura dos k -vizinhos mais próximos, isto é, fazer

$$k \left(\frac{n - (k - 1)}{2} \right) \quad (2.3)$$

comparações, o que significa que a regra k-NN tem uma complexidade temporal $T(n)$ de ordem $O(kn)$. Por outro lado, o armazenamento de n instâncias com d atributos, referente ao conjunto de treino, implica uma complexidade espacial $S(n,d)$ de ordem $O(n^d)$. Tendo em conta estas considerações, o custo computacional associado à implementação do classificador do vizinho mais próximo pode ser impraticável para muitas situações, dada a sua complexidade temporal e espacial.

Na tentativa de minimizar estes problemas e na sequência de muitos trabalhos de investigação, foram propostos alguns esquemas, tais como: associar um peso ao vizinho e/ou um peso a cada atributo, reduzir o cálculo de distâncias e reduzir o conjunto de treino através das chamadas técnicas de selecção de protótipos. Todas estas técnicas e esquemas podem ser combinados entre si, resultando num complexo sistema de implementação computacional para os métodos baseados em critérios de vizinhança.

Nas secções seguintes, pretende-se fazer um resumo que ilustre cada uma destas ideias.

2.4.1. Pesagem dos Vizinhos e dos Atributos

A partir da definição de classificador k-NN, compreendemos que para um novo objecto \mathbf{z} todos os seus k-vizinhos mais próximos têm igual contributo de voto, isto significa que todos os vizinhos e, conseqüentemente, os seus atributos têm peso $w_i = 1, i = 1, \dots, k$. No entanto, parece natural entender que aqueles que estão mais próximos de \mathbf{z} tenham um maior peso no voto. Podemos encontrar no artigo de Toussaint uma resenha histórica sobre os principais métodos de pesagem de vizinhos e atributos [Toussaint, 2002].

Pesagem dos Vizinhos

A ideia original sugerida por Royall, em 1966, reflecte-se no seguinte esquema: ao i -ésimo vizinho de \mathbf{z} é atribuído o peso w_i , onde

$$w_1 \geq w_2 \geq \dots \geq w_k \quad \text{e} \quad w_1 + w_2 + \dots + w_k = 1 \quad (2.4)$$

Dudani, em 1976, propôs um esquema em que o peso w_i atribuído ao vizinho \mathbf{x}_i de \mathbf{z} seja inversamente proporcional à distância entre eles:

$$w_i = \frac{1}{\text{dist}(\mathbf{z}, \mathbf{x}_i)}, \quad 1 \leq i \leq k \quad (2.5)$$

Outras ideias surgiram, como, por exemplo:

- Voto fixo – é atribuído um peso fixo w_1, w_2, \dots, w_k aos sucessivos k -vizinhos. Neste caso é irrelevante o valor da distância entre \mathbf{z} e os seus vizinhos;
- Voto ponderado – em caso de empate, é atribuída à classe menos provável, um maior peso. Neste caso é exigido que se conheçam as probabilidades, *a priori*, de cada classe a que pertencem os objectos (vizinhos de \mathbf{z}) em caso de empate.

Podemos encontrar revisões mais específicas sobre este tema em [Wettschereck, 1994].

Pesagem dos Atributos

Evidentemente que a ideia que se apresentou na subsecção anterior acabou por se propagar à pesagem dos atributos. Assim, é também natural pensar que nem todos os atributos de um objecto \mathbf{x} contribuam de igual modo para a mesma classe.

Se para um problema de classificação são analisados d atributos, também designados por variáveis predictoras X_1, X_2, \dots, X_d , e lhes são atribuídas os pesos

w_1, w_2, \dots, w_d , respectivamente, então, podemos afirmar que a distância entre dois objectos $\mathbf{x} = (x_1, x_2, \dots, x_d)$ e $\mathbf{y} = (y_1, y_2, \dots, y_d)$ é dada por:

$$dist_w(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^d w_i (x_i - y_i)^2 \quad (2.6)$$

significando que a cada atributo é associado um determinado peso w_i . Consequentemente, muitos trabalhos se debruçaram sobre o cálculo destes diferentes pesos. Podemos encontrar em [Sierra, 2000] a descrição de alguns métodos para a escolha de diferentes pesos dos atributos.

Contudo, a forma mais natural de o fazer é escolher dois tipos de pesos, $w_1 = 0$ e $w_2 = 1$. Os atributos seleccionados para intervir na aprendizagem do classificador seguem com peso 1, e aos restantes é associado o peso 0. Este processo é conhecido por *selecção de conjunto de atributos* (*feature subset selection* [Inza, 1999]).

Também podemos escolher três tipos de pesos, $w_1 = 0$, $w_2 = 0,5$ e $w_3 = 1$, que de alguma forma associam a importância dos atributos no processo de classificação e evita a sobreposição dos objectos [Kohavi & Sommerfield, 1995]. Este procedimento pode ser estendido a tantos quantos pesos se desejarem no intervalo $[0,1]$, normalmente equidistantes entre si.

2.4.2. Redução do Cálculo de Distâncias

Um dos vectores de intervenção para melhorar a pesquisa dos k -vizinhos mais próximos tem sido a procura de algoritmos que reduzam de alguma forma o cálculo de distâncias entre o novo objecto não classificado e os objectos do conjunto de treino. Com este intuito, Vidal, em 1986, propôs o algoritmo AESA – *Approximating Eliminating Search Algorithm* [Vidal, 1986]. Mais tarde, foram feitas duas versões melhoradas chamadas LAESA (*Linear Approximating – Eliminating Search Algorithm*) e TLAESA (*Tree Linear Approximating – Eliminating Search Algorithm*).

Basicamente, estes algoritmos desenvolvem-se em duas fases: *aproximação* e *eliminação*. Na primeira fase, a partir da eleição de um objecto, que esteja muito próximo do novo objecto a classificar, faz-se a actualização do vizinho mais próximo até ao momento. Na segunda fase, utiliza-se uma matriz triangular³ de

$$\frac{\|n\|(\|n\| - 1)}{2} \quad (2.7)$$

distâncias entre os protótipos do conjunto de treino para eliminar todos aqueles que não podem estar mais próximos do objecto a classificar até ao momento. Contudo, o armazenamento desta matriz implica um custo espacial muito elevado. Os algoritmos LAESA e TLAESA surgiram justamente como melhoramentos deste aspecto. Um estudo mais aprofundado de cada um destes algoritmos é feito em [Micó, 1996]. Existem outras versões de algoritmos que têm o mesmo objectivo dos aqui apresentados, cuja referência se encontra em [Fukunaga & Narendra, 1975], [Friedman et al., 1975] entre outras.

2.4.3. Selecção de Protótipos

Até aqui foram apresentadas estratégias e algoritmos que visam diminuir o custo computacional associado à regra do vizinho mais próximo, sem que haja, explicitamente, uma redução do próprio conjunto de treino. Todavia, é possível reduzir o número de objectos (do conjunto de treino), sem que isso agrave (consideravelmente) o desempenho do classificador. É comum denominar os objectos que formam o subconjunto resultante da redução por *protótipos*.

Entende-se por técnicas de *selecção de protótipos* todo um conjunto de procedimentos que conduzem à selecção de um subconjunto representativo do conjunto de treino. A aplicação destas técnicas para a redução do conjunto de treino tem por objectivo aliviar o custo computacional inerente às regras de classificação baseadas em critérios de vizinhança.

³ Esta matriz é calculada numa fase de pré-processamento.

Ao pesquisarmos técnicas de selecção de protótipos, somos confrontados com uma vasta literatura que se inicia, em 1968, com a proposta de *condensação de Hart*. Recentemente, podemos encontrar o desenvolvimento destas técnicas na área de *Machine Learning*.

Para uma melhor caracterização, as técnicas de selecção de protótipos dividem-se em duas grandes famílias: *técnicas de edição* e *técnicas de condensação*.

2.4.3.1. Técnicas de Edição

As técnicas de edição consistem em métodos que têm por objectivo eliminar inconsistências do conjunto de treino, isto é, eliminam fundamentalmente objectos que se encontram fora do espaço de representação da sua classe (*outliers*).

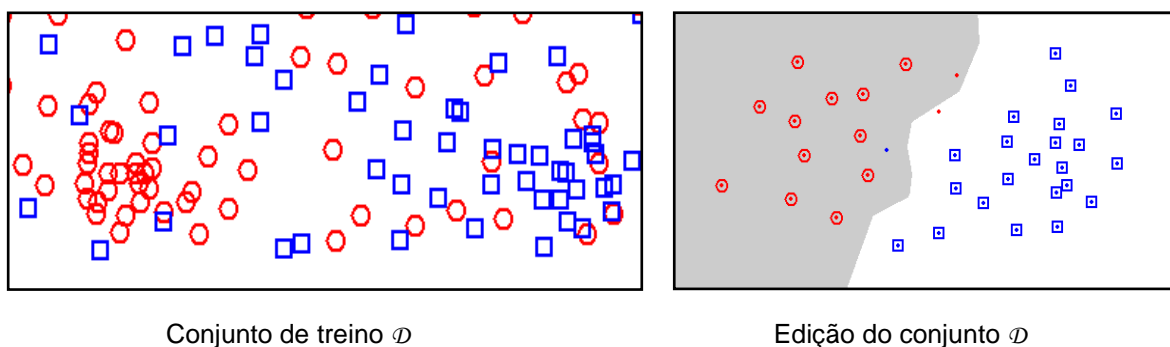


Figura 2.5. Resultado da Técnica de Edição ao conjunto de treino \mathcal{D}

A ‘limpeza’ das possíveis sobreposições entre regiões de classes distintas conduz a grupos mais homogêneos (*clusters*) entre os objectos de cada classe, fomentando a melhoria do desempenho do classificador. Por outro lado, a aplicação de alguma técnica de edição a um conjunto de treino produz fronteiras mais simples, favorecendo a aplicação de procedimentos posteriores, como as técnicas de condensação [Sánchez et al., 1997].

O algoritmo de *edição de Wilson* [Wilson, 1972], constitui a primeira proposta de redução do conjunto de treino mediante a eliminação de objectos mal classificados segundo a técnica de validação *leaving-one-out* (Capítulo 1, secção 1.3).

Em seguida, apresentamos o pseudo-código do algoritmo de edição de Wilson:

E1: Algoritmo de Edição de Wilson

Entrada: Conjunto de treino \mathcal{D}

Saída: Subconjunto editado \mathcal{D}_E

Passo 1: Para cada instância \mathbf{x} do conjunto de treino \mathcal{D}

1.1. Encontrar os k -vizinhos mais próximos de \mathbf{x} , no conjunto de treino \mathcal{D}

1.2. Classificar \mathbf{x} segundo a classe mais votada entre os seus k -vizinho mais próximos

Passo 2: Editar o conjunto de treino, eliminando todas as instâncias incorrectamente classificadas

Figura 2.6. Pseudo-código do algoritmo de edição de Wilson

Apesar da sua simplicidade, o custo computacional deste método de edição é de $O(n^2)$, o que pode ser, na prática, um impedimento para conjuntos de treino muito grandes. Existem muitos outros métodos que são, no fundo, derivações ou melhoramentos do algoritmo original de *Wilson*. Basicamente resumem-se:

- Edição repetitiva [Tomek, 1976a] – aplica sucessivamente o algoritmo de Wilson sobre o conjunto de treino obtido.
- Edição com re-etiqueta [Koplowitz & Brown 1981] – consiste em fazer uma nova avaliação dos grupos homogéneos de objectos, isto é, se um objecto estiver cercado por outros que pertencem a uma classe distinta da sua, então é-lhe atribuída a classe do grupo.
- Edição com rejeição [Tomek, 1976b] – propõe que um conjunto de treino de m classes se converta num de $m+1$ classes. O objectivo desta nova classe é guardar o conjunto de objectos rejeitados por alguma técnica de

edição e intervir, depois, no processo de classificação de uma nova instância.

- *Multiedit* [Devijver, 1980] – é uma versão repetitiva da *edição com rejeição*. Na realidade, este esquema de edição surge como uma alternativa para um problema que foi detectado na análise do comportamento assintótico do algoritmo de Wilson cujos detalhes podem ser consultados em [Sánchez, 2005]. O comportamento assintótico de *Multiedit* é ótimo, demonstrado, em 1982, por Devijver [Devijver, 1982]. Apesar de algumas objecções ao algoritmo de Wilson, na prática, este apresenta um melhor desempenho que o *Multiedit* para conjuntos de treino relativamente pequenos.

Mais recentemente, em 2000, foi proposto por Hattori e Takahashi uma modificação do esquema de edição cuja ideia consiste em manter apenas os objectos que são *fortemente classificados*, querendo dizer com isto que todos os k -vizinhos desse objecto x devem estar na mesma classe de x [Hattori & Takahashi, 2000].

2.4.3.2. Técnicas de Condensação

As técnicas de condensação têm por objectivo eliminar todos os objectos do conjunto de treino que não influenciam o desempenho do classificador. Isto significa que a classificação de um novo objecto z , a partir do subconjunto condensado, deverá ser igual ou muito próxima daquela que se obteria utilizando o conjunto de treino inicial.

Segundo esta perspectiva, Hart, em 1968, foi o primeiro a apresentar um método cujo objectivo é reduzir o conjunto de treino para a regra 1-NN chamado vizinho mais próximo condensado (*Condensed Nearest Neighbour - CNN*), [Hart, 1968]. Nesta aproximação inicial utiliza-se o conceito de *consistência* em relação ao conjunto de treino.

Seja $\mathcal{D} = \{ \langle \mathbf{x}^{(1)}, c^{(1)} \rangle, \langle \mathbf{x}^{(2)}, c^{(2)} \rangle, \dots, \langle \mathbf{x}^{(n)}, c^{(n)} \rangle \}$ um conjunto de treino com n objectos previamente classificados.

Definição 2.2. (Conjunto Consistente) Um subconjunto \mathcal{T} diz-se consistente em relação ao conjunto de treino \mathcal{D} se \mathcal{T} classifica correctamente todos os elementos de \mathcal{D} .

Outros dois conceitos, também, relevantes são os de *fronteira de decisão consistente* e *conjunto consistente mínimo*, definidos a seguir:

Definição 2.3. (Fronteira de Decisão Consistente) Um subconjunto \mathcal{T} que define a mesma fronteira de decisão que o conjunto inicial \mathcal{D} , diz-se que tem fronteira de decisão consistente.

Definição 2.4. (Conjunto Consistente Mínimo) Um subconjunto \mathcal{T} é um conjunto consistente mínimo se for o menor conjunto consistente.

Considerando as definições anteriores, um conjunto condensado deverá ser um conjunto de protótipos reduzido e consistente.

A Figura 2.7 exemplifica a aplicação da técnica de condensação a um conjunto de treino: (b) conjunto condensado e (c) conjunto consistente mínimo (conjunto de protótipos reduzido e consistente).

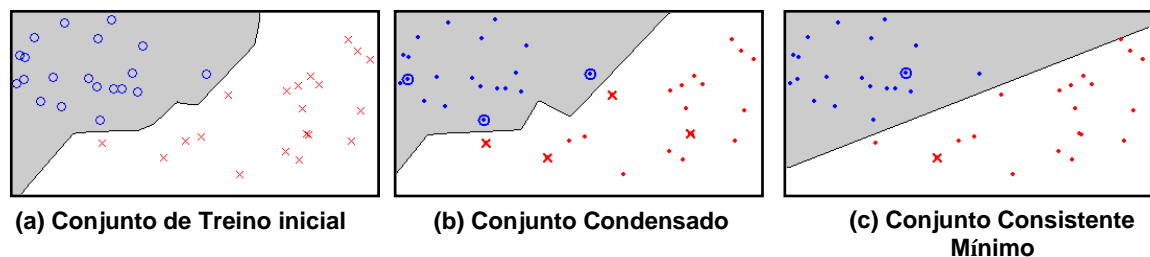


Figura 2.7. Conjunto Condensado e Conjunto Consistente Mínimo

O método de condensação de Hart elimina do conjunto de treino todos os objectos que não são necessários para a correcta classificação dos restantes objectos, mediante a regra do vizinho mais próximo (ver Figura 2.8). Este método baseia-se na ideia de que se um objecto é incorrectamente classificado, isso deve-se ao facto de se encontrar próximo da fronteira de decisão, logo deverá permanecer no subconjunto condensado (ver Figura 2.8, Passo 3.2.1).

C1: Algoritmo de Condensação de Hart

Entrada: Conjunto de treino \mathcal{D}

Saída: Subconjunto condensado \mathcal{D}_C

Passo 1: Iniciar: $\mathcal{D}_C \leftarrow \emptyset$;

Passo 2: Transferir aleatoriamente algumas instâncias do conjunto de treino \mathcal{D} para \mathcal{D}_C ;

Passo 3: Para cada instância \mathbf{x} de \mathcal{D} fazer:

3.1. Se a classe de \mathbf{x} obtida pela regra 1-NN sobre o conjunto \mathcal{D}_C é correcta:

3.1.1. não adicionar \mathbf{x} a \mathcal{D}_C ;

3.2. Se não

3.2.1. adicionar \mathbf{x} a \mathcal{D}_C ;

Figura 2.8. Pseudo-código do algoritmo de condensação de Hart

A inicialização aleatória do conjunto \mathcal{D}_C pode conduzir a diferentes subconjuntos condensados, naturalmente, todos eles consistentes, mas de tamanho diferente. Quando comparados com o conjunto de treino inicial são consideravelmente mais pequenos. Desta forma, o aspecto negativo deste método é não garantir a determinação do subconjunto mínimo consistente, pelo que motivou o aparecimento de outras alternativas ao método de condensação de Hart:

- Condensação reduzida [Gates 1972] - aplica repetidamente a condensação de Hart ao subconjunto que, entretanto, se obtém.

- Vizinho mais próximo Selectivo (*Selective nearest neighbour* - SNN) [Rittet et al., 1975] - constitui a primeira tentativa para se conseguir, por uma heurística própria, o subconjunto consistente mínimo.
- Condensação de Tomek [Tomek, 1976c] – é utilizada uma estratégia de selecção adequada do conjunto \mathcal{D}_C , em vez do método aleatório.
- Condensação de Voronoi [Toussaint & Poulsen, 1979] – é o único método capaz de satisfazer, simultaneamente, o critério de consistência para o subconjunto e fronteira de decisão. A ideia fundamental desta proposta é obter previamente o Diagrama de Voronoi sobre o conjunto de treino. Na prática, isto implica um custo computacional muito elevado, o que originou mais tarde uma nova proposta, condensação por grafo de proximidade.
- Condensação por grafo de proximidade [Toussaint et al., 1985] – é uma nova proposta que visa reduzir o custo computacional associado ao *diagrama de Voronoi* para eliminar os pontos que se encontram afastados da região onde se encontra a fronteira de decisão.

Estas duas últimas propostas serão objecto de estudo mais detalhado nos capítulos seguintes, uma vez que o propósito desta tese é estudar a aplicação de estruturas geométricas na condensação do conjunto de treino para a implementação da regra do vizinho mais próximo.

2.5. Algoritmos Eficientes para a procura do Vizinho mais Próximo

Na continuidade do que foi apresentado na secção 2.4, podemos constatar que, da necessidade de reduzir o custo do cálculo das distâncias para o classificador k-NN, surgiram ao longo do tempo uma grande variedade de técnicas de edição e condensação. Vimos, pois, que tais procedimentos conduzem à redução do

conjunto de treino, quer pela eliminação de instâncias erradamente classificadas (*edição*) quer pela eliminação de instâncias que não contribuem para o bom desempenho do classificador (*condensação*).

Logo que se tenha um conjunto de treino consistente (se possível mínimo), por aplicação de uma ou mais técnicas apresentadas na secção 2.4.3, resta aplicar um algoritmo que faça a procura do vizinho mais próximo de modo eficiente. Sob uma perspectiva algorítmica, podemos dividir o conjunto de algoritmos em duas famílias: os que se baseiam em técnicas da *Optimização Discreta*, mais particularmente, nos algoritmos de *ramificação e poda (branch and bound)* e os que se baseiam em técnicas da *Geometria Computacional*, mais propriamente, na implementação do Diagrama de Voronoi.

No que diz respeito à primeira família, na secção 2.4.2 foi mencionado o algoritmo AESA para a redução do cálculo de distâncias entre o novo objecto a classificar e os protótipos do conjunto de treino. No entanto, recorde-se que o principal inconveniente destes algoritmos é a complexidade espacial, de ordem $O(n^2)$, necessária para armazenar a matriz de inter-distâncias dos objectos do conjunto de treino. Acrescenta-se que, em 1990, Ramasubramanian, propôs o algoritmo S-Grid (*Spherical – Grid Intersection Detection Search*) cuja complexidade espacial é menor, de ordem $O(n)$ [Ramasubramanian, 1990].

Os progressos alcançados na década de 80 e 90, na área da informática e computação, contribuíram decisivamente para o tratamento da regra k-NN à luz de novas técnicas algorítmicas do domínio da Geometria Computacional. Assim, a segunda família de algoritmos propostos constitui uma alternativa baseada em princípios geométricos inerentes à Geometria Computacional. De facto, as estruturas de proximidade geométricas, como o Diagrama de Voronoi e seus subgrafos, constituem elegantes aproximações para a resolução dos sucessivos problemas que advêm do classificador vizinho mais próximo, anteriormente analisados:

- escolha do número de vizinhos;

- escolha dos atributos mais relevantes;
- redução da complexidade espacial associada ao tamanho do conjunto de treino;
- redução da complexidade temporal associada ao cálculo do k-vizinhos mais próximos.

O estudo das estruturas geométricas de proximidade segue no capítulo subsequente, para, em última instância, abordarmos a condensação através das mesmas.

Capítulo 3

Estruturas de Proximidade e Vizinhança

3.1. Introdução

Uma das principais e mais eficazes estruturas que permitem a resolução de problemas de proximidade é o Diagrama de Voronoi [Aurenhammer & Klein, 2000]. O seu estudo, ao contrário de outros problemas abordados pela Geometria Computacional, teve origem no século XVII, quando *R. Descartes* tratou da fragmentação do sistema solar no seu livro *Pricipia Philosophiae* (1644). É importante também destacar que os conceitos relacionados com esta estrutura surgiram de forma independente dos círculos matemáticos e, por isso, o Diagrama de Voronoi tem diversas designações na literatura: Tesselação de Dirichlet, zonas de Wigner-Seitz, polígonos de Thiessen, zonas próximas, entre outros.

Para um dado conjunto de n pontos no plano, podemos definir informalmente um Diagrama de Voronoi (DV) como sendo a subdivisão do plano em n regiões onde cada uma dessas regiões é formada pelo lugar geométrico dos pontos mais próximos a cada um dos pontos do conjunto dado. (Ver Figura 3.1)⁴.

Foi também provado que o DV está estreitamente ligado a outros problemas de

⁴ A Figura 3.1. foi construída num applet do site <http://www.cs.cornell.edu/Info/People/chew/Delaunay.html>

proximidade, como, por exemplo, o problema da determinação: do vizinho mais próximo, do par mais próximo, de árvores de longitude minimal e do máximo círculo vazio. De uma forma geral, esta classe de problemas caracteriza-se pela necessidade de calcular a distância euclidiana entre pontos, rectas, polígonos e círculos [Mendoza, 2006].

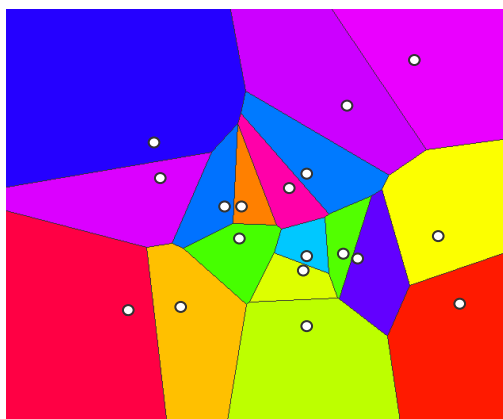


Figura 3.1. Diagrama de Voronoi

O Diagrama de Voronoi e o seu dual, Triangulação de Delaunay (TD), têm sido, ao longo dos últimos três séculos, importantes ferramentas na resolução de muitos outros problemas geométricos e podem ser redescobertos nos mais diversos campos da ciência: Biologia, Meteorologia, Metalurgia, Robótica, Cristalografia, Engenharia, Economia, etc., enfim, existe uma extensa bibliografia onde podemos encontrar a aplicação destas estruturas. Uma referência importante na compilação de muitos trabalhos sobre a aplicação de DV e seu dual é o livro escrito por Okabe, Boots e Sugihara [Okabe et al.,1992].

Na sequência do que foi referido, no Capítulo 2, as estruturas geométricas de proximidade como o Diagrama de Voronoi e seus subgrafos relativos evidenciam uma boa aproximação do problema do vizinho mais próximo [Bhattacharya et al.,2005]. Assim, ao longo deste capítulo, pretendemos estudar as seguintes estruturas de proximidade e vizinhança: diagrama de Voronoi e o dual – Triangulação de Delaunay, grafo de Gabriel e grafo de Vizinhança Relativa.

Uma vez que estas estruturas derivam do DV, serão primeiramente apresentadas, na secção 3.2, algumas das principais propriedades do Diagrama de Voronoi bem como as técnicas mais utilizadas para a sua construção. Em seguida, na secção 3.3, serão abordados os grafos de Gabriel e Vizinhança Relativa. Na secção 3.4, serão feitas algumas considerações finais sobre a relação entre as estruturas de proximidade e vizinhança estudadas neste capítulo.

3.2. Diagrama de Voronoi

3.2.1. Preliminares, Definições e Propriedades Elementares

Antes de introduzir as definições básicas, devemos destacar que os DV podem ser definidos para espaços de qualquer dimensão, contudo, com o objectivo de facilitar a exposição, o estudo que se segue refere-se apenas ao caso de \mathbb{R}^2 e a métrica que será utilizada é a distância Euclidiana entre dois pontos $p = (p_x, p_y)$ e $q = (q_x, q_y)$, assim definida:

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (3.1)$$

Também, sem perda de generalidade, assumimos que os conjuntos de pontos que analisaremos não apresentam configurações degeneradas, isto é, não existem três pontos colineares e nenhum conjunto de quatro pontos, são co-circulares.

Foi no século XIX que os matemáticos *Peter Dirichlet* e *Georgy Voronoi* introduziram formalmente o conceito de Diagramas de Voronoi [Aurenhammer & Klein, 2000].

Seja $S = \{p_1, p_2, \dots, p_n\}$ um conjunto de n pontos no plano; estes pontos são frequentemente designados na literatura especializada por geradores.

Definição 3.1. (Diagrama de Voronoi) O Diagrama de Voronoi de um conjunto de pontos S , $Vor(S)$, é uma partição⁵ do plano em n regiões poligonais V_1, \dots, V_n , tal que para cada i todos os pontos da região $V(p_i)$ estão mais próximos de p_i do que qualquer outro ponto de $S \setminus \{p_i\}$.

Ou seja,

$$V(i) = V(p_i) = \{x \in \mathbb{R}^2 \mid d(p_i, x) \leq d(p_j, x); \text{ para } i \neq j \text{ e } j = 1, 2, \dots, n\} \quad (3.2)$$

A Figura 3.2 mostra alguns exemplos de Diagramas de Voronoi.

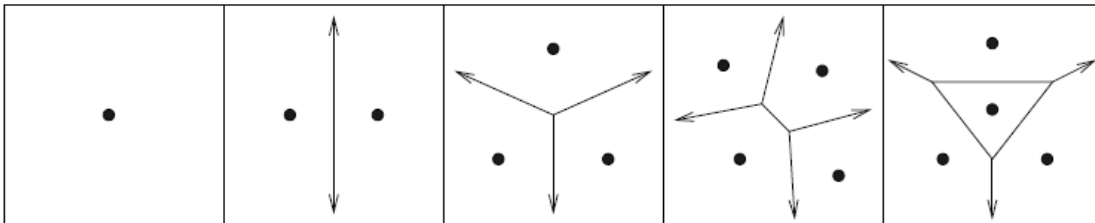


Figura 3.2. Alguns Diagramas de Voronoi com até 4 geradores

Sendo assim, é possível definir a região de Voronoi, $V(p_i)$, a partir da intersecção de semi-planos.

Dados dois pontos p_i e p_j , o conjunto de pontos que estão mais próximos de p_i do que p_j estão no semi-plano que contém p_i , definido pela mediatriz do segmento de recta $p_i p_j$. Seja $H(p_i, p_j)$ o semi-plano que contém o ponto p_i .

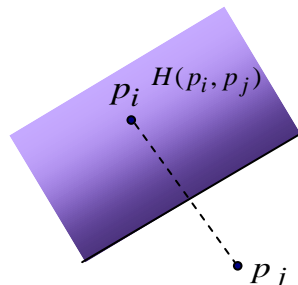


Figura 3.3. Semi-plano que contém p_i

⁵ Partição de uma região é uma decomposição cujas componentes não se sobrepõem, excepto na sua fronteira.

Deste modo, a região $V(p_i)$ pode ser obtida através da intersecção de $n-1$ semi-planos,

$$V(p_i) = \bigcap_{j \neq i} H(p_i, p_j) \quad (3.3)$$

Tendo em conta que a intersecção de conjuntos convexos é sempre um conjunto convexo, podemos afirmar que $V(p_i)$ é uma região convexa (limitada ou não limitada) com no máximo $n-1$ arestas e vértices. Também se diz que $V(p_i)$ é o **polígono de Voronoi** associado ao gerador p_i (ver Figura 3.4).

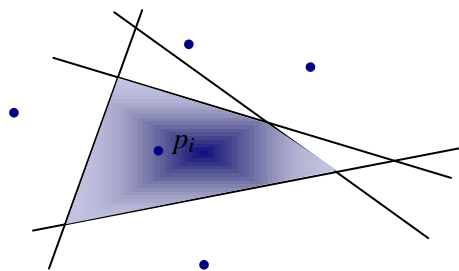


Figura 3.4. Polígono de Voronoi associado ao gerador p_i

A fronteira comum entre duas regiões de Voronoi denomina-se por **aresta de Voronoi**. É importante destacar que, partindo da definição de DV, podemos ter arestas limitadas ou não limitadas como se ilustra na Figura 3.5. Aos pontos de intersecção entre diferentes arestas designam-se **vértices de Voronoi**.

A Figura 3.5 ilustra o Diagrama de Voronoi para um conjunto de 14 geradores.

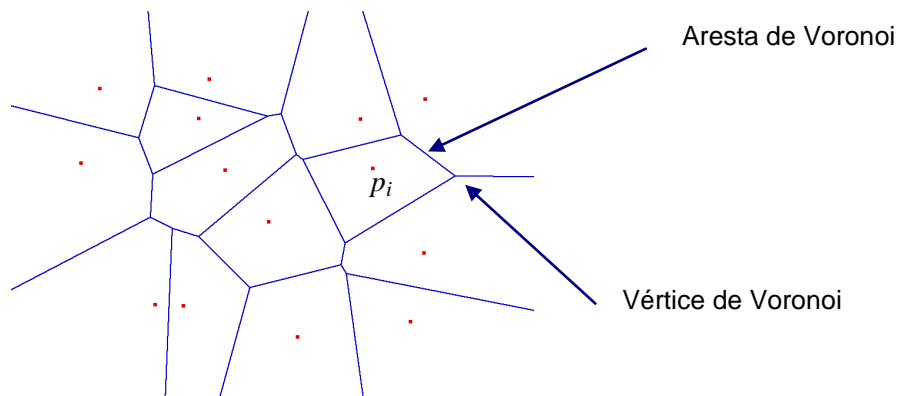


Figura 3.5. Diagrama de Voronoi

Teorema 3.2. (Vértice de Voronoi) Todo o vértice de Voronoi v está na intersecção de três arestas de Voronoi, ou seja um vértice de Voronoi tem exactamente grau⁶ três.

Prova: Dado o conjunto $S = \{p_1, p_2, \dots, p_n\}$ e v um vértice Voronoi, cada vértice de $Vor(S)$ obtém-se da intersecção entre as arestas de Voronoi. Sejam e_0, e_1, \dots, e_{k-1} ($k \geq 3$) as arestas incidentes no vértice v . A aresta e_i é comum aos $V(p_{i-1})$ e $V(p_i)$ para $i=1, \dots, k-1$. Como v pertence à aresta e_i , então, e_i é equidistante a p_{i-1} e p_i com $i=1, \dots, k-1$. Logo, v é equidistante a p_0, p_1, \dots, p_{k-1} ; conseqüentemente, estes pontos são *co-circulares*. Mas, uma vez que não existem quatro pontos co-circulares, vem que $k \leq 3$ (ver Figura 3.6).

Se $k=2$, então e_0 e e_1 são arestas partilhadas por $V(p_0)$ e $V(p_1)$. Portanto as arestas pertencem à mediatriz do segmento p_0p_1 . Logo v não é vértice de Voronoi.

Se $k=1$, então existe apenas uma semi-recta que divide a região $V(p_0)$ e, portanto, esta não é convexa, o que resulta numa contradição.

Logo, $k=3$, e todo o vértice de $Vor(S)$ tem exactamente grau 3.

□

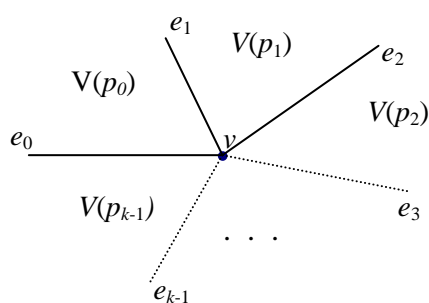


Figura 3.6. Arestas de Voronoi incidentes num Vértice de Voronoi

Do Teorema 3.2 resulta que se v é um vértice de Voronoi definido pela intersecção das regiões $V(p_1)$, $V(p_2)$ e $V(p_3)$ então v é o centro do círculo $C(v)$, definido por p_1 , p_2 e p_3 .

⁶ O grau de um vértice corresponde ao número de arestas que nele incidem.

Teorema 3.3. (Círculo Circunscrito) Seja v um vértice de Voronoi. Então, o círculo $C(v)$, definido por três vértices de Voronoi, não contém nenhum outro ponto de S .

Prova: Sejam p_1 , p_2 e p_3 os três geradores de S que determinam o círculo $C(v)$, como é ilustrado na Figura 3.7.

Se $C(v)$ contém outro gerador p_4 , então, este está mais próximo de v do que p_1 , p_2 e p_3 . Portanto, o polígono de Voronoi $V(p_4)$ deverá incidir sobre v , o que resulta numa contradição, pois, inicialmente assumimos que v é ponto comum a $V(p_1)$, $V(p_2)$ e $V(p_3)$. Logo, não há outros pontos de S dentro de $C(v)$.

□

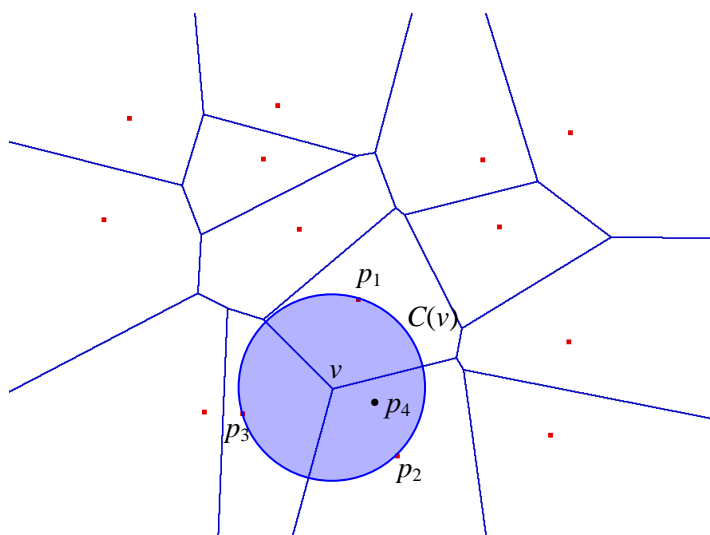


Figura 3.7. Círculo circunscrito

Para cada terno de pontos (p_i, p_j, p_k) de S , se o círculo que contém p_i , p_j e p_k na sua fronteira não contém nenhum outro ponto de S no seu interior, então, o centro do círculo é um vértice de Voronoi. A relação bijectiva entre o centro do círculo $C(v)$ e o vértice de DV conduz a um algoritmo elementar para a determinação dos vértices de Voronoi que pode ser consultado em [Aurenhammer & Klein, 2000].

Teorema 3.4. (Círculo Vazio) Sejam p_i e p_j dois geradores de S . As regiões $V(p_i)$ e $V(p_j)$ possuem uma aresta comum se e só se existe um círculo que contém apenas os dois geradores p_i e p_j .

Prova: Os polígonos $V(p_i)$ e $V(p_j)$ possuem uma aresta comum se e só se existem pontos que são equidistantes de p_i e p_j e, além disso, são mais próximos de p_i e p_j do que qualquer outro gerador de S . Isto implica que $V(p_i)$ e $V(p_j)$ têm uma aresta em comum se e só se existe um círculo que contém p_i e p_j e exclui todos os demais geradores (ver Figura 3.8). Considere-se um círculo $C(x)$ em que x é um ponto interior da aresta partilhada por $V(p_i)$ e $V(p_j)$, e com raio $d(x, p_i) = d(x, p_j)$.

□

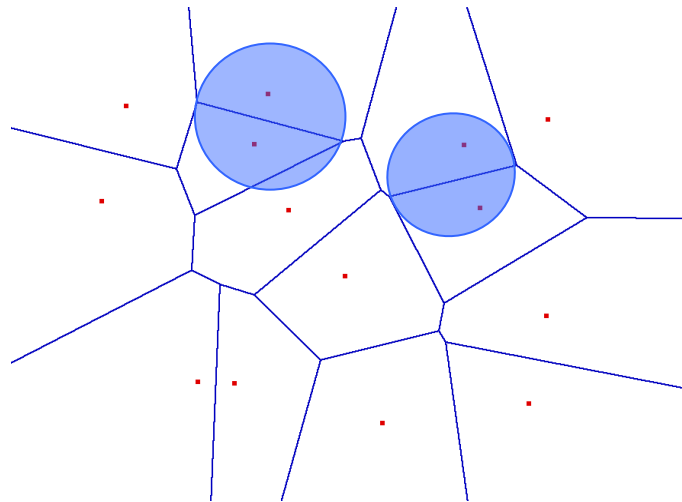


Figura 3.8. Todo o gerador mais próximo de p_i define uma aresta em $V(p_i)$

Teorema 3.5. (Região de Voronoi Limitada) A região de Voronoi $V(p_i)$ é limitada se e só se p_i é ponto interior do invólucro convexo⁷ de S , $conv(S)$.

⁷ Ao menor conjunto convexo que contém S chama-se invólucro convexo.

Prova: Suponhamos que p_i pertence ao interior de $\text{conv}(S)$. Então, existem geradores p_1, p_2 e p_3 em S tais que p_i pertence ao interior do triângulo $\Delta(p_1, p_2, p_3)$. Consideremos os círculos $C_{1,2}, C_{1,3}$ e $C_{2,3}$, determinados por p_i e por cada um dos três pares de geradores $\{p_1, p_2\}, \{p_1, p_3\}$ e $\{p_2, p_3\}$, respectivamente. Cada um destes círculos tem raio finito. Seja $A_{j,k}$ o arco circular (contido em $C_{j,k}$) definido entre p_j e p_k mas que não contém p_i ($1 \leq j < k \leq 3$). Seja C um círculo que contém $C_{1,2}, C_{1,3}$ e $C_{2,3}$ (ver Figura 3.9). Para um ponto x exterior a C , podemos afirmar que x está mais próximo de um dos geradores p_1, p_2 ou p_3 , do que de p_i . De facto, o segmento xp_i (para x exterior a C) intersecta, pelo Teorema de Jordan⁸, um dos lados do triângulo $\Delta(p_1, p_2, p_3)$, seja, por exemplo p_2p_3 . Logo xp_i intersecta $A_{2,3}$ em algum ponto u que está mais próximo de p_1, p_2 ou p_3 do que de p_i . Temos, então, que todo o ponto x exterior a C é mais próximo de um dos geradores p_1, p_2 ou p_3 do que de p_i , pelo que $V(p_i)$ está contida em C e, portanto, é limitada. Reciprocamente, suponhamos que $V(p_i)$ é limitada. Sejam e_1, e_2, \dots, e_k ($k \geq 3$), as arestas de $V(p_i)$ listadas em sentido anti-horário. Cada aresta e_j ($j=1, \dots, k$) está contida, por definição, em alguma mediatriz de um segmento de recta $p_i p'_j$, onde $p'_j \in S$. Assim, concluímos que p_i é um vértice no interior de $\text{conv}(p'_1, p'_2, \dots, p'_k)$.

□

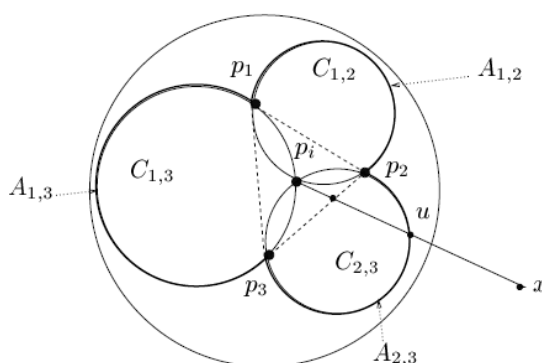


Figura 3.9. Sendo p_i um ponto interior do $\text{conv}(S)$, temos que $V(p_i)$ é limitada

⁸ O Teorema de Jordan afirma que toda a curva plana fechada simples divide o plano em duas regiões (o interior e o exterior da curva).

Uma vez que num diagrama de Voronoi identificamos arestas, vértices e regiões de Voronoi, parece ser natural questionar o seu tamanho. O resultado que se segue traduz a relação entre o número de vértices e arestas num $Vor(S)$.

Teorema 3.6. (Número de Vértices e Arestas em DV) O diagrama de Voronoi de um conjunto S com n geradores tem no máximo $2n-5$ vértices e $3n-6$ arestas.

Antes de apresentarmos a prova deste teorema é importante introduzir algumas definições e resultados da teoria dos grafos [Goodaire & Parmenter, 1997].

Um grafo diz-se planar se for possível desenhá-lo de tal forma que duas arestas não se intersectem excepto nos vértices inicial e final. Por outro lado, um grafo é conexo se, para quaisquer vértices v_i e v_j , existe um caminho (sucessão de vértices e arestas) que os une.

Um grafo planar conexo divide o plano em regiões à custa das suas arestas. Cada uma destas divisões é denominada por face do grafo. Dois pontos do plano estão na mesma face se existir uma curva do plano que os une sem intersectar nenhuma das arestas do grafo.

Prova: A prova do Teorema 3.6 tem como base a fórmula de Euler para grafos planares conexos o qual afirma que os números m_v ; m_e ; m_f , respectivamente, de vértices, arestas e faces do grafo satisfazem a igualdade:

$$m_v - m_e + m_f = 2 \quad (3.4)$$

Dado que $Vor(S)$ tem arestas “semi-infinitas” (ver Figura 3.5), implica que o resultado (3.4) não pode ser aplicado directamente a $Vor(S)$. Para ultrapassar o problema acrescenta-se um vértice v_∞ “no infinito”, ao conjunto dos vértices e considere-se que todas as arestas semi-infinitas de $Vor(S)$ estão ligadas a este vértice, conforme sugere a Figura 3.10.

Desta forma, obtém-se um grafo planar conexo. Da aplicação da fórmula (3.4) resulta:

$$(n_v + 1) - n_e + n = 2, \quad (3.5)$$

onde n_v , n_e e n designam o número de vértices, arestas e regiões de $Vor(S)$, respectivamente.

Cada aresta do grafo aumentado tem exactamente dois vértices então, se adicionarmos o grau de todos os vértices obtemos o dobro do número de arestas. Isto, porque tendo em conta o Teorema 3.2, todo o vértice, incluindo v_∞ , tem grau três, pelo que se tem:

$$2n_e \geq 3(n_v + 1) \quad (3.6)$$

conjuntamente com a equação (3.5) obtêm-se as condições enunciadas no teorema. □

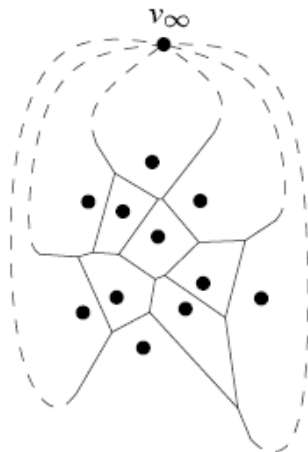


Figura 3.10. Construção do vértice v_∞

Tendo em conta as observações introduzidas na prova do Teorema 3.6, podemos considerar o $Vor(S)$ como sendo um grafo planar conexo. Consequentemente, será natural questionar qual o seu correspondente grafo dual. Por definição, o grafo dual de $Vor(S)$ é o grafo cujos vértices são os pontos de S tais que dois vértices são ligados por uma aresta se as regiões correspondentes partilham uma aresta de Voronoi.

Uma vez que os vértices de DV são de grau três, é fácil apurar que o grafo dual de DV é uma triangulação⁹, designada por *Triangulação de Delaunay*.

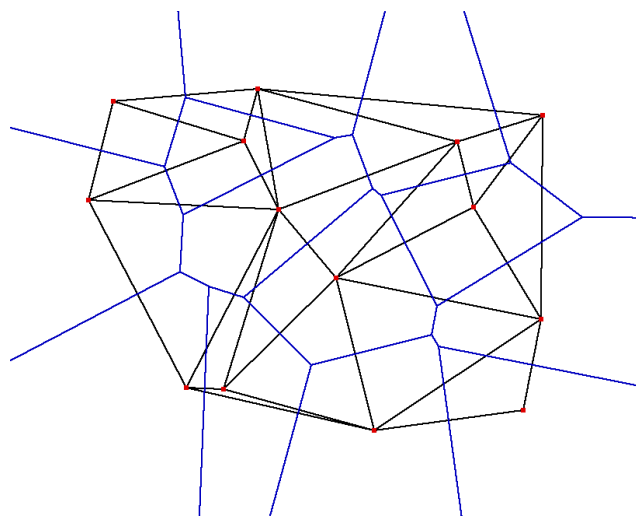


Figura 3.11. Triangulação de Delaunay

A Triangulação de Delaunay admite, à semelhança de DV, propriedades que são consequência da estrutura do seu dual. De seguida, listamos duas dessas propriedades.

Teorema 3.7. (Círculo Circunscrito) Três geradores p_1 , p_2 e p_3 de um conjunto S definem uma face da Triangulação de Delaunay de S se e só se existe um círculo que contém p_1 , p_2 e p_3 e não contém outro gerador de S .

Prova: Segue do Teorema 3.3 (ver Figura 3.12).

Teorema 3.8. (Círculo Vazio) Dois geradores p_1 e p_2 de um conjunto S são adjacentes da Triangulação de Delaunay de S , se e só se existe algum círculo que contém p_1 e p_2 e não contém outro gerador de S .

Prova: Segue do Teorema 3.4 (ver Figura 3.13).

⁹ Triangulação (de um polígono) é uma partição (do polígono) em triângulos.

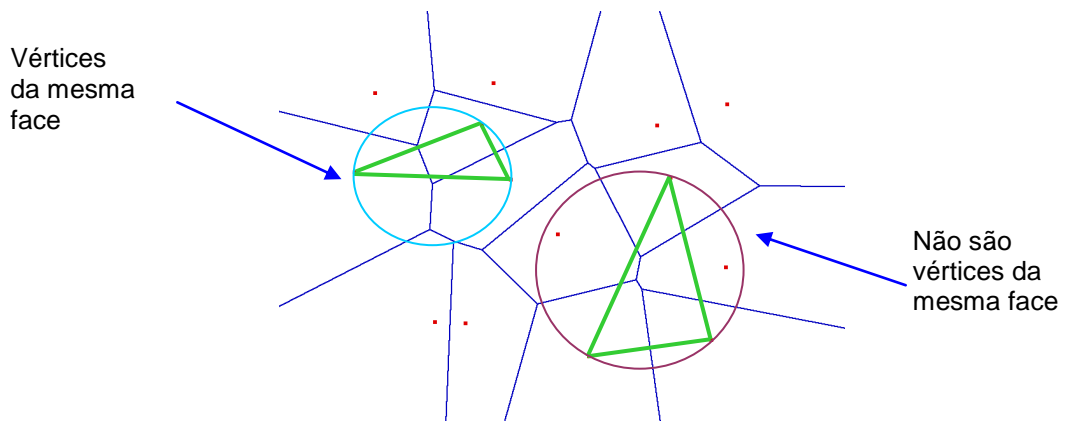


Figura 3.12. Vértices de uma face da Triangulação de Delaunay

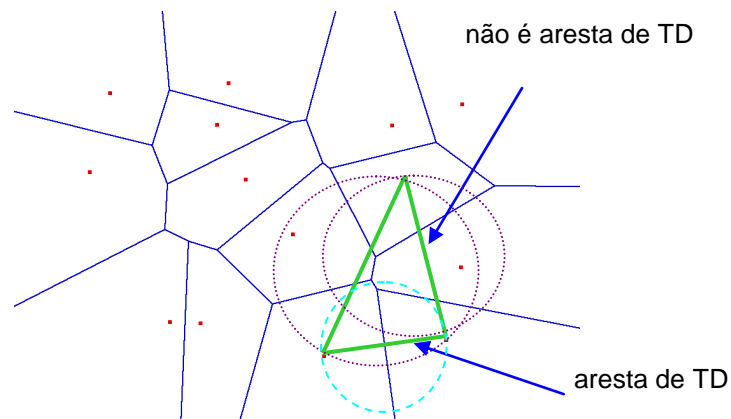


Figura 3.13. Arestas da Triangulação de Delaunay

O estudo das principais propriedades do DV evidenciam que esta estrutura contém toda a informação sobre proximidade, isto é, um DV para um conjunto S estabelece o conjunto de fronteiras de decisão (arestas de Voronoi) associadas a cada gerador, que permitem avaliar o conjunto de pontos mais próximos de cada gerador, ou seja, identificar cada região $V(p_i)$.

Esta é, pois, a “chave” da abordagem do problema do vizinho mais próximo por grafos de proximidade.

3.2.2. Construção do Diagrama de Voronoi

Nesta secção, faz-se o estudo das principais técnicas para a construção do Diagrama de Voronoi. Para facilitar a exposição, iremos apresentar a construção do DV em \mathbb{R}^2 .

Os algoritmos que serão analisados, a continuação, podem ser agrupados da seguinte forma: força bruta, incremental, divisão e conquista, e linha de varrimento.

A. Construção do DV por Intersecção de Semi-planos (Força-bruta)

Este método constitui a abordagem mais simples da construção de $Vor(S)$. Utilizando esta técnica, a região de Voronoi do gerador p_i , $V(p_i)$, é obtida a partir da intersecção dos $n-1$ semi-planos $H(p_i, p_j)$, $i \neq j$, $1 \leq j \leq n$. A seguir, será apresentado o pseudo-código que resume a ideia fundamental deste método.

DV1: DV por Força-bruta

Entrada: $S = \{p_1, p_2, \dots, p_n\}$

Saída: $Vor(S)$

Passo 1: Para cada i , com $i = 1, \dots, n$

Gerar $n-1$ semi-planos $H(p_i, p_j)$, $i \neq j$, $1 \leq j \leq n$

Construir $V(p_i) = \bigcap_{j \neq i} H(p_i, p_j)$

Passo 2: Devolver $\{V(p_1), \dots, V(p_n)\}$

Figura 3.14. Pseudo-código do algoritmo A (força bruta)

A construção de um semiplano $H(p_i, p_j)$ entre p_i e p_j requer um gasto de tempo constante. Consequentemente, para cada p_i , o tempo necessário para construir os $n-1$ semi-planos é $O(n)$. Como a intersecção entre os semi-planos é

construtiva, isto é, sempre que é gerado um novo semi-plano opera-se a intersecção deste com o resultado dos anteriores, então a obtenção de $V(p_i)$ exige $O(n^2)$. Deste modo, aplicando sucessivamente o processo a todos os pontos de S , podemos verificar que a construção de $Vor(S)$ pelo algoritmo de intersecção de semi-planos é $O(n^3)$.

B. Construção do DV pelo Método Incremental

Como o próprio nome sugere, esta técnica baseia-se na construção sucessiva do Diagrama de Voronoi à medida que se adicionam novos pontos. A ideia central deste método é actualizar o diagrama de Voronoi construído para i pontos quando é adicionado um novo ponto p_{i+1} . Esta actualização é feita de acordo com o pseudo-código apresentado na Figura 3.15.

DV2: DV pelo método incremental

Entrada: $Vor(p_1, p_2, \dots, p_i), p_{i+1}$

Saída: $Vor(p_1, p_2, \dots, p_i, p_{i+1})$

Passo 1: Localizar o ponto p_{i+1} em $Vor(p_1, p_2, \dots, p_i)$

Seja $V(p_k)$ a região de $Vor(p_1, p_2, \dots, p_i)$ que contém p_{i+1}

Passo 2: Determinar a mediatriz entre p_{i+1} e p_k

Determinar sucessivamente as mediatrizes entre p_{i+1} e os restantes geradores cuja fronteira é intersectada pela mediatriz

Passo 3: Eliminar as porções das arestas e vértices contidos na região $V(p_{i+1})$

Figura 3.15. Pseudo-código do passo $i+1$ do algoritmo B

Não é difícil verificar que a complexidade deste algoritmo é $O(n^2)$.

A Figura 3.16 apresenta o modo como o método incremental faz a construção do DV quando é adicionado um ponto p_{i+1} .

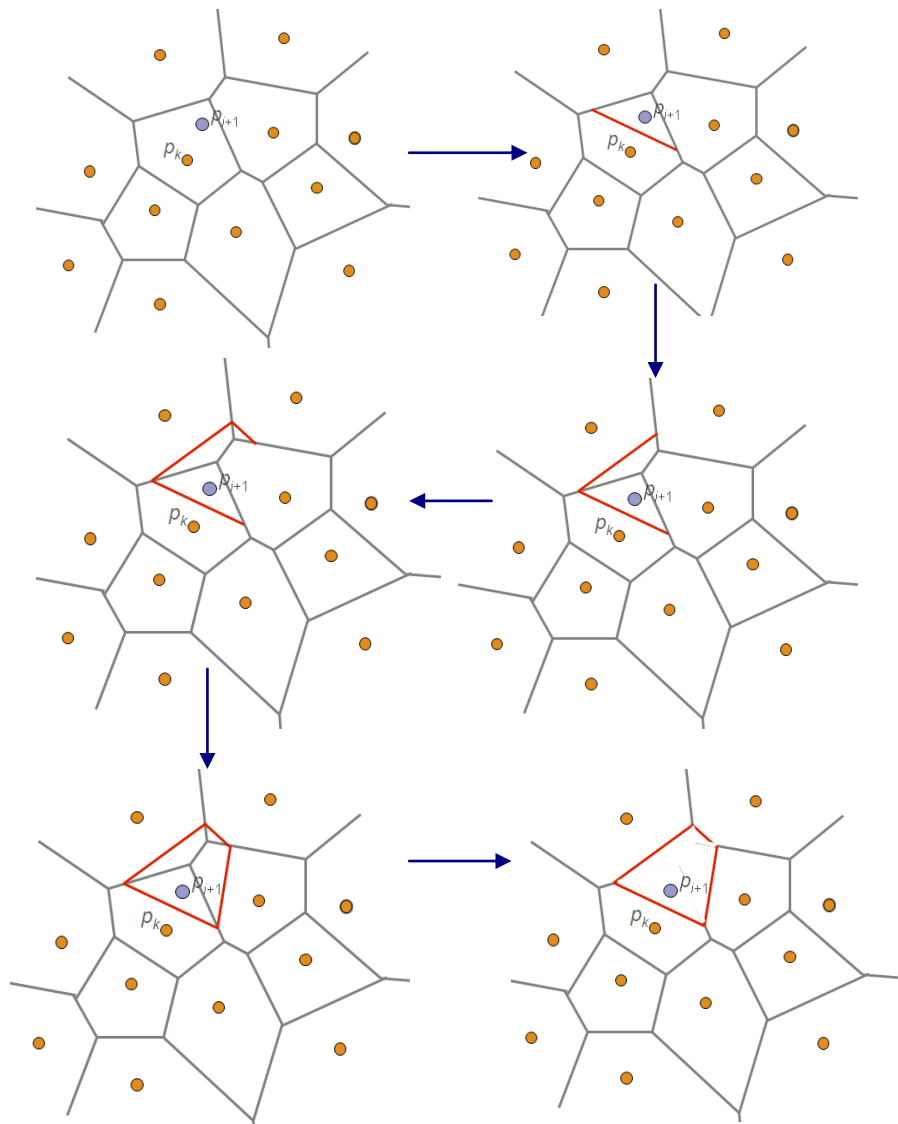


Figura 3.16. Método Incremental para a construção do DV_{i+1}

C. Construção do DV pelo Método de Divisão e Conquista

O método do tipo divisão e conquista é muito utilizado na resolução de problemas, em particular, em problemas de Geometria Computacional. O método para a construção de DV que será analisado a seguir foi proposto por Shamos e Hoey em 1975 [Shamos & Hoey, 1975].

A ideia subjacente a qualquer algoritmo que recorra a esta técnica consiste em três fases: dividir, conquistar e combinar.

Na primeira fase, divide-se o conjunto S em dois subconjuntos S_1 e S_2 (mediante uma linha vertical L) com, aproximadamente, o mesmo tamanho.

Na fase de conquistar, constrói-se recursivamente o $Vor(S_1)$ e $Vor(S_2)$, para cada um dos subconjuntos S_1 e S_2 .

Na última fase, procede-se à concatenação de $Vor(S_1)$ e $Vor(S_2)$, a fim de obter $Vor(S)$.

A Figura 3.17 ilustra a execução das fases acima mencionadas.

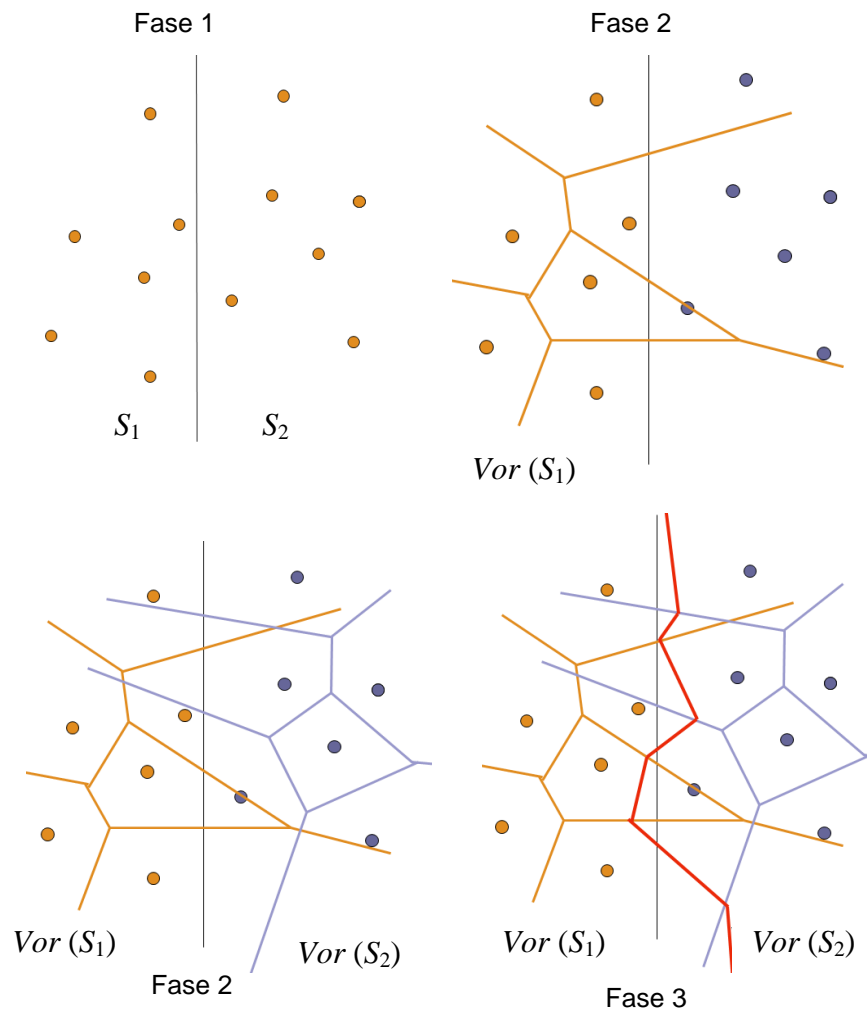


Figura 3.17. Fases do método C: 1 - Dividir, 2 - Conquistar, 3 - Combinar

É evidente que, para poder construir correctamente o DV, devemos responder à seguinte questão: como garantir que a fase de combinação confere a verdadeira estrutura de $Vor(S)$?

Dada a partição $\{S_1, S_2\}$ de S e seja $\sigma(S_1, S_2)$ a cadeia poligonal formada pelo conjunto das arestas de Voronoi que são partilhadas entre os polígonos $V(i)$ e $V(j)$ de $Vor(S)$, para $p_i \in S_1$ e $p_j \in S_2$.

Nota Importante: Os resultados enunciados nos Teorema 3.10 e 3.11 serão apresentados sem prova, pois estas são excessivamente técnicas. Contudo, podem ser encontradas em [Preparata & Shamos, 1985].

Teorema 3.10. Se S_1 e S_2 são linearmente separáveis, então, a cadeia $\sigma(S_1, S_2)$, ou simplesmente σ , é uma cadeia monótona¹⁰.

Teorema 3.11. Se S_1 e S_2 estão linearmente separáveis por uma linha vertical, onde S_1 está à esquerda de S_2 , então, o diagrama de Voronoi, $Vor(S)$, é a união de $Vor(S_1) \cap \pi_R$ e $Vor(S_2) \cap \pi_L$. (π_R e π_L designam, respectivamente, a porção do plano que se encontra à direita e à esquerda da cadeia σ).

DV3: DV por divisão e conquista

Entrada: $S = \{p_1, p_2, \dots, p_n\}$

Saída: $Vor(S)$

Passo 1: Dividir S em dois subconjuntos S_1 e S_2 , através da mediana x-coordenas

Passo 2: Construir $Vor(S_1)$ e $Vor(S_2)$ recursivamente

Passo 3: Construir a cadeia poligonal σ

Passo 4: Eliminar todas as arestas de $Vor(S_2)$ que estão à esquerda de σ e todas as arestas de $Vor(S_1)$ que estão à direita de σ

Figura 3.18. Pseudo-código do algoritmo C

¹⁰ Uma cadeia diz-se monótona em relação a um dos eixos coordenados se, para toda a recta perpendicular ao eixo considerado, intersecta a cadeia em no máximo um ponto.

Admitindo os resultados anteriores e considerando que é possível condicionar a construção de σ , de tal forma que S_1 e S_2 são linearmente separáveis, temos então que a fase de combinar $Vor(S_1)$ e $Vor(S_2)$ conduz à estrutura final de $Vor(S)$. Da análise da Figura 3.18 (pseudo-código da construção C) podemos concluir que o passo 3 constitui a chave deste algoritmo.

Para a construção da cadeia σ , Shamos & Honey (1975) e Lee (1978) propuseram uma forma eficiente que a seguir se resume:

1. Determinar as mediatrizes (m_1 e m_2) dos segmentos de recta tangentes (superior t_1 e inferior t_2) aos invólucros convexos de S_1 e S_2 (ver Figura 3.19);
2. Inicializar a cadeia poligonal σ entre $Vor(S_1)$ e $Vor(S_2)$, tomando a parte superior de m_1 (até a intersecção com a tangente superior t_1);
3. Para cada par de geradores pertencentes a $Vor(S_1)$ e $Vor(S_2)$:
 - 3.1. Determinar a mediatriz entre os geradores das duas regiões encontradas;
 - 3.2. Se a mediatriz intersectar uma aresta de $Vor(S_1)$, então, actualizar o gerador de $Vor(S_2)$ (ver Figura 3.20), caso contrário, actualizar o gerador de $Vor(S_1)$;
4. Concluir a construção da cadeia poligonal σ quando encontra m_2 .

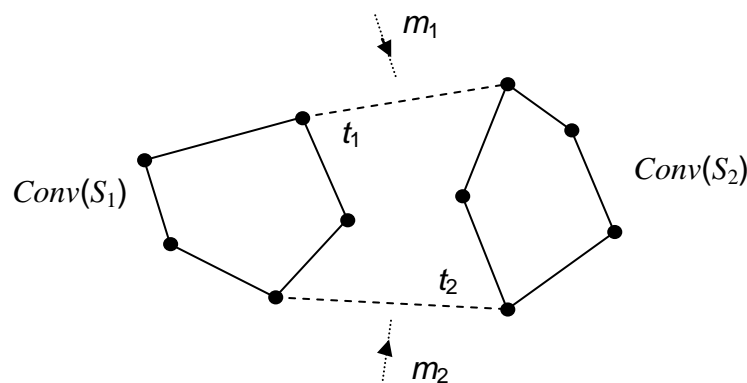


Figura 3.19. Inicialização da cadeia poligonal σ

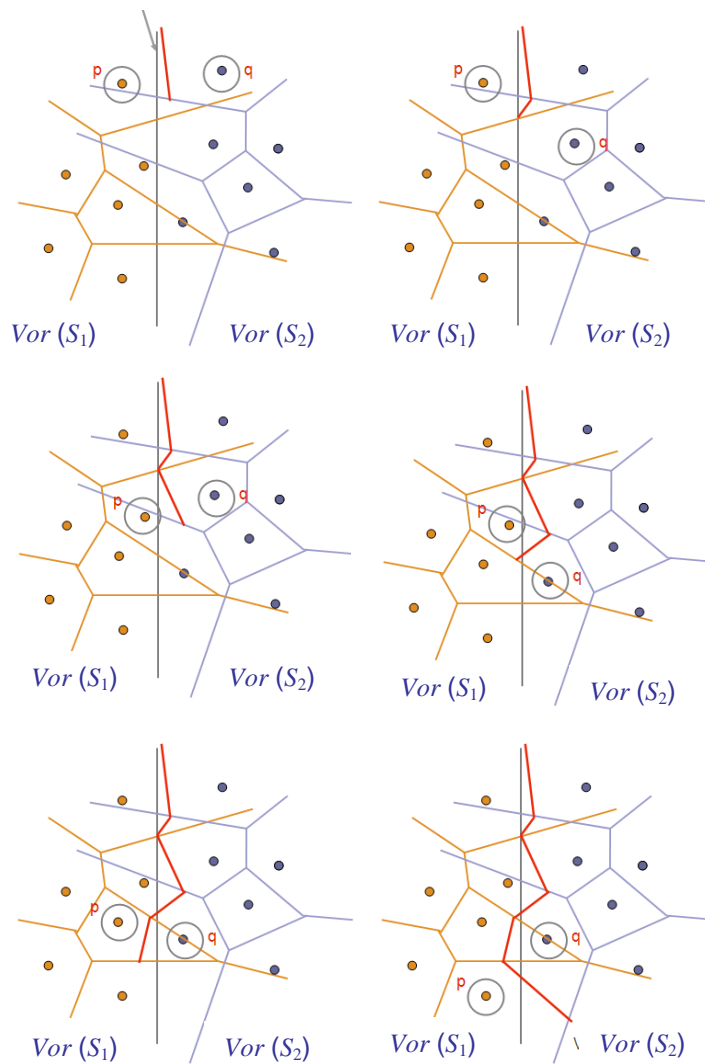


Figura 3.20. Ilustração do processo de construção da cadeia divisória σ

Tendo em atenção o resultado do teorema 3.6 (número de arestas de um $Vor(S)$ não é superior a $3n-6$), a construção da cadeia σ tem um custo $O(n)$. Por outro lado, é conhecido [Cormen et al., 2001] que, se um algoritmo recursivo do tipo Divir-para-Conquistar decompõe uma instância de tamanho n em duas instâncias de tamanho $n/2$ do mesmo problema, e se o processamento necessário à execução das etapas de Dividir e Combinar tem, no total, complexidade $O(n)$, o algoritmo resultante tem complexidade $O(n \log n)$. Sendo assim, podemos enunciar o seguinte resultado:

Teorema 3.12. O Diagrama de Voronoi para um conjunto S de pontos do plano, pode ser construído em $O(n \log n)$.

Antes de analisar o próximo método, é conveniente fazer a seguinte observação: dado um $Vor(S)$ não é difícil determinar o par de pontos mais próximo de S . Basta, para isso, examinar os pontos de S cujas regiões partilham uma aresta de Voronoi. Pelo teorema 3.6 sabemos que será necessário examinar $O(n)$ pares de pontos (geradores), pois o número de arestas de $Vor(S)$ é $O(n)$. É conhecido [Preparata & Shamos, 1985] que o Problema do Par Mais Próximo tem cota inferior $O(n \log n)$ pelo que temos então o seguinte teorema:

Teorema 3.13. Qualquer algoritmo para construir o diagrama de Voronoi de um dado conjunto com n geradores é no mínimo $O(n \log n)$.

D. Construção do DV pelo Método Linha de Varrimento

Em 1987, Steven Fortune propôs um algoritmo muito interessante para construir o Diagrama de Voronoi. A sua proposta foi inovadora, na medida em que a construção do DV por um método de linha de varrimento é diferente dos, já então conhecidos, algoritmos da mesma família [Fortune, 1987].

A concepção do algoritmo tem por base as seguintes observações geométricas:

- o diagrama de Voronoi para n pontos do plano pode ser interpretado como a projecção da intersecção de n cones cujo vértice é cada um dos geradores; a curva resultante da intersecção dos cones é uma hipérbole, contida num plano vertical, cuja projecção no plano é a mediatriz definida entre os geradores p e q .

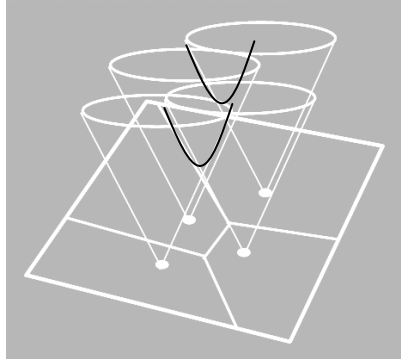


Figura 3.21. Ideia de S. Fortune

Para obter tal resultado, considera:

- um plano de varrimento que se move de cima para baixo com inclinação igual à dos cones e que intersecta:
 - o plano, segundo a linha de varrimento (*Sweep line*)
 - cada um dos cones, segundo uma hipérbole, cuja projecção no plano é uma parábola com foco no gerador e de directriz a linha de varrimento.

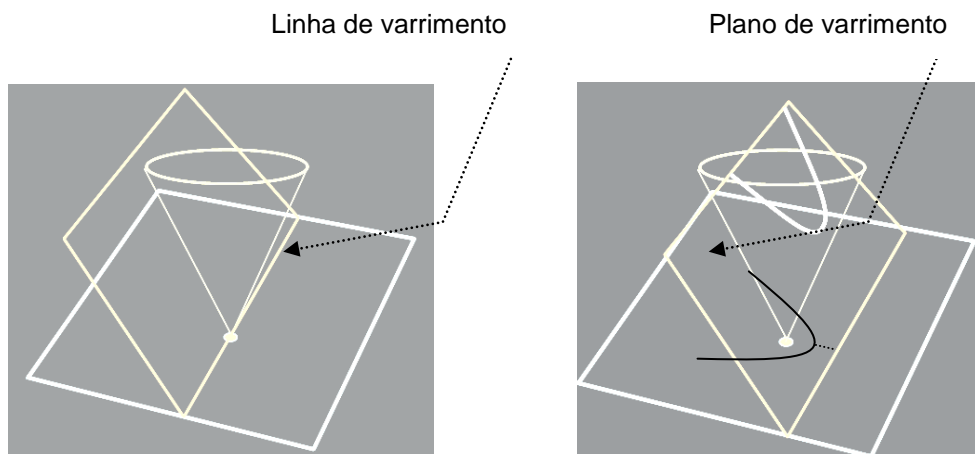


Figura 3.22. Plano de varrimento e linha de varrimento

A maior dificuldade da construção deste algoritmo reside no facto de como é possível prever, antecipadamente, vértices de Voronoi para regiões cujo gerador ainda não foi “lido” pela linha de varrimento (ver Figura 3.23)

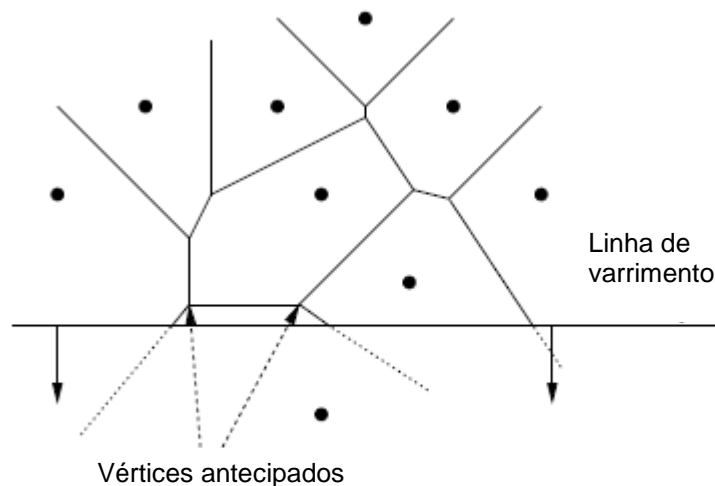


Figura 3.23. Vértices antecipados de uma região de Voronoi cujo gerador está abaixo da linha de varrimento

Para resolver esta questão Steven Fortune propôs o seguinte:

o semi-plano acima da linha de varrimento L é subdividido em duas regiões: o conjunto de pontos q que está mais perto de um gerador p do que da própria linha de varrimento L e o conjunto de pontos equidistantes de um gerador p e da linha de varrimento L . O primeiro conjunto diz respeito à parcela do DV que já não é afectada por nenhum gerador que se encontre abaixo da linha de varrimento L e o segundo conjunto constitui a chamada *frente parabólica* ou *linha de praia* (*beach line*) (ver Figura 3.24) [Berg et al., 2000].

À medida que a linha de varrimento L desliza (de cima para baixo), para cada ponto p do plano é gerada uma parábola, inicialmente com abertura nula, que cresce à medida que a linha de varrimento se afasta do gerador p . É fácil observar que os sucessivos arcos de parábola da linha de praia intersectam-se segundo um conjunto de pontos, que são simultaneamente equidistantes de dois geradores e da linha de varrimento L . Isto significa, que à medida que L desliza, a linha de praia segue imediatamente atrás, alterando-se continuamente. Este movimento é caracterizado por dois tipos de acontecimentos: *site events* e *circle events* (ou *vertex events*).

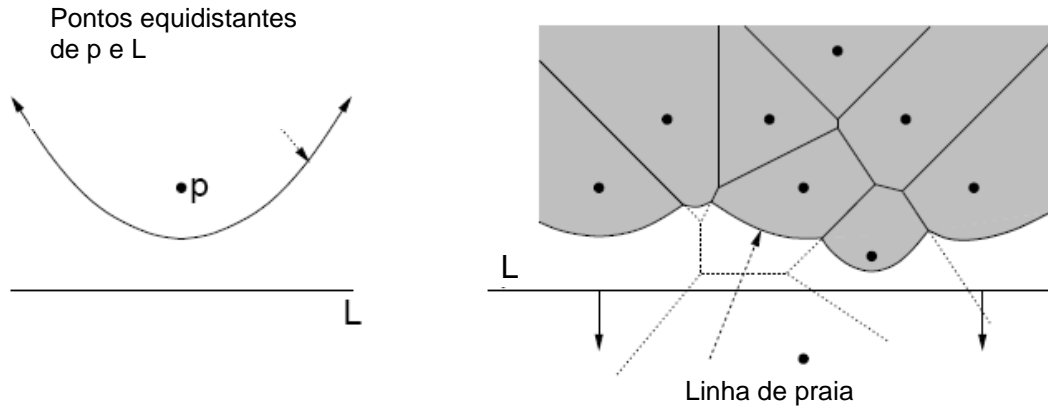


Figura 3.24. Construção do Diagrama de Voronoi acima da linha de praia

Quando a linha de varrimento passa por um gerador, um novo arco de parábola é adicionado à linha de praia – *site events* (ver Figura 3.25).

No início, começa por ser uma parábola degenerada com abertura zero (segmento de recta entre o novo gerador, encontrado por L , e a linha de praia), depois, à medida que a linha de varrimento se afasta, este novo arco de parábola vai dilatando, mudando a própria linha de praia.

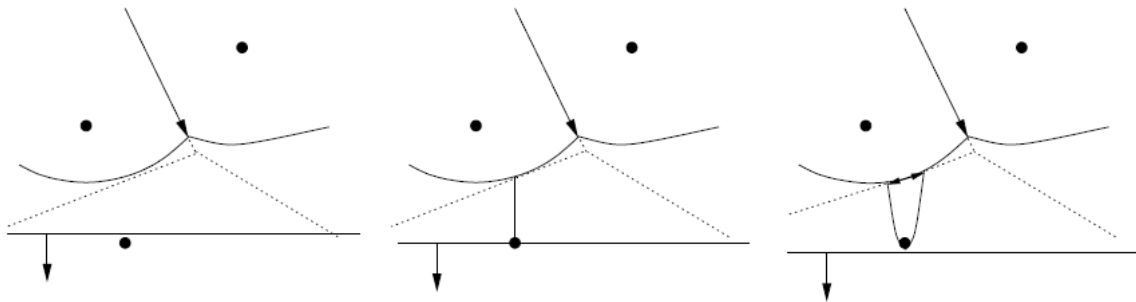


Figura 3.25. Início de *site events*

Por outro lado, se o comprimento do arco de parábola tende para zero, o arco desaparece da linha de praia e dá origem a um vértice de Voronoi – *circle events*.

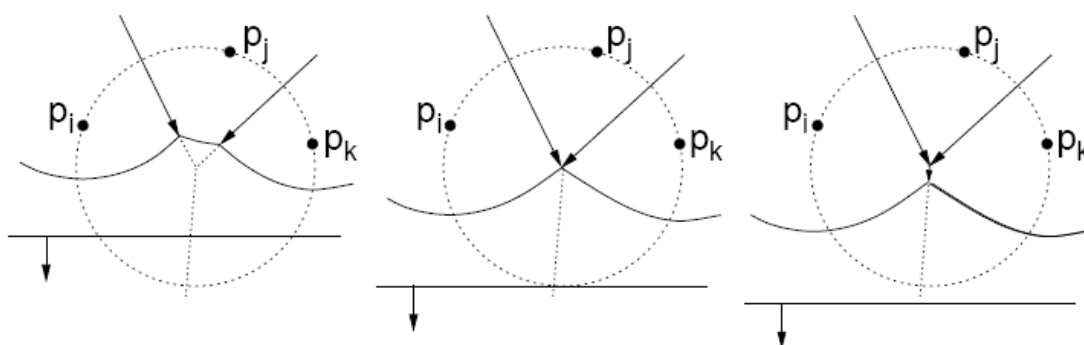


Figura 3.26. Desenvolvimento de um *circle events*

O algoritmo de construção reconhece um *circle events* quando a circunferência que contém na sua fronteira os três geradores p_i , p_j e p_k , está vazia no momento em que é tangente à linha de varrimento. Nesse preciso instante, o centro da circunferência é equidistante dos três geradores e da linha de varrimento. O que implica que o arco de parábola gerado por p_j desapareça da linha de praia e origine um vértice de Voronoi. Em termos de DV, isto significa que as mediatrizes de $p_i p_j$ e de $p_j p_k$ acabam de se intersectar num ponto, que é o vértice de Voronoi (ver Figura 3.26).

Para implementar o algoritmo de *Fortune* são necessárias três estruturas diferentes para armazenar os diferentes tipos de informação:

- parcela DV que vai sendo construída – DCEL (*doubly-connecter edge list*);
- linha de praia – árvore de busca binária balanceada (*balanced binary search tree*);
- *site* e *circle events* – lista de prioridade de acontecimentos (*priority queue list*).

A exposição de cada uma destas estruturas pode ser consultada mais detalhadamente em [Berg et al., 2000].

Finalmente em [Fortune, 1987] prova-se, pela utilização interdinâmica das estruturas anteriormente referidas, que:

Teorema 3.14. A complexidade temporal do algoritmo de *Fortune* é $O(n \log n)$ e utiliza $O(n)$ espaço.

Tendo em consideração o Teorema 3.13, podemos asseverar que o algoritmo de *Fortune* é ótimo.

A título de comentário final desta secção, consideramos importante indicar que os algoritmos de construção do Diagrama de Voronoi em espaços d -dimensionais, têm complexidade espacial $O(n^{\lceil d/2 \rceil})$ e complexidade temporal $O(n \log n + n^{\lceil d/2 \rceil})$. Estas cotas podem ser consultadas em [Berg et al., 2000] e em [Aurenhammer & Klein, 2000].

3.3. Grafos de Vizinhaça

Grafo é uma estrutura geométrica que representa a relação entre pontos.

Formalmente, podemos definir um grafo [Hernández, 2003] como um par $G = (V, A)$, onde V é um conjunto finito não vazio (cujos elementos designaremos por vértices) e A uma família finita de pares não ordenados de vértices de V (cujos elementos designaremos por aresta ou arcos).

Grafos de Vizinhaça, também denominados Grafos de Proximidade [Toussaint, 1991], capturam a proximidade entre pontos através da conexão dos vértices por arestas que obedecem a alguma propriedade de vizinhaça.

Seja V um conjunto de n pontos do plano.

Suponhamos que para cada par de pontos $(p, q) \in V \times V$ se encontra associada uma certa relação de vizinhaça, $R_{p,q} \subset \mathbb{R}^2$.

Seja, ainda, \wp uma propriedade definida sobre $Y = \{R_{p,q} : (p, q) \in V \times V\}$.

Um **grafo de vizinhança** $G_{Y,\wp}(V,A)$, definido pela propriedade \wp , é um grafo formado por um conjunto de **vértices** ou **nós**, V , e um conjunto de **arestas** ou **arcos**, A , tal que $(p,q) \in A$ se e só se $R_{p,q}$ tem a propriedade \wp .

Deste modo, diz-se que p e q são vizinhos de grafo e partilham entre si uma aresta.

Da definição anterior, decorre que a propriedade \wp , definida inicialmente, determina diferentes tipos de grafo de vizinhança (ver figura 3.27).

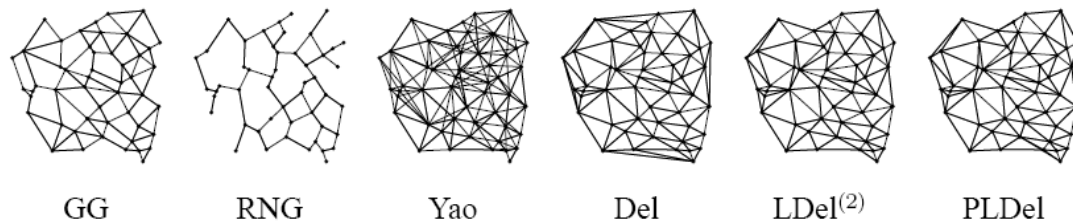


Figura 3.27. Exemplos de diferentes tipos de Grafos de Vizinhança

Do conjunto de grafos apresentados na Figura 3.27, apenas iremos estudar dois: o grafo de Gabriel e o grafo de Vizinhança Relativa

3.3.1. Grafo de Gabriel

O Grafo de Gabriel (GG) também conhecido por *Least Squares Adjacency Graphs*, foi aplicado, inicialmente, na análise da variação geográfica por Gabriel & Sokal em 1969, como é referido em [Marchette, 2004].

Desde então, a sua aplicação tem sido a mais variada, desde problemas de reconhecimento de padrões até à redução do conjunto de treino supracitado em [Toussaint et al., 1985].

Definição 3.15. (Grafo de Gabriel) Grafo de Gabriel (GG) sobre um conjunto V é o grafo $G(V, A_{GG})$, onde A_{GG} é o conjunto de arestas definidas entre dois pontos que satisfazem a seguinte propriedade:

$$(p, q) \in A_{GG} \Leftrightarrow d^2(p, q) \leq d^2(p, r) + d^2(q, r) \quad \forall r \in V, \quad r \neq p, q \quad (3.7)$$

Geometricamente, isto significa que p e q são vizinhos de Gabriel se o círculo de influência¹¹ de diâmetro \overline{pq} com relação a V está vazio, ou seja,

$$(p, q) \in A_{GG} \Leftrightarrow C\left(\frac{p+q}{2}, \frac{d(p,q)}{2}\right) \cap V = \emptyset \quad (3.8)$$

A seguir, a Figura 3.28 ilustra no plano o conceito de vizinhos Gabriel.

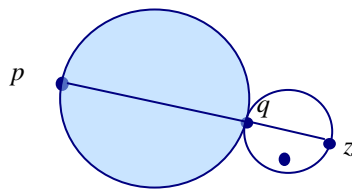


Figura 3.28. (p, q) são vizinhos Gabriel;
 (q, z) não são vizinhos Gabriel

A Figura 3.29 ilustra como o Grafo de Gabriel é um subgrafo da Triangulação de Delaunay, para um dado conjunto de doze pontos.

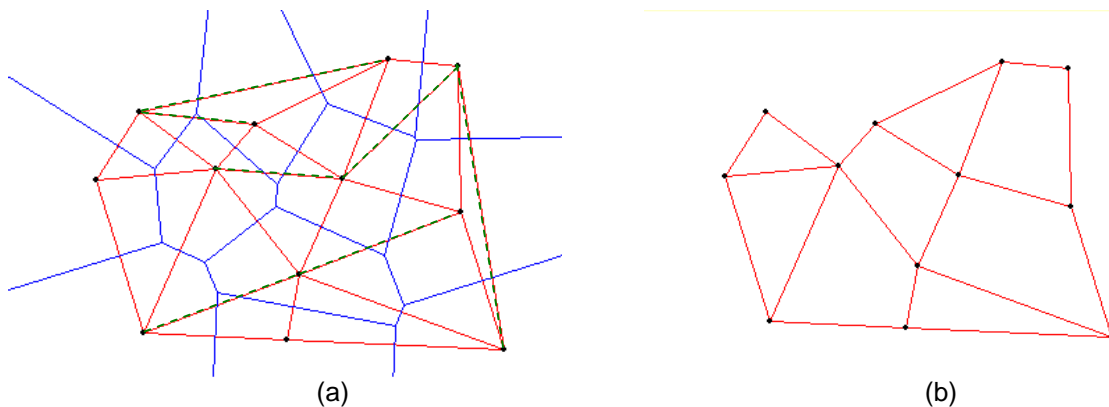


Figura 3.29. (a) Triangulação de Delaunay (eliminação de arestas a tracejado)
 (b) Grafo de Gabriel

¹¹ Na generalização da definição a \mathcal{R}^d chama-se hiper-esfera de influência ou hiper-esfera diametral.

Construção do Grafo de Gabriel

No livro [Matula & Sokal 1980] é apresentado um método para a construção do grafo de Gabriel a partir do Diagrama de Voronoi associado a um dado conjunto S de pontos. Este método tem complexidade $O(n^{d/2})$, pelo que a sua implementação em espaços de representação superiores a duas dimensões torna-se impraticável.

Uma alternativa, para a construção do GG directamente de S , consiste em determinar exhaustivamente quais os pontos do plano que são vizinhos Gabriel, utilizando um método de força bruta. Assim, dispensamos a construção do Diagrama de Voronoi como pré-processamento do algoritmo. Os principais passos dessa construção estão mencionados na Figura 3.30.

O passo 1 do algoritmo requer $O(n^2)$ operações para considerar os $O(n^2)$ possíveis pares de pontos do conjunto de n pontos. Por outro lado, o passo 2 requer $O(dn)$ operações sobre cada par de pontos considerado. Assim, o custo total do algoritmo é $O(dn^3)$.

GG1: Grafo de Gabriel por força bruta

Entrada: Um conjunto S de pontos

Saída: Grafo de Gabriel

Passo 1: Para cada par de pontos $(p_i, p_j) \in S \times S$ testar

Passo 2: Se não existe um ponto $p_k \in S$, $k \neq i \neq j$ tal que

$$d^2(p_i, p_j) > d^2(p_i, p_k) + d^2(p_j, p_k)$$

então os pontos (p_i, p_j) são vizinhos de Gabriel: $A_{GG} \leftarrow (p_i, p_j)$

Figura 3.30. Pseudo-código de algoritmo de força bruta do GG

É fácil verificar que, se a dimensionalidade d do conjunto S for reduzida, este método é mais rápido que o procedimento que utiliza o DV, contudo, se o

número de pontos n for muito grande, este algoritmo não é eficiente.

Face a este problema, Bhattacharya [Bhattacharya,1982] propôs um algoritmo baseado numa técnica aproximada (heurística).

A observação fundamental desta heurística é a seguinte:

A eficiência computacional do algoritmo de força-bruta pode ser melhorado se, de alguma forma, se conseguir diminuir o número de pares de pontos a testar no passo 1. De facto, o número de pares de pontos vizinhos de Gabriel, num conjunto de n pontos, é menor do que aqueles que são considerados no passo 1, em número de $n(n-1)/2$. A partir desta constatação, Bhattacharya apresentou um procedimento (teste de rejeição rápida) cujo objectivo é eliminar pontos que não sejam, à partida, vizinhos de Gabriel do par de pontos a testar.

Considerando a Figura 3.31 como suporte visual do método heurístico e tomando a mesma notação que o Autor, podemos proceder como segue: seja p ($p \in S$) o ponto para o qual pretendemos encontrar os seus vizinhos de Gabriel; considerando, um ponto $q \in S$, traça-se $B(p,q)$, recta perpendicular a pq e que passa por q . Seja $RH(B,p)$ o semi-espaço¹² determinado por $B(p,q)$ que não contém p .

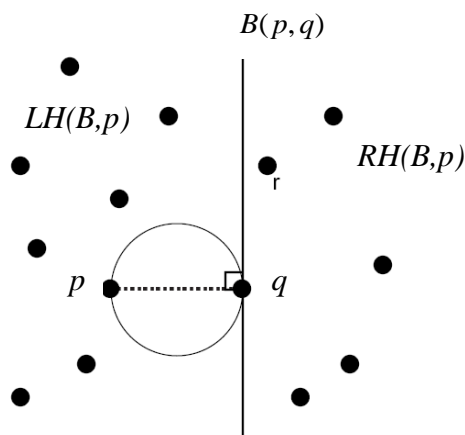


Figura 3.31. Técnica heurística para o cálculo dos vizinhos de Gabriel do ponto p

¹² Naturalmente em \mathfrak{R}^2 denota-se por semi-plano.

A partir desta construção apresenta-se o seguinte resultado:

Lema 3.16. Nenhum ponto de S contido no semiplano $RH(B,p)$ poderá ser um vizinho Gabriel do ponto p .

Prova: Consideremos algum ponto r pertencente a $RH(B,p)$. Seja $D(p,r)$ a circunferência de diâmetro igual à distância entre p e r . Uma vez que r está contido em $RH(B,p)$, o ângulo formado pelos segmentos de recta qp e qr será maior que 90° e, portanto, q está contido em $D(p,r)$. Consequentemente, r não poderá ser vizinho Gabriel do ponto p . \square

Tendo em conta as ideias desta heurística, podemos refinar o algoritmo para a construção do GG, cuja complexidade temporal está próxima de $O(dn^2)$ (ver Figura 3.32).

GG2: Grafo de Gabriel por heurística

Entrada: Um conjunto S de pontos

Saída: Grafo de Gabriel

Para cada ponto p_i de S fazer:

Passo 1: Considerar $N_i = \{p_1, p_2, \dots, p_{i-1}, p_{i+1}, \dots, p_n\}$ o conjunto dos candidatos a vizinhos de Gabriel do ponto p_i

Passo 2: Para cada potencial candidato p_r de N_i fazer:

Para todo o ponto p_k de S , $p_k \neq p_i \neq p_r$:

(i) Se $d^2(p_i, p_r) > d^2(p_i, p_k) + d^2(p_r, p_k)$ então eliminar o ponto p_r do conjunto N_i . Retomar o passo 1.

(ii) Se $p_k \in N_i$, então verificar se p_r está dentro do círculo de influência definido por p_i e p_k . Se sim, eliminar p_k do conjunto N_i

Passo 3: Devolver o conjunto final N_i como sendo o conjunto dos vizinhos de Gabriel de p_i

Figura 3.32. Pseudo-código para o algoritmo de GG utilizando o método heurístico

3.3.2. Grafo de Vizinhaça Relativa

Grafo de Vizinhaça Relativa (*Relative Neighborhood Graph, RNG*) foi, inicialmente, utilizado para o reconhecimento de padrões e só depois aplicado a diversos problemas de Geometria Computacional [Toussaint,1980].

Para o grafo de Vizinhaça Relativa devemos associar uma certa relaçaõ de vizinhaça $R_{p,q}$ que neste caso é a *lente* entre os ponto p e q , à semelhança do que foi feito para o grafo de Gabriel.

Definiçaõ 3.17. (Lente) Seja V um conjunto de pontos de \mathfrak{R}^d . Para qualquer par $(p, q) \in V \times V$, o conjunto $\Lambda_{p,q} = C(p, d(p,q)) \cap C(q, d(p,q))$ chama-se lente.

A Figura 3.33 (a) ilustra em \mathfrak{R}^2 o conceito de lente e, (b) dois pontos vizinhos relativos.

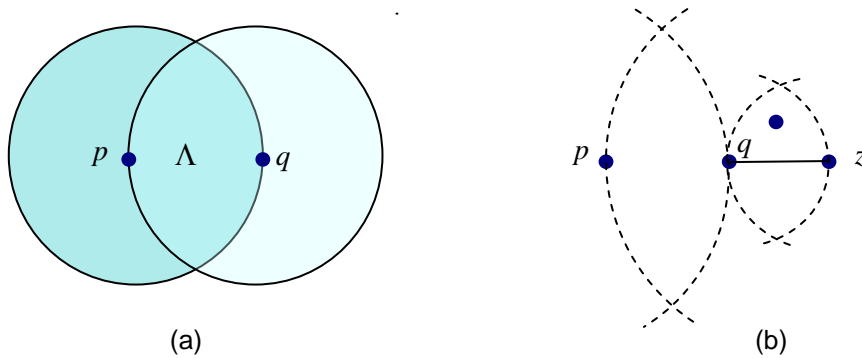


Figura 3.33. (a) lente Λ definida entre p e q

(b) p e q são vizinhos relativos; q e z não são vizinhos relativos

Definiçaõ 3.18. (Grafo de Vizinhaça Relativa) Grafo de Vizinhaça Relativa sobre um conjunto V é grafo $G(V, A_{RNG})$, onde A_{RNG} é o conjunto de arestas definidas entre dois pontos que satisfazem a seguinte propriedade:

$$(p, q) \in A_{RNG} \Leftrightarrow \Lambda_{p,q} \cap V = \emptyset \quad (3.9)$$

Na definição anterior é importante mencionar que a lente definida entre p e q deve ser aberta, caso contrário, p e q estariam nela e a definição teria que ser ajustada.

Construção do Grafo de Vizinhaça Relativa

Como é referido por Marchette, é sempre possível construir RNG a partir da Triangulação de Delaunay, já que não é difícil verificar que o RNG é um subgrafo do grafo induzido pela Triangulação de Delaunay como se mostra na Figura 3.34 [Marchette, 2004].

O algoritmo de força bruta para implementar a construção do grafo de Vizinhaça Relativa é idêntico ao descrito para o grafo de Gabriel, modificando-se apenas os procedimentos relativos à relação de vizinhaça em questão.

Por outro lado, em [Sánchez, 2005] é assinalado que existem diversos algoritmos eficientes para a construção de RNG, contudo, apenas, são válidos para o espaço bidimensional.

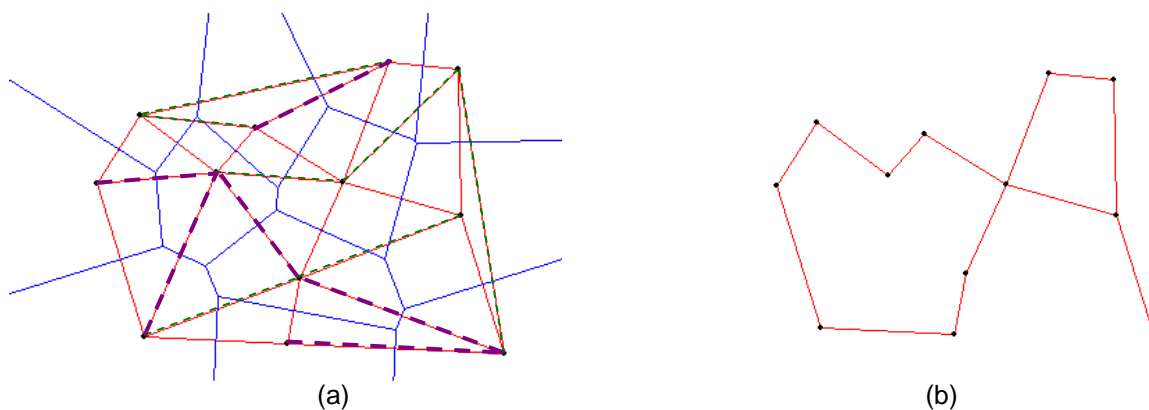


Figura 3.34. (a) Eliminação (tracejado) de arestas na Triangulação de Delaunay
(b) Grafo de Vizinhaça Relativa

3.4. Relações entre os Grafos

Nas secções 3.2 e 3.3 foi apresentado, essencialmente, a estrutura, propriedades e algoritmos de construção de cada um dos grafos que interessa, face ao contexto deste trabalho. Contudo, importa ainda fazer uma breve menção à relação entre eles, uma vez que foi referido, ao longo desta secção, que tanto GG como RNG são subgrafos da Triangulação de Delaunay.

Em 1992, Jaromczyk [Jaromczyk, 1992] sintetizou todo o trabalho que até então se tinha feito em torno dos grafos de vizinhança. Uma das primeiras conclusões refere-se à hierarquia entre os grafos:

$$RNG \subseteq GG \subseteq TD \quad (3.10)$$

Esta relação é importante na medida em que a partir de pequenas alterações no algoritmo de construção de TD se pode obter facilmente qualquer um dos outros grafos. Os algoritmos eficientes para a construção destas estruturas resultam do facto de TD ter no plano tamanho $O(n)$ e poder ser construída em tempo $O(n \log n)$. É preciso observar que estas cotas de complexidade não são válidas em dimensões maiores [Aurenhammer & Klein, 2000].

Para além disso, notamos a evidente redução do número de arestas de cada grafo, naturalmente aliado ao tipo de relação de vizinhança que se define sucessivamente para cada caso. Estas considerações conduzem a outro resultado importante que se relaciona com o tamanho dos grafos.

É perceptível, da análise das Figuras 3.29 e 3.34, que o número de arestas de RNG está limitado pelo número de arestas de GG e, de igual modo, este está limitado pelo número de arestas de TD.

$$|A_{RNG}| \leq |A_{GG}| \leq |A_{TD}| \quad (3.11)$$

A análise desta relação encontra-se formalmente estudada em [Marchette, 2004]. Analogamente, o número de arestas é:

$$3n - 10 \leq 3n - 8 \leq 3n - 6 \quad (3.12)$$

Cabe ainda dizer, nesta secção, que sendo GG e RNG subgrafos da Triangulação de Delaunay, fazem parte de uma família de grafos denominada por β -esqueletos. Os β -esqueletos estão definidos por uma relação de vizinhança baseada em discos vazios. Para o caso de $\beta = 1$, temos um círculo vazio que origina um conjunto de arestas cuja estrutura é o GG (subsecção 3.3.1); para $\beta = 2$ a propriedade de vizinhança será a lente vazia, pelo que conduz ao RNG (subsecção 3.3.2). Na Figura 3.35 mostra-se a hierarquia das relações de vizinhança definidas para os β -esqueletos.

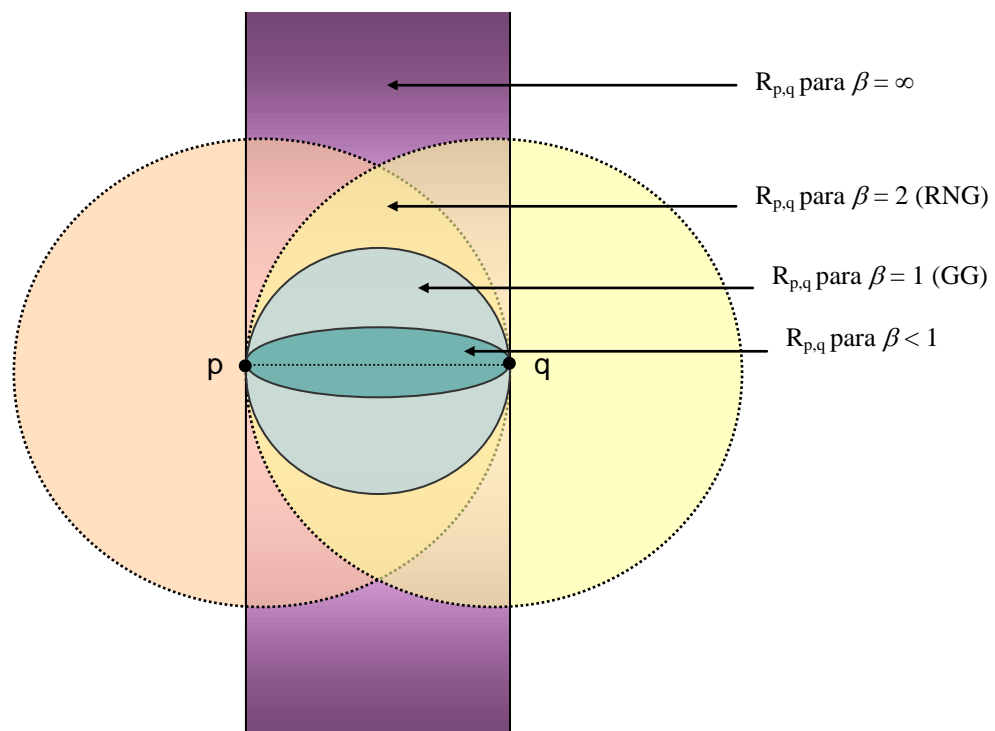


Figura 3.35. Família dos β -esqueletos

A família de β -esqueletos foi introduzida por Kirkpatrick e Radke [Kirkpatrick & Radke, 1985] como sendo uma classe de *grafos de vizinhança vazia*. Recentemente, tem aumentado o seu interesse por lhes ser reconhecida aplicação na área do reconhecimento de padrões, nomeadamente, na condensação do conjunto de treino para a regra de decisão k-NN, assunto que iremos tratar no próximo capítulo.

Aplicação de Estruturas de Proximidade nos Algoritmos de Condensação da Regra k-NN

4.1. Introdução

O objectivo deste capítulo é fazer o enlace entre as estruturas de proximidade e a condensação de um conjunto de treino para a regra de decisão do vizinho mais próximo. Apesar destes dois temas pertencerem a áreas diferentes, revela-se que é viável a aplicação de técnicas estudadas na área da Geometria Computacional à aprendizagem baseada em instâncias.

A simplicidade e o bom desempenho do classificador do vizinho mais próximo concebido por Fix e Hodges (1951), constitui uma das mais atraentes regras de decisão. Contudo, o tamanho do conjunto de treino e o número de distâncias a calcular entre o novo objecto e os restantes elementos do conjunto de treino, revelam desvantagens na implementação deste classificador.

Face a estas dificuldades, a aplicação do Diagrama de Voronoi, do grafo de Gabriel e do grafo de Vizinhaça Relativa, referido no Capítulo 3, mostram ser boas estratégias para a condensação do conjunto de treino, sem que se degrade, significativamente, o desempenho do classificador.

4.2. Condensação por Estruturas de Proximidade

Sem perda de generalidade, a apresentação das técnicas de condensação por grafos de proximidade far-se-á em \mathfrak{R}^2 .

Seja $\mathcal{D} = \{ \langle \mathbf{x}^{(1)}, c^{(1)} \rangle, \langle \mathbf{x}^{(2)}, c^{(2)} \rangle, \dots, \langle \mathbf{x}^{(n)}, c^{(n)} \rangle \}$ um conjunto de treino com n objectos (pontos em \mathfrak{R}^2) previamente classificados e seja \mathbf{p} um novo objecto que se pretende classificar. A regra de decisão do vizinho mais próximo classifica \mathbf{p} na classe mais representada entre os seus vizinhos mais próximos. De acordo com o exposto nos capítulos anteriores e ilustrado na Figura 4.1 (a), o resultado da classificação está associado a uma partição do espaço dos atributos definida pela fronteira de decisão.

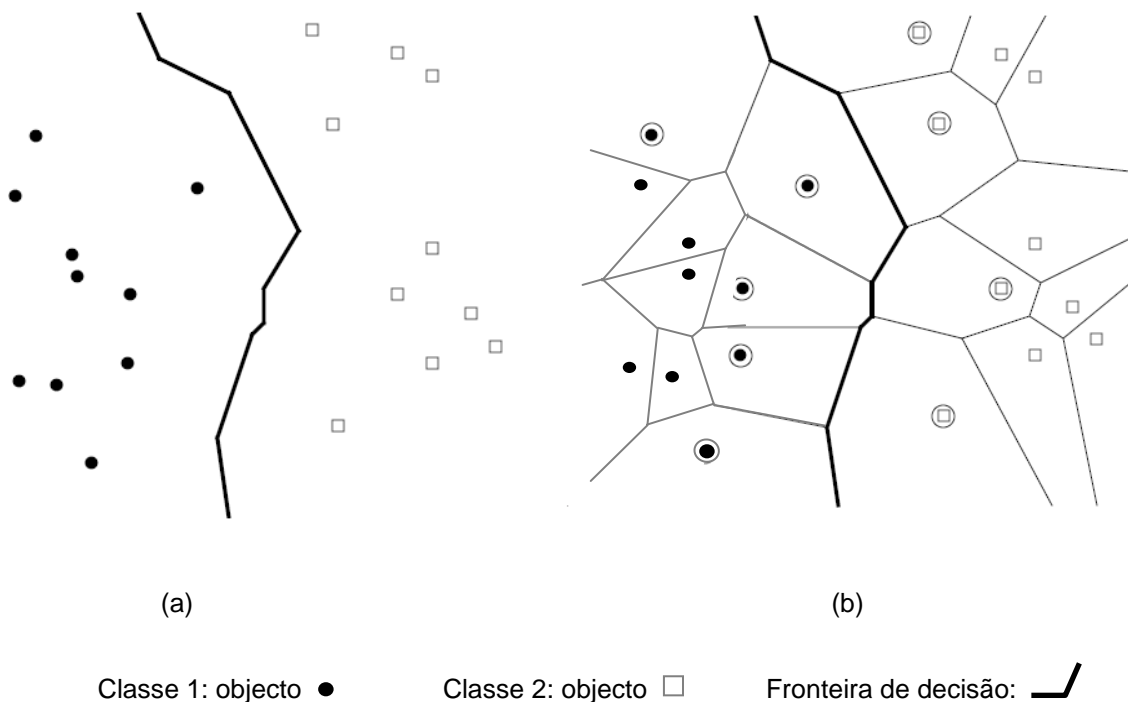


Figura 4.1. (a) Fronteira de decisão no espaço de atributos
(b) Fronteira de decisão no Diagrama de Voronoi

No espaço dos atributos, podemos associar cada objecto (um ponto $\mathbf{p} \in \mathbb{R}^2$) uma região $\mathcal{V}(\mathbf{p})$ que contém todos aqueles objectos que estão mais próximos de si do que de outro objecto. O lugar geométrico assim determinado, define-se como *região de Voronoi* (ou *polígono de Voronoi*) associada ao gerador \mathbf{p} (ver Capítulo 3, secção 3.1). Assim, a fronteira de decisão induzida pelo classificador 1-NN, pode ser determinada como sendo o conjunto das arestas de Voronoi que separam os pontos (objectos) que pertencem a classes diferentes [Bhattacharya et al., 2005]. (Esta observação está ilustrada na Figura 4.1 (b)).

Para simplificar a exposição que se segue, passaremos a chamar à fronteira de decisão induzida pelo classificador do vizinho mais próximo como *NN-fronteira* ou simplesmente como *fronteira de decisão*.

A fronteira de decisão constitui, pois, a parte mais importante do espaço de atributos, já que, o seu conhecimento seria suficiente para a classificação de novos objectos. O algoritmo de Dasarathy e White [Dasarathy & White, 1978] é o único algoritmo conhecido que gera a fronteira de decisão explicitamente e directamente a partir do conjunto de treino, cuja complexidade é de $O(dn^{d/2})$. Por outro lado, é referido em [Toussaint et al., 1992] que este algoritmo parece não ser aplicável a dimensões superiores a $d=3$. Assim, a fronteira raramente é determinada, pelo que a maioria dos algoritmos procuram reter apenas os objectos que geram uma fronteira de decisão idêntica.

Face ao exposto, a condensação por estruturas de proximidade é desejável e será tanto melhor, quanto menos variações provocar na fronteira de decisão. Isto significa que os pontos que estão próximos da *NN-fronteira* devem ser mantidos no conjunto de treino, enquanto que os que estão mais longe devem ser eliminados. Tomando por exemplo a Figura 4.1(b), os pontos da classe 1 e da classe 2, assinalados por um círculo, são os pontos que se encontram mais próximos da fronteira de decisão. São estes os pontos que devem permanecer no conjunto de treino para serem, posteriormente, usados na classificação de um novo ponto. Naturalmente que a eliminação dos restantes pontos pode suscitar,

nalguns casos, alterações na fronteira de decisão, pelo que é necessário encontrar o equilíbrio certo entre a redução do conjunto de treino e o bom desempenho do classificador. Este é, pois, o desafio da condensação por estruturas de proximidade como a seguir se comenta.

4.2.1. Condensação de Voronoi

Em 1979 Toussaint e Poulsen [Toussaint & Poulsen, 1979] usaram o diagrama d -dimensional de Voronoi para eliminar objectos “redundantes” do conjunto de treino \mathcal{D} . O subconjunto de \mathcal{D} obtido produzia, exactamente, a mesma fronteira de decisão que todo o conjunto de treino. Chamaram, ao seu método, condensação de Voronoi (*Voronoi condensing*), que a seguir se descreve:

CV: Algoritmo de condensação de Voronoi

Entrada: Conjunto de treino \mathcal{D} com n pontos classificados

Saída: Conjunto de Voronoi condensado: \mathcal{D}_V

Passo 1: Construir o diagrama de Voronoi sobre os pontos de \mathcal{D}

Passo 2: Para cada ponto \mathbf{p} de \mathcal{D} , determinar os seus vizinhos de Voronoi;

2.1. Se existir algum vizinho Voronoi que não seja da mesma classe do ponto \mathbf{p} , então, este é marcado.

Passo 3: Eliminar todos os pontos que não foram marcados

Figura 4.2. Pseudo-código do algoritmo de condensação de Voronoi

Esta técnica consiste em construir, numa primeira fase, o Diagrama de Voronoi e, a seguir, avaliar a classe dos vizinhos Voronoi de cada ponto do conjunto de treino \mathcal{D} . Como se pode observar na Figura 4.3, os pontos marcados com um círculo são aqueles pontos que têm pelo menos um vizinho Voronoi com uma classe diferente da sua e, por conseguinte, são mantidos no conjunto de treino, enquanto que os restantes (os que não foram marcados) são eliminados. O resultado da condensação de Voronoi é ilustrado na Figura 4.4.

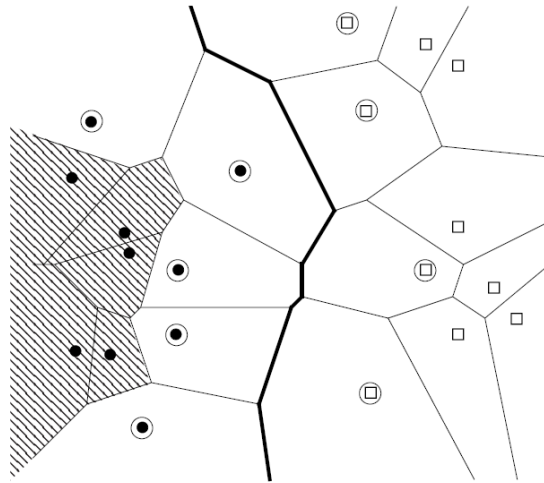


Figura 4.3. A sombreado está representado o conjunto de polígonos de Voronoi da classe 1 que são eliminados pela condensação de Voronoi.

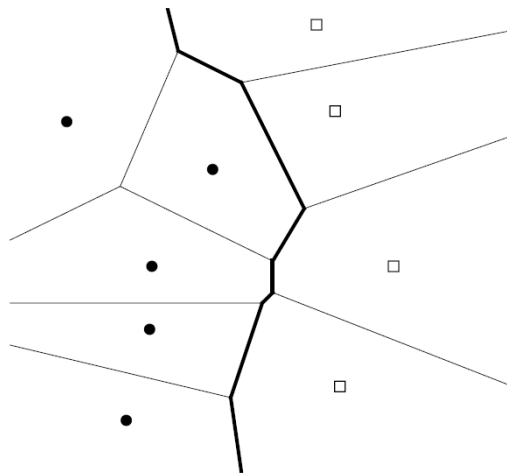


Figura 4.4. Resultado da condensação de Voronoi. A fronteira de decisão mantém-se inalterável.

Entre as duas maiores vantagens do algoritmo de condensação de Voronoi podemos referir que este produz uma *fronteira de decisão consistente* e, conseqüentemente, um conjunto de referência¹³ consistente. Para além disso,

¹³ Os autores no seu artigo [Toussaint et al.,1992] denominam ao subconjunto reduzido por “*reference set*”.

quando os pontos estão em posição geral¹⁴, a condensação de Voronoi produz o menor *conjunto fronteira de decisão consistente*. De facto, esta técnica é a única que reduz o *conjunto de treino* \mathcal{D} de tal forma que a fronteira de decisão definida pelo *conjunto de referência*, \mathcal{D}_V , é exactamente a mesma que a do conjunto de treino inicial. A demonstração destes resultados pode ser encontrada em [Toussaint et al., 1992]. Contudo, quando os pontos não estão em posição geral a condensação de Voronoi não produz um conjunto consistente mínimo. Podemos observar, na Figura 4.5, que seriam suficientes dois pontos, um por classe, para implementar a fronteira de decisão.



Figura 4.5. Exemplo patológico

Outro aspecto a referir na condensação de Voronoi é que esta trata todas as regiões de Voronoi de igual forma. Consequentemente, pode manter um certo número de pontos que, na prática, são irrelevantes e que, intuitivamente, até seriam eliminados. É o caso, por exemplo, quando alguns pontos estão distribuídos uniformemente entre duas circunferências (ver Figura 4.6).

Para este exemplo, a condensação de Voronoi mantém pontos bem separados, assim como grande parte da fronteira de decisão fora da região de interesse (região interior às circunferências).

¹⁴ Em Geometria Computacional, um conjunto de pontos está em posição geral quando não existem situações degeneradas, como, por exemplo: três pontos colineares ou quatro pontos co-circulares.

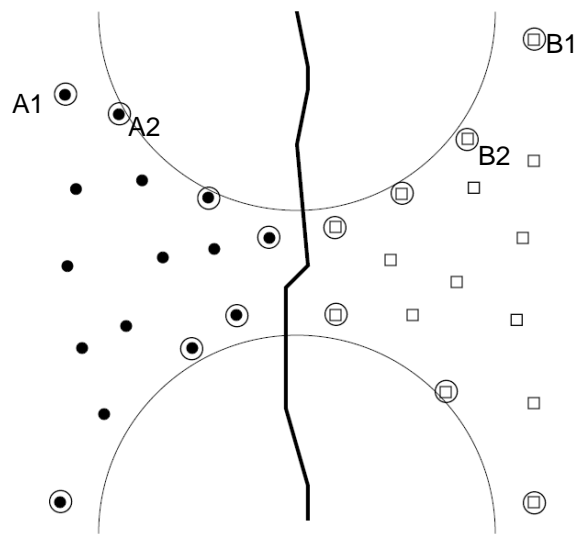


Figura 4.6. A1, A2, B1 e B2 são pontos bem separados, mas mantidos por condensação de Voronoi

A complexidade temporal e espacial, associada à construção do Diagrama de Voronoi, e o igual tratamento que é feito entre as regiões de Voronoi constituem os principais problemas desta técnica de condensação. Estas desvantagens levaram a que novas alternativas fossem propostas, nomeadamente, na procura de outras técnicas com custos inferiores e que reduzissem o conjunto de treino de uma forma mais criteriosa.

4.2.2. Condensação de Gabriel

A condensação de Gabriel é semelhante à condensação de Voronoi [Toussaint et al., 1992]. A diferença reside na utilização do grafo de Gabriel em vez do Diagrama de Voronoi, para a redução de pontos do conjunto de treino. Consequentemente, a eliminação dos pontos faz-se segundo a relação de vizinhança de Gabriel (Capítulo 3, secção 3.3.1).

A seguir mostra-se o pseudo-código do método de condensação por grafo de Gabriel.

CG: Algoritmo de condensação de Gabriel

Entrada: Conjunto de treino \mathcal{D} Saída: Conjunto de Gabriel condensado: \mathcal{D}_G

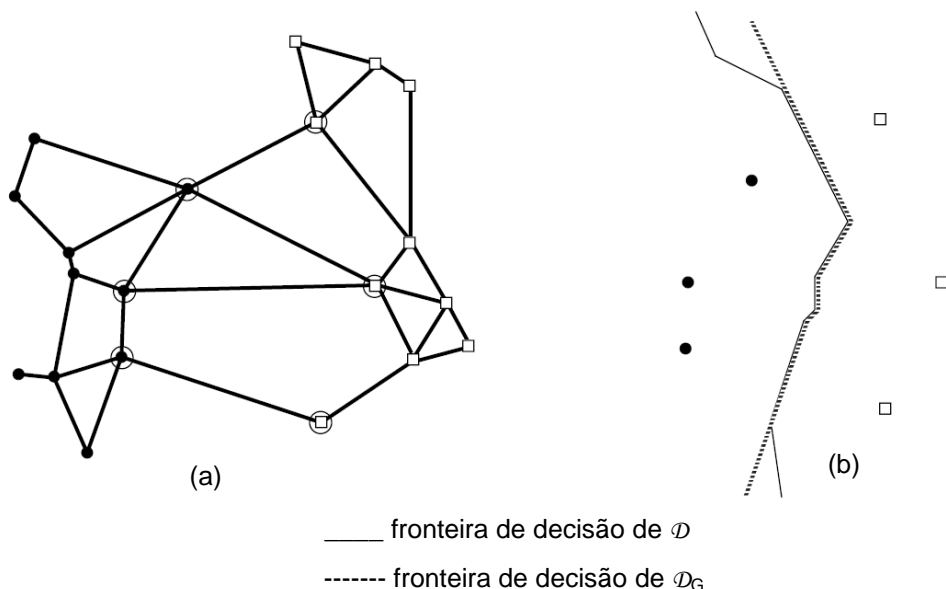
Passo 1: Construir o grafo de Gabriel sobre os pontos de \mathcal{D} Passo 2: Para cada ponto de \mathcal{D} determinar os seus vizinhos de Gabriel

- 2.1. Se existir algum vizinho Gabriel que não seja da mesma classe do ponto, então este é marcado

Passo 3: Eliminar todos os pontos que não foram marcados

Figura 4.7. Pseudo-código do algoritmo de condensação de Gabriel

A Figura 4.8 mostra o resultado da aplicação da condensação de Gabriel ao conjunto de pontos, tomados como exemplo neste estudo. Os pontos assinalados em (a) têm pelo menos um vizinho de Gabriel que não pertence à sua classe, portanto são aqueles que ficam após a condensação de Gabriel (b).

**Figura 4.8.** (a) Grafo de Gabriel (b) Condensação de Gabriel

Ao compararmos as Figuras 4.4 e 4.8(b), observamos que o número de pontos do conjunto editado por grafo de Gabriel é menor em relação à condensação

Voronoi. Todavia, o melhoramento na redução de pontos do conjunto de treino produz um conjunto condensado que não é fronteira de decisão consistente. Apuramos, pela Figura 4.8(b), que existem ligeiras diferenças entre as duas NN-fronteiras. Esta diferença reside apenas na região exterior ao invólucro convexo do conjunto dado [Toussaint et al.,1992], ou seja, na região de interesse as NN-fronteiras são coincidentes. Este facto traduz que não houve significativa degradação do desempenho do classificador.

Podemos dizer, ainda, que a condensação de Gabriel mostra ser um bom método para um conjunto de pontos idênticos aos descritos na Figura 4.6. Neste caso, a aplicação da condensação de Gabriel mantém, exactamente, a fronteira de decisão na região de interesse e reduz, consideravelmente, o conjunto de pontos para quatro (dois por classe - ver Figura 4.9). Pelo que este método, ao contrário da condensação de Voronoi, elimina pontos bem separados.

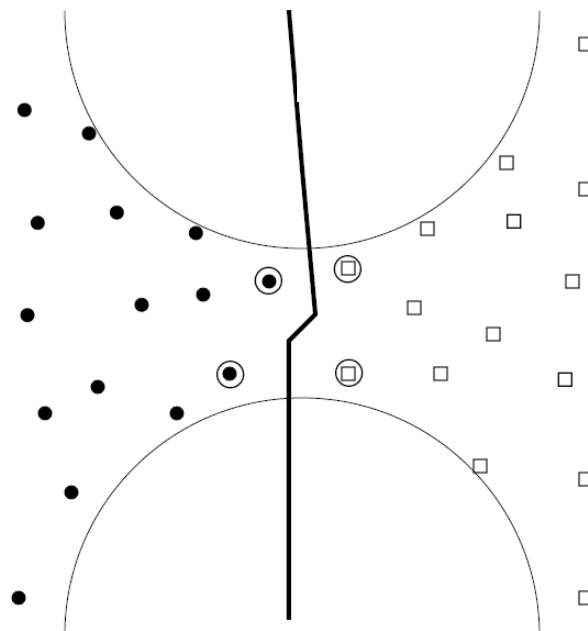


Figura 4.9. Num conjunto de pontos uniformemente distribuídos entre duas circunferências, a condensação de Gabriel reduz para quatro pontos e mantém a NN-fronteira na região de interesse

Comparando os dois tipos de condensação, até agora tratados, podemos observar que:

1. O conjunto condensado de Gabriel é menor, logo o custo computacional associado ao classificador 1-NN é também menor;
2. O conjunto condensado de Gabriel não é um conjunto consistente, mas, na prática, não degrada o desempenho do classificador.

Resultados práticos obtidos por Toussaint, Bhattacharya e Poulsen [Toussaint et al., 1992] mostram que a condensação de Gabriel é mais eficiente em termos computacionais que a condensação de Voronoi, tanto mais quando é adoptada a modificação heurística na construção do grafo de Gabriel, analisada na secção 3.3.1 do Capítulo 3.

4.2.3. Condensação de Vizinhaça Relativa

O grafo de Vizinhaça Relativa, analisado na secção 3.3.2 do Capítulo 3, é um subgrafo do grafo de Gabriel. Logo, é natural que também se deseje analisar a aplicação desta estrutura na condensação de conjuntos.

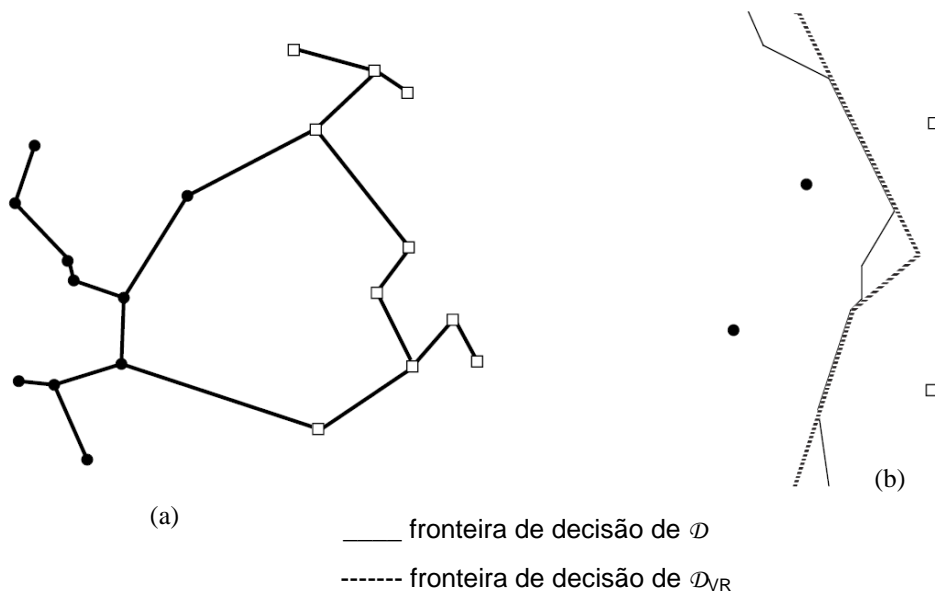


Figura 4.9. (a) Grafo de RNG (b) Condensação de Vizinhaça Relativa

A Figura 4.9(a) mostra o grafo de Vizinhaça Relativa do conjunto de treino inicial \mathcal{D} , e em (b) o resultado da aplicaçaõ da condensaçaõ Vizinhaça Relativa \mathcal{D}_{VR} .

É fácil verificar que o algoritmo de condensaçaõ Vizinhaça Relativa é semelhante ao algoritmo de condensaçaõ Gabriel, uma vez que apenas muda a relaçaõ de vizinhaça (ver secçaõ 3.3.2 do Capítulo 3) definida entre os pontos. Neste caso, o conjunto condensado contém ainda menos pontos que o conjunto condensado de Gabriel, o que implica, por sua vez, uma diminuicãõ dos custos computacionais associados à implementaçãõ do classificador. Mas será na realidade um benefício?

A NN-fronteira gerada pela condensaçaõ de vizinhaça relativa, quando confrontada com a NN-fronteira da condensaçaõ de Voronoi, difere, consideravelmente, não só na regiãõ exterior ao invólucro convexo como na regiãõ interior do mesmo. Consequentemente, o conjunto condensado de Vizinhaça Relativa não é fronteira de decisãõ consistente nem é um conjunto de referênciã consistente.

Portanto, a classificaçaõ de um novo ponto pode ser incorrecta mesmo que seja um ponto da regiãõ de interesse. Isto significa que pode existir uma degradaçaõ do desempenho do classificador 1-NN, quando se aplica a condensaçaõ de Vizinhaça Relativa.

É importante referir que Toussaint, Bhattacharya e Poulsen ressalvam no seu artigo [Toussaint et al., 1992] que os esquemas de condensaçaõ apresentados ao longo desta secçaõ podem ser estendidos à regra k-NN. Para tal, sugerem o método de re-etiqueta dos objectos do conjunto de treino pela regra k-NN, seguida da aplicaçaõ da regra 1-NN conjuntamente com um dos esquemas de condensaçaõ por grafo de proximidade.

4.3. Resultados Experimentais

Os resultados experimentais obtidos em 1992 por Toussaint, Bhattacharya e Poulsen [Toussaint et al., 1992] e em 1997 por Sánchez, Pla e Ferri [Sánchez et al., 1997] sugerem que a condensação de Gabriel é a melhor para a condensação do conjunto de treino para o classificador k-NN.

Os conjuntos de dados utilizados são entendidos pela Comunidade Científica como uma escolha universal, para a análise experimental do desempenho de algoritmos de aprendizagem baseada em instâncias (IBL). O primeiro conjunto de dados é o clássico conjunto “*Iris Data*”, obtido a partir do repositório de *UCI Machine Learning Repository*¹⁵. O segundo conjunto de dados, “*Cervical Cell Data*”, descreve as células do cólon proveniente do Laboratório de Processamento de Imagem Biomédica da Universidade de McGill. O conjunto “*Iris Data*” é composto por 150 instâncias. Cada instância representa uma flor da íris, descrita através de 4 atributos e classificada em três classes (*Virgínica*, *Setosa e Versicolor*). O conjunto “*Cervical Cell Data*” inclui 1999 instâncias, descritas através de 4 atributos e classificadas em duas classes (*normal* e *anormal*).

As Tabelas 1 e 2 mostram os resultados da aplicação do esquema de condensação por estruturas de proximidade (Diagrama de Voronoi, grafo de Gabriel e grafo de Vizinhança Relativa) para cada conjunto de dados, respectivamente. A taxa de erro para o classificador 1-NN foi estimada utilizando o método *bootstrap* (com janela uniforme¹⁶).

Da análise da Tabela 1, observamos que a taxa de erro para o classificador 1-NN, usando o método de condensação de Voronoi e de Gabriel, é igual à taxa de erro obtida, usando todo o conjunto de dados inicial.

¹⁵ C. J. Merz e P. M. Murphy. UCI repository of machine learning database. Internet <http://www.ics.uci.edu/~mllearn/MLRepository.html>, Department of Information and Computer Science, University of California

¹⁶ Para mais detalhes sobre o método de bootstrap com janela uniforme consultar o artigo [Toussaint et al, 1992].

	Conjunto Inicial	Condensação Voronoi	Condensação Gabriel	Condensação Vizinhança Relativa
Tamanho	150	109	39	21
Taxa de erro	1,3%	1,3%	1,3%	2,1%
Variância	0,31E-4	0,31E-4	0,31E-4	0,81E-4

Tabela 1. Resultados para o conjunto “*Iris Data*”

Isto significa que a condensação de Gabriel é a que mostra melhores resultados, uma vez que consegue uma excelente redução do conjunto de treino (até 26% do tamanho inicial) sem que se degrade o desempenho do classificador. Este resultado deve-se ao facto da condensação Gabriel preservar a fronteira de decisão na região de interesse, como foi anteriormente referido.

Quanto à condensação por Vizinhança Relativa, este método consegue uma redução maximal do conjunto de treino (até 14% do tamanho inicial), mas o desempenho do classificador reduz-se em, aproximadamente, 100%. Podemos concluir que ao eliminarmos adicionalmente 70 dados do conjunto condensado de Voronoi, pela condensação de Gabriel, a taxa de erro permanece a mesma (1,3%) e agrava-se, praticamente para o dobro, quando são removidos mais 18 dados pela condensação de Vizinhança Relativa.

Portanto, o conjunto condensado de Gabriel parece ter informação suficiente para classificar “*Iris Data*” (os dados da flor da íris) através da regra 1-NN.

Por outro lado, os resultados obtidos para o conjunto “*Cervical Cell Data*” reforçam a ideia de que a condensação de Gabriel é a melhor escolha, no que diz respeito à relação entre a redução do conjunto inicial e a correspondente taxa de erro (ver Tabela 2).

	Conjunto Inicial	Condensação Voronoi	Condensação Gabriel	Condensação Vizinhança Relativa
Tamanho	1999	1313	820	452
Taxa de erro	5,9%	5,9%	6,0%	9,7%
Variância	0,71E-4	0,71E-4	0,67E-4	0,45E-4

Tabela 2. Resultados para o conjunto “*Cervical Cell Data*”

4.4. Estudos Recentes

Recentemente, outros trabalhos foram desenvolvidos no âmbito do estudo da aplicação de novos métodos de condensação para os problemas de aprendizagem baseada em instâncias e, em particular, para o classificador do vizinho mais próximo ([Brighton & Mellish, 2000], [Houle, 2003], [Mukherjee, 2004] [Toussaint, 2002], [Wilson & Martinez, 2000]).

Brighton e Mellish, em 2000, propuseram um novo método híbrido de condensação, o algoritmo ICF (*Iterative Case Filtering*). Este novo método inclui duas fases. Na primeira fase, é aplicada a já conhecida *edição de Wilson* ao conjunto de treino e, em seguida, na segunda fase, é executada a *condensação adaptativa* (conhecida por *filtering step*). Este último procedimento envolve duas funções que determinam o *conjunto alcançável* (*reachable set*) e o *conjunto de cobertura* (*coverage set*) de cada objecto do conjunto de treino.

O *conjunto alcançável*, $R(\mathbf{x}_k, c_k)$, de um objecto (\mathbf{x}_k, c_k) consiste em todos os objectos que se encontram no interior da hiper-esfera centrada em \mathbf{x}_k e cujo raio é igual à distância entre \mathbf{x}_k e o seu vizinho mais próximo que pertence a outra classe. O *conjunto de cobertura*, $C(\mathbf{x}_k, c_k)$, de um objecto (\mathbf{x}_k, c_k) consiste em

todos os objectos do conjunto de treino que têm o objecto (\mathbf{x}_k, c_k) , no seu conjunto *alcançável*.

A condensação pelo algoritmo ICF pode ser resumida da seguinte forma: “*eliminar todos os objectos cuja cardinalidade do seu conjunto alcançável é superior à do seu conjunto de cobertura*”. A concepção desta regra pressupõe que um objecto pode ser removido quando pode ser representado por outros objectos. Esta nova ideia de condensação combinada com a condensação de Gabriel e, conjuntamente, suportadas por novas estruturas de pesquisa do k-vizinho mais próximo (SASH – *Spatial Approximation Sample Hierarchy* [Houle, 2003]) originaram um novo algoritmo denominado por algoritmo *híbrido do grafo de Gabriel* (*hybrid Gabriel graph algorithm*).

Segue-se um resumo das principais fases do algoritmo híbrido do grafo de Gabriel. Um estudo mais aprofundado pode ser encontrado em [Mukherjee, 2004].

Pré-processamento:

Passo 1: Aplicar ao conjunto de treino inicial a técnica de *edição de Wilson*, mas considerando a relação de vizinhança Gabriel;

Passo 2: Reduzir o conjunto obtido no passo 1 por *condensação de Gabriel*;

Passo 3: Aplicar o algoritmo ICF ao conjunto obtido no passo 2.

Regra de decisão: classificar um novo objecto \mathbf{z} segundo a classe mais votada entre os seus vizinhos Gabriel.

Na realidade, a versão de implementação do algoritmo não constrói o grafo de Gabriel, mas uma aproximação do grafo de Gabriel, uma vez que o custo da construção do grafo de Gabriel, para conjuntos de treino muito grandes, é elevado (Capítulo 3, secção 3.3). Para este efeito, utilizam uma modificação da estrutura SASH que chamaram GSASH – *Gabriel Spatial Approximation Sample Hierarchy*.

Resultados experimentais obtidos em conjuntos de dados *standards* (Tabela 3) sugerem que o *algoritmo híbrido do grafo de Gabriel* é o melhor quando

comparado com outros algoritmos, nomeadamente, com a condensação de Gabriel e de ICF.

Ao analisarmos os valores da taxa de erro e armazenamento dos três algoritmos, podemos observar que o *algoritmo híbrido do grafo de Gabriel* (na Tabela 3, simplesmente Híbrido) combina a maior redução de pontos (9% e 11,06% para a “*Iris Data*” e “*Cervical Cell Data*”, respectivamente) com o bom nível de desempenho do classificador (taxa de erro 4,67% para o conjunto “*Iris Data*” e taxa de erro 9,17% para “*Cervical Cell Data*”).

	Gabriel		Híbrido		ICF	
	Taxa de erro	Armazenamento	Taxa de erro	Armazenamento	Taxa de erro	Armazenamento
“ <i>Iris Data</i> ”	4%	20,81%	4,67%	9,0%	4,67%	13,33%
“ <i>Cervical Cell Data</i> ”	8,93%	35,37%	9,17%	11,06%	10,1%	15,36%

Tabela 3. Resultados da aplicação dos algoritmos de condensação: Gabriel, Híbrido e ICF, usando o método de validação cruzada com 10-folds para estimar a taxa de erro.

Em conjuntos de dados sintéticos (por exemplo 1000 pontos gerados aleatoriamente num esquema rectangular) os resultados obtidos (ver Tabela 4) testemunham que o algoritmo híbrido de grafo de Gabriel contrabalança entre a agressiva redução realizada pelo algoritmo ICF (cerca de 95%) e o similar desempenho do classificador¹⁷ k-NN (taxa de erro 10,3%) no conjunto condensado de Gabriel (taxa de erro 10,4%).

¹⁷ À semelhança dos trabalhos realizados por Sánchez, Pla e Ferri, a escolha de k variou entre um ou três [Sánchez et al., 1997].

	Gabriel	Híbrido	ICF
Nº de pontos	124	75	50
Armazenamento	13,6%	8,65%	6,64%
Taxa de erro	10,4%	10,3%	11,8%

Tabela 4. Resultados da aplicação dos algoritmos de condensação: Gabriel (aproximado), Híbrido e ICF em dados sintéticos, usando o método de validação cruzada com 10 - folds para estimar a taxa de erro

Em conclusão, podemos dizer que, perante este pequeno resumo de resultados e um vasto conjunto de experiências realizadas, o algoritmo híbrido do grafo de Gabriel é o melhor, no que diz respeito à condensação de conjunto de treino para o classificador k-NN.

Capítulo 5

Considerações Finais

No presente trabalho foi estudado o problema da condensação do conjunto de treino por aplicação de estruturas de proximidade e vizinhança para a implementação da regra k-NN. Este classificador, de fácil implementação e conceptualmente simples, apresenta um comportamento assintótico óptimo, resultado demonstrado por Cover & Hart (1967), cuja probabilidade de erro é majorada por duas vezes o erro de Bayes. Por outro lado, o classificador k-NN não tem fase de treino, por isso, se diz ser um algoritmo preguiçoso (*lazy algorithm*). Consequentemente, a sua fase de implementação (classificação) é lenta e pode apresentar um elevado custo computacional, principalmente em conjuntos de treino muito grandes (n objectos com d atributos). Assim, logo que chega um novo objecto para ser classificado, é preciso calcular todas as distâncias entre o novo objecto e os elementos do conjunto de treino. Este processo envolve elevados custos computacionais associados ao espaço de armazenamento de todo o conjunto de treino e ao tempo de cálculo de todas as distâncias. Todos estes aspectos têm sido alvos de atenção desde os anos 50. Desde então, várias aproximações têm sido propostas, na tentativa de melhorar o desempenho do classificador k-NN.

A redução do custo computacional pode ser concretizada, tendo em consideração três áreas de intervenção: i) *redução do cálculo de distâncias* através do

armazenamento das inter-distâncias entre os elementos do conjunto de treino, tarefa realizada pela família de algoritmos AESA; ii) *atribuição de diferentes pesos aos atributos e/ou diferentes pesos aos vizinhos mais próximos*, na tentativa de reduzir a dimensionalidade do conjunto de treino; iii) *selecção de protótipos* através das técnicas de edição e/ou condensação, cuja referência principal é *Wilson's algorithm* e *Hart's algorithm*, respectivamente, para a desejada redução do conjunto de treino.

Recentemente, os métodos associados a estruturas geométricas - Diagrama de Voronoi, grafo de Gabriel e grafo de Vizinhança Relativa - revelaram ser os melhores, no que diz respeito à condensação para a implementação da regra k-NN, como é referido em [Bhattacharya et al., 2005], [Sánchez et al., 1997] e [Toussaint et al., 1992].

Para melhor compreendermos a aplicação destas estruturas geométricas na condensação, primeiro, foi realizado um estudo detalhado das propriedades e dos métodos de construção associados a cada estrutura para, em seguida, estudarmos as técnicas geométricas de condensação, propriamente ditas.

A *condensação de Voronoi* é a única que mantém os dois critérios de consistência exigidos – *fronteira de decisão consistente* e *conjunto de referência consistente*. Esta seria a melhor opção, não fosse o elevado custo computacional associado à construção do Diagrama de Voronoi em espaços d -dimensionais e o igual tratamento que é feito entre as regiões de Voronoi. Estas duas desvantagens são claramente melhoradas quando se faz a condensação de Gabriel. Mas, em contrapartida, a condensação de Gabriel (num conjunto de treino) não satisfaz os critérios de consistência. Ainda assim, consegue preservar a fronteira de decisão na região de interesse de classificação pelo que o desempenho do classificador não se degrada significativamente. A condensação de Vizinhança Relativa reduz, consideravelmente, o conjunto de treino, mas degrada o desempenho do classificador, uma vez que, neste caso, a NN-fronteira, quando comparada com a do conjunto condensado de Voronoi, é muito diferente. Face ao exposto, a

condensação de Gabriel parece ser a que melhores resultados produz, não fosse a complexidade algorítmica ainda bastante elevada.

Finalmente, fez-se referência aos últimos resultados alcançados no desenvolvimento de algoritmos de aprendizagem baseados em instâncias (*IBL – instance based learning*). Um dos algoritmos mais promissórios, desenvolvidos na actualidade, é o algoritmo *Híbrido do grafo de Gabriel*. Este algoritmo foi proposto em 2004 por Mukherjee [Mukherjee, 2004] e concerta a implementação de vários métodos: i) edição de *Wilson-Gabriel*; ii) condensação de *Gabriel*; iii) condensação pelo *ICF*; iv) regra de decisão do vizinho *Gabriel mais próximo* e, finalmente, v) a utilização de uma estrutura chamada *GSASH*, adequada à computação do grafo de aproximação de *Gabriel*. A combinação das várias técnicas conduz a resultados nunca antes alcançados.

Em resumo, podemos afirmar que face aos resultados experimentais apresentados em vários trabalhos, o grafo de *Gabriel* parece ser a estrutura mais natural subjacente a qualquer método de condensação para a implementação do classificador *k-NN*, já que pelas suas propriedades preserva a parte da fronteira de decisão mais importante, aquela que é interior ao invólucro convexo do conjunto dado. Posto isto, a contribuição do grafo de *Gabriel* continua a ser fundamental na redução do conjunto de treino para a implementação da regra de decisão *k-NN*.

Apresenta-se como linha de trabalho futuro a investigação do problema da condensação por novas estruturas geométricas de vizinhança pertencentes à família β -esqueletos. Provavelmente, em estudos futuros, poderá vir a existir um valor de β para o qual o grafo associado à condensação elimine pontos adicionais que são irrelevantes para o desempenho do classificador.

Outro enriquecimento científico para esta área de pesquisa futura poderia ser a investigação da aplicação do *Diagrama de Voronoi de ordem- k* ao problema da condensação para a regra *k-NN*.

Bibliografia

- [Aha et al., 1991] D. W. Aha, D. Kibler e M. K. Albert. Instance-Based Learning Algorithms. *Machine Learning*, 6, pp. 37-66, (1991).
- [Aurenhammer & Klein, 2000] F. Aurenhammer e R. Klein. Voronoi diagrams, in J. R. Sack, J. Urrutia (Eds.), *Handbook of Computational Geometry*, Elsevier, Amsterdam, pp. 201–290, (2000).
- [Berg et al., 2000] M. Berg, M. van Kreveld, M. Overmars e O. Schwarzkopf. *Computacional Geometry: Algorithms and Applications*. Springer Verlag, (2000).
- [Bhattacharya,1982] B. K. Bhattacharya. *Application of computational geometry to pattern recognition problems*. Ph.d. thesis, School of Computer Science, McGill University, (1982).
- [Bhattacharya et al., 2005] B. K. Bhattacharya, K. Mukherjee e G. T. Toussaint. Geometric Decision Rules for Instance-based Learning Problems, in *Proceedings of Frist International Conference on Pattern Recognition and Machine Intelligence (PReMI'05)*, Kolkata, Índia, (2005).
- [Bolstad, 2004] W. M. Bolstad. *Introduction to Bayesian Statistics*. Wiley-IEEE, (2004).

- [Brighton & Mellish, 2000] H. Brighton e C. Mellish. Advances in Instance Selection for Instance-Based Learning Algorithms. *Data Mining and Knowledge Discovery*, 6, pp. 156-172, (2000).
- [Cormen et al., 2001] T. H. Cormen, C. E. Leiserson e R. L. Rivest. *Introduction to Algorithms*. MIT Press, Second Edition, (2001).
- [Cover & Hart, 1967] T. M. Cover e P. E. Hart. Nearest Neighbor Pattern Classification, *IEEE Transactions on Information Theory*, vol. IT-13, No.1, pp. 21-27, (1967).
- [Dasarathy & White, 1978] B. Dasarathy e L. J. White. A characterization of nearest neighbor rule decision surfaces and a new approach to generate them, *Pattern recognition*, vol 10, pp. 41-46, (1978).
- [Devijver, 1980] P. A. Devijver e J. Kittler. On the edited nearest neighbour rule, in *Proc. of the 5th International Conference on Pattern Recognition*, pp. 72-80, (1980).
- [Devijver, 1982] P. A. Devijver e J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall, Englewood Cliffs, NJ, (1982).
- [Devroye, 1988] L. Devroye. Automatic pattern recognition: a study of the probability of error, *IEEE Trans. on Pattern Analysis and Machine Intelligence* PAMI-10, pp. 530-543, (1988)
- [Devroye et al., 1996] L. Devroye, L. Györfi e G.Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, NY, (1996).
- [Duda & Hart, 1973] R. O. Duda e P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., (1973).

- [Fix & Hodges, 1951] E. Fix e L. Hodges Jr. *Discriminatory analysis, nonparametric discrimination*. USAF school Aviation Medicine, Randolph field, Project 21-49-004, Rept4, (1951).
- [Fortune,1987] S. J. Fortune. *A Sweepline Algorithm for Voronoi Diagrams*. *Algorithmica*, 2, pp. 153-174, (1987).
- [Friedman et al., 1975] J. H. Friedman, F. Baskett e L. J. Shustek. An algorithm for finding Nearest Neighbours, *IEEE Transactions on Computers*, 10, pp. 1000-1006, (1975).
- [Fu, 1982] K. S. Fu. *Syntatic Pattern Recognition and Application*. Prentice-Hall, Inc., Englewood Cliffs, (1982).
- [Fukunaga, 1990] K. Fukunaga. *Introduction to Statistial Pattern Recognition*. Academic Press, San Diego, CA, (1990).
- [Fukunaga & Narendra, 1995] K. Fukunaga e M. Narendra. A Branch and Bound algorithm for computing k Nearest neighbours, *IEEE Transactions on Computers*, 10, pp. 750-735, (1995).
- [Gates, 1972] G. W. Gates. The reduced nearest neighbor rule, *IEEE Trans. On Information Theory* IT-18, pp. 431-433, (1972).
- [Goodaire & Parmenter 1997] E. G. Goodaire e M. M. Parmenter. *Discrete Mathematics with Graph Theory*. Prentice-Hall, (1997).
- [Hart, 1968] P. E. Hart. The condensed nearest neighbor rule, *IEEE Trans. on Information Theory* IT-14, pp. 515-516, (1968).

- [Hattori & Takahashi, 2000] K. Hattori e M. Takahashi. A new edited k.nearest neighbor rule in the pattern classification problem. *Pattern Recognition*, 13, pp. 521-528, (2000).
- [Hernández, 2003] G. Hernández. *Teoría de Grafos*, Fundación General de la UPM, España, (2003).
- [Houle, 2003] M. Houle. SASH: a Spatial Approximation Sample Hierarchy for Search. *IBM Tokyo Research Laboratory Report RT-0517*, (2003).
- [Inza, 1999] I. Inza. *Feature Weighting for Nearest Neighbor Algorithm by Bayesian Networks based Combinatorial Optimization*. Proceedings of the Student Session of Advanced Course on Artificial Intelligence. ACAI 99, pp. 33-35, (1999).
- [Jaromczyk, 1992] J. W. Jaromczyk e G.T. Toussaint. Relative neighbourhood graphs and their relatives. *Proc. IEEE* 80, pp. 1502-1517, (1992).
- [Kirkpatrick & Radke, 1985] D. G. Kirkpatrick e J. D. Radke. A framework for computational morphology, in G. T. Toussaint, editor, *Computational Geometry*, NorthHolland, Amsterdam, Netherlands, pp. 217-248, (1985).
- [Kohavi & Sommerfield,1995] R. Kohavi e D. Sommerfield. Feature Subset Selection using the wrapper model: overfitting and dynamic search space topology. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, KDD'95, Montreal, Canada, pp. 192-197, (1995).
- [Koplowitz & Brown, 1981] J. Koplowitz e T. A. Brown. On the relation of performance to editing in nearest neighbor rules. *Pattern Recognition* , 13, pp. 251-255, (1981).

- [Lunts, 1967] A. Lunts e V. Brailovsky. Evaluation of attributes obtained in statistical decision rules. *Engineering Cybernetics*, 3, pp. 98-109, (1967).
- [Matula & Sokal, 1980] D. W. Matula e R. R. Sokal. Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical Analysis*, 12, pp. 205-222 ,(1980).
- [Mendoza, 2006] F. R. Mendoza. *Geometria Computacional*. Universidad de Los Andes, Facultad de Ciencias, Departamento de Matemáticas, Mérida Venezuela, pp. 52-71, (2006).
- [Micó, 1996] M. L. Micó. *Algoritmos de búsqueda de vecinos más próximos en espacios métricos*. Tesis doctoral. Universidad Politécnica de València, (1996).
- [Michie et al., 1994] D. Michie, D. J. Spiegelhalter e C. C. Taylor. *Machine learning, neural and statistical classification*. Ellis, New York, (1994).
- [Mitchell, 1997] T. Mitchell. *Machine Learning*. McGraw-Hill, (1997).
- [Mitchell, 2006] T. M. Mitchell. *The discipline on Machine Learning*. School on Computer Science, Carnegie Mellon University, Pittsburgh, USA, (2006).
- [Mukherjee, 2004] K. Mukherjee. *Application of the Gabriel graph to instance-based learning*. M.sc. Project, School of Computing Science, Simon Fraser University, (2004).
- [Okabe et al.,1992] A. Okabe, B. Boots e K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrama*. John Wiley & Chichester, UK, (1992).

- [Preparata & Shamos, 1985] F. P. Preparata e M. I. Shamos. *Computational Geometry: An Introduction*, Texts and Monographs in Computer Science, Springer-Verlag, New York, (1985).
- [Ramasubramanian, 1990] V. Ramasubramanian e K.K. Paliwal. An efficient approximation-elimination algorithm for fast nearestneighbor search based on a spherical distance coordinate formulation, in *Signal Processing V: Theories and Applications*. L. Torres-Urgell and M.A.Lagunas-Hernández (eds.), North-Holland, pp. 1323-1326, (1990).
- [Rittet et al., 1975] G. L. Ritter, H. N. Woodruff, S. R. Lowri e T.L. Isenhour. An algorithm for a selective Nearest-Neighbour Decision Rule, *IEEE Transaction on Theory*, 21, pp. 665-669, (1975).
- [Sánchez, 2005] J. S. Sánchez Garreta. *Aprendizaje y Clasificación basadas en Criterios de Vecindad. Métodos Alternativos y Análisis Comparativo*. Tesis Doctoral. Universitat Jaume I. Departament d'Informàtica, (2005).
- [Sánchez et al., 1997] J. S. Sánchez, F. Pla e F. J. Ferri. Prototype selection for the nearest neighbour rule through proximity graphs. *Pattern Recognition Letter*, 18, pp. 507-513, (1997).
- [Schalkoff, 1992] R. Schalkoff. *Pattern Recognition – Statistic and Structural Neural Approaches*. John Willey & Sons, (1992).
- [Shamos & Hoey, 1975] M. I. Shamos e D. Hoey. Closest point problems, *Proc. 16th Annual IEEE Symposium in Foundations of Computer Science*, pp. 151–162, (1975).
- [Sierra, 2000] B. Sierra. *Aportaciones metodológicas a la Clasificación Supervisada*. Tesis Doctoral. Departamento Ciencias de la Computación e Inteligencia Artificial, Universidad del País Vasco, (2000).

- [Stone, 1974] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society* 36, pp. 111-147, (1974).
- [Tomek, 1976a] I. Tomek. An experiment with the edited nearest neighbour rule, *IEEE Transaction on Systems, Man and Cybernetics* SMC-6, pp. 448-452, (1976).
- [Tomek, 1976b] I. Tomek. A generalization of the k -NN rule, *IEEE Transaction on Systems, Man and Cybernetics* SMC-6, pp. 121-126, (1976).
- [Tomek, 1976c] I. Tomek. Two modifications of CNN, *IEEE Transaction on Systems, Man and Cybernetics* SMC-6, pp. 769-772, (1976).
- [Toussaint, 1980] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12, pp. 261-268, (1980).
- [Toussaint, 1991] G. T. Toussaint. *Some Unsolved Problems on Proximity Graphs*. School of Computer Science, McGill university, Montreal, Canada, (1991).
- [Toussaint, 2002] G. T. Toussaint. Proximity graphs for nearest neighbour decision rules: Recent Progress, in *Proceedings of 34th Symposium on Computing and Statistics*, Montreal, Canada, (2002).
- [Toussaint et al., 1985] G. T. Toussaint, B. K. Bhattacharya e R. S. Poulsen. The Application of Voronoi diagrams to nonparametric decision rules, in *Computer Science and Statistics: The Interface*, L. Billard (ed.), Elsevier Science, North-Holland, Amsterdam, pp. 97-108, (1985).
- [Toussaint et al., 1992] G. T. Toussaint, B. K. Bhattacharya e R. S. Poulsen. Application of Proximity Graphs to editing Nearest neighbor Decision Rules, *International Symposium on Information Theory*, Santa Mónica, (1992).

- [Toussaint & Poulsen, 1979] G. T. Toussaint e R. S. Poulsen. Some new algorithms and software implementation methods for pattern recognition research, in *Proc. 3rd International COMPSAC*, pp. 55-63, (1979).
- [Vidal , 1986] E. Vidal. An algorithm for finding nearest neighbours in (approximately) constant average time. *Pattern Recognition Letters*, 4, pp. 145-157, (1986).
- [Wettschereck, 1994] D. Wettschereck. *A study of distance-based Machine Learning Algorithms*. Ph.D. Thesis, Oregon State University, (1994).
- [Wilson, 1972] D. L. Wilson. Asymptotic properties of nearest neighbour rules using edited data. *IEEE Trans. On Systems, Man and Cybernetics SMC-2*, pp. 408-421, (1972).
- [Wilson & Martinez, 2000] D. R. Wilson e T. R. Martinez. Reduction Techniques for instance based learning algorithms. *Machine Learning*, 38, pp. 257-286, (2000).