CC-226 Introdução à Análise de Padrões

Estimação Não-Paramétrica e Aprendizado por Instâncias

Carlos Henrique Q. Forster[1]

[1]Divisão de Ciência da Computação
Instituto Tecnológico de Aeronáutica

16 de março de 2008

Outline

Estimação de Densidade

- Estimação de densidade é o problema de estimar a função de densidade de probabilidade. É um problema intimamente relacionado aos problemas de classificação e de regressão.

Estimação por votação em grade

- Consideramos intervalos consecutivos de comprimento fixo h.
- Esses intervalos em 2D formam uma grade de quadrados de lado h.
- No caso de dimensão M (atributos) formam-se hipercubos de lado h e volume h^M.
- Para N instâncias, a probabilidade de uma instância estar numa célula é:

$$p_i = \frac{n_i}{Nh^M}$$

Número de amostras em cada cubo

- Dado um total de N observações obtidas com distribuição $p(\mathbf{x})$.
- Cada ponto tem probabilidade P de cair na região R.
- O total K de pontos na região R segue uma distribuição binomial:

$$Bin(K|N,P) = \frac{N!}{K!(N_K)!} P^K (1-P)^{N-K}$$

Densidade estimada

- A esperança de K é dada por NP.
- A variância é $NP(1 - P)$
- Assumindo $p(x)$ aproximadamente constante em torno da região R e R suficientemente pequena, aproximamos $P \simeq p(\mathbf{x})V$.
- Substituindo, obtemos para o volume V da região R:

$$p(x) = \frac{K}{NV}$$

Densidade estimada

- Como veremos, há duas formas de explorar esse resultado:
- Deixar K constante e permitir variar V, o que dá origem a um método de k-vizinhos mais próximos (k-nearest neighbors);
- Fixar V e determinar K a partir dos dados, o que dá origem ao método dos kernels.

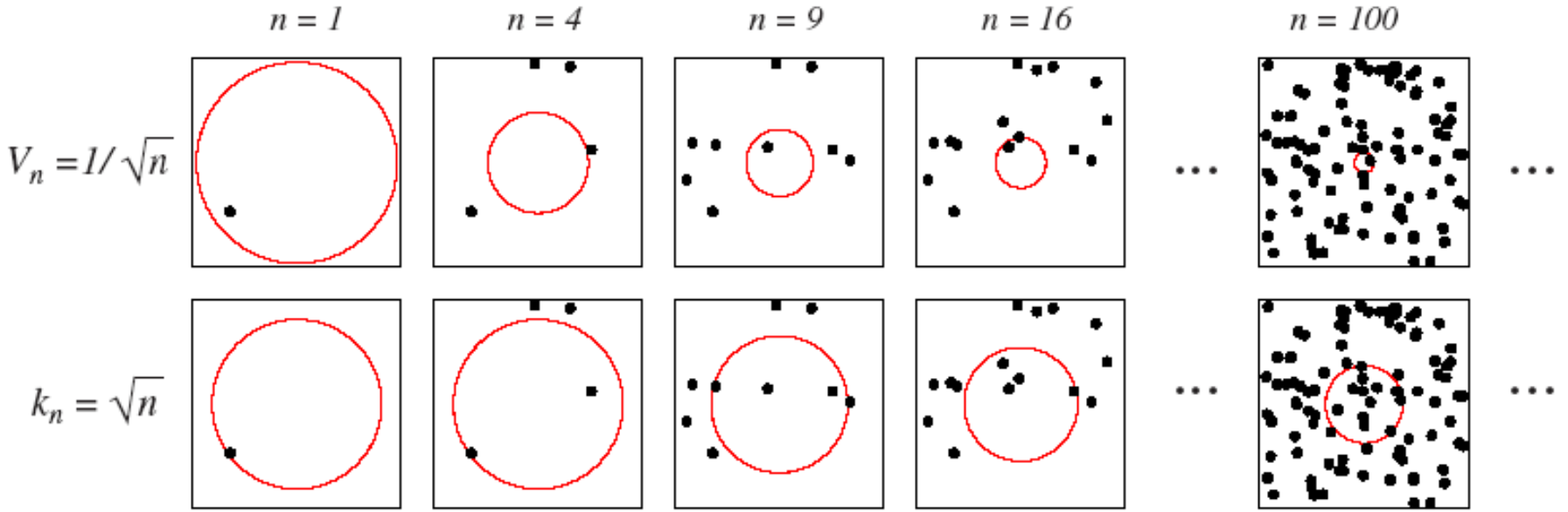


FIGURE 4.2. There are two leading methods for estimating the density at a point, here at the center of each square. The one shown in the top row is to start with a large volume centered on the test point and shrink it according to a function such as $V_n = 1/\sqrt{n}$. The other method, shown in the bottom row, is to decrease the volume in a data-dependent way, for instance letting the volume enclose some number $k_n = \sqrt{n}$ of sample points. The sequences in both cases represent random variables that generally converge and allow the true density at the test point to be calculated. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Classificação

- Podemos classificar uma observação cujos atributos remetam a uma célula da grade.
- Classificamos de acordo com a maioria das observações da amostra de treinamento.

Maldição da Dimensionalidade

- O numero de células a dividir o espaço cresce exponencialmente com a dimensão (número de atributos).
- Lembrando o problema de regressão múltipla, considerar o que acontece com o número de interações entre fatores quando a dimensão (número de fatores) aumenta.

Função de Janela

- Cada célula pode ser representada através de uma função janela ou *gate* definida da forma:

$$\phi(\mathbf{u}) = \begin{cases} 1, & |u_i| \leq 1/2, i = 1, \ldots, M, \\ 0, & \text{caso contrário} \end{cases}$$

- O número de pontos dentro do hipercubo de lado h será dado por:

$$K = \sum_{i=1}^{N} \phi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

.

Densidade estimada para a janela

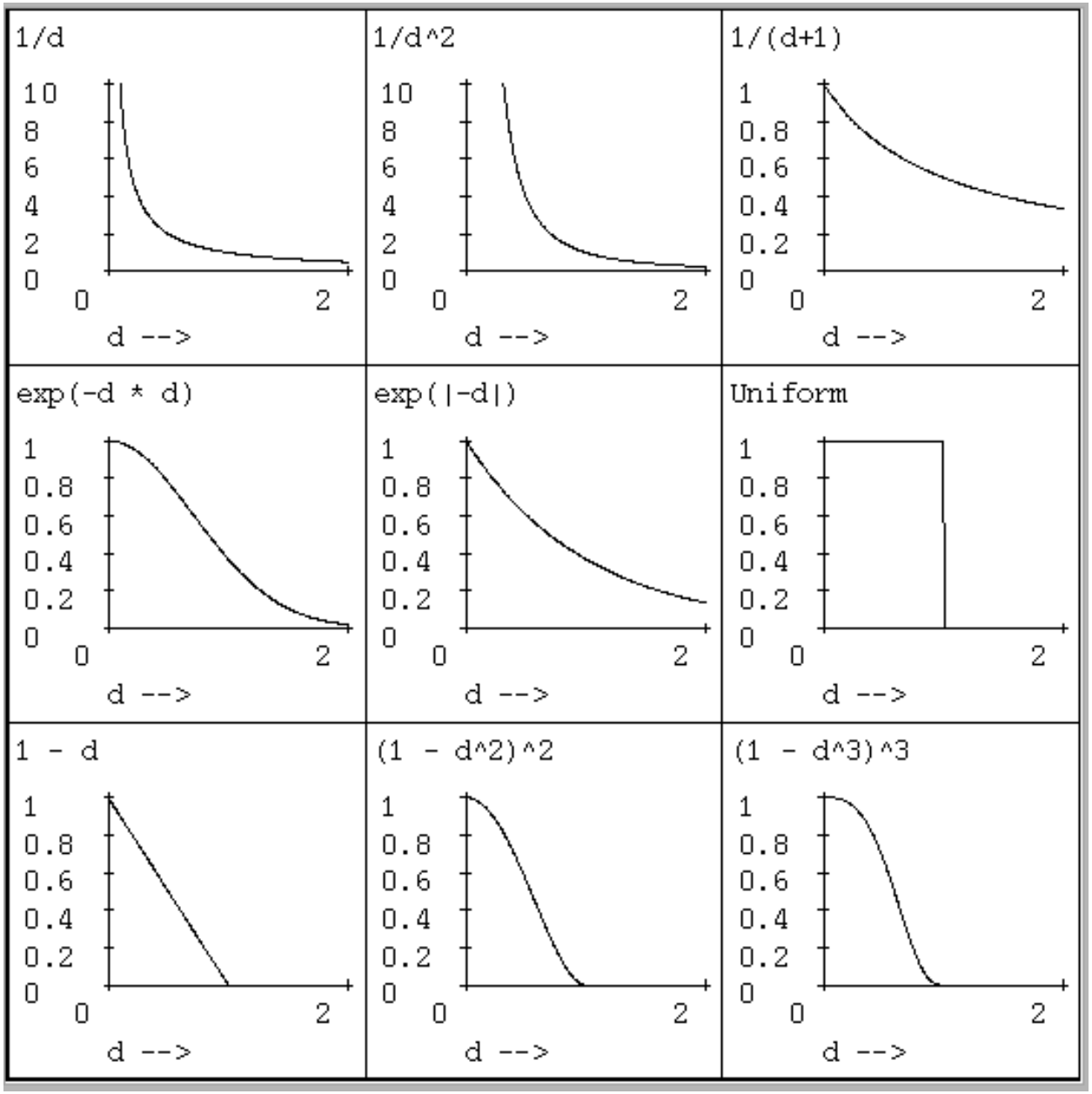
- Substituindo:

$$p(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^M} \phi \left(\frac{\mathbf{x} - \mathbf{x}_i}{h} \right)$$

Janela de Parzen

- Janela de Parzen ou Kernel.
- Na forma geral, não há necessidade de ser uma função *gate* abrupta. Podem-se utilizar funções de base radial com sobreposições.
- Lembrar da regressão com funções RBF da aula passada.
- Por exemplo, utilizando uma Gaussiana, a densidade estimada é então:

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{(2\pi h^2)^{1/2}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2h^2} \right)$$


1/d
10
8
6
4
2
0
0
2
d -->
1/d^2
10
8
6
4
2
0
0
2
d -->
1/(d+1)
1
0.8
0.6
0.4
0.2
0
0
2
d -->
exp(-d * d)
1
0.8
0.6
0.4
0.2
0
0
2
d -->
exp(|-d|)
1
0.8
0.6
0.4
0.2
0
0
2
d -->
Uniform
1
0.8
0.6
0.4
0.2
0
0
2
d -->
1 - d
1
0.8
0.6
0.4
0.2
0
0
2
d -->
(1 - d^2)^2
1
0.8
0.6
0.4
0.2
0
0
2
d -->
(1 - d^3)^3
1
0.8
0.6
0.4
0.2
0
0
2
d -->

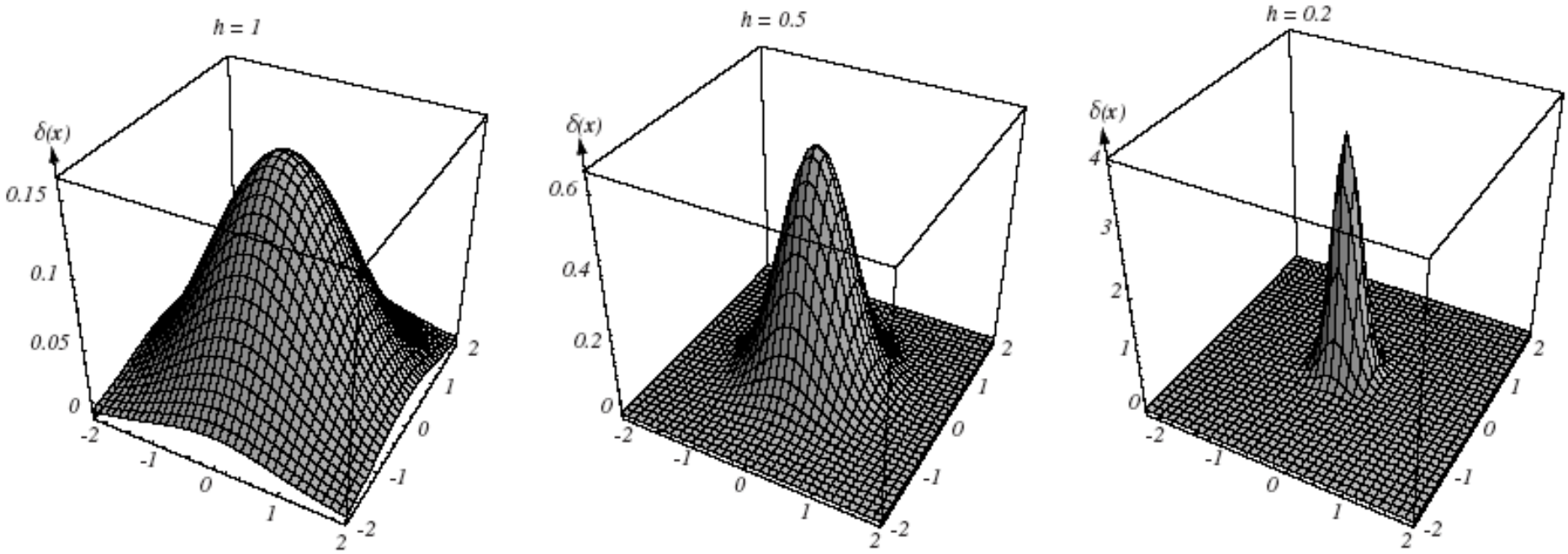
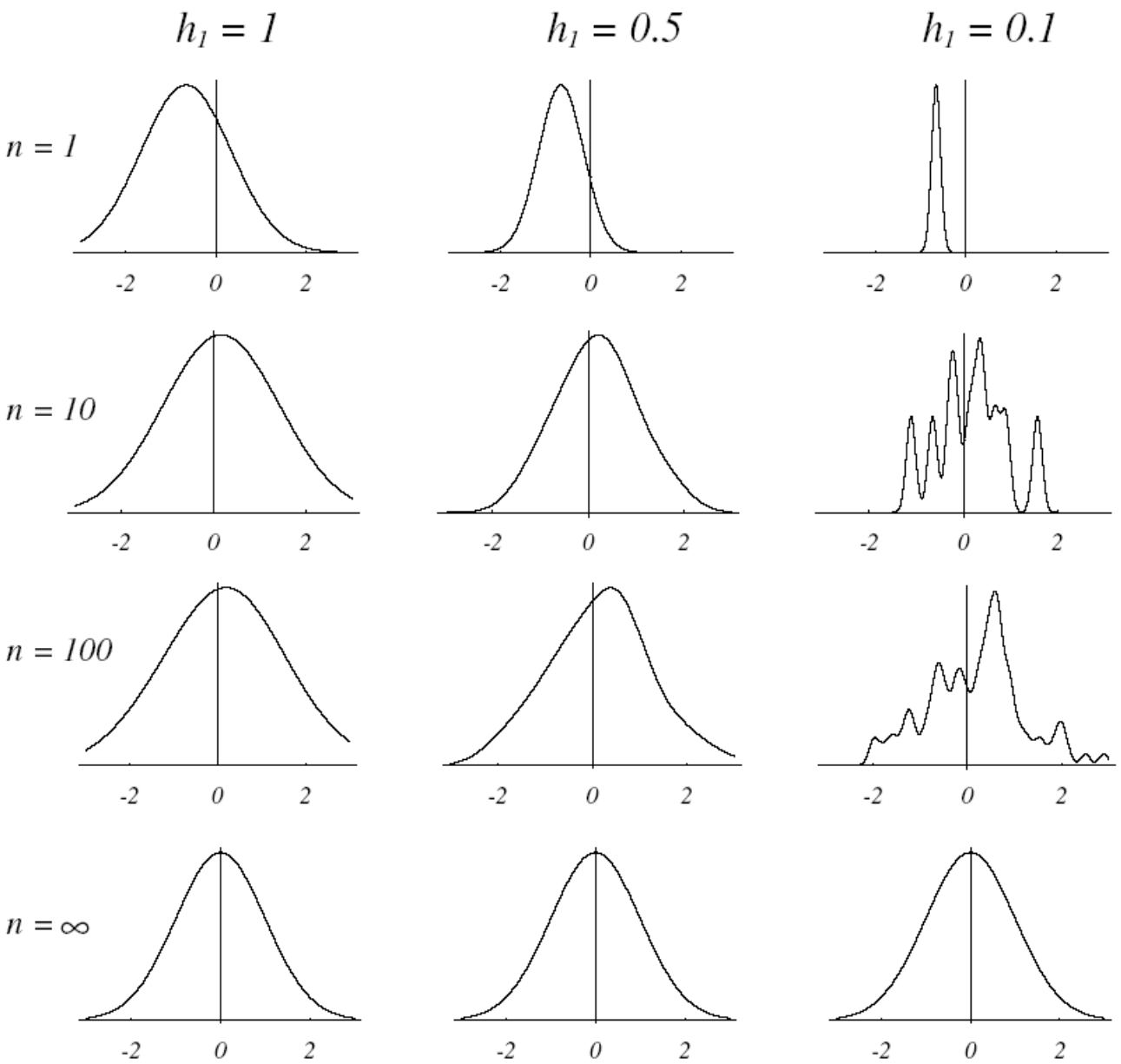
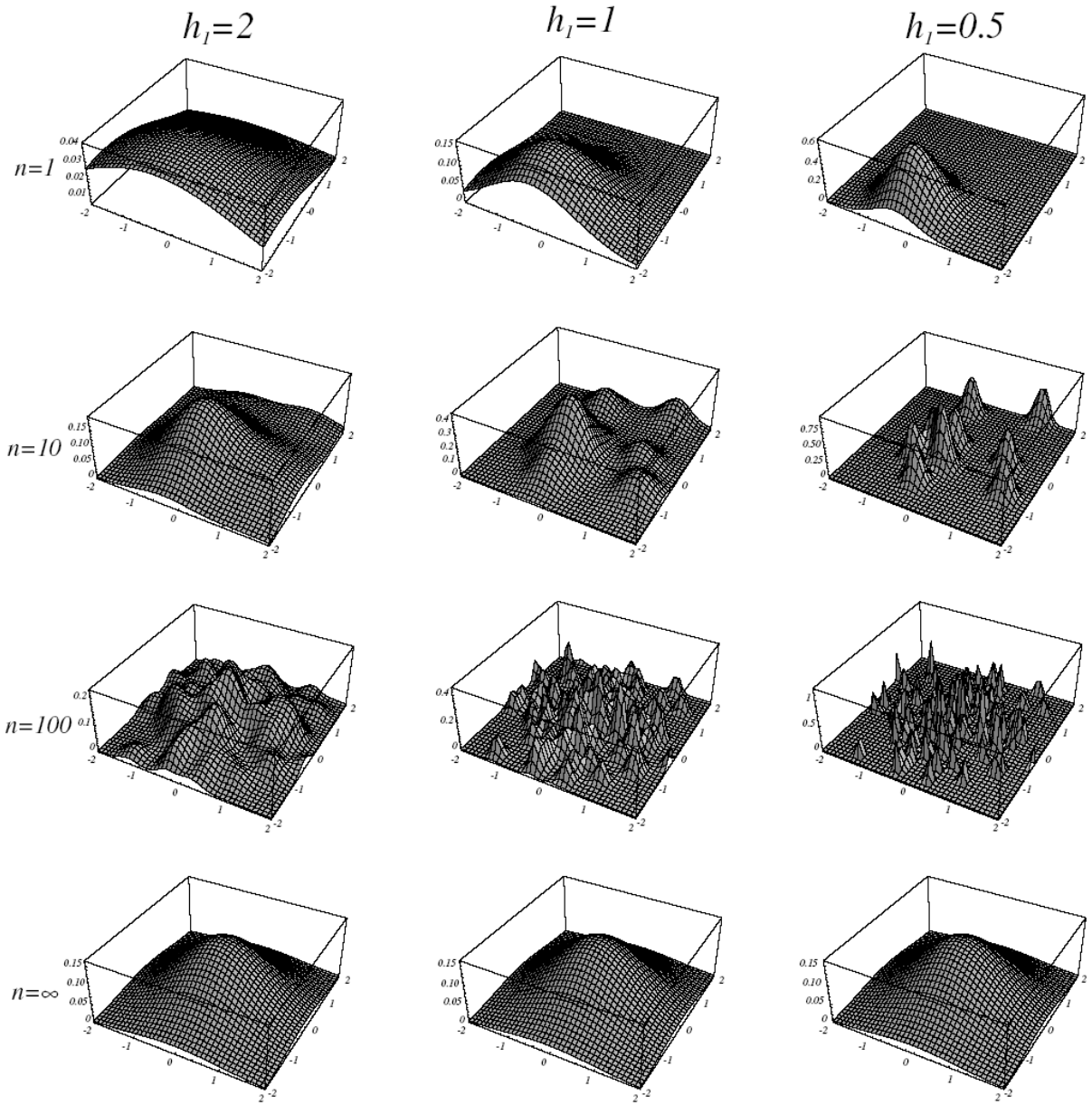


FIGURE 4.3. Examples of two-dimensional circularly symmetric normal Parzen windows for three different values of h. Note that because the $\delta(\mathbf{x})$ are normalized, different vertical scales must be used to show their structure. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Método da Janela de Parzen

- Consideramos a Janela de tamanho fixo centrada no ponto para o qual queremos estimar a densidade.
- Outra interpretação é somar todas as janelas centradas nas instâncias.


$h_1 = 1$
$h_1 = 0.5$
$h_1 = 0.1$
$n = 1$
$n = 10$
$n = 100$
$n = \infty$
-2
0
2


$h_1=2$
$h_1=1$
$h_1=0.5$
$n=1$
$n=10$
$n=100$
$n=\infty$

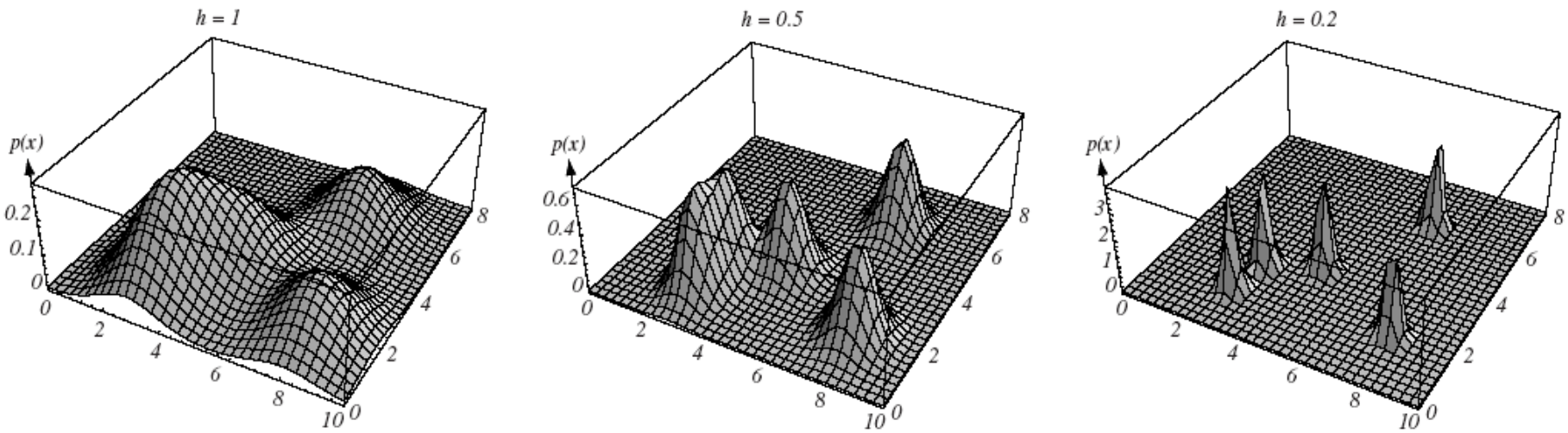
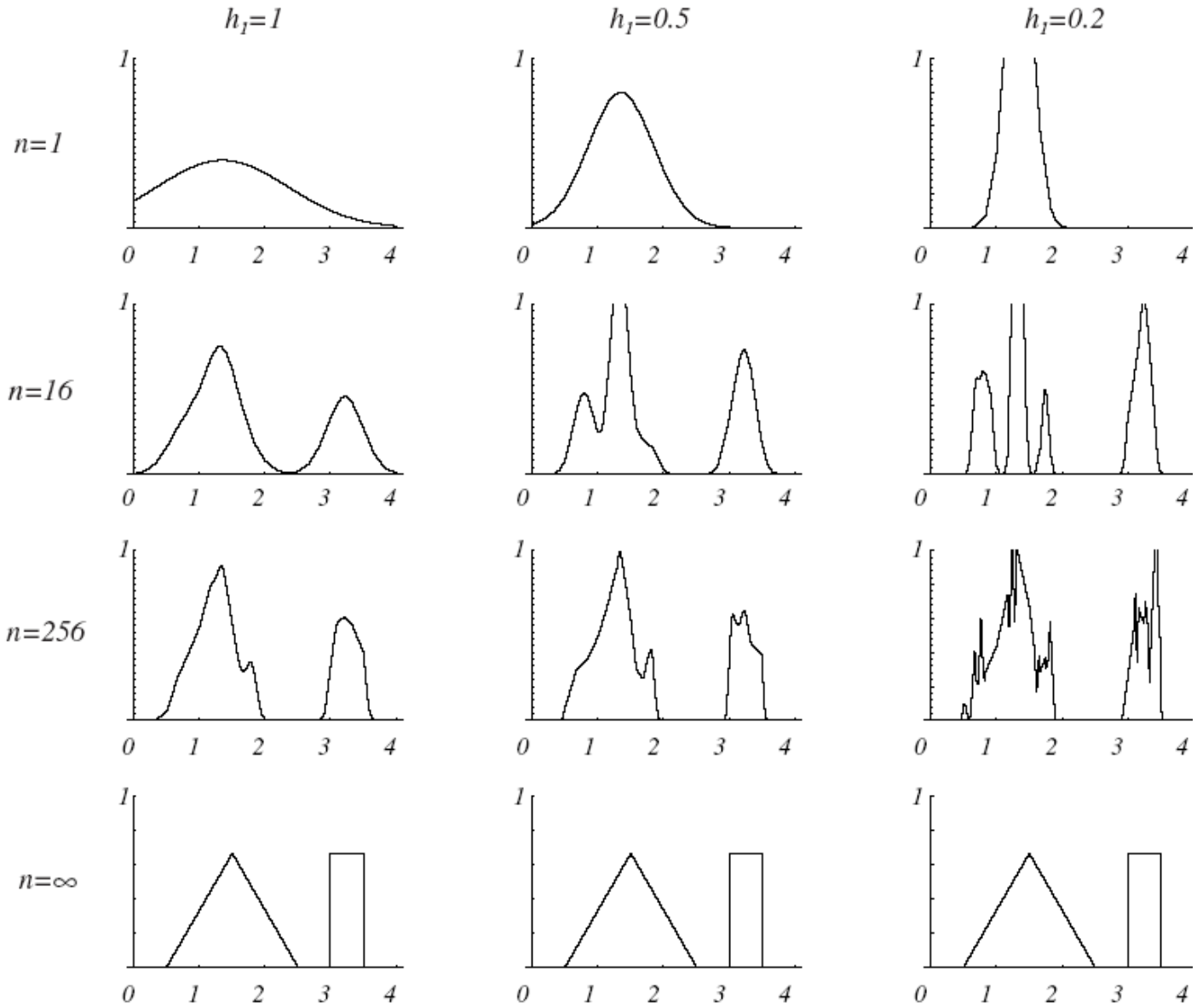
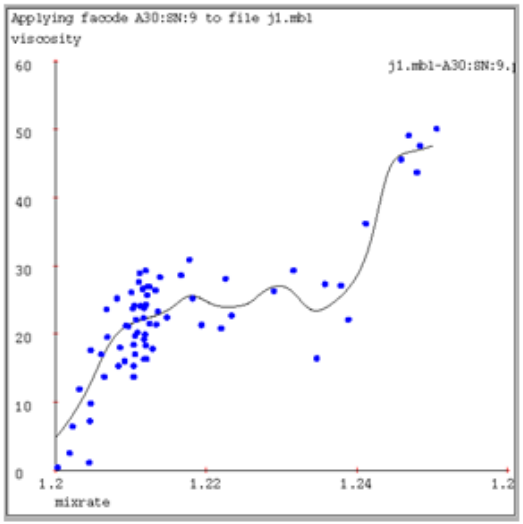
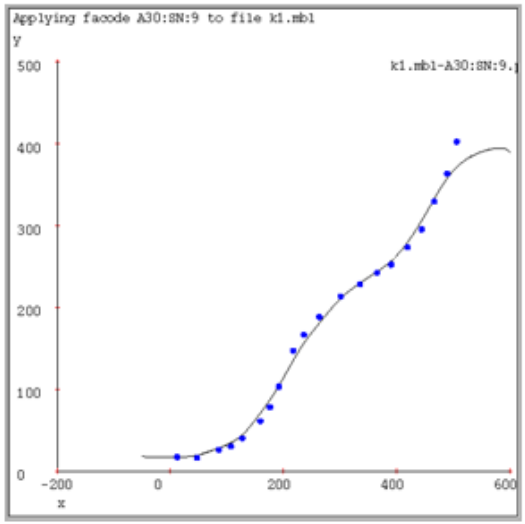


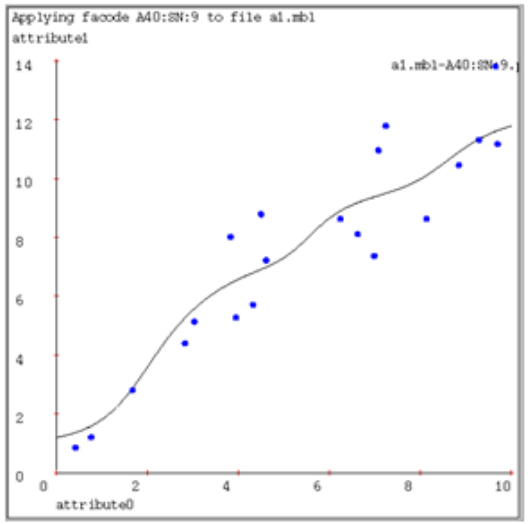
FIGURE 4.4. Three Parzen-window density estimates based on the same set of five samples, using the window functions in Fig. 4.3. As before, the vertical axes have been scaled to show the structure of each distribution. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.


h_1=1
h_1=0.5
h_1=0.2
n=1
n=16
n=256
n=∞
1
0 1 2 3 4

Kernel Regression on our test cases







KW=1/32 of x-axis width.

It's nice to see a smooth curve at last. But rather bumpy. If Kw gets any higher, the fit is poor.

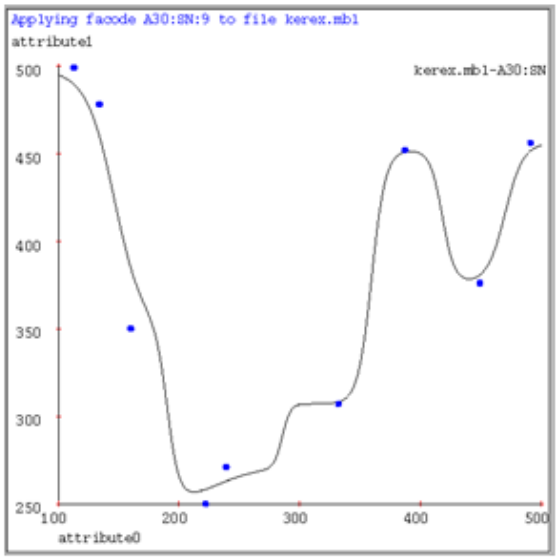
KW=1/32 of x-axis width.

Quite splendid. Well done, kernel regression. The author needed to choose the right K_W to achieve this.

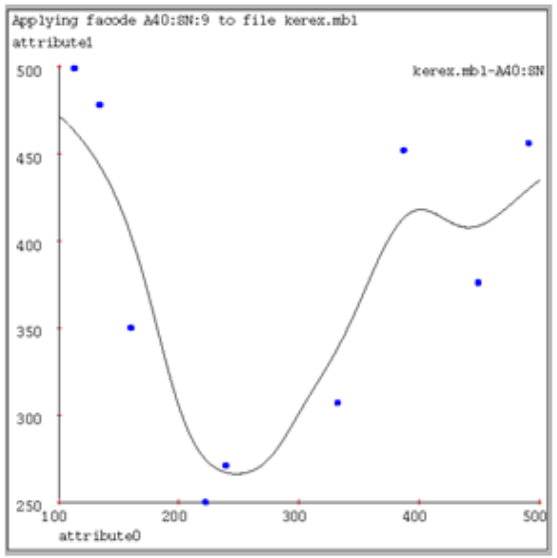
KW=1/16 axis width.

Nice and smooth, but are the bumps justified, or is this overfitting?

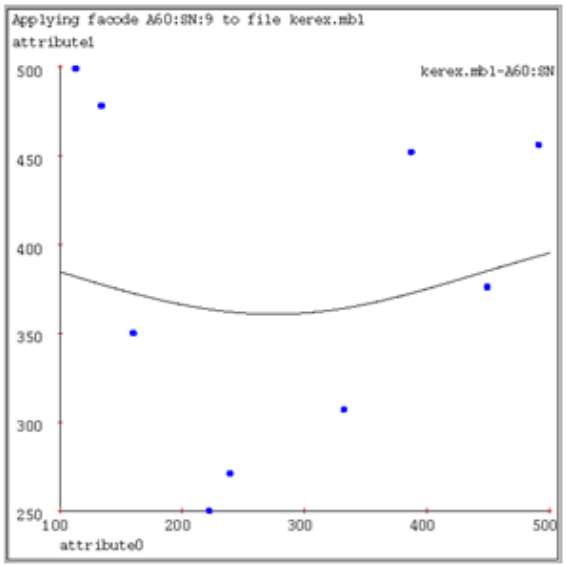
Kernel Regression Predictions



K_W=10

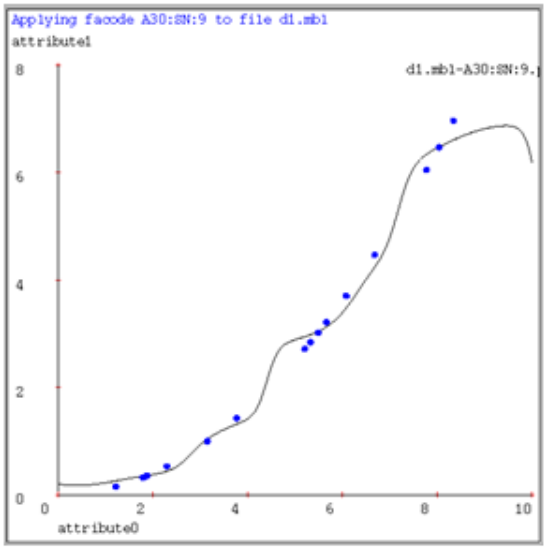


K_W=20



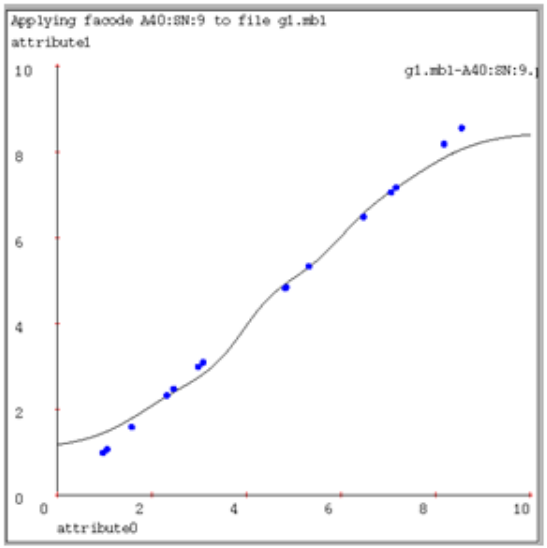
K_W=80

Kernel Regression can look bad



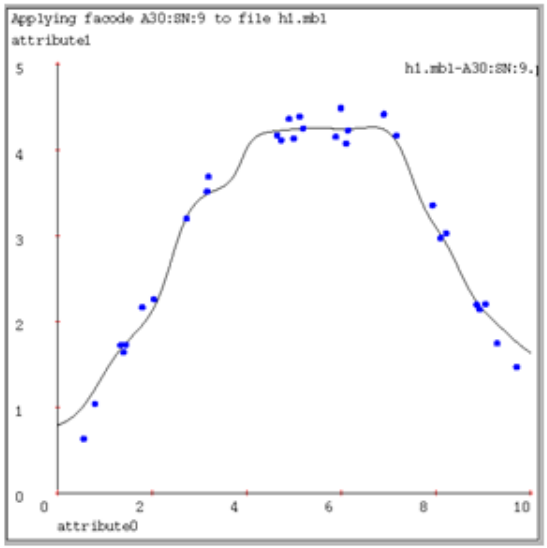
KW = Best.

Clearly not capturing the simple structure of the data.. Note the complete failure to extrapolate at edges.



KW = Best.

Also much too local. Why wouldn't increasing Kw help? Because then it would all be "smeared".

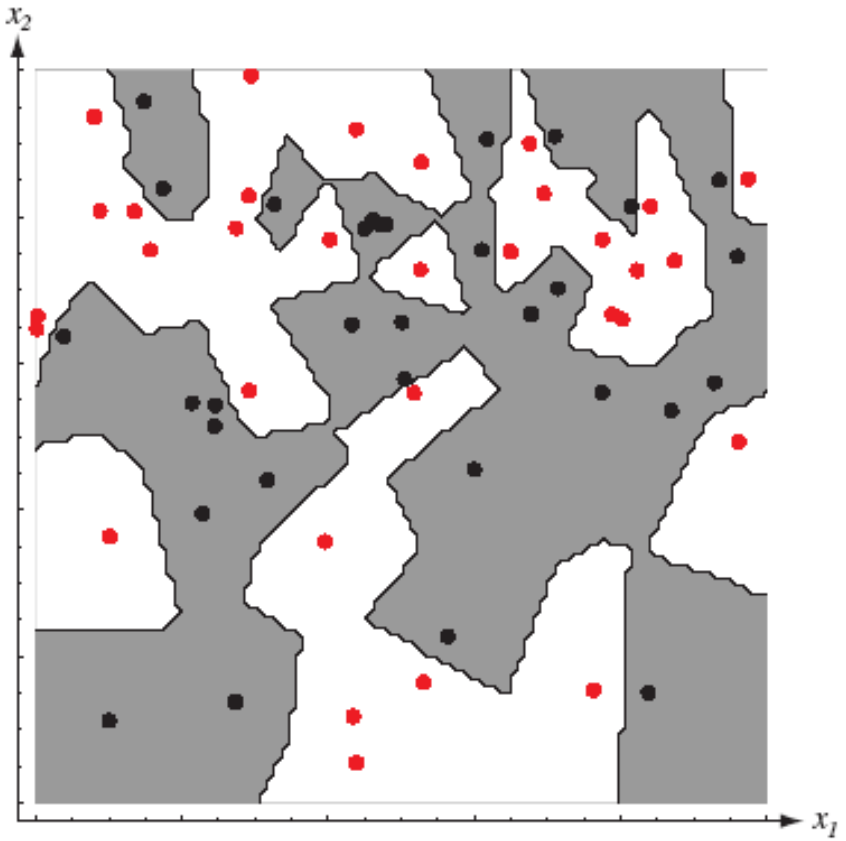


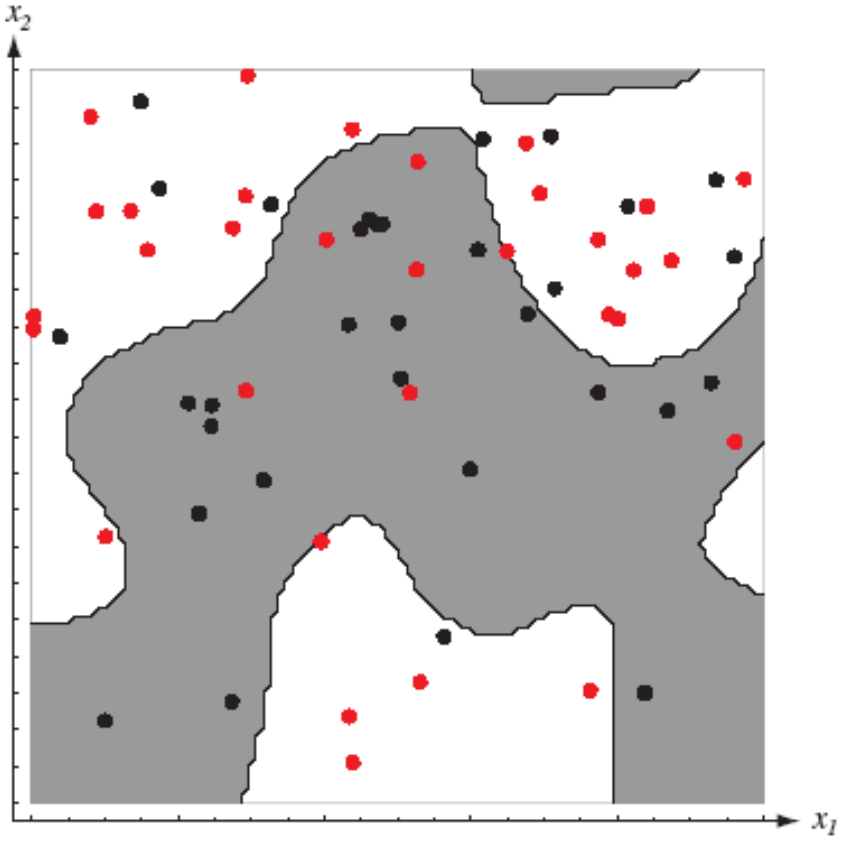
KW = Best.

Three noisy linear segments. But best kernel regression gives poor gradients.

Classificação pela Janela de Parzen

- No caso da função *gate*, podemos classificar um ponto de acordo com a maioria das amostras dentro da janela centrada no ponto a classificar.
- No caso suave, haverá uma ponderação da importância desses pontos (votação ponderada).


x_2
x_1

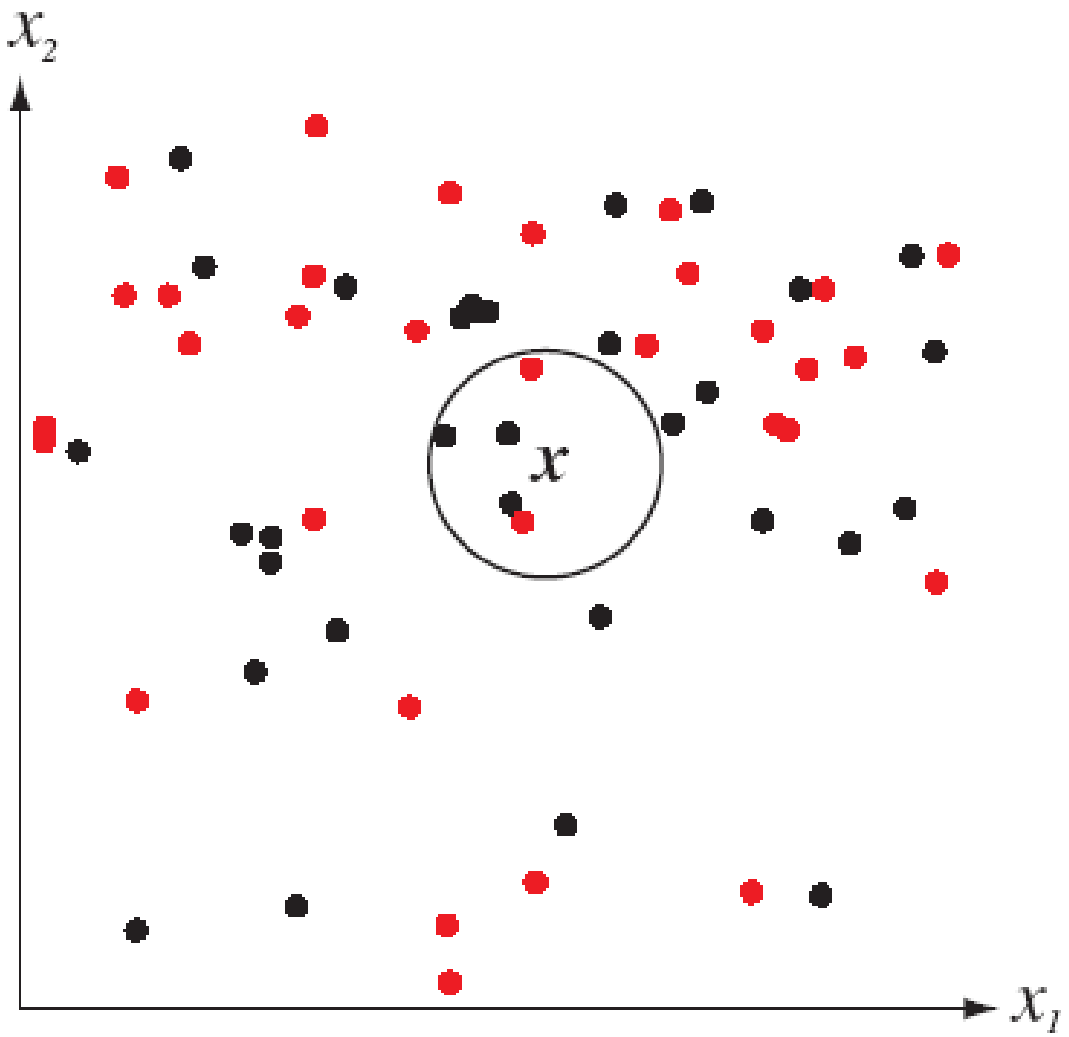

x_2
x_1

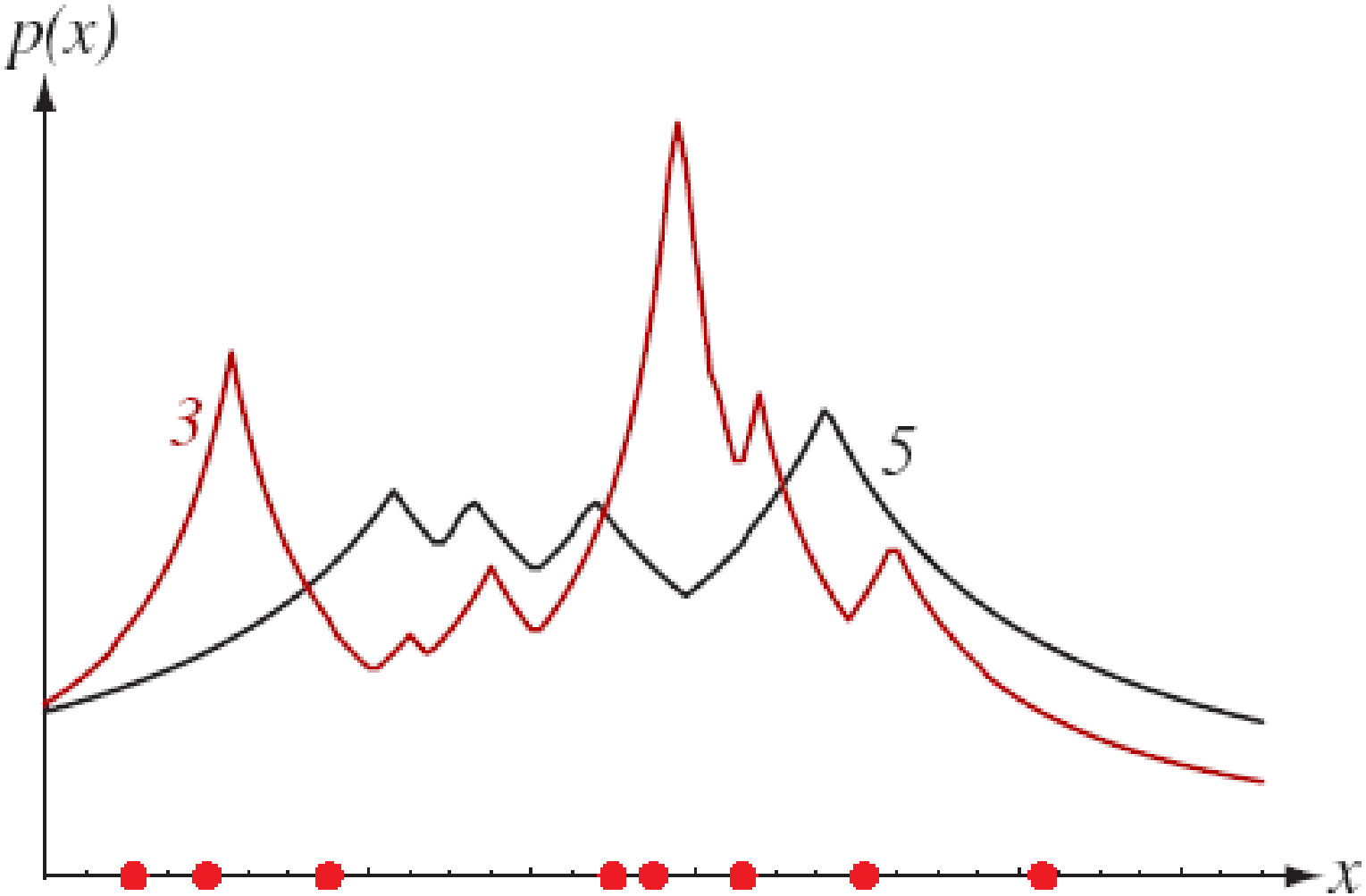
Problema com o método da Janela de Parzen

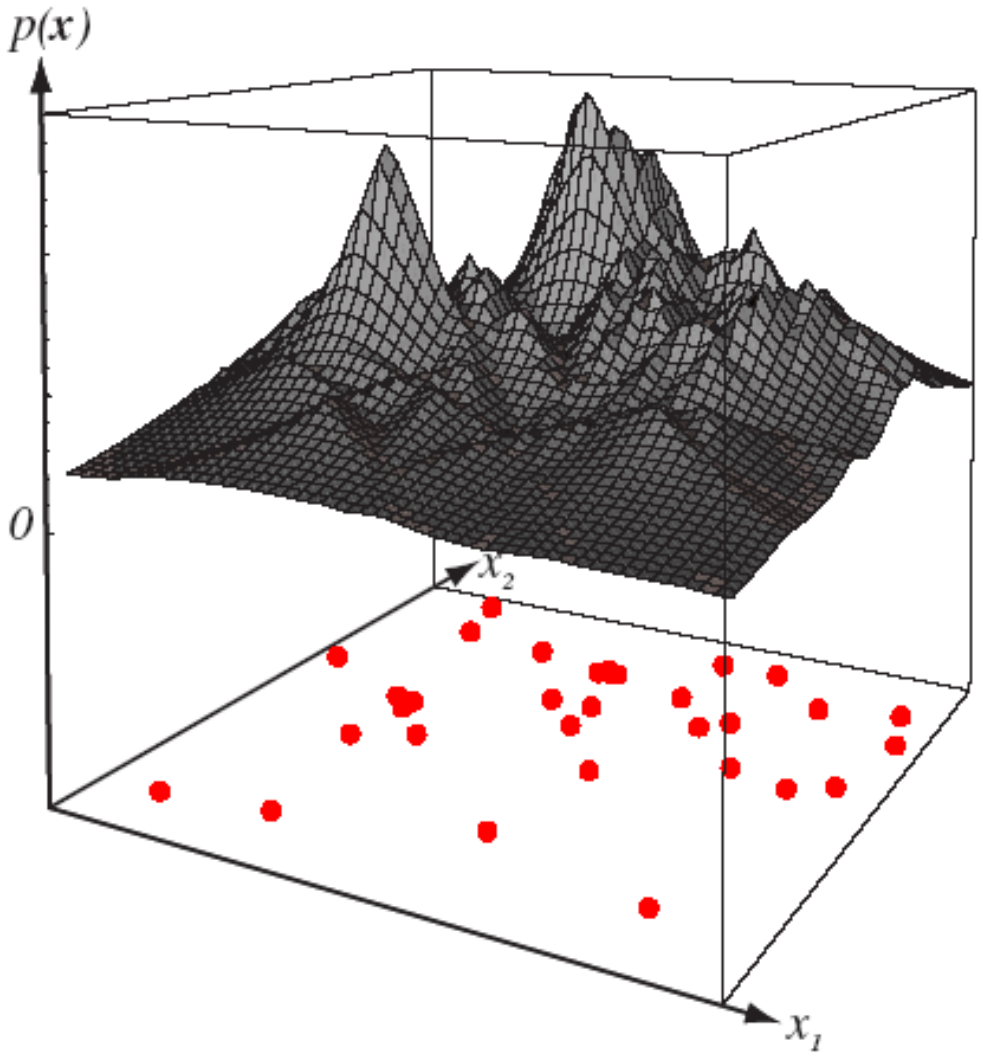
- Considere o problema de propor um tamanho para a janela.
- O que fazer se há regiões do espaço em que as amostras são muito esparsas ou inexistentes?
- Que tal utilizar um volume adaptativo?

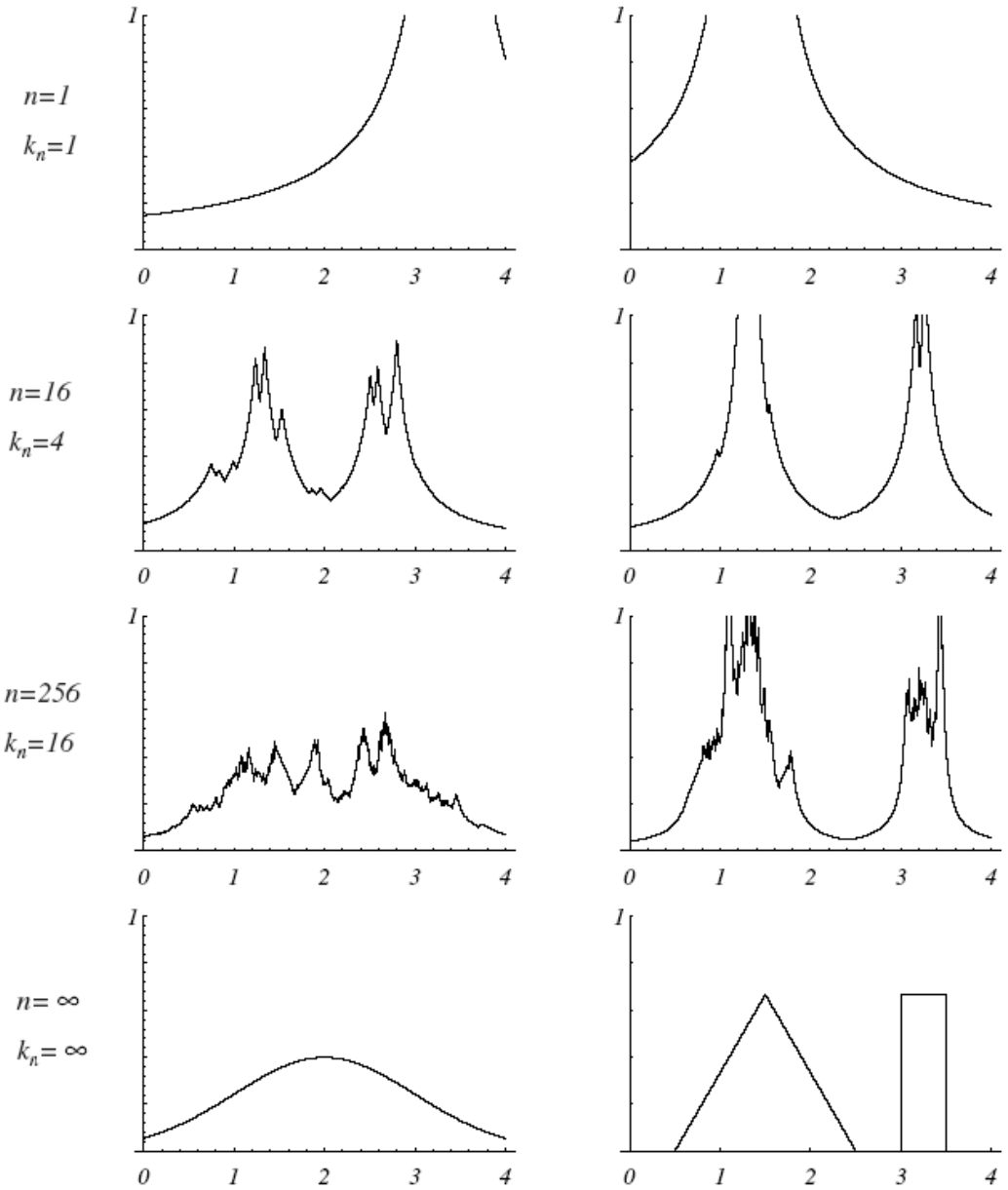
O Método dos Vizinhos mais Próximos

- Considere que podemos alterar o tamanho da janela.
- Fixamos então um número de vizinhos mais próximos em k e procuramos o menor volume centrado no ponto a se estimar a densidade que contém pelo menos k instâncias.


x_2
x
x_1


p(x)
3
5
x

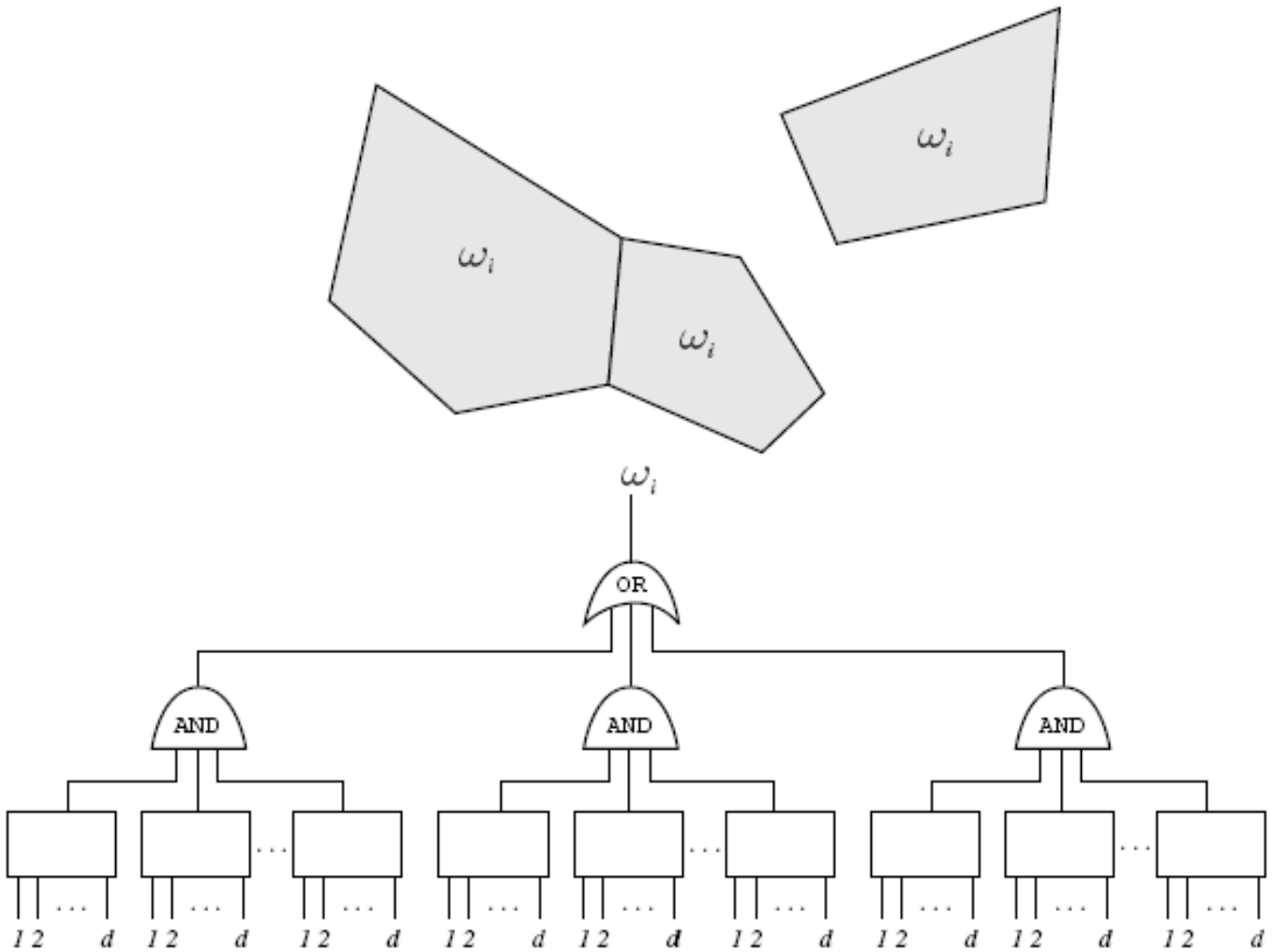

p(x)
0
x2
x1


n=1
k_n=1
n=16
k_n=4
n=256
k_n=16
n= ∞
k_n= ∞
1
0
1
2
3
4

Classificação pelos vizinhos mais próximos

Algoritmo:

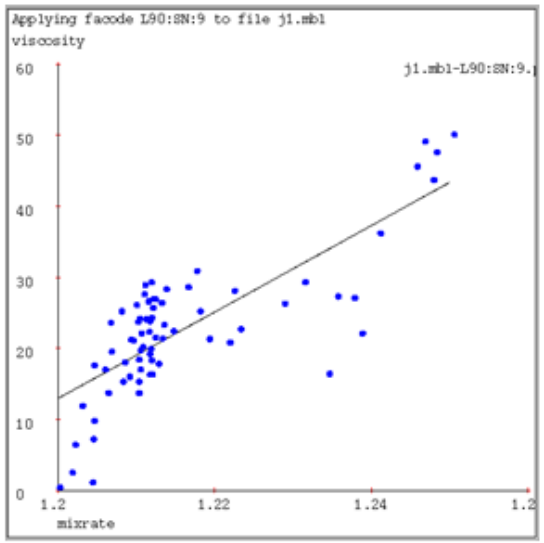
- Fixamos k.
- Centramos a janela no ponto x a ser classificado.
- Procuramos os k vizinhos mais próximos dada uma medida de distância.
- Se a maioria desses pontos pertence à classe C_i classifique x como C_i.


ω_i
ω_i
ω_i
ω_i
OR
AND
AND
AND
1 2 d
1 2 d
1 2 d
1 2 d
1 2 d
1 2 d
1 2 d
1 2 d
1 2 d

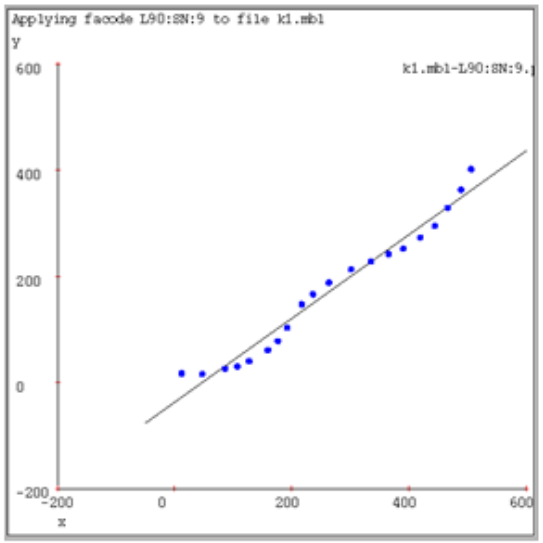
A Escolha de k

- Quanto maior k, mais pontos influenciam a regressão.
- Pode-se, portanto, esperar uma suavização da curva aproximadora.

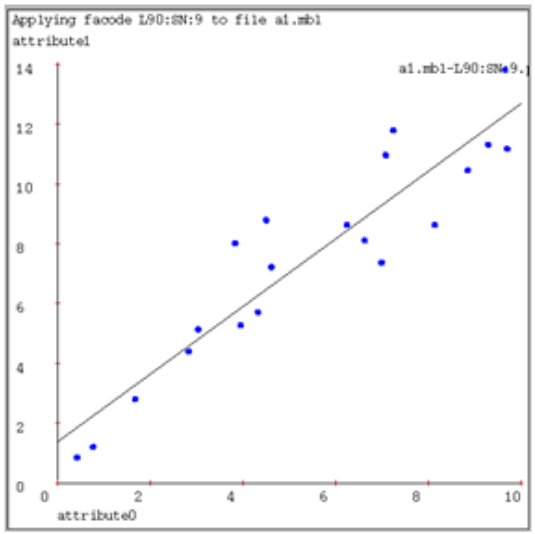
Why not just use Linear Regression?



Here, linear regression manages to capture a significant trend in the data, but there is visual evidence of bias.

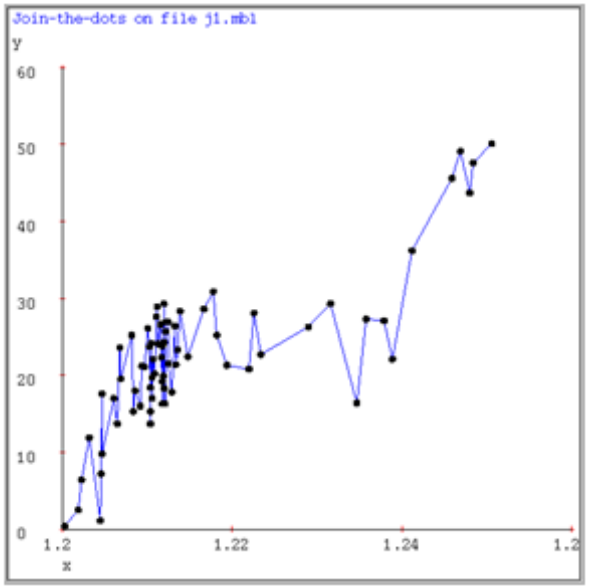


Here, linear regression appears to have a much better fit, but the bias is very clear.

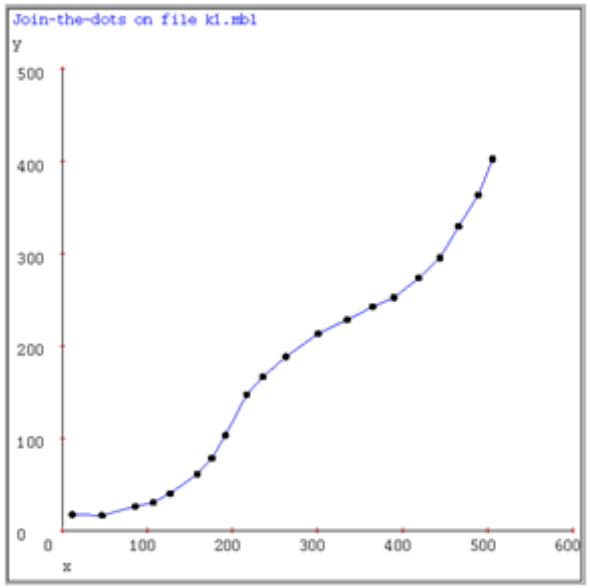


Here, linear regression may indeed be the right thing.

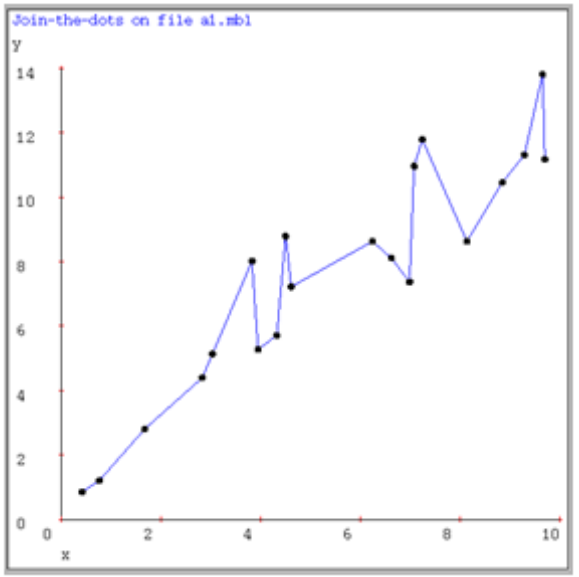
Why not just Join the Dots?



Here, joining the dots is clearly fitting noise.



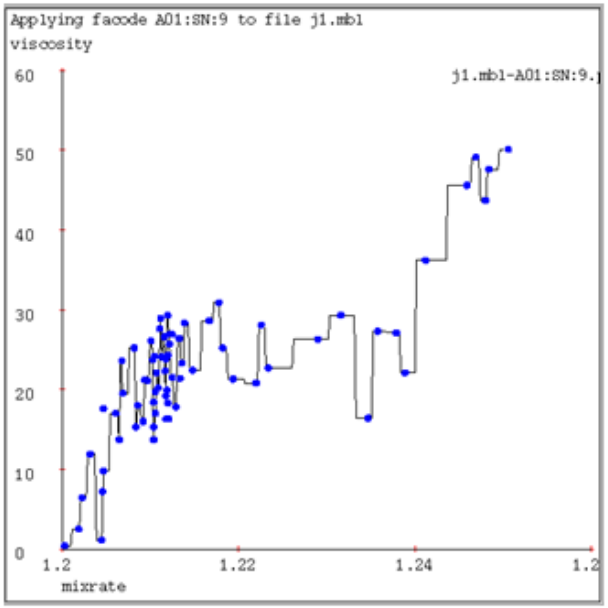
Here, joining the dots looks very sensible.

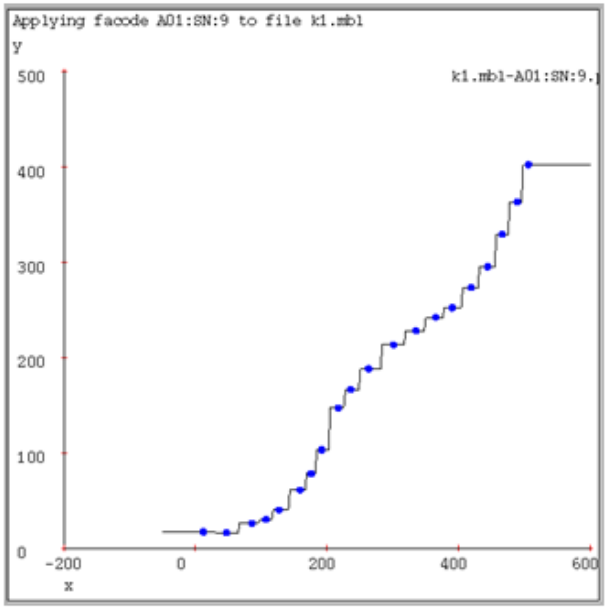


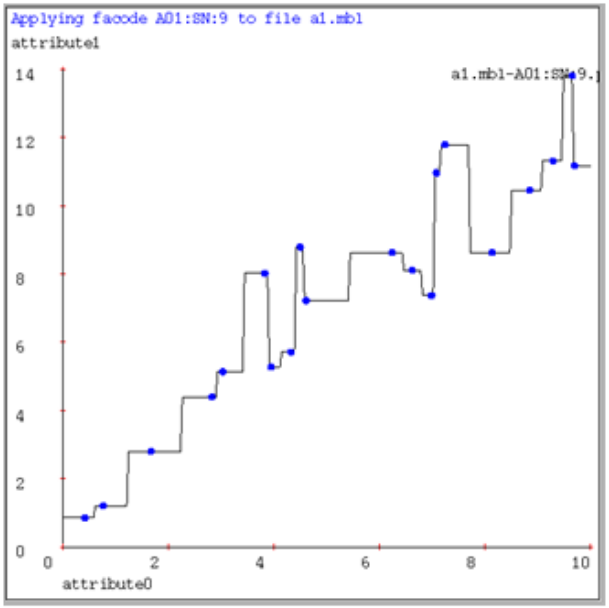
Again, a clear case of noise fitting.

One-Nearest Neighbor

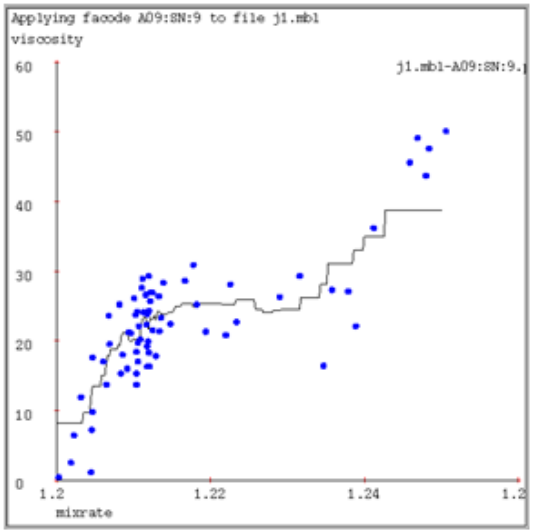
...One nearest neighbor for fitting is described shortly...



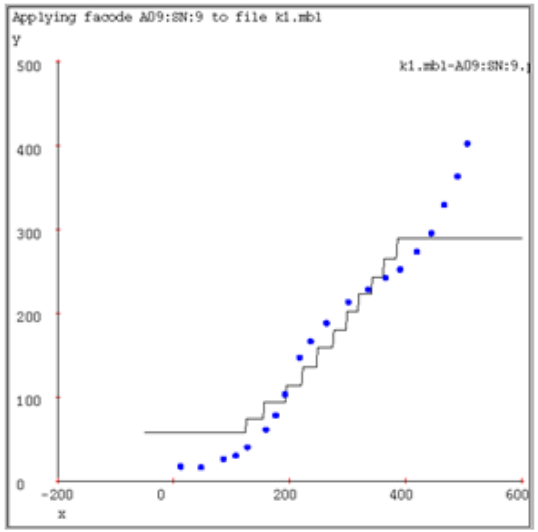




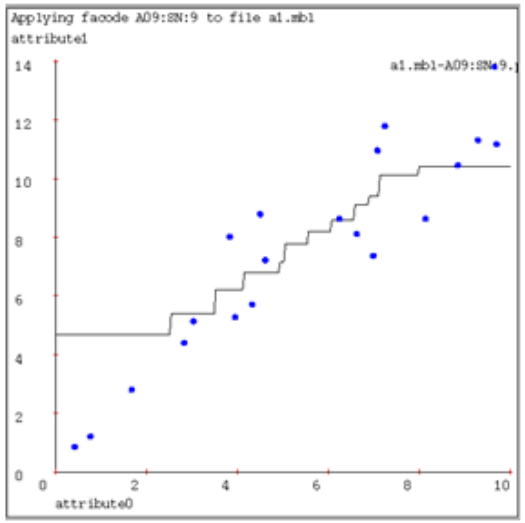
k-Nearest Neighbor (here k=9)



A magnificent job of noise-smoothing. Three cheers for 9-nearest-neighbor.
But the lack of gradients and the jerkiness isn't good.



Appalling behavior! Loses all the detail that join-the-dots and 1-nearest-neighbor gave us, yet smears the ends.



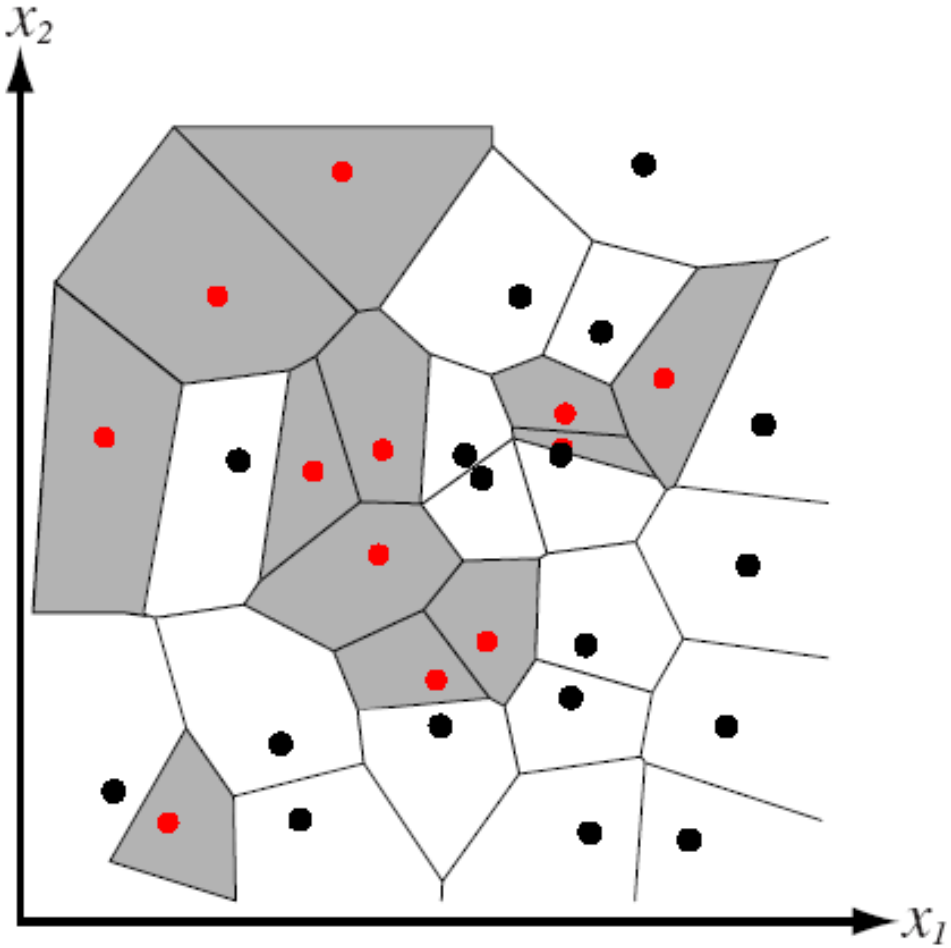
Fits much less of the noise, captures trends. But still, frankly, pathetic compared with linear regression.

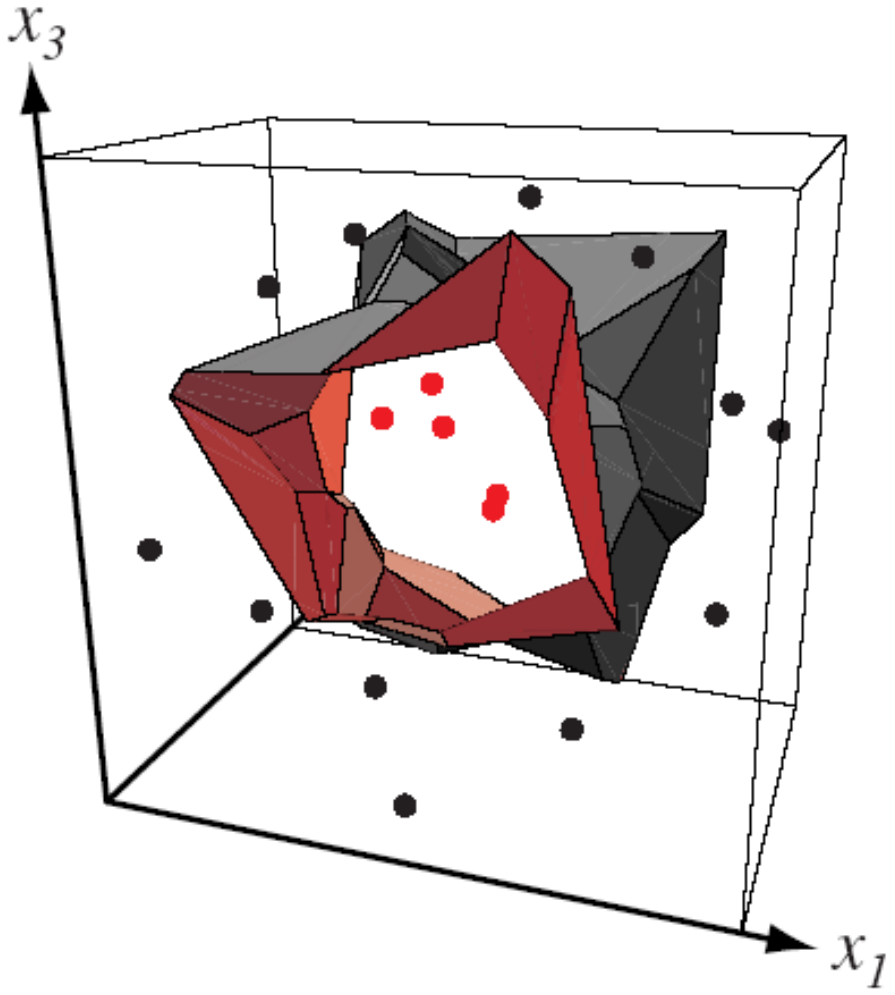
A Regra do vizinho mais próximo

- Quando fazemos $k = 1$ temos o método 1-NN que associa x à classe do elemento de treinamento mais próximo.

Mosaico de Voronoi

- O Diagrama de Voronoi contém os pontos de igual menor distância a um conjunto de pontos.
- Cada célula do diagrama corresponde a um dos pontos e representa a vizinhança desse ponto que classificaria qualquer elemento da célula como da mesma classe segundo a regra do vizinho mais próximo.


x_2
x_1


x_3
x_1

Escolha da função distância

- Propriedades de invariância:
- translações
- escala

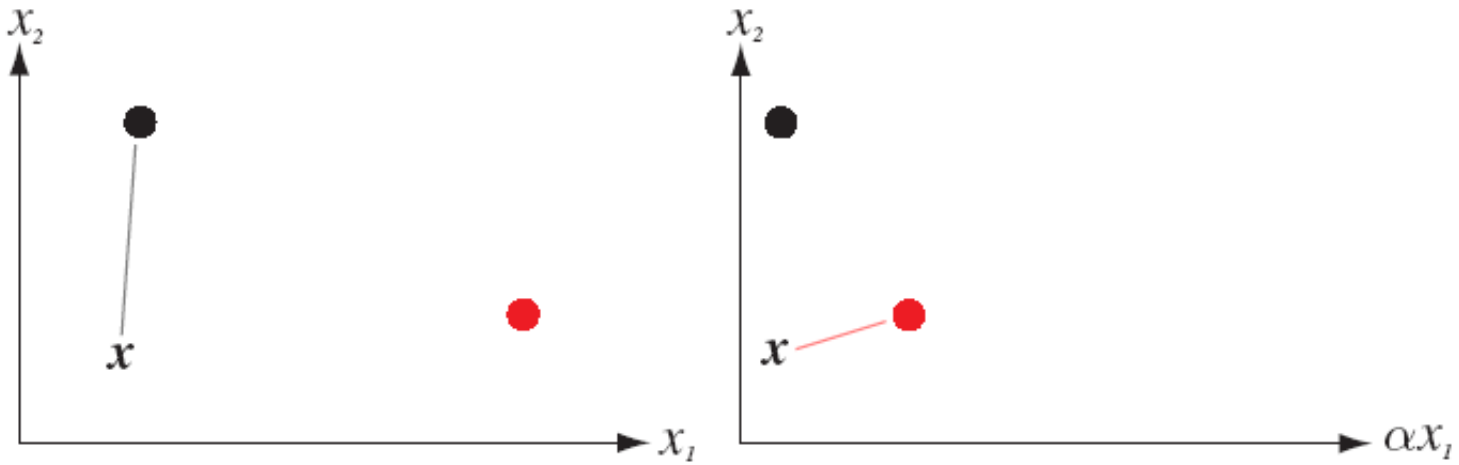
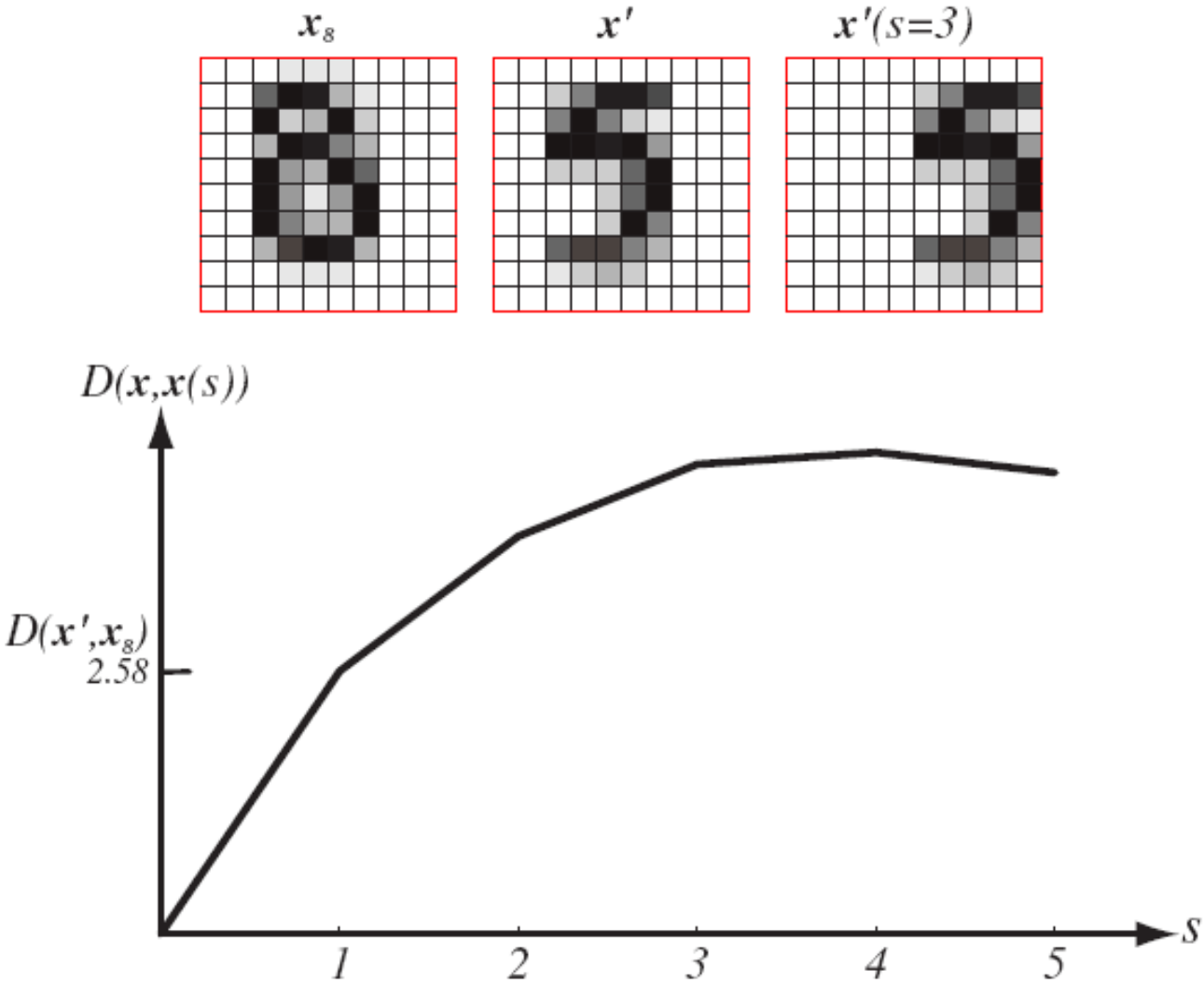
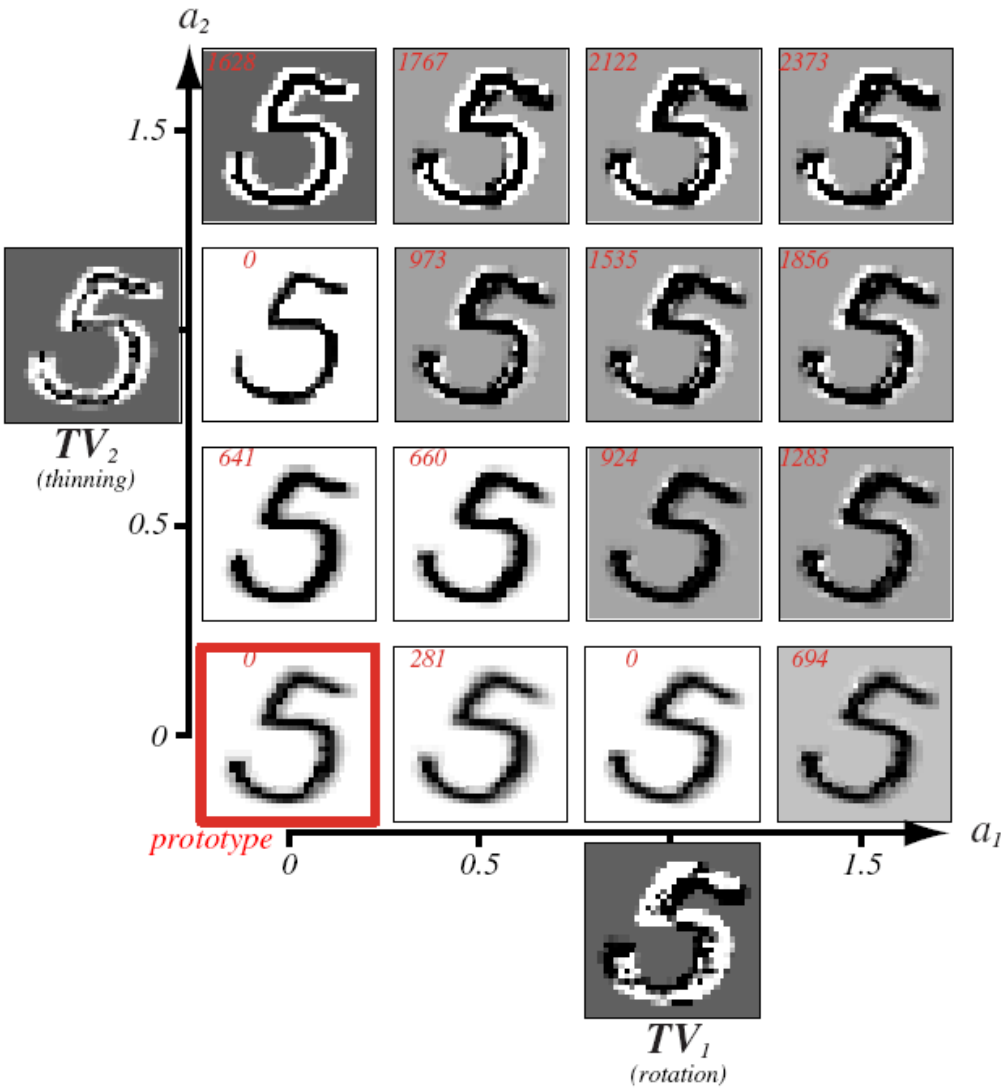


FIGURE 4.18. Scaling the coordinates of a feature space can change the distance relationships computed by the Euclidean metric. Here we see how such scaling can change the behavior of a nearest-neighbor classifer. Consider the test point **x** and its nearest neighbor. In the original space (left), the black prototype is closest. In the figure at the right, the x_1 axis has been rescaled by a factor 1/3; now the nearest prototype is the red one. If there is a large disparity in the ranges of the full data in each dimension, a common procedure is to rescale all the data to equalize such ranges, and this is equivalent to changing the metric in the original space. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.


x_8
x'
$x'(s=3)$
$D(x,x(s))$
$D(x',x_8)$
2.58
1
2
3
4
5
s


a_2
1.5
0.5
0
1628
1767
2122
2373
0
973
1535
1856
641
660
924
1283
0
281
0
694
TV_2
(thinning)
prototype
0
0.5
1.5
a_1
TV_1
(rotation)

Lazy learning

- Lazy Learning ou Aprendizado baseado em Instâncias ou baseado em Memória
- Como é chamado o aprendizado sem treinamento.
- É necessário armazenar todas as instâncias e buscá-las no momento da classificação.
- O fato de não haver treinamento onera a fase de classificação.
- Estruturas de dados facilitam a recuperação dos vizinhos mais próximos a um dado ponto.

Regressão Localmente Ponderada

- Função objetivo da regressão

$$E = \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2$$

- Minimizar o erro quadrático apenas para os k vizinhos mais próximos.

$$E(x_q) = \frac{1}{2} \sum_{x \in k \text{ nearest nbrs of } x_q} (f(x) - \hat{f}(x))^2$$

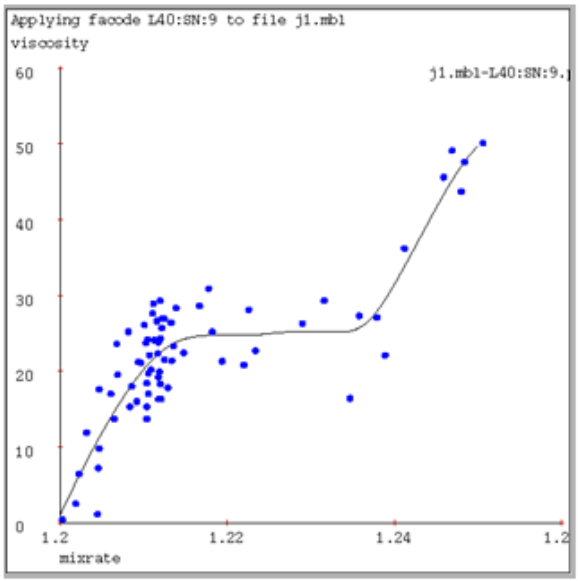
- Ou minimizar o erro quadrático ponderado por uma função da distância.

$$E(x_q) = \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2 K(d(x_q, x))$$

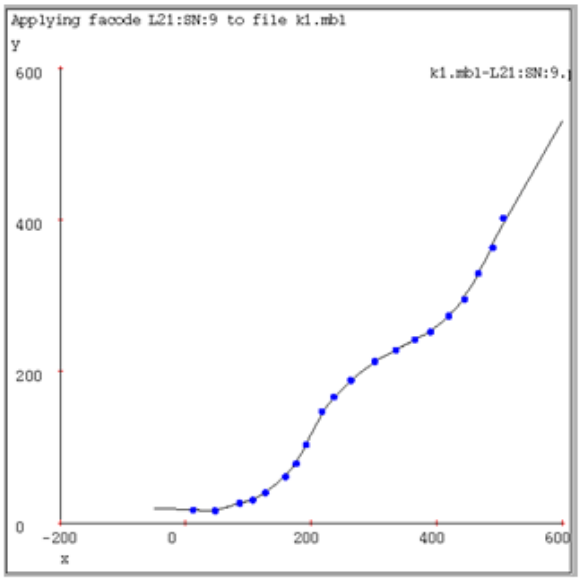
Para determinar uma aproximação $\hat{y}$ para o ponto x_q:

1. Obtenho N instâncias na forma (y_i, x_i);
2. Utilizo $K(d(x_q, x))$ para determinar pesos para cada instância i, baseados na distância d ao ponto x_q;
3. Obtenho os parâmetros β de regressão ponderada utilizando os pesos calculados;
4. Obtenho a estimativa $\hat{y}$.
5. Para outro ponto x_p, refaço do passo 2 em diante.

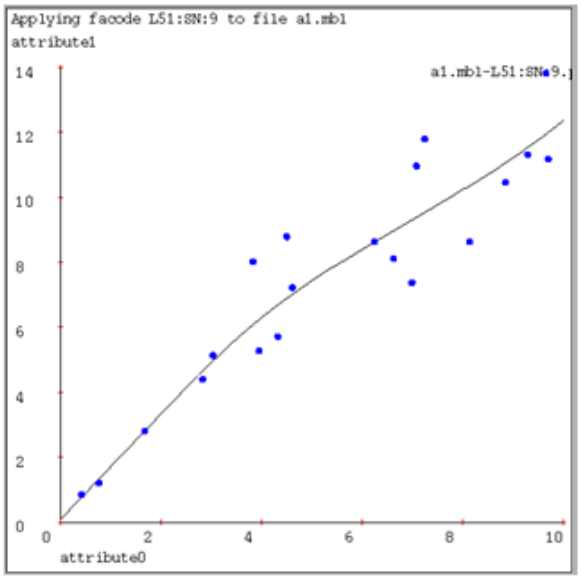
LWR on our test cases



KW = 1/16 of x-axis width.

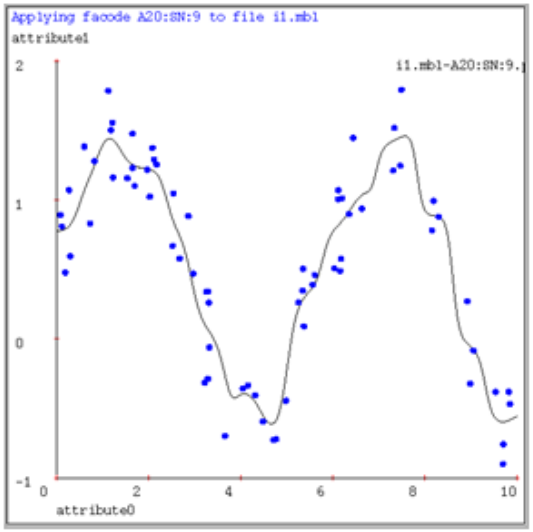


KW = 1/32 of x-axis width.



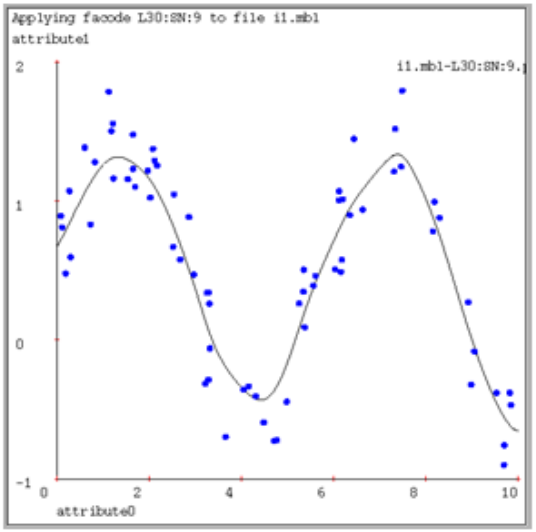
KW = 1/8 of x-axis width.

Locally weighted Polynomial regression



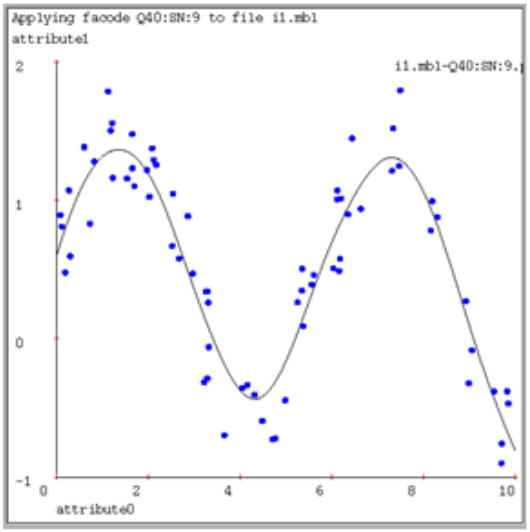
Kernel Regression
Kernel width K_W at optimal level.

KW = 1/100 x-axis



LW Linear Regression
Kernel width K_W at optimal level.

KW = 1/40 x-axis



LW Quadratic Regression
Kernel width K_W at optimal level.

KW = 1/15 x-axis

Agradecimentos

- Agradeço ao Prof. Andrew Moore por deixar utilizar seus slides (a maioria dos slides que estão em inglês)
- Outros slides em inglês são do livro Duda, Hart e Stork.