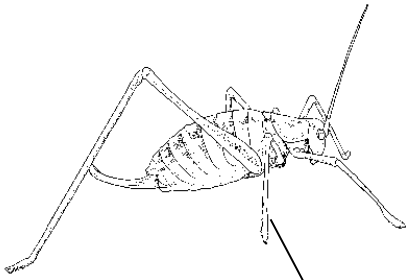
Mineração de Dados
2017.2

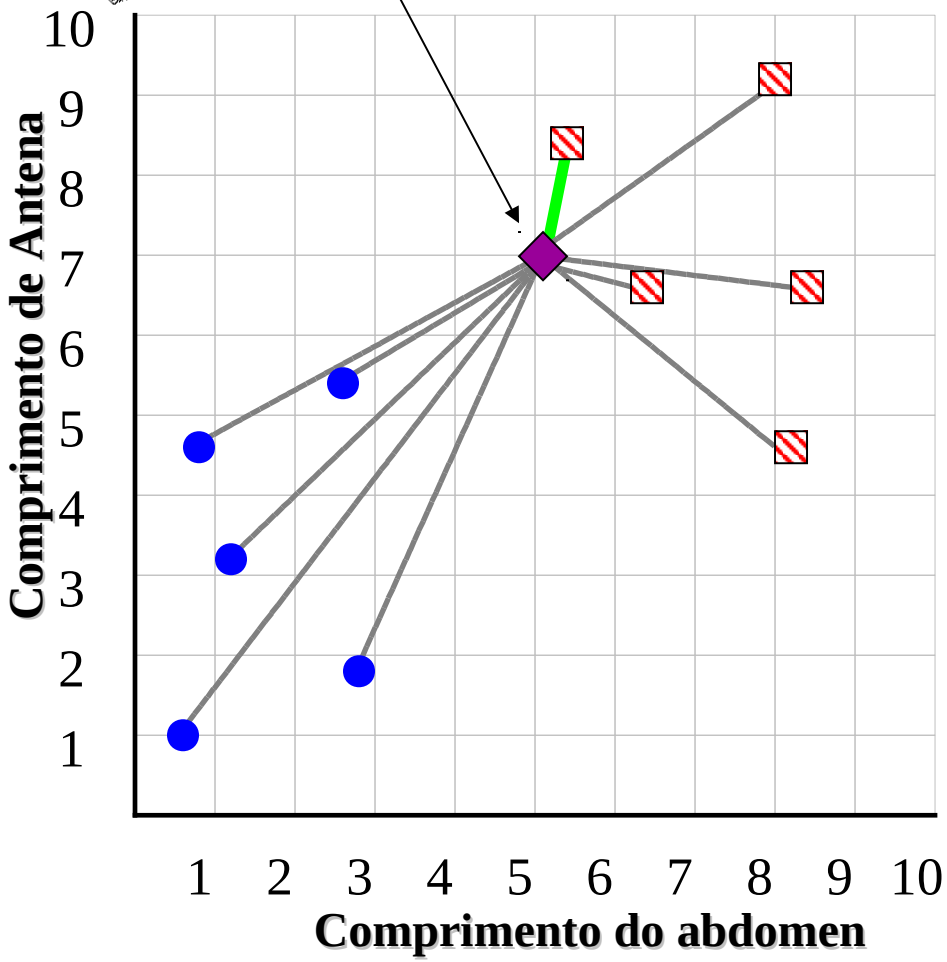
Classificador Vizinhos Mais próximos

Thiago Ferreira Covões

(slides baseados no material do Prof. Eamonn Keogh [eamonn@cs.ucr.edu])

Classificador Vizinho Mais Próximo





Evelyn Fix
1904-1965

Joe Hodges
1922-2000

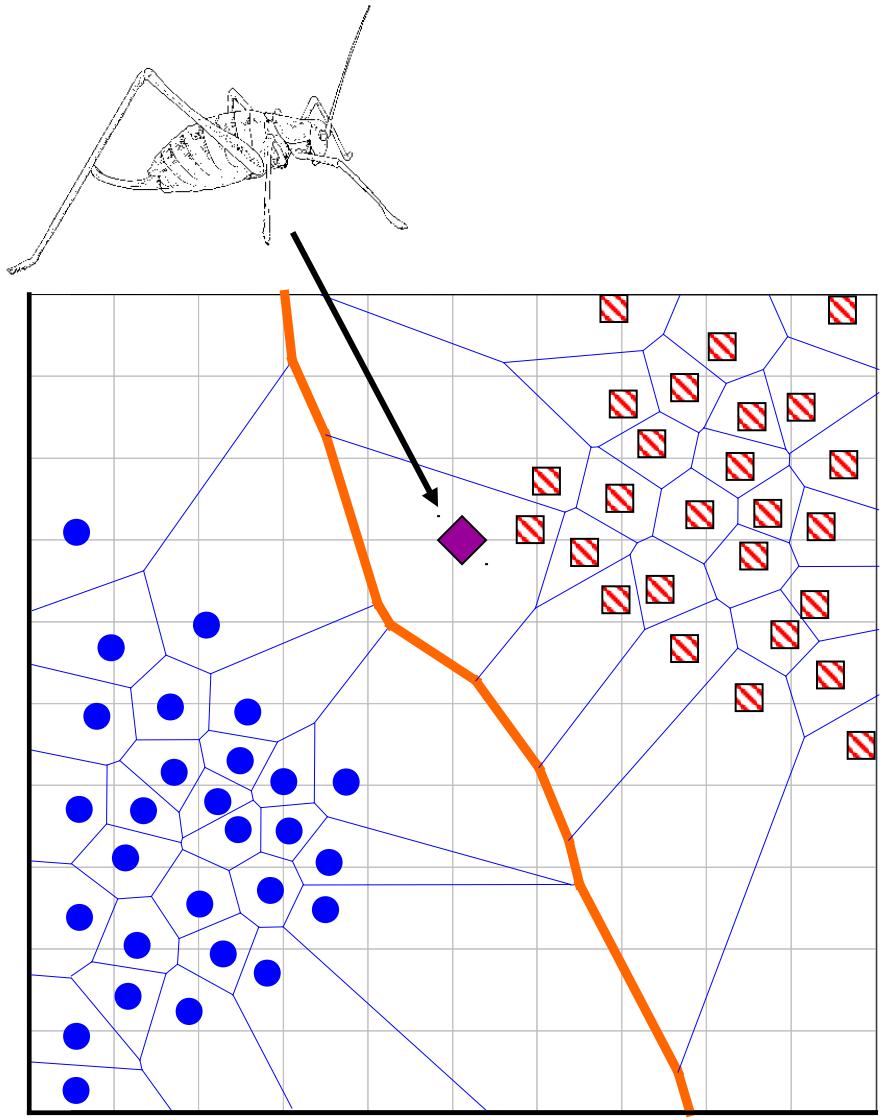
Se o exemplo **mais próximo** de um **exemplo não visto antes é uma Esperança**
a classe é **Esperança**
Senão
a classe é **Gafanhoto**

Esperança
Gafanhotos

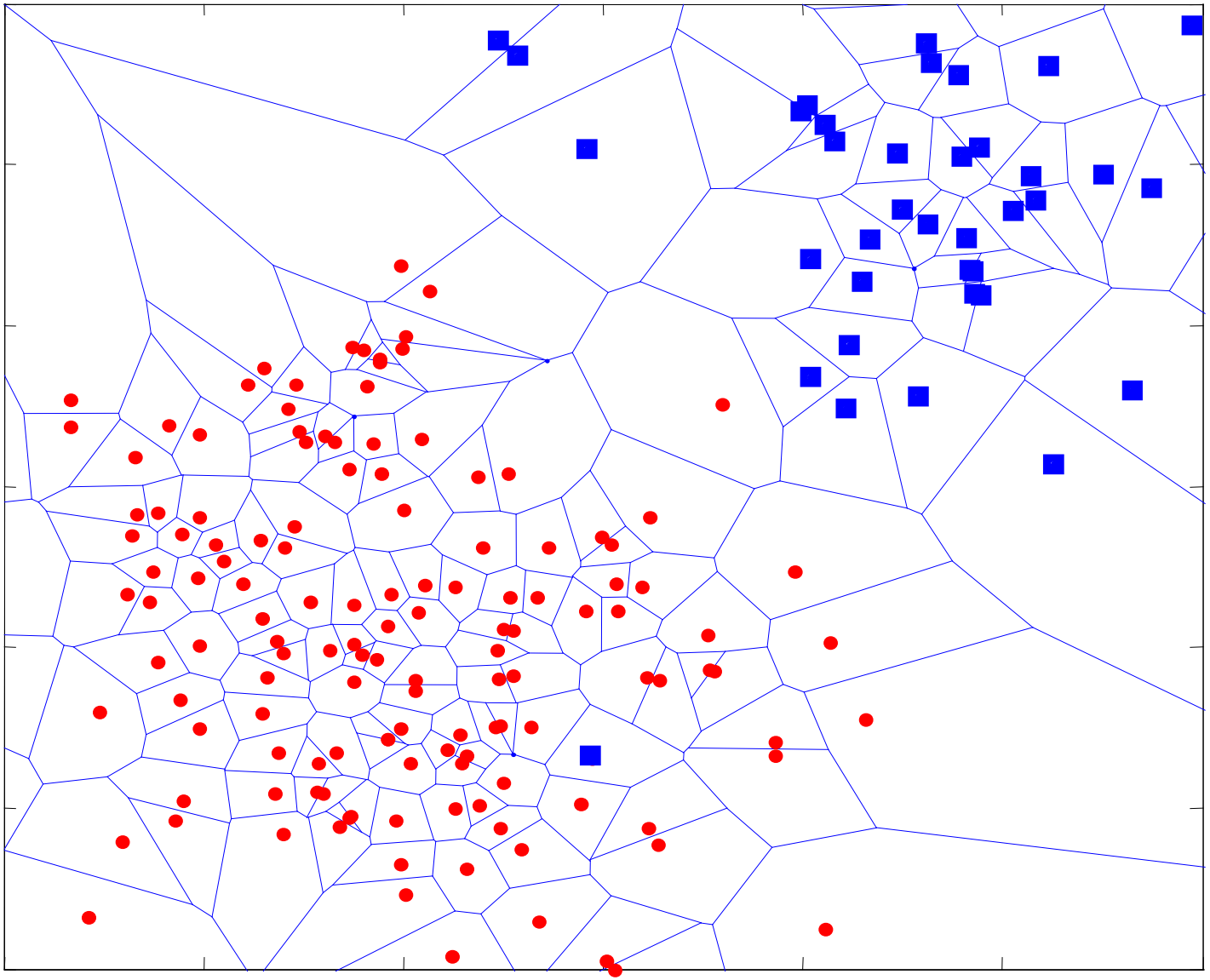
Podemos visualizar o algoritmo do vizinho mais próximo em termos de uma superfície de decisão…

Note que não precisamos realmente construir essas superfícies, elas são simplesmente os limites implícitos que dividem o espaço em regiões que "pertencem" a cada exemplo.

Esta divisão de espaço é chamada de Dirichlet Tessellation (ou diagrama de Voronoi, ou regiões Theissen).

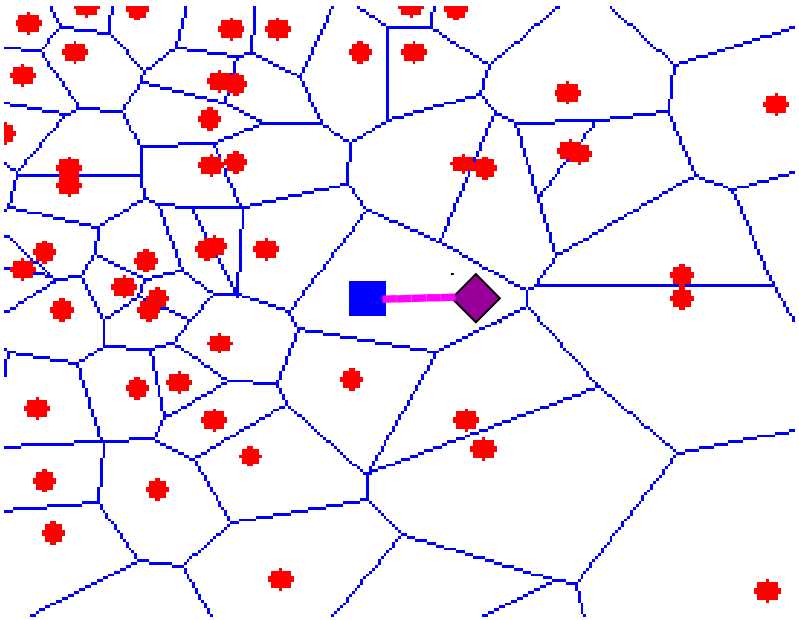


O alg. do vizinho mais próximo é sensível a “exceções”…

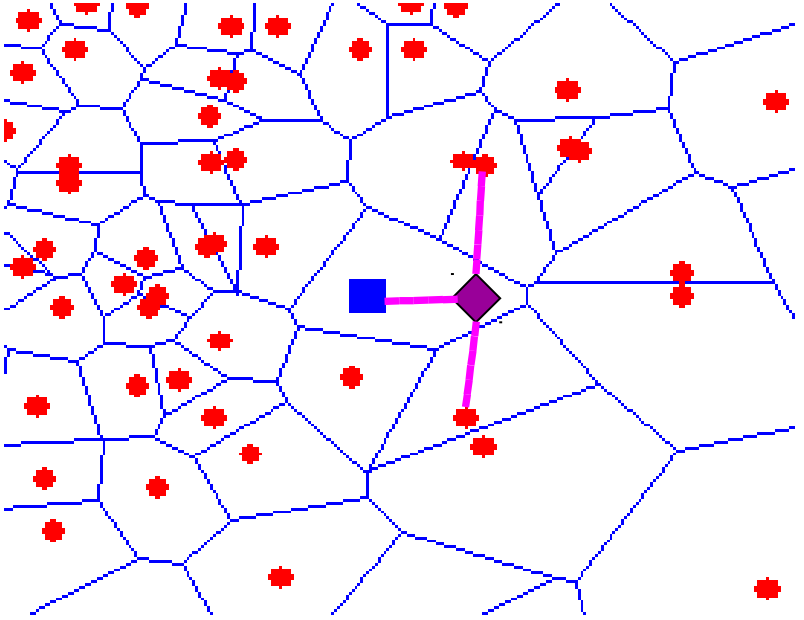


A solução é…

Podemos generalizar o algoritmo do vizinho mais próximo para o algoritmo do k-vizinhos mais próximos (KNN). Medimos a distância até os *k* exemplos mais próximos e as deixamos votar. *k* é tipicamente escolhido como um número ímpar.



k = 1

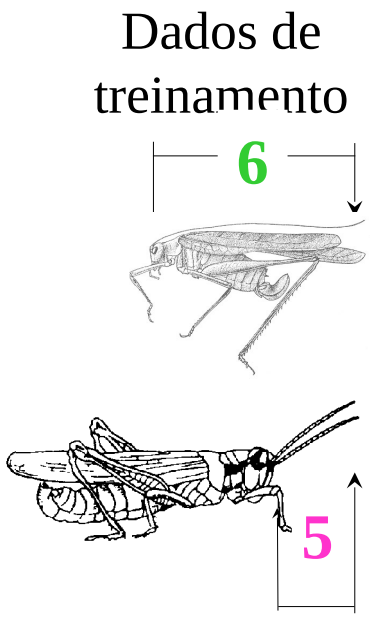


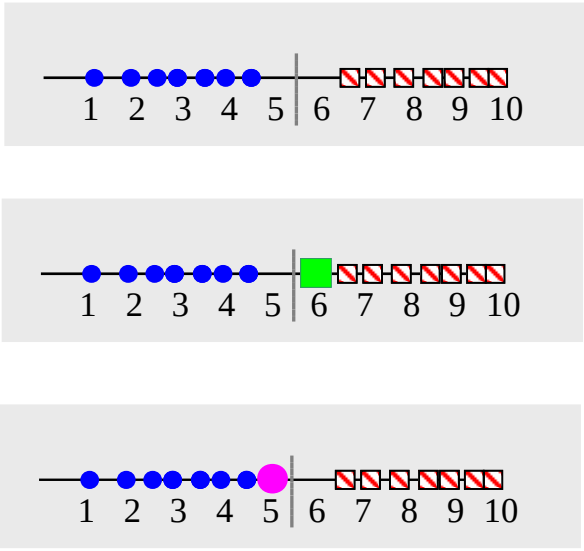
k = 3

O algoritmo do vizinho mais próximo é sensível a características irrelevantes…

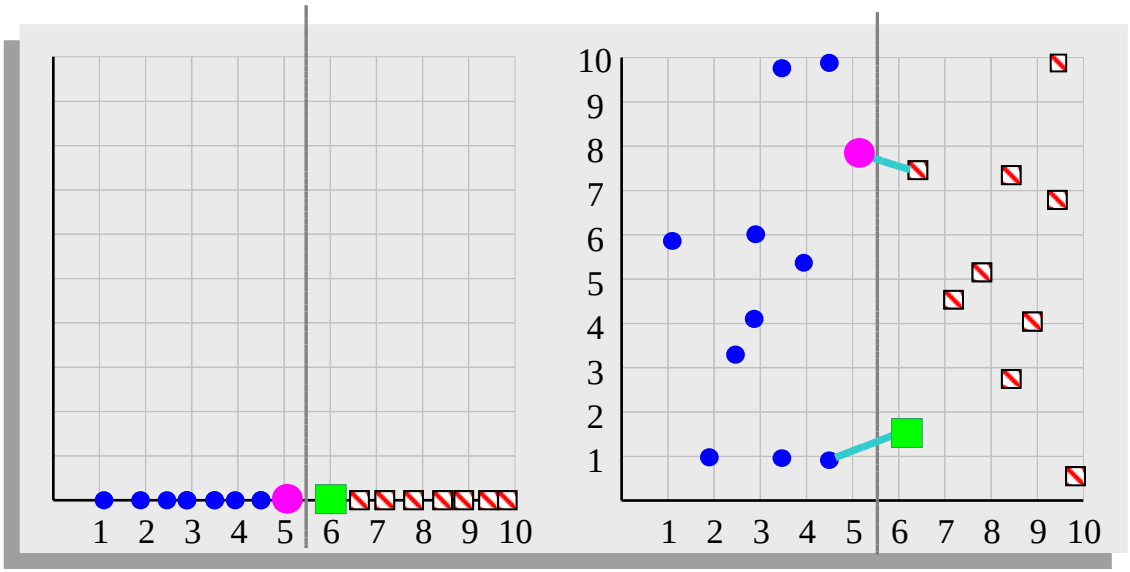
Suponha que o seguinte é verdadeiro, se a antena de um inseto é maior que 5.5 ele é um **Esperança**, senão ele é um **Gafanhoto**.

Usando somente o comprimento de antena conseguimos classificação perfeita!





O algoritmo do vizinho mais próximo é sensível a características irrelevantes…



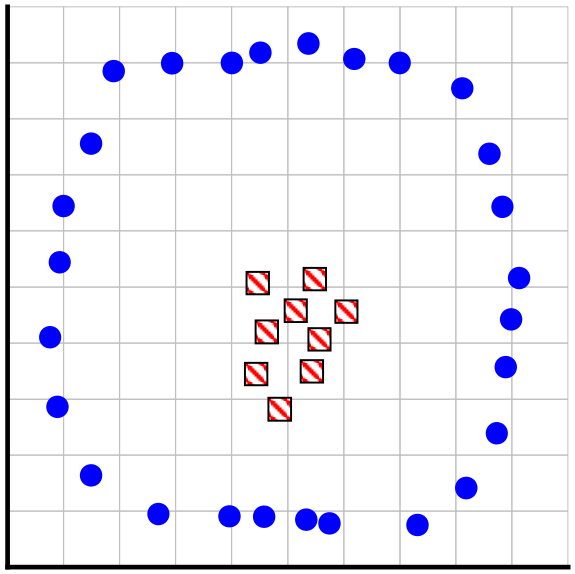
Suponha entretanto, que adicionemos uma característica **irrelevante**, por exemplo, a massa de um inseto.
Usando o comprimento da antena e a massa dos insetos com o algoritmo 1-NN obtemos a classificação errada!

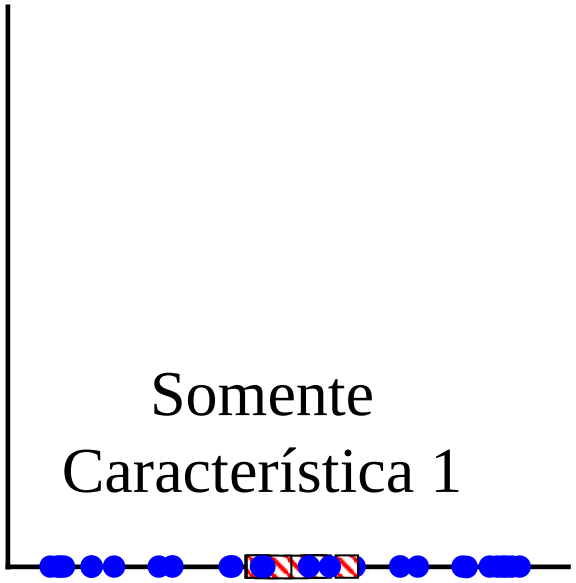
Como amenizamos a sensibilidade dos algoritmos do vizinho mais próximo a características irrelevantes?

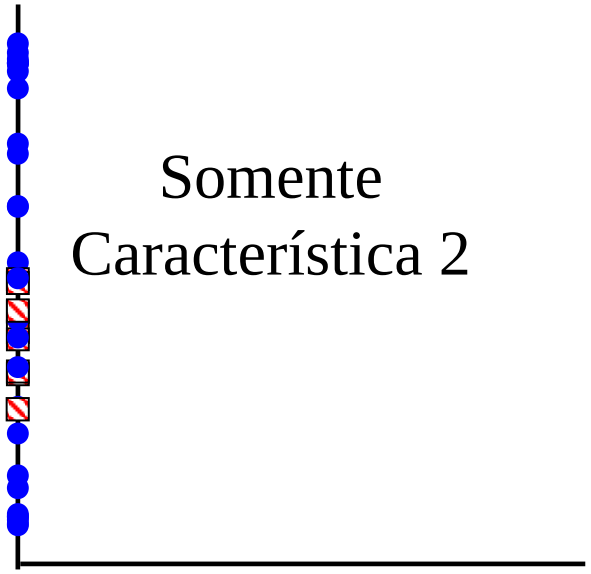
- Perguntando a um especialista quais características são relevantes para a tarefa
- Usando testes estatísticos para tentar determinar quais características são úteis
- Procurando sub-conjuntos de características (no próximo slide veremos porque isto é difícil)

Por que procurar sub-conjuntos de características é difícil

Suponha que você tenha o seguinte problema de classificação, com 100 características, e aconteça que as Características 1 e 2 (o X e Y abaixo) dão classificação perfeita, mas todas as outras 98 características são irrelevantes…





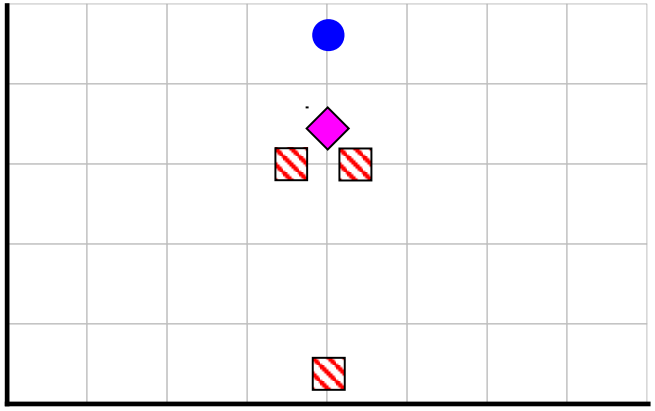


Por que procurar sub-conjuntos de características é difícil

Suponha que você tenha o seguinte problema de classificação, com 100 características, e aconteça que as Características 1 e 2 (o X e Y abaixo) dão classificação perfeita, mas todas as outras 98 características são irrelevantes…

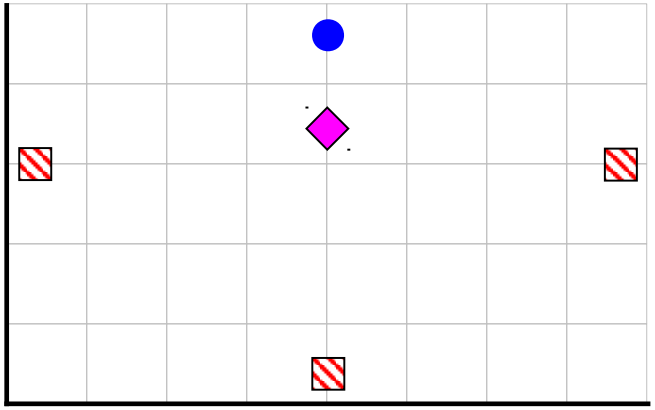
Usar todas as 100 características dará resultados pobres, mas também dará se usarmos somente a Característica 1, e também usando somente a Característica 2! Dos $2^{100} - 1$ possíveis sub-conjuntos de características, somente um realmente funcionará.

O algoritmo do vizinho mais próximo é sensivel a unidades de medida



Eixo X medido em **centímetros**
Eixo Y medido em dólares

O vizinho mais próximo ao exemplo **cor-de-rosa** desconhecido é **vermelha**.



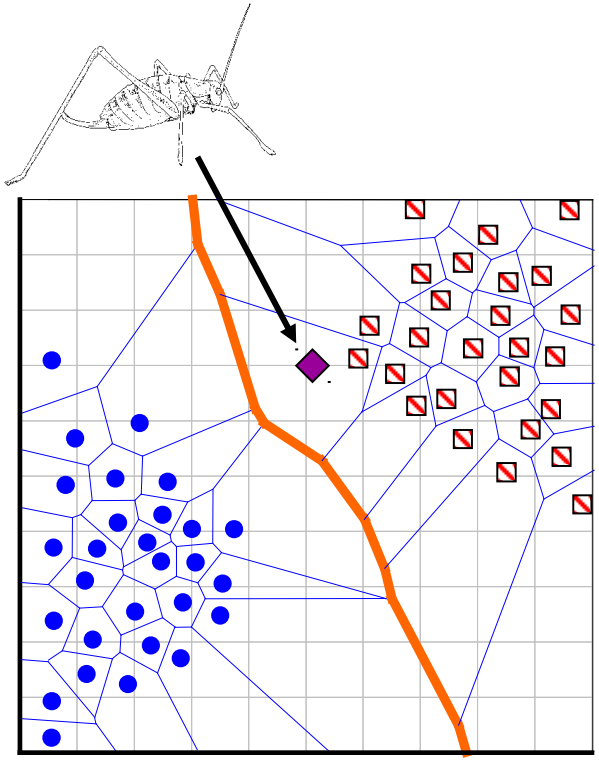
Eixo X medido em **milímetros**
Eixo Y medido em dólares

O vizinho mais próximo ao exemplo **cor-de-rosa** desconhecido é **azul**.

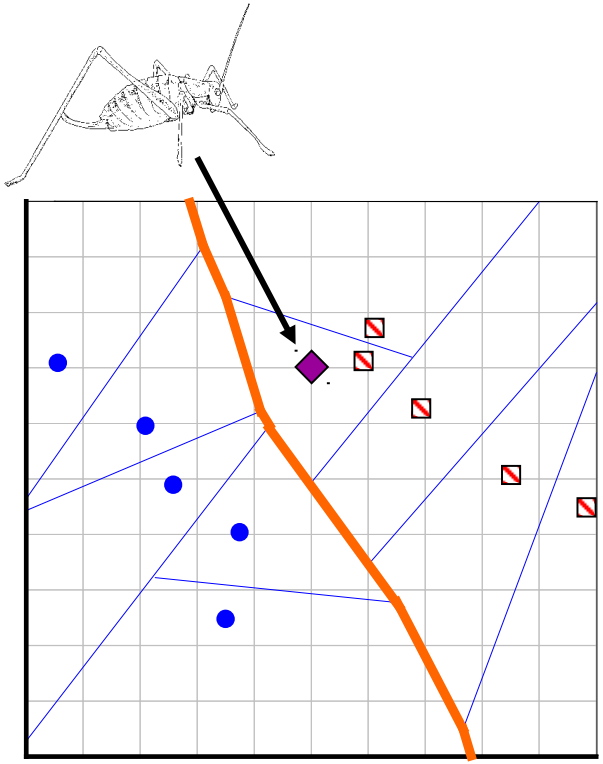
Uma solução é normalizar as unidades para números puros. Tipicamente as características são Z-normalizadas para ter uma média de zero e um desvio padrão de um. $X = (X - \text{mean}(X))/\text{std}(x)$

Podemos acelerar o algoritmo do vizinho mais próximo “jogando fora” alguns dados. Quais?

Podemos acelerar o algoritmo do vizinho mais próximo “jogando fora” alguns dados. Isto é chamado de limpeza de dados.



Uma abordagem possível. Apagar todos os exemplos que estão rodeados por membros das suas próprias classes.

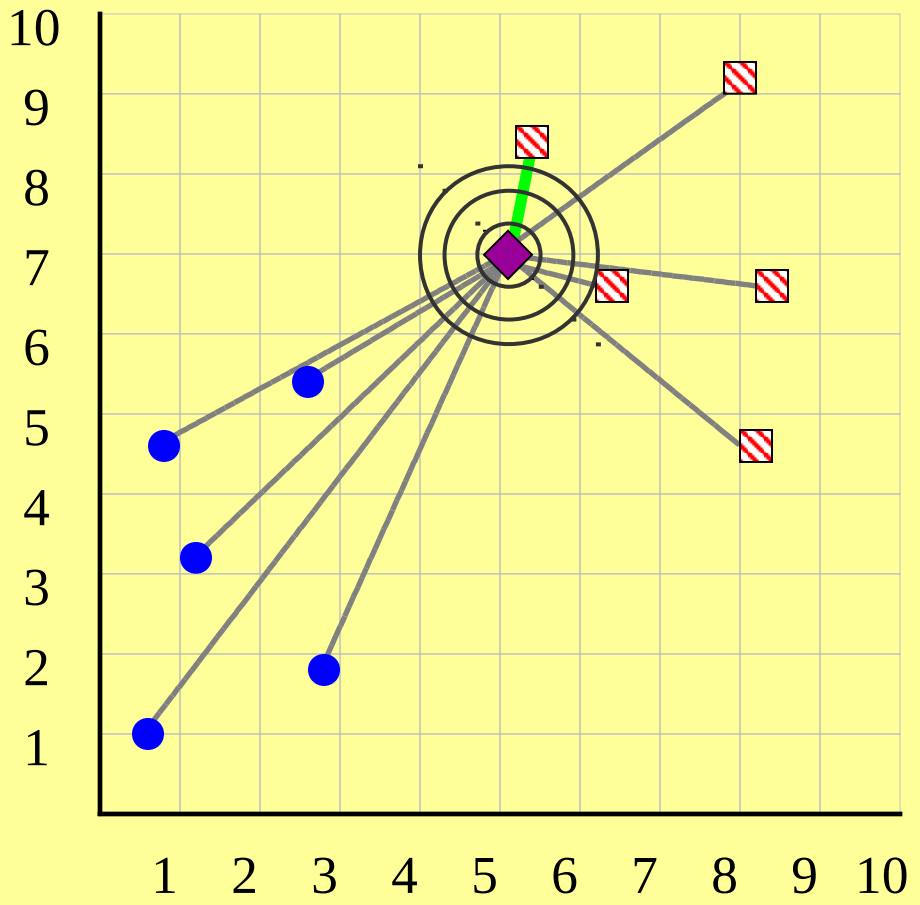


Condensed Nearest Neighbor (Hart, 1968)

1. Seleciona x ∈ **X**; D(1) = **X** \ {x}, **E** = {x}, pass=1;
2. D(pass+1) = ∅, count = 0;
3. Seja x ∈ D(pass), classificar x usando 1NN na base de dados **E**
    1. Se classificação de x está correta então:
        1. D(pass + 1) = D(pass + 1) ∪ {x}
    2. Senão
        1. **E** = **E** ∪ {x}, count++
4. D(pass)= D(pass) \ {x},
5. Se D(pass) != ∅ vá para 3
6. Se count > 0
    1. pass++, vá para o passo 2.
7. Retornar **E**

Até agora assumimos que o algoritmo do vizinho mais próximo usa a Distância Euclidiana, entretanto, este pode não ser o caso…

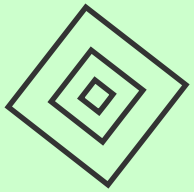
$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2}$$



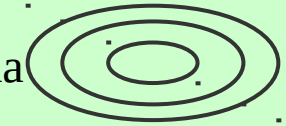
$$D(Q,C) \equiv \sqrt[p]{\sum_{i=1}^{n} (q_i - c_i)^p}$$

Max (p=inf)

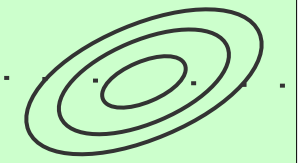
Manhattan (p=1)



Euclidiana Balanceada



Mahalanobis



…De fato, podemos usar o algoritmo do vizinho mais próximo com quaisquer funções de distância/similaridade

Por exemplo, "*Faloutsos*" é grego ou irlandês? Podemos comparar o nome "*Faloutsos*" com uma base de dados de nomes usando a distância de edição de seqüências de caracteres…

editar(*Faloutsos*, *Keogh*) = 8
editar(*Faloutsos*, *Gunopulos*) = 6

Com sorte, a semelhança do nome (particularmente o sufixo) com outros nomes gregos pode significar que o vizinho mais próximo é também um nome grego.

ID	Name	Classe
1	Gunopul**os**	**Grego**
2	Papadopoul**os**	**Grego**
3	Kolli**os**	**Grego**
4	Dardan**os**	**Grego**
5	Keo**gh**	**Irlandês**
6	Gou**gh**	**Irlandês**
7	Greenhau**gh**	**Irlandês**
8	Hadlei**gh**	**Irlandês**

Exemplo de Distância de Edição

É possível transformar qualquer string *Q* em uma string *C*, usando somente *Substituição*, *Inserção* e *Deleção*. Assuma que cada um destes operadores tem um custo associado.

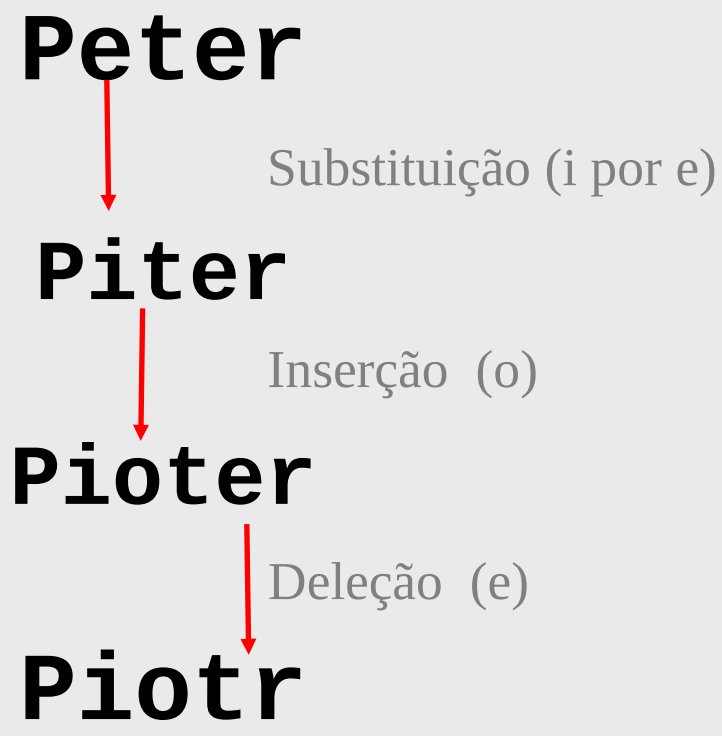
A similaridade entre duas strings pode ser definida como o custo da transformação mais barata de *Q* para *C*.

Quão semelhantes são os nomes “Peter” e “Piotr”?

Assuma a seguinte função de custo

Substituição	1 Unidade
Inserção	1 Unidade
Deleção	1 Unidade

D(**`Peter`**,**`Piotr`**) é 3



K-Vizinhos mais Próximos como Classificador Bayesiano

- Se considerarmos o kernel

$$K\left(\frac{d(\mathbf{x_i},\mathbf{x})}{d_k(\mathbf{x})}\right) = K(u) = \begin{cases} 1, \text{se } \ u \leq 1 \\ 0, \ \ \text{caso contrário} \end{cases}$$

$$P(\mathbf{x}) = \frac{k}{N d_k(\mathbf{x})}$$

K-Vizinhos mais Próximos como Classificador Bayesiano

- N_i como o número de objetos na classe c_i
- k_i como o número de objetos da classe c_i entre os k vizinhos mais próximos do objeto de teste

$$P(\mathbf{x}|c_i) = \frac{k_i}{N_i d_k(\mathbf{x})}$$

$$P(c_i|\mathbf{x}) = \frac{k_i}{k}$$

$$P(c_i) = \frac{N_i}{N}$$

K-Vizinhos Mais Próximos

- Os votos de cada vizinho não precisam ter o mesmo peso
  - Ponderação pelo inverso da distância
- O algoritmo pode ser utilizado para regressão
- Maldição de dimensionalidade prejudica qualidade dos resultados

K-Vizinhos Mais Próximos

- Limite superior do erro proporcional ao erro bayesiano conforme *N* aumenta (constante multiplicativa pode chegar a 2)
- Custo computacional alto
  - *O(NkD)*
  - Indexação pode melhorar
    - KD-TREE

Referências

- Alpaydin Introduction to Machine Learning, **Seção 8.4**
- Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar. Introduction to Data Mining, **Seção 4.3**

- Tomek, I. Two Modifications of CNN, in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-6, no. 11, pp. 769-772, Nov. 1976. doi: 10.1109/TSMC.1976.4309452
- Silverman, B., & Jones, M. (1989). E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951). International Statistical Review / Revue Internationale De Statistique, 57(3), 233-238. doi:10.2307/1403796
- T. Cover and P. Hart, "Nearest neighbor pattern classification," in IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21-27, January 1967. doi: 10.1109/TIT.1967.1053964