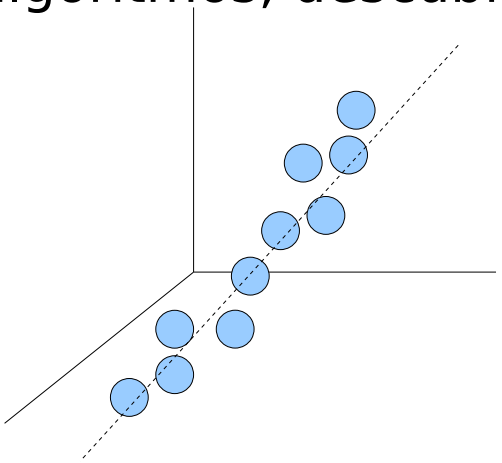
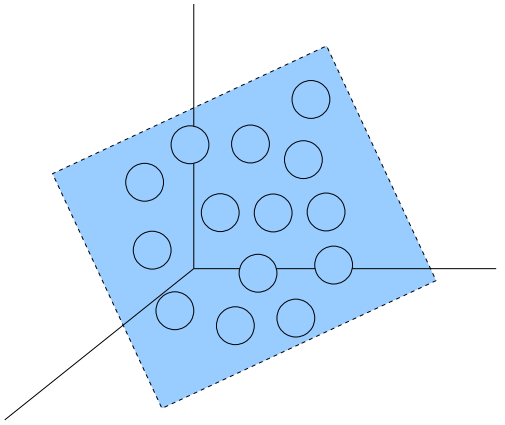
Tema 9. Reducción da dimensionalidade

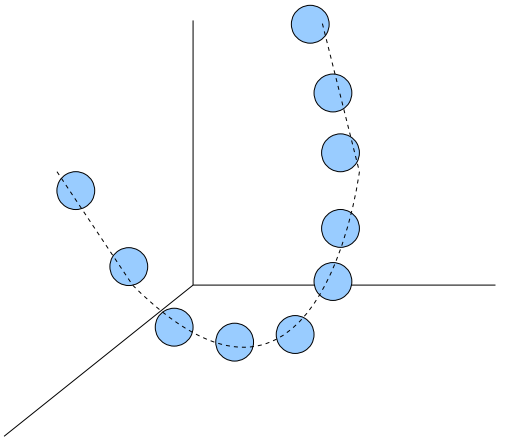
- Suponse que a dimensionalidade intrínseca dos datos é inferior á súa dimensionalidade orixinal (n)
- Obxectivo: atopar a dimensionalidade mínima d<n que recolle a mesma información (aproximadamente) que os datos orixinais
- Proxectar a $\mathbb{R}^d$ conservando certas propiedades dos datos en $\mathbb{R}^n$
- Aplicacións: compresión de datos, aumento na velocidade dos algoritmos, descubrimiento de información



Espazo linear 1D



Espazo linear 2D



Espazo 1D curvo

Reducción da dimensionalidade (II)

- Dependendo das propiedades que nos interesa preservar, existen técnicas de reducción supervisadas e non supervisadas:

a) Supervisadas: mapean a un espazo d-dimensional (d<n) tentando separar os patróns con saídas desexadas (clases en clasificación, valores continuos en regresión) distintas e mantendo xuntos patróns con saídas desexadas similares

b) Non supervisadas: mapean tentando conservar certas propiedades, p.ex. as distancias entre patróns en $\mathbb{R}^n$

- Existen distintas medidas de calidade para o proceso de reducción de dimensionalidade:

a) Varianza das entradas orixinais conservada no espazo reducido

b) Acerto ou RMSE dun clasificador ou regresor no espazo reducido, comparado co espazo orixinal en problemas de clasificación ou regresión

Análise en compoñentes principais (PCA)

- Tamén chamada transformación de Karhunen-Loève
- Proxecta os patróns a un espazo no cal as dimensións (eixos ou compoñentes principais) son perpendiculares entre si
- Estos eixos maximizan a varianza dos datos proxectados
- Proxección non supervisada
- Para proxectar a $\mathbb{R}^d$ necesitas unha matriz $\mathbf{A}_{dxn}$, de modo que $\mathbf{y}=\mathbf{Ax}$, con $y \in \mathbb{R}^d$
- Supoñemos datos con media nula: $\sum_{i=1}^{N} \boldsymbol{x}_i = \mathbf{0}$
- Consideremos o caso d=1 ($\mathbf{A}_{1xn}=\mathbf{a}^T$): proxectamos a $\mathbb{R}^1$ maximizando a varianza. A función de custe (varianza) en $\mathbb{R}^d$ é:

$$J(\boldsymbol{a}) = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{a}^T \boldsymbol{x}_i\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{a}^T \boldsymbol{x}_i \boldsymbol{x}_i^T \boldsymbol{a} = \boldsymbol{a}^T \Sigma \boldsymbol{a} \qquad \Sigma = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i \boldsymbol{x}_i^T$$

$\mathbf{\Sigma}$ é a matriz de covarianza dos datos, de ten orde nxn

PCA (II)

- A proxección non é única porque multiplicando a matriz **A** por calquera número obtés unha proxección equivalente
- Imponse a condición $|\mathbf{A}|^2=|\mathbf{a}|^2=\mathbf{a}^T\mathbf{a}=1$ para obter unha única solución
- Función lagranxiana: $L(\boldsymbol{a},\lambda)=\boldsymbol{a}^T\Sigma\boldsymbol{a}+\lambda(\boldsymbol{a}^T\boldsymbol{a}-1)$

λ é o multiplicador de Lagrange

- Derivando a respeito de **a** e igualando a **0** temos $\boldsymbol{\Sigma a}=\lambda\mathbf{a}$
- A dimensión na que hai que proxectar é un autovector da matriz $\boldsymbol{\Sigma}$
- Como ten que ser $\mathbf{a}^T\mathbf{a}=1$, multiplicando por λ temos que:

$$\boldsymbol{a}^T(\lambda\boldsymbol{a})=\lambda\Rightarrow\boldsymbol{a}^T\Sigma\boldsymbol{a}=\lambda$$

- Como $\mathbf{a\Sigma a}$ (a varianza) debe maximizarse, λ ten que ser máximo
- Polo tanto, a dirección que maximiza a varianza das entradas é a do autovector **a** asociado ao autovalor λ máximo (autovector principal)

PCA (III)

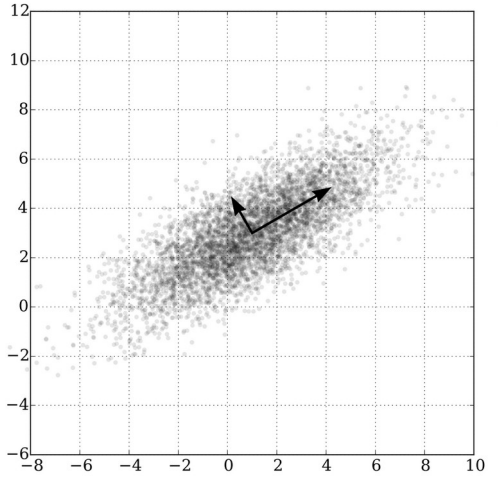
- Como a matriz de covarianza $\mathbf{\Sigma}$ é simétrica e definida positiva, tódolos seus n autovalores son reais e non negativos
- Se N>n, $\mathbf{\Sigma}$ é invertíbel e tódolos seus autovalores son positivos
- A 2ª compoñente principal: mesmo problema de optimización con $\mathbf{a}_2^T\mathbf{a}_1=0$ para que $\mathbf{a}_2$ sexa perpendicular a $\mathbf{a}_1$
- É fácil ver que $\mathbf{a}_2$ é o autovector de $\mathbf{\Sigma}$ correspondente ao 2º meirande autovalor λ_2
- O autovalor λ_i mide a varianza retida pola compoñente proxectada i=1..n
- A proxección PCA a $\mathbb{R}^d$ está dada pola matriz $\mathbf{A}=[\mathbf{a}_1..\mathbf{a}_d]$, de orde nxd, definida polos d autovectores $\mathbf{a}_1..\mathbf{a}_d$ principais, asociados aos d meirandes autovalores $\lambda_1>..>\lambda_d$
- A proxección a $\mathbb{R}^d$ execútase como $\mathbf{y}=\mathbf{Ax}$, $\mathbf{y}\in\mathbb{R}^d$, $\mathbf{x}\in\mathbb{R}^n$, ou $\mathbf{Y}=\mathbf{XA}$, con $\mathbf{X}$ de orde Nxn e $\mathbf{Y}$ de orde Nxd

PCA (IV)

- O PCA proxecta os datos a un espazo $\mathbb{R}^d$ no que as compoñentes son perpendiculares entre si e reteñen a máxima varianza posíbel con d compoñentes

$$\sum_{i=1}^{d} \lambda_i$$

- Pódense considerar máis ou menos compoñentes principais
- Unha alternativa frecuente é usar as d compoñentes necesarias para sumar o 95% da varianza dos datos
- Outras veces o nº d de compoñentes a reter está predeterminada
- Se proxectas a d=n, o que fas é rotar o espazo orixinal de modo que os novos eixos sexan os que maximizan as varianzas



Redución da dimensionalidade con LDA

- O LDA reduce a d<C a dimensión dun problema de clasificación (supervisado)
- Proxecta ao espazo xerado polos d autovectores asociados aos meirandes autovalores da matriz $\mathbf{S}_B^{-1}\mathbf{S}_W$
- $\mathbf{S}_B$=matriz covarianza entre (between) clases (maximiza)
- $\mathbf{S}_W$=matriz covarianza intra (within)-clase (minimiza)
- Con C=2 clases, o LDA so proxecta a d=1 dimensión

```
d=input('d? ');x=load('datos.dat');
y=x(:,end);x(:,end)=[];n=size(x,2);
if d<n; error('d<n'); end
C=numel(unique(y));
St=cov(x);Sw=zeros(n);
for i=1:C
    j=find(y==i);
    Sw=Sw+numel(j)*cov(x(j,:))/C;
end
Sb=St-Sw;m=min(C-1,d);
[V,L]=eig(Sb,Sw);
L=sort(diag(L),'descend');
A=V(:,L(1:m));y=x*A;
```

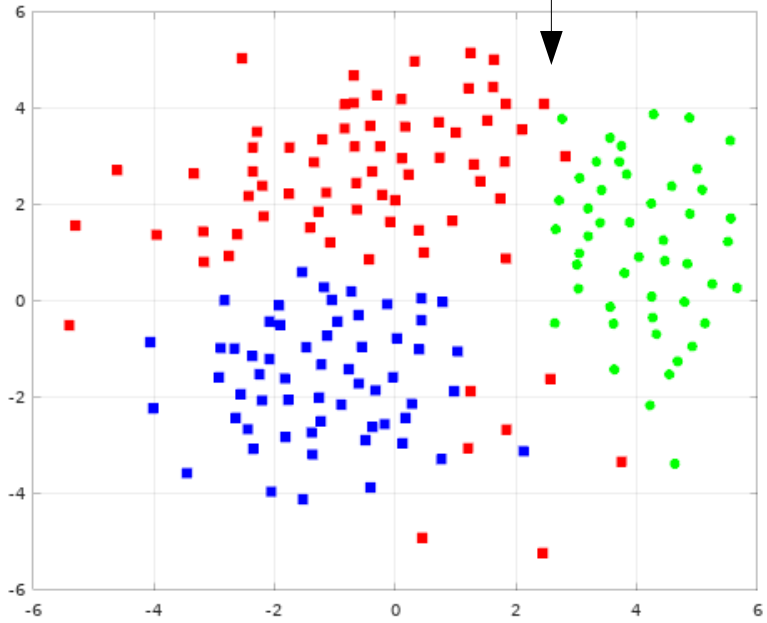
V=matriz de autovectores
L=matriz diagonal cos autovalores

Proxección de Sammon

- Conserva a estructura de distancias entre patróns
- Inicializa os N patróns $\{\mathbf{y}_i\}_{i=1}^{N}$ en $\mathbb{R}^d$ de modo aleatorio
- Actualiza iterativamente os $\mathbf{y}_i$ usando descenso de gradiente para minimizar a función de estrés de Kruskal:

$$\varphi = \frac{\sum_{i<j} \frac{(d\,\boldsymbol{x}_{ij} - d\,\boldsymbol{y}_{ij})^2}{d\,\boldsymbol{x}_{ij}}}{\sum_{i<j} d\,\boldsymbol{x}_{ij}}$$

Mapa de clasificación en dataset wine



- $d\mathbf{x}_{ij}=\mathbf{x}_i-\mathbf{x}_j$ e $d\mathbf{y}_{ij}=\mathbf{y}_i-\mathbf{y}_j$: trata de minimizar a diferencia cadrática entre as distancia $d\mathbf{x}_{ij}$ en $\mathbb{R}^n$ e $d\mathbf{y}_{ji}$ en $\mathbb{R}^d$
- Sumado sobre o conxunto de datos e relativo ás diferencias orixinais en $\mathbb{R}^n$
- Proxecta problemas de clasificación ou regresión a calquer dimensión d<n

Neighborhood component analysis (NCA)

- Método supervisado: usa as saídas verdadeiras
- Transformación linear: $\mathbf{y}=\mathbf{A}\mathbf{x}$
- Aprende a distancia de Mahalanobis definida pola matriz $\mathbf{A}$ que maximiza o erro LOO do clasificador 1NN no espazo proxectado:

$$d(\mathbf{y}_i,\mathbf{y}_j)=(\mathbf{y}_i-\mathbf{y}_j)^T(\mathbf{y}_i-\mathbf{y}_j)=[\mathbf{A}(\mathbf{x}_i-\mathbf{x}_j)]^T[\mathbf{A}(\mathbf{x}_i-\mathbf{x}_j)]=d\mathbf{x}_{ij}^T\mathbf{A}^T\mathbf{A}d\mathbf{x}_{ij}=(\mathbf{A}d\mathbf{x}_{ij})^2$$

- Probabilidade p_{ij} de que un patrón $\mathbf{x}_j$ sexa o veciño do patrón $\mathbf{x}_i$ (función softmin como aproximación derivábel do mínimo):

$$p_{ij}=\frac{e^{-(\mathbf{A}d\mathbf{x}_{ij})^2}}{\sum_{k\neq i} e^{-(\mathbf{A}d\mathbf{x}_{ij})^2}}$$

- Probabilidade p_i de que un patrón $\mathbf{x}_i$ se clasifique correctamente:

$$p_i=\sum_{y_j=y_i} p_{ij}(\mathbf{A})$$

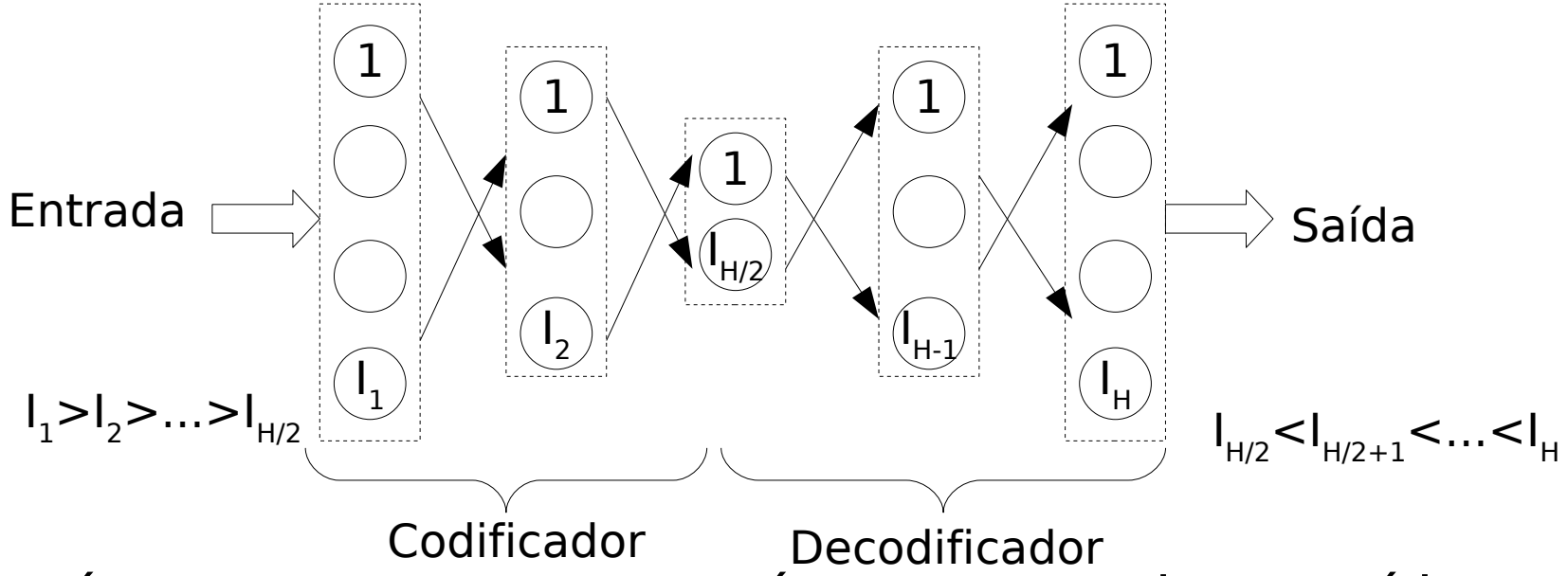
- Erro LOO:

$$\Phi(\mathbf{A})=\sum_{i=1}^{N} p_i(\mathbf{A})$$

NCA minimiza $\Phi(\mathbf{A})$ usando descenso de gradente

Redución da dimensionalidade con redes neuronais autocodificadoras

- Reducen a dimensionalidade mapeando os patróns da capa de entrada á capa oculta media H/2



- Entrénase co mesmo patrón na entrada e saída, para reconstruir este patrón
- A primeira metade da rede realiza a redución da dimensionalidade

Selección de características

- Selecciona as características (entradas) de meirande utilidade para o obxectivo concreto. Non as transforma
- Adición/eliminación de características:

1) Con alta ou baixa varianza
2) Con altas ou baixas puntuacións en test estatístico (p.ex. Chi2)
3) Moi ou pouco importantes para un clasificador ou regresor (selección baseada en modelo): sequential feature selection (SFS)

   Entrena e avalía un clasificador ou regresor rápido cunha soa característica usando validación cruzada. Repite para tódalas características, elexindo a que acada mellor calidade

   Repite o proceso ata que se acada o nº desexado de características. Pode engadir (forward SFS) ou eliminar (backward SFS) características