Aula12

DATA SCIENCE IPT

TURMA 02

LINKED EDUCATION

V2019

Aprendizado Supervisionado :

Há as amostras (X) e os resultados corretos para elas (Y). Queremos descobrir uma função que mapeie X a Y e possa prever, para novos X', novos Y'. O nome decorre do fato que, após o aprendizado, podemos verificar se o algoritmos aprendeu (supervisionar o aprendizado) usando novas amostras (X) para as quais conhecemos as repostas (Y).

Exemplos : Regressão e Classificação

Aprendizado Não Supervisionado :

Há as amostras (X) .Queremos descobrir alguma estrutura ou distribuição de dados entre elas. Não há resposta certa ou errada verificável, como no caso dos algoritmos de aprendizado supervisionado.

Clustering : o objetivo é descobrir possíveis agrupamentos entre os dados. Exemplo : K-Means

Association : o objetivo é descobrir tendências do tipo : comprou A então comprou B (para usar em Cross –Sell, por exemplo). Exemplo : Apriori

Clustering

Conteúdo

Conglomerados (Clusters)

Métodos Hierárquicos

- Aglomerativos
- Divisivo

Métodos não hierárquicos : Ex: **K-Means**

Clustering

Conteúdo

Algoritmos Hierárquicos

Criam uma hierarquia de conjuntos de classes por fusão de classes menores em classes maiores (ascendente) ou por divisão de classes maiores em classes menores (descendente).

O resultado de um algoritmo hierárquico é uma árvore ou dendograma.

Cortando a árvore num determinado nível é obtida uma partição dos indivíduos em k classes.

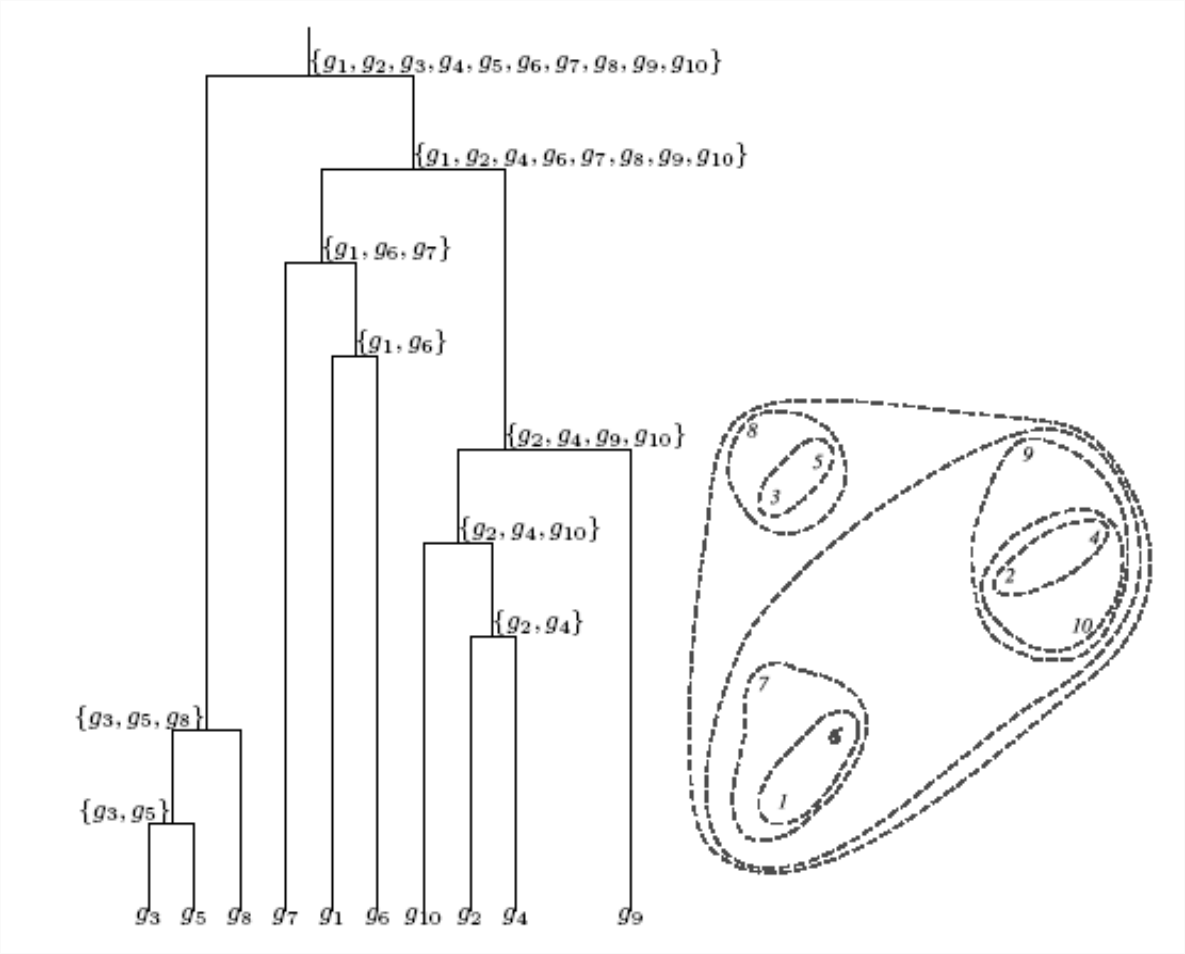
Hierárquicos aglomerativos: Partem de **n** individuos agrupados em n classes, cada classe com 1 indivíduo. Agrupam as classes sucessivamente até se obter uma única classe.

Hierárquicos divisivos: Partem de uma única classe que inclui os n indivíduos. As classes são sucessivamente divididas em classes menores até se obterem n classes, cada uma com um indivíduo.

Clustering

Conteúdo

Exemplo : Cluster Hierárquico Aglomerativo



Fonte : An Introduction to Bioinformatics Algorithms

Clustering

Conteúdo

Como é o K-Means em “pseudocódigo”

Defina o número de clusters (k)

Defina os centróides dos k clusters

Faça

Forme os k clusters associando cada objeto a seu centróide mais próximo

Recompute o centróide de cada cluster

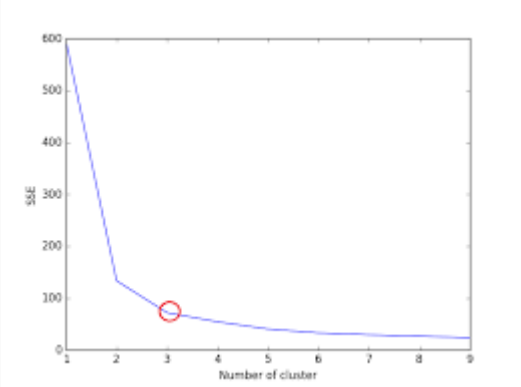
Enquanto mudarem os objetos dos clusters

Clustering

Conteúdo

Partindo de Cluster-carro-custo.ipynb, obtenha o custo de kmeans para 2 a 20 clusters (desenvolva a função loss).

Tente identificar o “cotovelo” (número de clusters onde o custo tende a “estabilizar”). É um dos critérios para se determinar o número ideal de clusters...



Ainda sobre clusters

- Há várias métricas para avaliar a “qualidade” de uma partição do dataset em clusters: Dunn, Silhouette etc...

Dunn Index

O índice Dunn é dado pelo quociente entre a menor *distância entre pontos de clusters diferentes e a maior distância entre pontos do mesmo cluster.

$$\text{Dunn}=\frac{\min\ dist\ pontos\ entre\ diferentes\ clusters}{\max \begin{matrix} dist\ pontos\ do \\ mesmo\ cluster \end{matrix}}$$

Quanto maior o índice, melhor a partição (mais compacto e separado é o cluster)

Silhouette

Para um ponto “i” de um cluster (com k pontos), a média das distâncias dele a cada um dos (k-1) outros pontos do cluster é dada por **a(i)**.

Para um ponto “i” de um cluster, a (menor) média das distâncias dele a cada um dos pontos de um outro cluster é dada por **b(i)**.

Silhouettte para um ponto “i” é $S(i)=\frac{b(i)-a(i)}{\max(a(i),b(i))}$

Podemos calcular Silhouette para o cluster todo pela média dos pontos e para a partição toda, pela média dos clusters...

Quanto mais próximo de 1 o índice, melhor a partição.

Silhouette

Atividade :

1)Partindo de silhouette_res.ipynb com k=2.

Obter (na mão) o índice Silhouette para o pontos 0 e compará-lo com o gerado pelo Scikit

2)Rodar com k=3. Comparar índices com k=3..qual é a melhor partição? Discussão.

Silhouette

Silhouette para ponto 0:

a=1.414

b = 5.65+6.4+4+5.09/4=5.27

s=(5.27-1.414)/5.27=0.73

Métricas para Classificação

Conteúdo

Na classificação, uma métrica usual é a **acurácia**, dada pelo quociente entre o número de amostras corretamente classificadas e o número total de amostras classificadas.

Essa métrica pode ser inadequada quando há grande desbalanceamento entre o número de positivos e negativos. Exemplo : doenças raras.

É importante considerar as métricas precision, recall e F1-Score...

Métricas para Classificação

A acurácia pode ser uma métrica inadequada se há grande desbalanceamento entre o número de elementos em cada classe. Suponha uma doença muito rara, que afeta 0.001% da população. Se o algoritmo sempre classificar uma pessoa como negativa (sem a doença) em um milhão de pessoas, por exemplo, teremos :
Positivos ≈ 10
negativos ≈ 999990
O algoritmo que considerar todos negativos terá Acurácia : 999990/1000000=99.999% .....por outro lado, não encaminhou ninguém para o tratamento...
Foi fraco em identificar os positivos.

Métricas para Classificação

Conteúdo

CONFUSION MATRIX

TP = True Positives
TN = True Negatives
FP = False Positives
FN = False Negatives

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

$$\textbf{F1-score} = \frac{2 * precision * recall}{precision + recall}$$

Precision mostra a parcela de TP entre os que a classificação considerou P

Recall (ou TP rate) mostra a parcela de TP identificados pela classificação entre os que são realmente P

F1-Score combina precision e recall em métrica única, pois normalmente há um trade-off entre eles. (quando se varia o threshold, por exemplo).

Métricas para Classificação

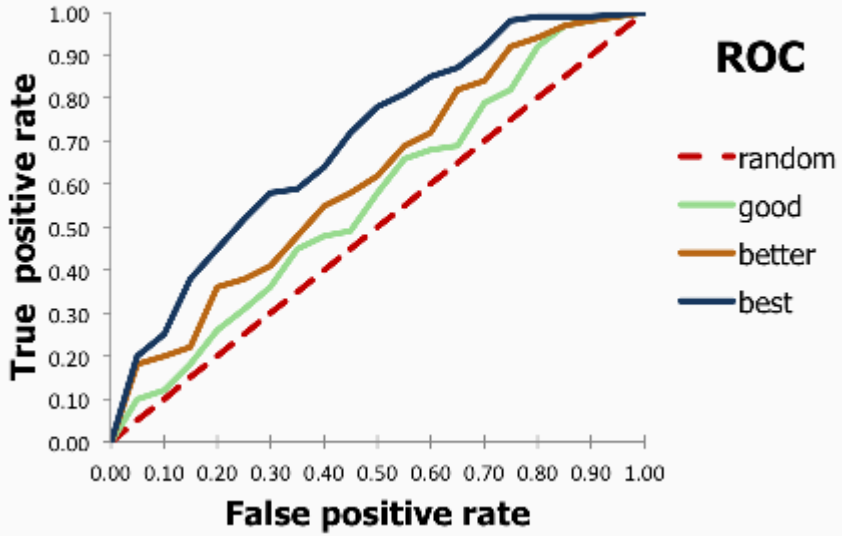
Atividade :

Partindo de metrics.ipynb

1)Montar a Confusion Matrix e obter Acurácia, Precision, Recall e F1-Score para
a lista de thresholds=[0.10,0.16,0.20,0.50]
Monte também os gráficos das predições

Curva ROC

A Curva ROC (Receiver Operating Characteristic) é útil para comparar classificadores.
Ela é traçada com FP rate em x e TP rate (Recall) em y para diferentes thresholds.
O ponto FP rate=0 e TP rate=1 é o ideal...quando possível.
Quanto maior a área sob a curva (mais próxima de 1.0), melhor o classificador.
A diagonal (FP rate = TP rate) indica que o classificador é apenas “randômico”.



True Positive Rate (TPR) = Recall = Sensitivity (Prob. de detecção) = TP/(TP+FN)
False Positive Rate(FPR) = False alarm ratio = FP/(FP+TN)
True Negative Rate(TNR) = Specificity = TN/(FP+TN) = 1-FPR

Curva ROC

Atividade : partindo de ROC_res.ipynb

Observar código da geração das curvas ROC de Logistic Regression e KNN

2)Comparar as áreas sobre as curvas ROC de logistic regression e KNN

Autovalores e Autovetores

Seja M uma matriz, v um vetor e λ um escalar.

Dizemos que v é um autovetor de M e λ é seu autovalor associado quando:

$$Mv = \lambda v$$

Assim, a transformação linear causada por M em um vetor é equivalente ao produto de um escalar por esse vetor...

Autovalores e Autovetores

$$Mv = \lambda v$$
$$(M - \lambda I)v = 0$$

Essa equação (sistema de equações lineares homogêneos) tem uma solução trivial (vetor v=0). Para que existam outras soluções além da trivial, o determinante de M- λI deve ser nulo. Dessa imposição, chegamos em uma equação que nos permite encontrar os autovalores e, depois, os autovetores (obviamente).

Autovalores e Autovetores

Exemplo :

$M = \begin{matrix} 1 & -2 \\ 3 & 9 \end{matrix}$

$M = M - \lambda I = \begin{matrix} 1-\lambda & -2 \\ 3 & 9-\lambda \end{matrix}$

$Det(M - \lambda I) = 0 \quad (1-\lambda)(9-\lambda)+6=0$

$9 - \lambda - 9\lambda + \lambda^2 + 6 = 0$

$\lambda^2 - 10\lambda + 15 = 0$

Raízes (aprox): 8.16 e 1.84....são os autovalores

Autovalores e Autovetores

Para λ=8.16, o autovetor correspondente é :

(M- λI)v=0

$$\begin{matrix} -7.16 & -2 \\ 3 & 0.84 \end{matrix} \begin{matrix} x \\ y \end{matrix} = \begin{matrix} 0 \\ 0 \end{matrix}$$

-7.16x-2y=0

Exemplo: para x=1 v={ 1, -3.58}

Teste: $\begin{matrix} 1 & -2 \\ 3 & 9 \end{matrix} \begin{matrix} 1 \\ -3.58 \end{matrix}$ = 8.16 $\begin{matrix} 1 \\ -3.58 \end{matrix}$? Vamos ao Python!

Autovalores e Autovetores

Analogamente, para λ=1.84, o autovetor correspondente é :

(M- λI)v=0

$$\begin{matrix} -0.84 & -2 \\ 3 & 7.16 \end{matrix} \begin{matrix} x \\ y \end{matrix} = \begin{matrix} 0 \\ 0 \end{matrix}$$

-0.84x– 2y =0
y=-0.42x

Exemplo: para x=1 v={ 1, -0.42}

Teste: $\begin{matrix} 1 & -2 \\ 3 & 9 \end{matrix} \begin{matrix} 1 \\ -0.42 \end{matrix}$ = 1.84 $\begin{matrix} 1 \\ -0.42 \end{matrix}$? Sim! A conferir no Python

Autovalores e Autovetores

É interessante deixar os autovetores com módulo 1. Para isso, basta obter a norma e dividir seus componentes por essa norma...

λ=8.16 o vetor v={1, -3.58} tem norma: 3.72
e fica v={ 0.27, -0.96) com norma 1

λ=1.84 o vetor v={1, -0.42} tem norma: 1.08
e fica v={ 0.92, -0.39) com norma 1

Vamos conferir com numpy! autovalores_res.ipynb

Propriedade Importante:

O produto dos autovalores é o determinante da matriz:

M= $\begin{matrix} 1 & -2 \\ 3 & 9 \end{matrix}$ det(M)= 9+6=15 = 8.16*1.84

Hermitian Matrix

A matriz M é ‘Hermitiana” se a transposta da conjugada é a própria M (?)

Exemplos:

$$\begin{bmatrix} 3 & 2+i & -i \\ 2-i & 5 & 5-i \\ i & 5+i & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 2 & -i & 10+i \\ 2 & 1 & 0 & -5i \\ i & 0 & -2 & 0 \\ 10-i & 5i & 0 & 5 \end{bmatrix} \quad \begin{bmatrix} 0 & i & i \\ -i & 0 & i \\ -i & -i & 0 \end{bmatrix}$$

É fácil identificá-las: Na diagonal principal só há reais e os elementos simétricos são conjugados (a+bi , a-bi)

Toda matriz real simétrica é Hermitiana.

Toda Matriz hermitiana tem autovalores reais

Os autovetores associados a autovalores diferentes são ortogonais

Matriz de Covariância

Conteúdo

Covariância:

$$\sigma(x, y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

Matriz de covariância (para 3 features)

$$\begin{bmatrix} \text{Var}_1 & \text{Cov}_{1,2} & \text{Cov}_{1,3} \\ \text{Cov}_{1,2} & \text{Var}_2 & \text{Cov}_{2,3} \\ \text{Cov}_{1,3} & \text{Cov}_{2,3} & \text{Var}_3 \end{bmatrix}$$

Matrizes de Covariância são reais e simétricas, portanto Hermitianas

Todas as matrizes de covariância são positivas semi definidas. Isso implica que todos os autovalores são positivos.

Uma matriz positiva semi definida A => $v_t Av >= 0$ para qualquer v diferente de zero

PCA : Principal Component Analysis

A Análise de Componentes Principais (PCA) é um
ferramenta de redução de dimensionalidade que pode ser usada para
reduzir um grande conjunto de variáveis para um pequeno conjunto
que ainda contém a maioria das informações em
o conjunto grande

PCA : Principal Component Analysis

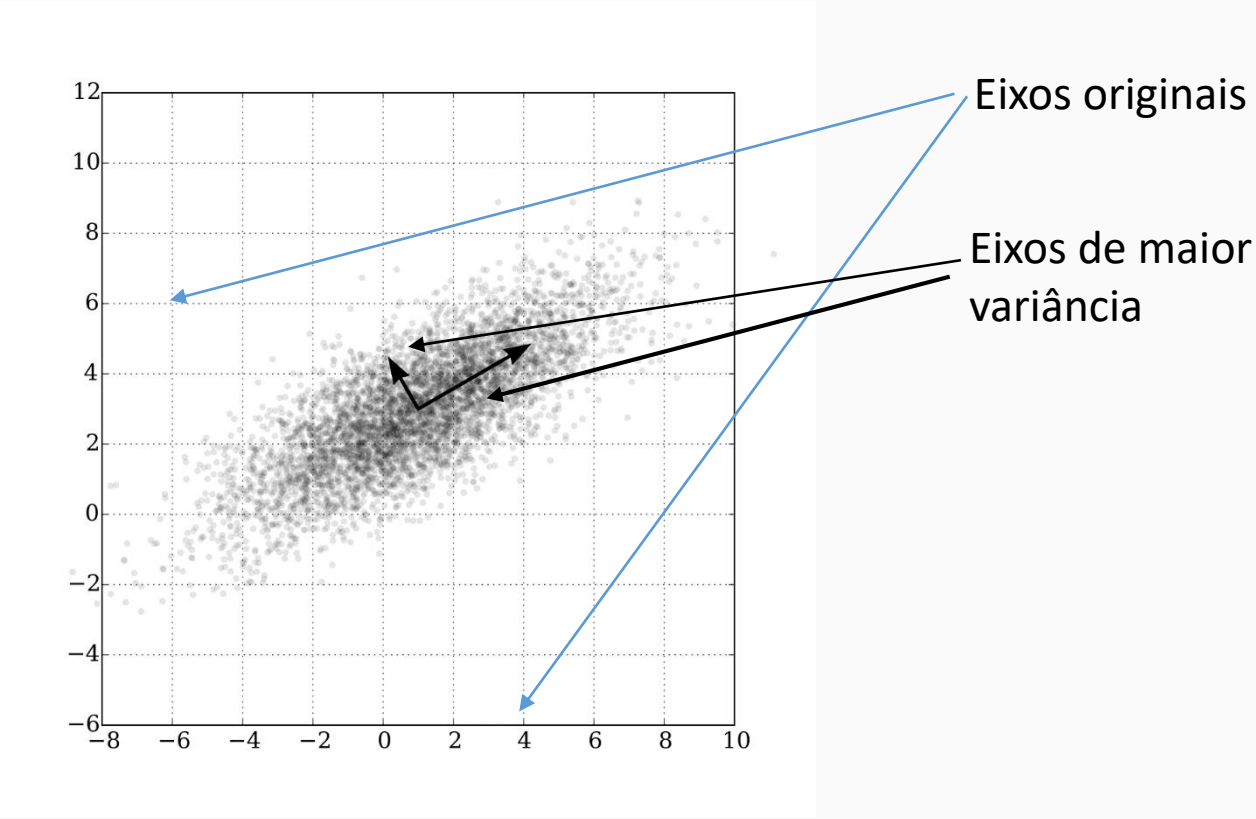
Basicamente, partimos da matriz de covariância entre as features e extraímos seus autovalores e autovetores correspondentes.

Os autovetores associados a maiores autovalores indicarão maior "variância naquele eixo" => mais importância...mais variância, mais informação

Podemos representar o dataset com menos eixos (dimensões) que o inicial desprezando os eixos com autovalores baixos.

É possível voltar ao dataset incial, mas com perdas....

PCA : Principal Component Analysis



PCA : Principal Component Analysis

Conteúdo

Partindo de pca.ipynb

1)Analisar o código sem e com scikit

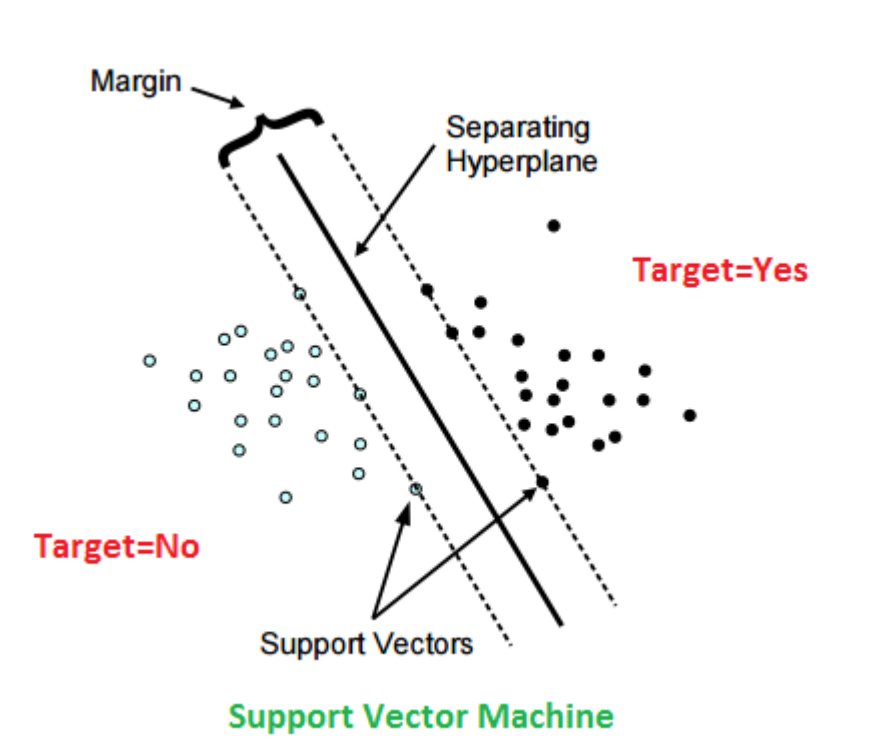
2)Verificar a matriz de covariância antes e depois das transformações

3)Calcular a acurácia (logistic regression) para (binário) versicolor só com dois componentes e com 4 componentes

4)Reconstruir o dataset (de 2 para 4 features) e observar perdas (é uma lossy compression)

SVM (Support Vector Machines)

Introdução (Linear SVM)...as classes na figura abaixo são claramente linearmente separáveis. Pra criar a “separação” (hiperplano) que deixe as classes o mais “distante” possível ,procuramos uma reta que maximiza a distância de cada um dos pontos próximos `a fronteira (os de mais difícil classificação), ou seja, maximiza a margem. Esses pontos são os “Support Vectors” (vectors?)

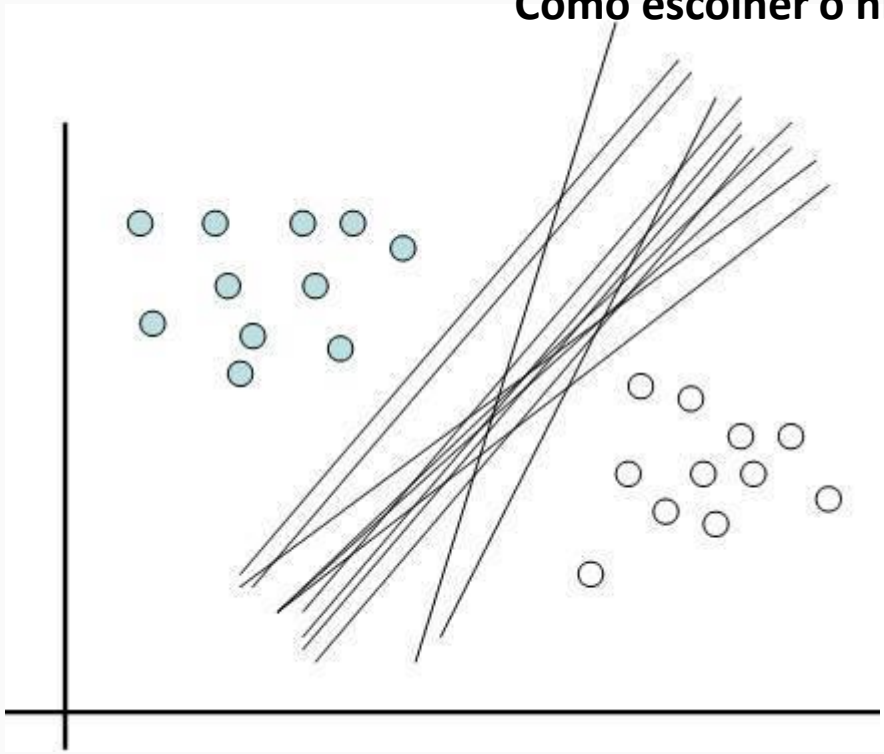


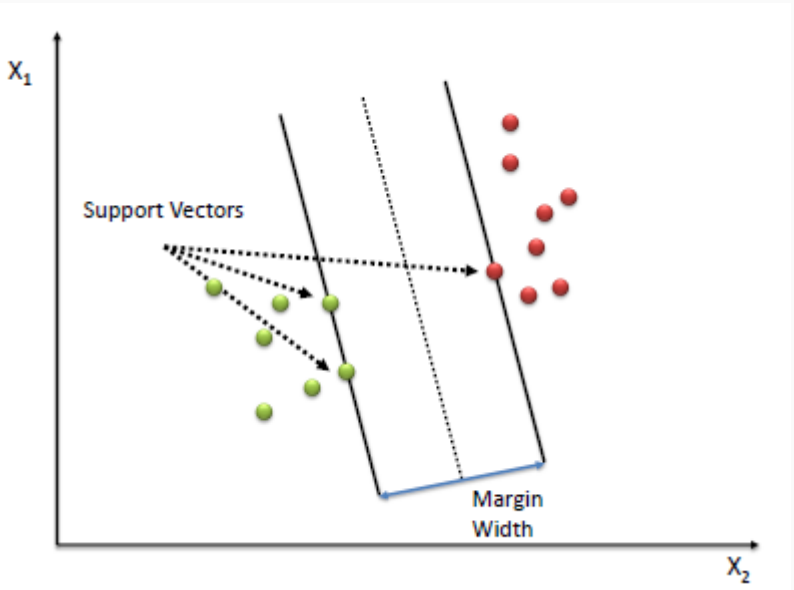
SVM (Support Vector Machines)

Support Vector Machine é um tipo de algoritmo bastante utilizado para classificação, tipicamente. O principal fundamento desse algoritmo é buscar um hiperplano que crie uma maior margem (distância) entre as classes. Os pontos considerados (matematicamente) para maximizar essa margem são os support vectors.

SVM (Support Vector Machines)

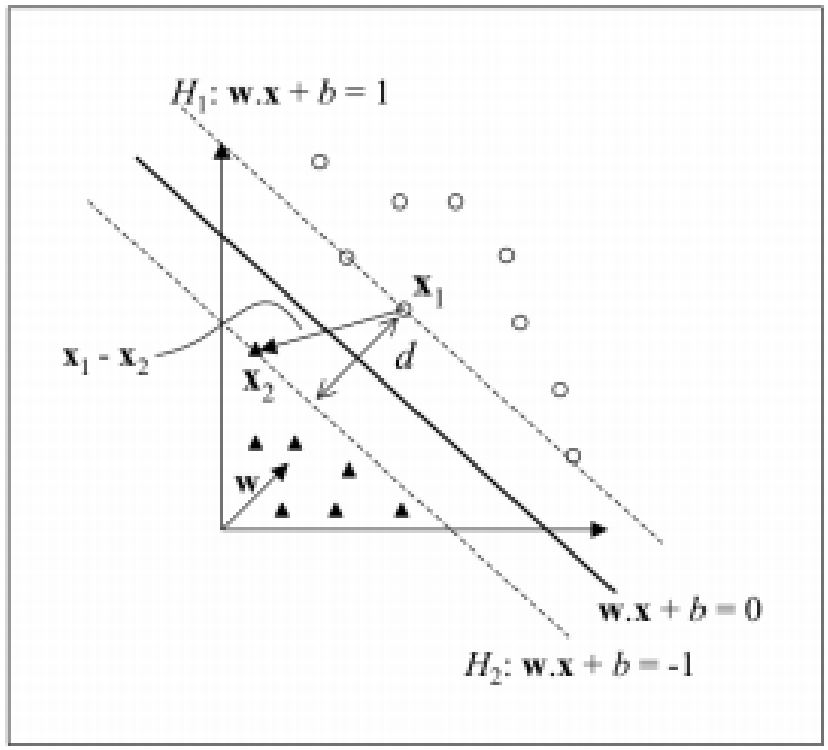
Como escolher o hiperplano?





SVM (Support Vector Machines)

Como escolher o *hiperplano?



d (margem) é a decomposição de **x1 – x2** na direção do vetor normal **w**

A Margem é o Módulo da Projeção de **x1-x2** sobre **w**

$$\frac{||x1 - x2|| * < (x1 - x2), w >}{||x1 - x2|| * ||w||}$$

<x1,w> é 1-b
-<x2,w> é 1+b

Assim, a margem é : $\frac{2}{||w||}$

Para maximizar a margem, devemos minimizar o módulo de w...

Minimizar $||\mathbf{w}||$ **ou** $\frac{1}{2}||\mathbf{w}||^2$ chega no mesmo resultado. Optamos pela forma quadrática por ser um problema matemático de otimização bem estudado.

O problema final é (margens rígidas):

$$\underset{\mathbf{w},b}{\text{Minimizar}} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

Com as restrições: $y_i\,(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \geqslant 0, \quad \forall i = 1, \ldots, n$

Matematicamente, criamos uma função Lagrangeana (?) que engloba a função a ser minimizada e as restrições.

SVM (Support Vector Machines)

Conteúdo

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1\right)$$

Equação 1

Impondo:

$$\frac{\partial L}{\partial b} = 0 \quad \text{e} \quad \frac{\partial L}{\partial \mathbf{w}} = 0$$

Equação 2

Chegamos a:

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

Substituindo 3 em 1, Chegamos ao problema dual:

$$\underset{\boldsymbol{\alpha}}{\text{Maximizar}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right)$$

$$\text{Com as restrições:} \begin{cases} \alpha_i \geqslant 0, \quad \forall i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

SVM (Support Vector Machines)

Conteúdo

Sendo α* vindo do problema dual e os correspondentes w* e b*....há as condições de Kühn-Tucker para problemas de otimização:

$$\alpha_i^* \left(y_i \left(\mathbf{w}^* \cdot \mathbf{x}_i + b^* \right) - 1 \right) = 0, \ \forall i = 1, \ldots, n$$

Para que as condições sejam satisfeitas com α*> 0, xi deverá estar nas bordas....será um Support Vector!

A função decisora será o sinal da fórmula abaixo...

$$g\left(\mathbf{x}\right) = \operatorname{sgn}\left(f\left(\mathbf{x}\right)\right) = \operatorname{sgn}\left(\sum_{\mathbf{x}_i \in \mathrm{SV}} y_i \alpha_i^* \mathbf{x}_i \cdot \mathbf{x} + b^*\right)$$

SVM (Support Vector Machines)

Conteúdo

Partindo de lousa-svm_res.ipynb

Atividade1 : analisar código

Atividade 2 : mostrar que a reta é x1=1.5

Atividade 3 : apresentar vetores de suporte

Atividade 4: criar função decisora com base em support vectors e coeficientes da solução dual...

SVM (Support Vector Machines)

SVM continua na próxima aula

SVM Analisando o código da classe MKMeans (Prof. Leston)

Cursos com Alta Performance de Aprendizado