Computational Intelligence: Methods and Applications

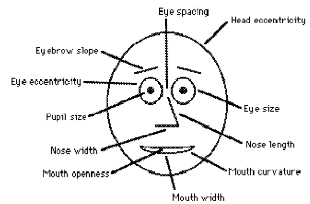
Lecture 5
EDA and linear transformations.

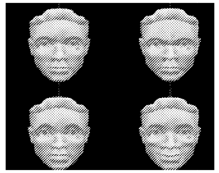
Włodzisław Duch
SCE, NTU, Singapore
Google: Duch

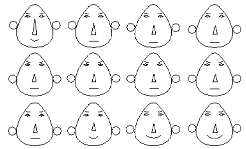
Chernoff faces

Humans have specialized brain areas for face recognition.

For d < 20 represent each feature by changing some face elements.







Interesting applets:

http://www.cs.uchicago.edu/~wiseman/chernoff/

http://hesketh.com/schampeo/projects/Faces/chernoff.html

Other EDA techniques

NIST Engineering Statistics Handbook has a chapter on exploratory data analysis (EDA).

http://www.itl.nist.gov/div898/handbook/index.htm

Unfortunately many visualization programs are written for X-Windows only, are in Fortran, or S or R languages.

Sonification: data converted to sounds!
Example of sound of EEG data.
More: http://www.techfak.uni-bielefeld.de/~thermann/projects/
Think about potential applications!

CI approach to visualization

Scatterograms: project all data on two features.
Find more interesting directions to create projections.
Linear projections:

- Principal Component Analysis,
- Discriminant Component Analysis,
- Projection Pursuit – “define interesting” projections.

Non-linear methods – more advanced, some will appear later.

Statistical methods: multidimensional scaling.
Neural methods: competitive learning, Self-Organizing Maps.
Kernel methods, principal curves and surfaces.
Information-theoretic methods.

Distances in feature spaces

Data vector, d-dimensions $\mathbf{X}^T = (X_1, \ldots X_d)$, $\mathbf{Y}^T = (Y_1, \ldots Y_d)$
Distance, or metric function, is a 2-argument function that satisfies:

$$d(\mathbf{X},\mathbf{Y}) = \|\mathbf{X}-\mathbf{Y}\| \geq 0; \quad d(\mathbf{X},\mathbf{Y}) = d(\mathbf{Y},\mathbf{X})$$
$$d(\mathbf{X},\mathbf{Y}) \leq d(\mathbf{X},\mathbf{Z}) + d(\mathbf{Z},\mathbf{Y})$$

Distance functions measure (dis)similarity.

Popular distance functions:

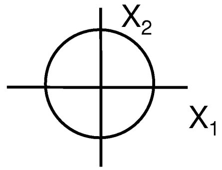
Euclidean distance (L_2 norm) $$\|\mathbf{X}-\mathbf{Y}\|_2 = \left(\sum_{i=1}^{d}(X_i - Y_i)^2\right)^{1/2}$$

Manhattan (city-block) distance (L_1 norm) $$\|\mathbf{X}-\mathbf{Y}\|_1 = \sum_{i=1}^{d}|X_i - Y_i|$$

Two metric functions

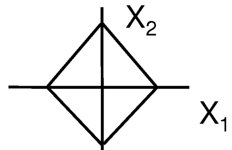
Equidistant points in 2D: $\|\mathbf{X}-\mathbf{P}\|_L = \|\mathbf{Y}-\mathbf{P}\|_L$

Euclidean case: circle or sphere



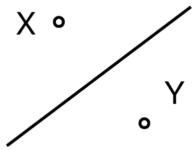
isotropic

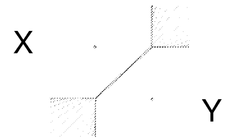
Manhattan case: square



non-isotropic

Identical distance between two points X, Y: imagine that in 10 D !



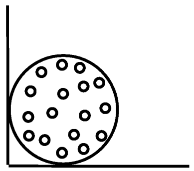


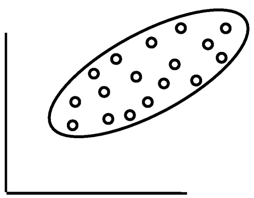
All points in the shaded area have the same Manhattan distance to X and Y!

Linear transformations

2D vectors **X** in a unit circle with mean (1,1); **Y** = **A**∗**X**, **A** = 2x2 matrix

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$$





The shape and the mean of data distribution is changed.
Scaling (diagonal a_{ii} elements); rotation (off-diag), mirror reflection.
Distances between vectors are not invariant: $\|Y^1-Y^2\| \neq \|X^1-X^2\|$

Invariant distances

Euclidean distance is not invariant to linear transformations **Y** = **A**∗**X**,
scaling of units has strong influence on distances.
How to select scaling/rotations for simplest description of data?

$$\left\|\mathbf{Y}^{(1)}-\mathbf{Y}^{(2)}\right\|^2 = \left(\mathbf{Y}^{(1)}-\mathbf{Y}^{(2)}\right)^T \left(\mathbf{Y}^{(1)}-\mathbf{Y}^{(2)}\right)$$
$$= \left(\mathbf{X}^{(1)}-\mathbf{X}^{(2)}\right)^T \mathbf{A}^T\mathbf{A}\left(\mathbf{X}^{(1)}-\mathbf{X}^{(2)}\right)$$

Orthonormal matrices: $A^TA = I$, are inducing rigid rotations.

To achieve full invariance requires therefore standardization of data (scaling invariance) and should use covariance matrix.
Mahalanobis metric will replace A^TA by inverse of the covariance matrix.

Data standardization

For each vector component $\mathbf{X}^{(j)T}=(X_1^{(j)}, \ldots X_d^{(j)})$, $j=1 \,..\, n$

calculate mean and std: n – number of vectors, d – their dimension

$$\bar{X}_i = \frac{1}{n}\sum_{j=1}^{n} X_i^{(j)}; \quad \bar{\mathbf{X}} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{X}^{(j)}$$

Vector of mean feature values.

Averages over rows.

	$\mathbf{X}^{(1)}$	$\mathbf{X}^{(2)}$	$\cdots$	$\mathbf{X}^{(n)}$
$\bar{X}_1$	$X_1^{(1)}$	$X_1^{(2)}$	$\cdots$	$X_1^{(n)}$
$\bar{X}_2$	$X_2^{(1)}$	$X_2^{(2)}$	$\cdots$	$X_2^{(n)}$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$\bar{X}_d$	$X_d^{(1)}$	$X_d^{(2)}$	$\cdots$	$X_d^{(n)}$

Standard deviation

Calculate standard deviation:

$$\bar{X}_i = \frac{1}{n}\sum_{j=1}^{n} X_i^{(j)}$$

Vector of mean feature values.

$$\sigma_i^2 = \frac{1}{n-1}\sum_{j=1}^{n}\left(X_i^{(j)} - \bar{X}_i\right)^2$$

Variance = square of standard deviation (std), sum of all deviations from the mean value.

Why $n-1$, not n ? If true mean was known it should be n, but if the mean is calculated the formula with $n-1$ converges to <u>the true variance</u>!

Transform X => Z, standardized data vectors:

$$Z_i^{(j)} = \left(X_i^{(j)} - \bar{X}_i\right)/\sigma_i$$

Standardized data

Std data: zero mean and unit variance.

$$\bar{Z}_i = \frac{1}{n}\sum_{j=1}^{n} Z_i^{(j)} = \frac{1}{n}\sum_{j=1}^{n}\left(X_i^{(j)} - \bar{X}_i\right)/\sigma_i = 0$$

$$\sigma_{Z,i}^2 = \frac{1}{n-1}\sum_{j=1}^{n}\left(Z_i^{(j)} - \bar{Z}_i\right)^2 = \frac{1}{n-1}\sum_{j=1}^{n}\left(X_i^{(j)} - \bar{X}_i\right)^2 / \sigma_i^2 = 1$$

Standardize data after making data transformation.

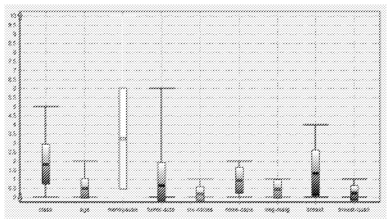
Effect: data is invariant to scaling only; for diagonal transformations distances after standardization are invariant, are based on identical units.

Note: it does not mean that all data models are performing better!

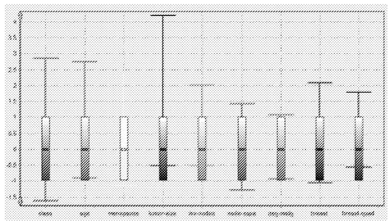
How to make data invariant to any linear transformations?

Std example

Before std



After std



Mean and std are shown using a colored bar; minimum and max values may extend outside.

Some features (ex. yellow), have large values; some (ex: gray) have small values; this may depend on units used to measure them.

Standardized data have all mean 0 and σ=1, thus contribution from different features to similarity or distance calculation is comparable.