analiza danych

1. Wprowadzenie
2. Regresja liniowa – przypomnienie i najprostszy klasyfikator - regresja logistyczna
3. Metody wyboru zmiennych oparte na równaniu regresji
4. Metody preselekcji zmiennych
5. **Transformacje – embedding ①**
6. Transformacje – embedding ②
7. Prezentacja projektów

Copyright ©2025, Joanna Szyda

Plan wykładu

1. Embedding
2. Embedding algebraiczny
   - PCA
   - SVD
   - NMF
3. Embedding z uczenia maszynowego
   - Model uczenia głębokiego
   - AE
   - CNN

Copyright ©2025, Joanna Szyda

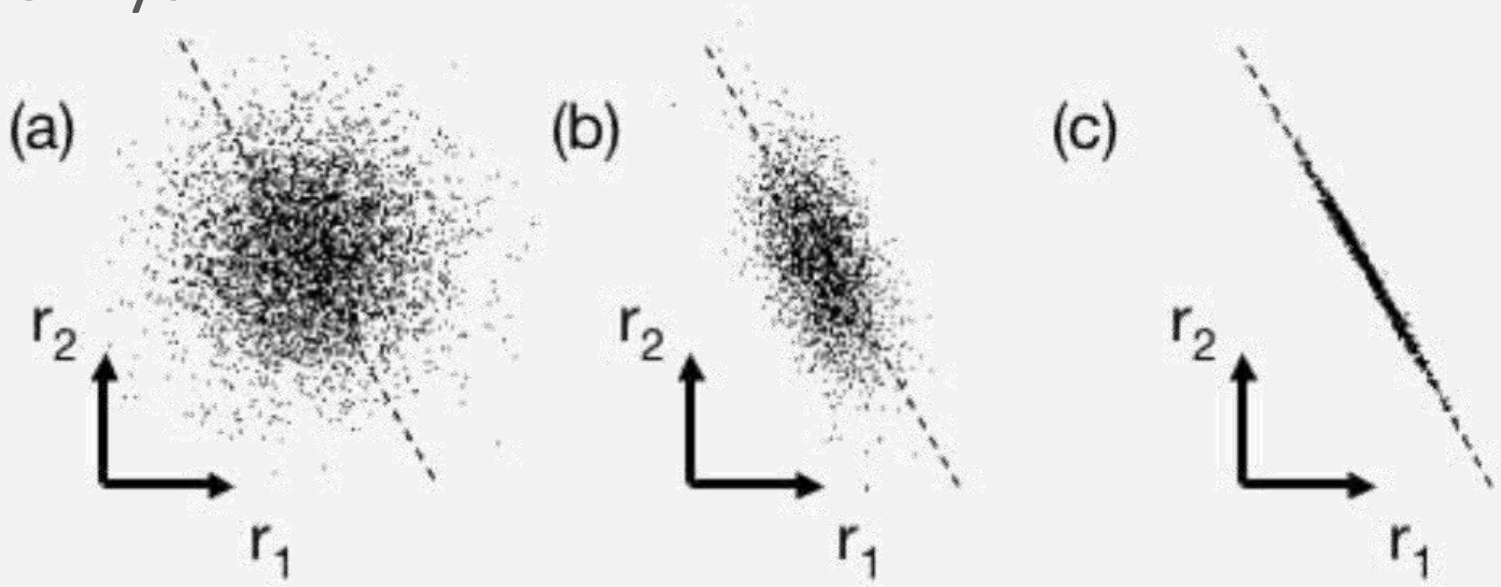
1. **Embedding**
2. Embedding algebraiczny
   - PCA
   - SVD
   - NMF
3. Embedding z uczenia maszynowego
   - Model uczenia głębokiego
   - AE
   - CNN

Copyright ©2025, Joanna Szyda

Embedding

Zmniejszenie wymiarowości danych

- ~~wybór najważniejszych zmiennych~~
- transformacja wszystkich zmiennych
- wyrażenie oryginalnych wartości przez ich skondensowaną reprezentację
- kumulowanie zmiennych o wysokiej korelacji
- odwzorowanie zestawu zmiennych do przestrzeni o mniejszej liczbie pseudozmiennych

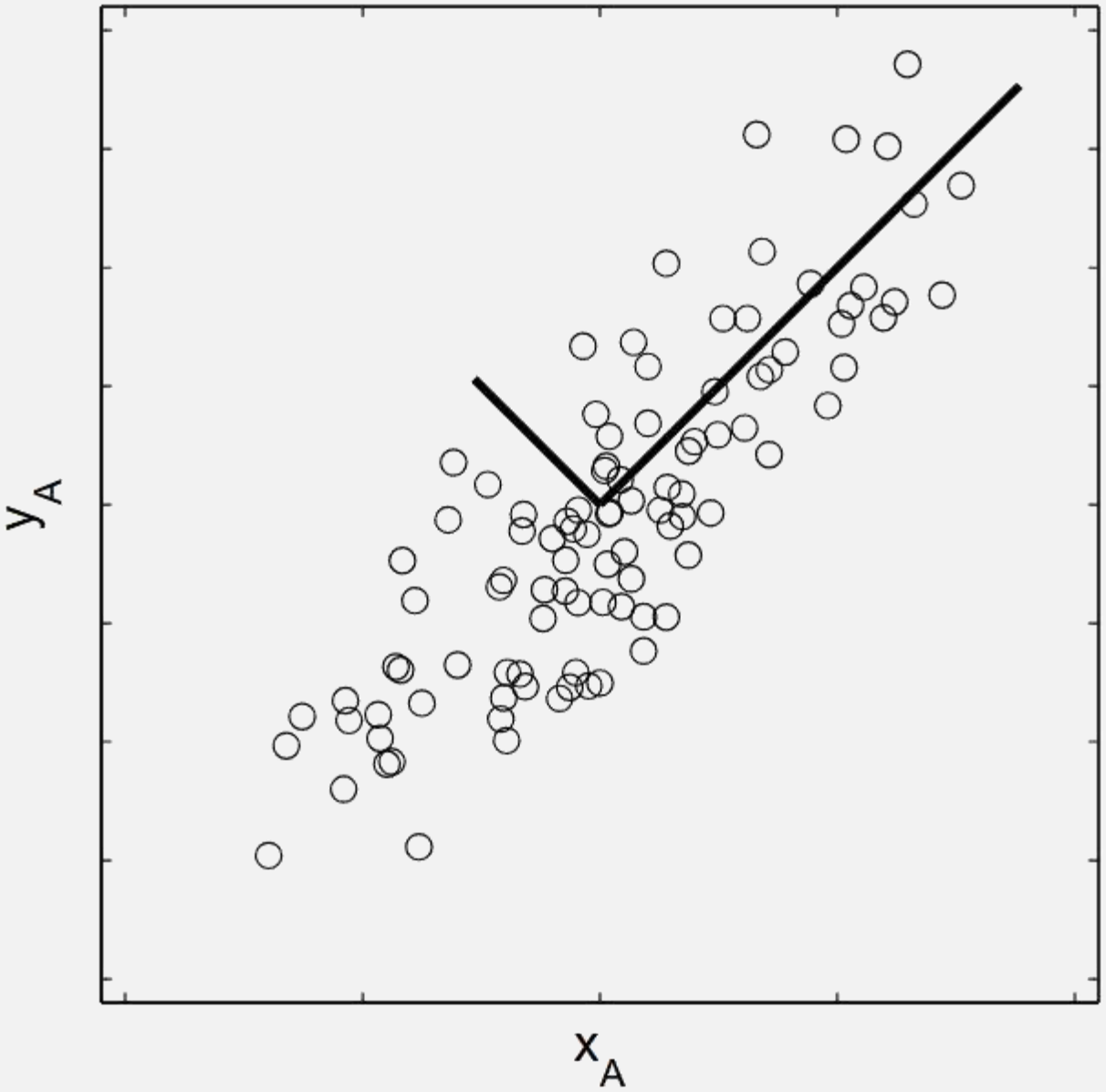


Jonathon Shlens

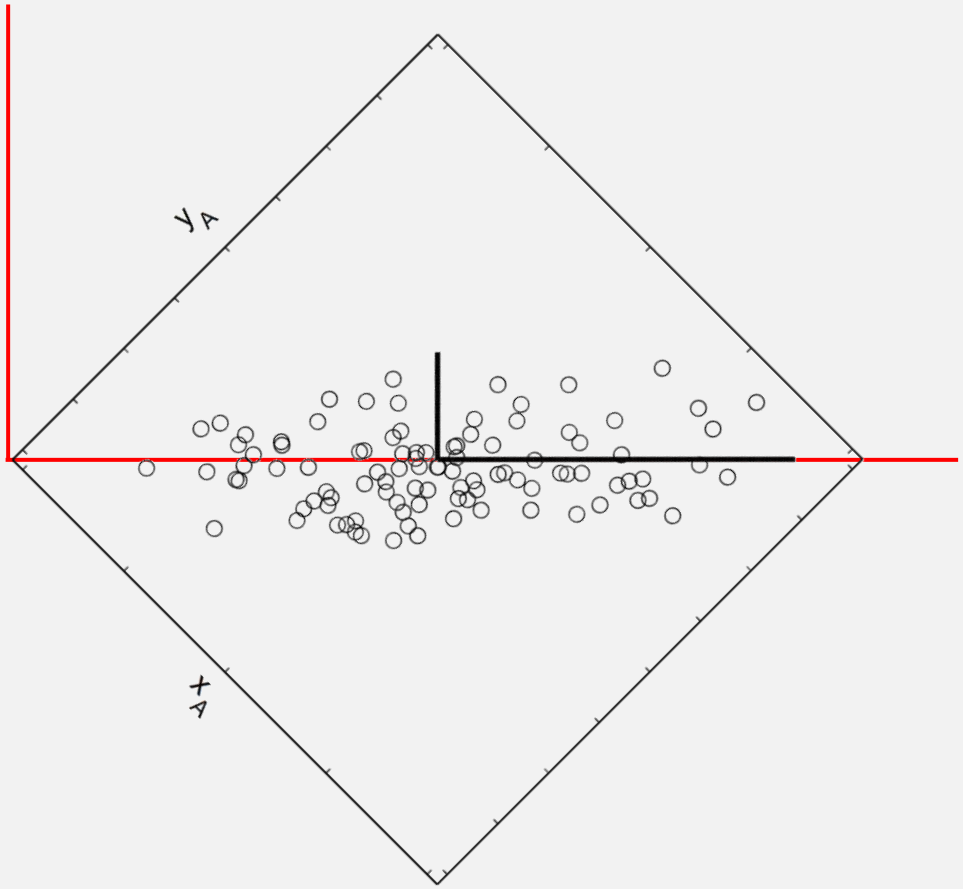
1. Embedding
2. Embedding algebraiczny
   - **PCA – principal component analysis, analiza składowych głównych**
   - SVD
   - NMF
3. Embedding z uczenia maszynowego
   - Model uczenia głębokiego
   - AE
   - CNN

Copyright ©2025, Joanna Szyda

PCA



– Transformacja danych → wybór nowych koordynatów p1, p2, o największej zmienności danych



Jonathon Shlens

PCA

Nowa macierz danych:

- embedding
- n wierszy – próbki

Brak kowariancji między zmiennymi:

$$\mathbf{C_Y} = \frac{\mathbf{Y}\mathbf{Y}^{\boldsymbol{T}}}{n-1} = \boldsymbol{I}\sigma_i^2$$

$$\mathbf{Y} = \mathbf{X}\mathbf{P}$$

Macierz transformująca $\mathbf{X}$ na $\mathbf{Y}$

Macierz składowych głównych:

- m wierszy
- ~ n kolumn

Oryginalna macierz danych:

- n wierszy – próbki
- m kolumn – zmienne

Kowariancja między zmiennymi:

$$\mathbf{C_X} = \frac{\mathbf{X}\mathbf{X}^{\boldsymbol{T}}}{n-1}$$

Copyright ©2025, Joanna Szyda

PCA

$$\mathbf{Y} = \mathbf{XP}$$

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ p_{31} & p_{32} \end{bmatrix}$$

x_{ij} = oryginalna wartość zmiennej j dla próbki i

p_{ij} = transformowana wartość zmiennej j dla próbki i

embedding → np. próbka 1 opisana zmiennymi y_{11} i y_{12} zamiast x_{11}, x_{12} i x_{13}

Copyright ©2025, Joanna Szyda

PCA

```
library(FactoMineR)
library(factoextra)

# PCA of the original data matrix X
X_pca <- PCA(X_df)

# get embedding
embedding <- X_pca$ind
```

Copyright ©2025, Joanna Szyda

PCA

- Embedding bazuje na dekompozycji macierzy kowariancji oryginalnych danych (maksymalna wariancja) → $\mathbf{C_X} = \frac{\mathbf{X}\mathbf{X}^{\boldsymbol{T}}}{n-1}$
- wymaga danych o średniej „0” → centered data, o rozkładzie normalnym
- problem dla danych rzadkich → sparse $\mathbf{X}$ → dużo „0”
- algorytm wydajny obliczeniowo → analiza dużych rozmiarów danych

Copyright ©2025, Joanna Szyda

PCA

kernel PCA

- zastosowanie PCA do przetransformowanych danych = $\boldsymbol{X}^*$
- kernel transformation
- Gaussian kernel: $\boldsymbol{X}^* = e^{-\gamma\sqrt{\sum_{i=1}^{m}(\boldsymbol{x1}_i - \boldsymbol{x2}_i)^2}}$
- $\boldsymbol{X}^*$ macierz n x n

PCA

$$\boldsymbol{X}^* = e^{-\gamma\sqrt{\sum_{i=1}^{m}(\boldsymbol{x1}_i - \boldsymbol{x2}_i)^2}}$$

```
library(kernlab)

# transformation of the original data matrix X with
Gaussian kernel
# PCA of the transformed data matrix X_star
X_star_pca <- kpcam(X, kernel = "rbfdot", kpar = list(sigma
= 0.1))

# get embedding
embedding <- X_star_pca$ind
```

Copyright ©2025, Joanna Szyda

1. Embedding
2. Embedding algebraiczny
   - PCA
   - **SVD – singular value decomposition, rozkład wartości osobliwych**
   - NMF
3. Embedding z uczenia maszynowego
   - Model uczenia głębokiego
   - AE
   - CNN

Copyright ©2025, Joanna Szyda

SVD

$$\mathbf{X} = \mathbf{U}\boldsymbol{S}\mathbf{V}^T$$

- n x m
- zawiera lewe wektory osobliwe
- kolumny → reprezentacja oryginalnych danych wzdłuż nowych koordynatów

- m x n
- zawiera prawe wektory osobliwe
- kolumny → wkład oryginalnych zmiennych do embeddingu

- m x m
- macierz diagonalna
- elementy przekątnej – pierwiastek z wartości własnych macierzy $\mathbf{X}^T\mathbf{X}$
- elementy przekątnej reprezentują zmienność ~ odchylenie standardowe oryginalnych danych wzdłuż nowych koordynatów

Copyright ©2025, Joanna Szyda

SVD

$$\mathbf{X} = \mathbf{U}\boldsymbol{S}\mathbf{V}^T$$

- embedding → $\mathbf{US}$
- wybrane kolumny macierzy $\mathbf{U}$ (k < m) odpowiadające wysokiej wariancji – elementy na przekątnej macierzy $\mathbf{S}$

Copyright ©2025, Joanna Szyda

SVD

```
# SVD of the original data matrix X
X_svd <- svd(X)
U <- X_svd$u
S <- diag(X_svd$d)
V <- t(X_svd$v)

# get embedding for k components
k <- 10
embedding <- U[, 1:k] %*% S[1:k, 1:k]
```

Copyright ©2025, Joanna Szyda

SVD

- embedding bazuje na dekompozycji oryginalnej macierzy $\mathbf{X}$ → poszukiwanie macierzy $\mathbf{U}$ i $\mathbf{S}$ które najlepiej odwzorują $\mathbf{X}$ → poszukiwanie minimum $\mathbf{X} - \mathbf{U}\boldsymbol{S}\mathbf{V}^T$
- nie wymaga danych o średniej „0”
- nie wymaga danych o rozkładzie normalnym
- SVD = PCA dla danych o rozkładzie normalnym i średniej 0
- pozwala na dekompozycję danych rzadkich
- bardziej wymagający obliczeniowo niż PCA

Copyright ©2025, Joanna Szyda

1. Embedding
2. Embedding algebraiczny
   - PCA
   - SVD
   - **NMF – nonnegative matrix factorisation, nieujemna faktoryzacja macierzy**
3. Embedding z uczenia maszynowego
   - Model uczenia głębokiego
   - AE
   - CNN

Copyright ©2025, Joanna Szyda

NMF

$$\mathbf{X} \approx \boldsymbol{W}\boldsymbol{H}$$

- oryginalna macierz danych może zawierać tylko elementy ≥ 0

- n x m
- kolumny → reprezentacja oryginalnych danych wzdłuż nowych koordynatów

- m x n
- wkład oryginalnych zmiennych do embeddingu

Copyright ©2025, Joanna Szyda

NMF

$$\mathbf{X} \approx \boldsymbol{W}\boldsymbol{H}$$

- embedding $\rightarrow \boldsymbol{W}\boldsymbol{H}$
- wybrane kolumny macierzy $\boldsymbol{W}$ (k < m)

Copyright ©2025, Joanna Szyda

NMF

```
library(NMF)

# NMF of the original data matrix X
X_nmf <- nmf(X,3,method="brunet") #k=3 embedding components
W <- X_nmf$W
H <- X_nmf$H

# get embedding
embedding <- W %*% H
```

Copyright ©2025, Joanna Szyda

NMF

- embedding bazuje na aproksymacji dekompozycji oryginalnej macierzy $\mathbf{X}$ → poszukiwanie macierzy $\mathbf{W}$ i $\mathbf{H}$ które najlepiej odwzorują $\mathbf{X}$ → poszukiwanie minimum $\mathbf{X} - \mathbf{WH}$ →
- metoda iteracyjna → potencjalne problemy: brak zbieżności, różne wyniki
- wymaga danych ≥ 0
- NMF podobne do SVD dla danych ≥ 0
- nie wymaga danych o rozkładzie normalnym
- pozwala na dekompozycję danych rzadkich
- bardziej wymagający obliczeniowo niż PCA i SVD
- $\mathbf{H}$ jest macierzą rzadką → łatwiejszy wybór ważnych zmiennych, lepsza interpretowalność

Copyright ©2025, Joanna Szyda

Inne metody

- Uniform Manifold Approximation and Projection - UMAP
- t-distributed Stochastic Neighbor Embedding - tSNE
- MDS - Multidimensional Scaling
- Sammon Mapping
- Isomap
- LLE - Local Linear Embedding

Copyright ©2025, Joanna Szyda

1. Embedding
2. Embedding algebraiczny
   - PCA
   - SVD
   - NMF
3. **Embedding z uczenia maszynowego**
   - Model uczenia głębokiego
   - AE
   - CNN

Copyright ©2025, Joanna Szyda

1. Embedding
2. Embedding algebraiczny
   - PCA
   - SVD
   - NMF
3. Embedding z uczenia maszynowego
   - Model uczenia głębokiego
   - AE
   - CNN

1. Co to jest embedding ?
2. Jakie są najważniejsze ograniczenia PCA ?
3. Jakie są najważniejsze ograniczenia NMF ?