

академия
больших
данных



Machine Learning, Lecture 4

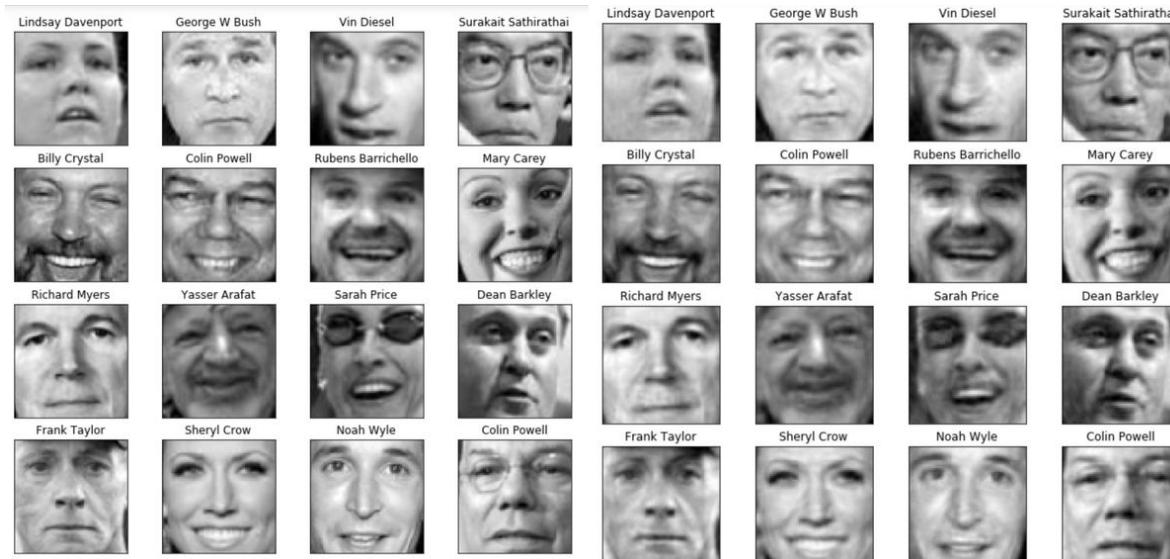
Support-Vector Machine Principal Component Analysis

Radoslav Neychev, Fall 2021

1. Support Vector Machine (SVM)
 - Linearly separable and inseparable cases
2. Dimensionality reduction and PCA
 - Connections with SVD

PCA in practice: examples

- Eigenfaces: reconstruction with $n=250$



Recap: linear classification

$$X^l = (x_i, y_i)_{i=1}^l, x_i \in R^n, y_i \in \{-1, 1\}$$

- Linear classification model

$$a(x; w, w_0) = \text{sign}(\langle x, w \rangle - w_0) \quad w \in R^n, w_0 \in R$$

- Loss function – empirical risk

$$\sum_{i=1}^{\ell} [a(x_i; w, w_0) \neq y_i] = \sum_{i=1}^{\ell} [M_i(w, w_0) < 0] \rightarrow \min_{w, w_0} .$$

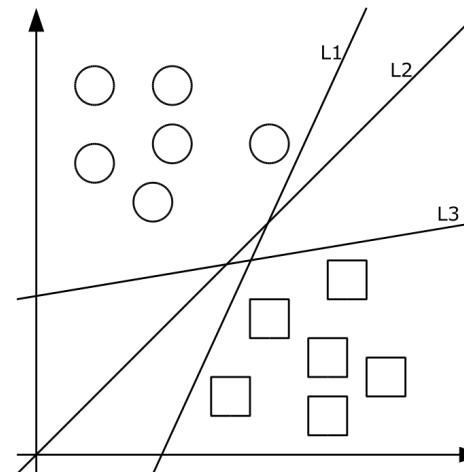
- Margin $M_i(w, w_0) = (\langle x_i, w \rangle - w_0)y_i$

Optimal classification hyperplane

- In linearly separable case:

$$\exists w, w_0 : M_i(w, w_0) = y_i(\langle w, x_i \rangle - w_0) > 0, \quad i = 1, \dots, \ell$$

- Trivia: which classification hyperplane is optimal?



Optimal classification hyperplane

- In linearly separable case:

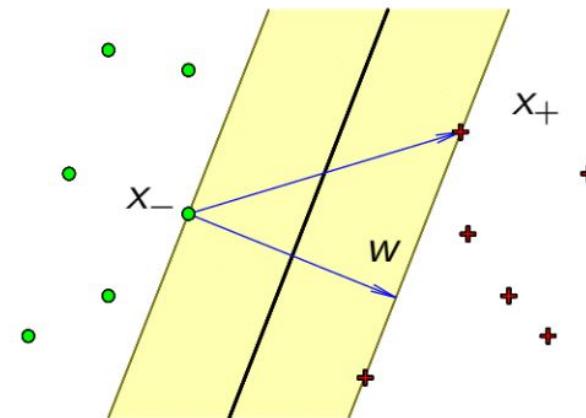
$$\exists w, w_0 : M_i(w, w_0) = y_i(\langle w, x_i \rangle - w_0) > 0, \quad i = 1, \dots, \ell$$

Weight normalization: $\min_{i=1, \dots, \ell} M_i(w, w_0) = 1$

Maximum margin hyperplane:

$$\forall x_+ : \langle w, x_+ \rangle - w_0 \geq 1$$

$$\forall x_- : \langle w, x_- \rangle - w_0 \leq -1$$



Optimal classification hyperplane

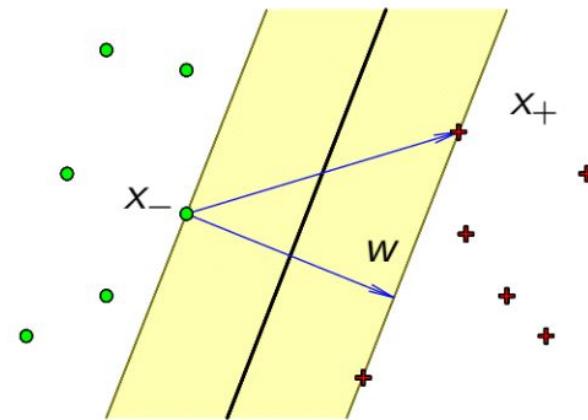
Maximum margin hyperplane:

$$\forall x_+ : \langle w, x_+ \rangle - w_0 \geq 1$$

$$\forall x_- : \langle w, x_- \rangle - w_0 \leq -1$$

Goal: maximize the margin

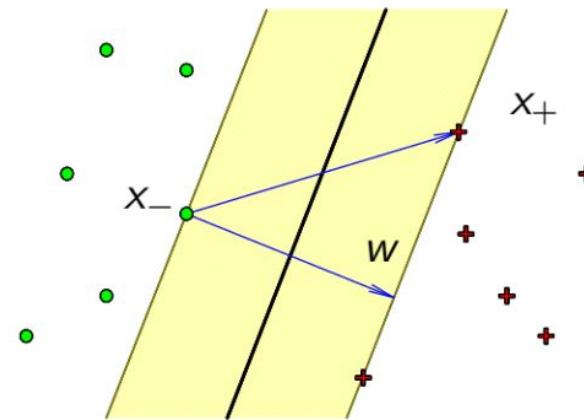
$$\frac{\langle x_+ - x_-, w \rangle}{\|w\|} \geq \frac{2}{\|w\|} \rightarrow \max$$



Optimization problem in SVM

- Separable data optimization problem:

$$\begin{cases} \frac{1}{2} \|w\|^2 \rightarrow \min_{w, w_0}; \\ M_i(w, w_0) \geq 1, \quad i = 1, \dots, \ell. \end{cases}$$



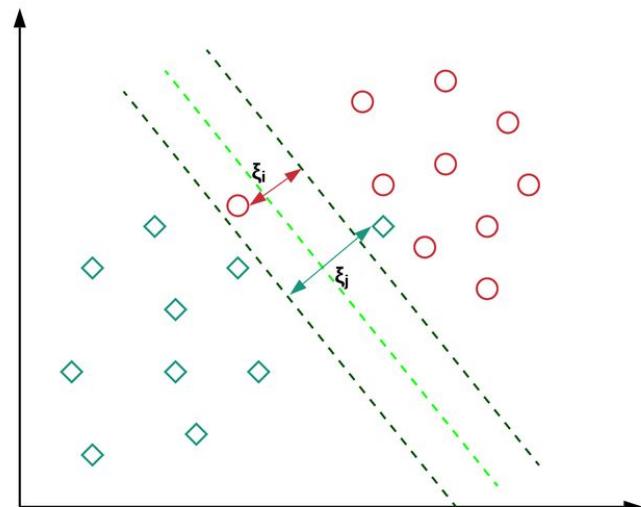
Inseparable case

- Separable data optimization problem:

$$\begin{cases} \frac{1}{2} \|w\|^2 \rightarrow \min_{w, w_0}; \\ M_i(w, w_0) \geq 1, \quad i = 1, \dots, \ell. \end{cases}$$

- Heuristic: introduce penalties for violating the margin:

$$M_i(w, w_0) \geq 1 - \xi_i, \quad i = 1, \dots, \ell;$$



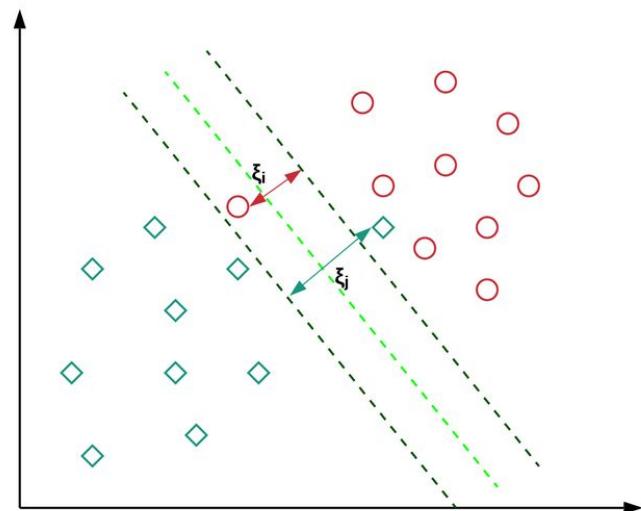
Inseparable case

- Optimization task (with constraints)

$$\begin{cases} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{\ell} \xi_i \rightarrow \min_{w, w_0, \xi}; \\ M_i(w, w_0) \geq 1 - \xi_i, \quad i = 1, \dots, \ell; \\ \xi_i \geq 0, \quad i = 1, \dots, \ell. \end{cases}$$

- Equivalent unconstrained target

$$C \sum_{i=1}^{\ell} (1 - M_i(w, w_0))_+ + \frac{1}{2}\|w\|^2 \rightarrow \min_{w, w_0}.$$



Karush-Kuhn-Tucker conditions

- Consider nonlinear optimization problem

$$\begin{cases} f(x) \rightarrow \min_x; \\ g_i(x) \leq 0, \quad i = 1, \dots, m; \\ h_j(x) = 0, \quad j = 1, \dots, k. \end{cases}$$

- The following conditions are necessary for any local minimum

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial x} = 0, \mathcal{L} = f(x) + \sum_{i=1}^m \mu_i g_i(x) + \sum_{j=1}^k \lambda_j h_j(x); \\ g_i(x) \leq 0; h_j(x) = 0; \\ \mu_i(x) \geq 0; \\ \mu_i g_i(x) = 0; \end{cases}$$

Karush-Kuhn-Tucker conditions for SVM

- Lagrangian function

$$\mathcal{L}(w, w_0, \xi, \lambda, \eta) = \frac{1}{2}||w|| - \sum_{i=1}^l \lambda_i(M_i(w, w_0) - 1) - \sum_{i=1}^l \xi_i(\lambda_i + \eta_i - C)$$

- Necessary conditions for an optimum

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0, \frac{\partial \mathcal{L}}{\partial w_0} = 0, \frac{\partial \mathcal{L}}{\partial \xi} = 0; \\ \xi_i \geq 0, \lambda_i \geq 0, \eta_i \geq 0; \\ \lambda_i = 0 \vee M_i(w, w_0) = 1 - \xi_i; \\ \eta_i = 0 \vee \xi_i = 0; \end{cases}$$

Karush-Kuhn-Tucker conditions for SVM

- Lagrangian function

$$\mathcal{L}(w, w_0, \xi, \lambda, \eta) = \frac{1}{2}||w|| - \sum_{i=1}^l \lambda_i(M_i(w, w_0) - 1) - \sum_{i=1}^l \xi_i(\lambda_i + \eta_i - C)$$

- Necessary conditions for an optimum

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i=1}^l \lambda_i y_i x_i = 0 \implies w = \sum_{i=1}^l \lambda_i y_i x_i;$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = -\sum_{i=1}^l \lambda_i y_i = 0 \implies \sum_{i=1}^l \lambda_i y_i = 0;$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = -\lambda_i - \eta_i + C = 0 \implies \eta_i + \lambda_i = C, \quad i = 1, \dots, l.$$

SVM: object classification

1. $\lambda_i = 0; \eta_i = C; \xi_i = 0; M_i \geq 1.$

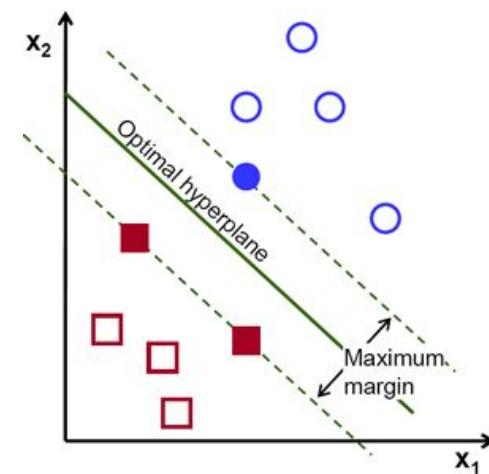
Non-informative objects (they don't contribute to a solution vector!)

2. $0 < \lambda_i < C; 0 < \eta_i < C; \xi_i = 0; M_i = 1.$

support objects (borderline)

3. $\lambda_i = C; \eta_i = 0; \xi_i > 0; M_i < 1.$

support objects (violators)



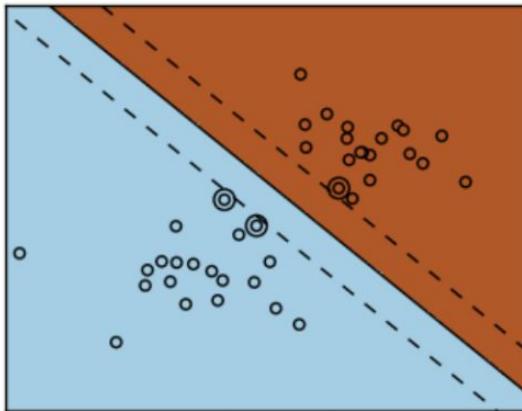
- Can be treated as upper-bound of the empirical risk

$$Q(w, w_0) = \sum_{i=1}^l [M_i(w, w_0) < 0] \leq$$

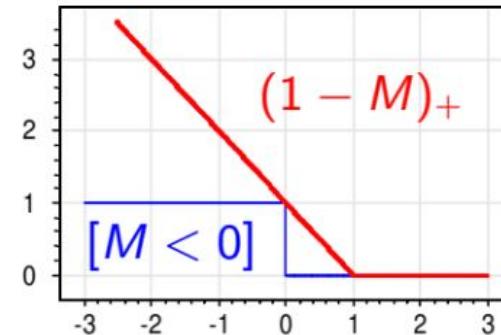
$$\leq \sum_{i=1}^l (1 - M_i(w, w_0))_+ + \frac{1}{2C} \|w\|^2 \rightarrow \min$$

Approximation *Regularization*

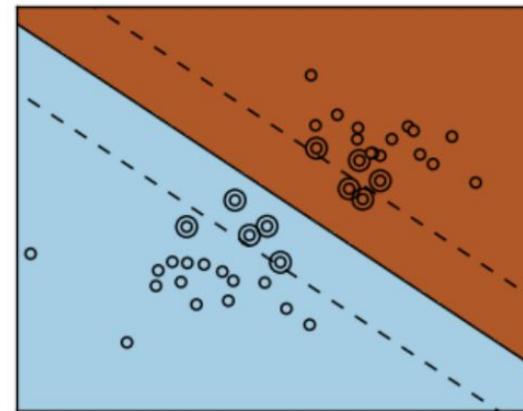
C is large: weak regularization



Hinge loss



C is small: strong regularization



Nonlinear SVM

- Can we use other similarity functions apart from dot product?
- Yes, if it can be presented as dot product in another Hilbert space!

- **Kernel function**

$$K(x, x') : X \times X \rightarrow R$$

$\exists \psi : X \rightarrow H : K(x, x') = \langle \psi(x), \psi(x') \rangle$, H - Hilbert space

- **Kernel examples**

$K(x, x') = \langle x, x' \rangle^2$ - quadratic

$K(x, x') = \langle x, x' \rangle^d$ - polynomial with degree d

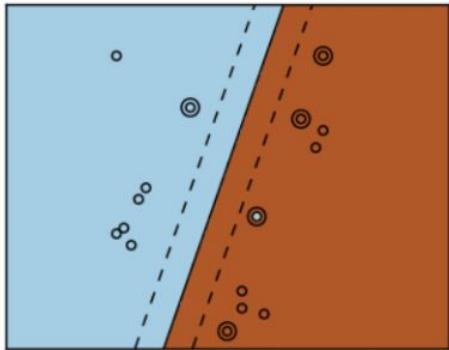
$K(x, x') = (\langle x, x' \rangle + 1)^d$ - polynomial with degree $\leq d$

$K(x, x') = \exp(-\gamma \|x - x'\|^2)$ - Radial Basis Functions (RBF) kernel

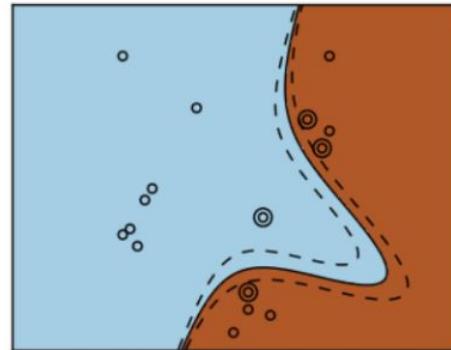
SVM: kernel examples

- SVM finds a linear boundary in linearized feature space
- But for initial features it may be nonlinear!

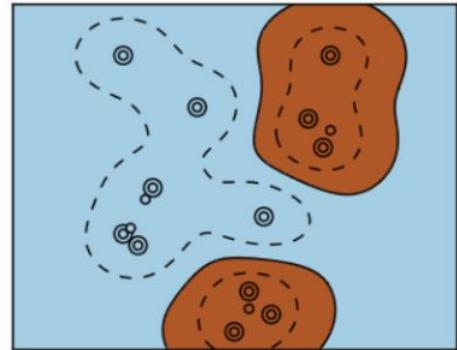
$$\langle x, x' \rangle$$



$$(\langle x, x' \rangle + 1)^d, \quad d=3$$



$$\exp(-\gamma \|x - x'\|^2)$$



Principal Component Analysis

Dimensionality reduction

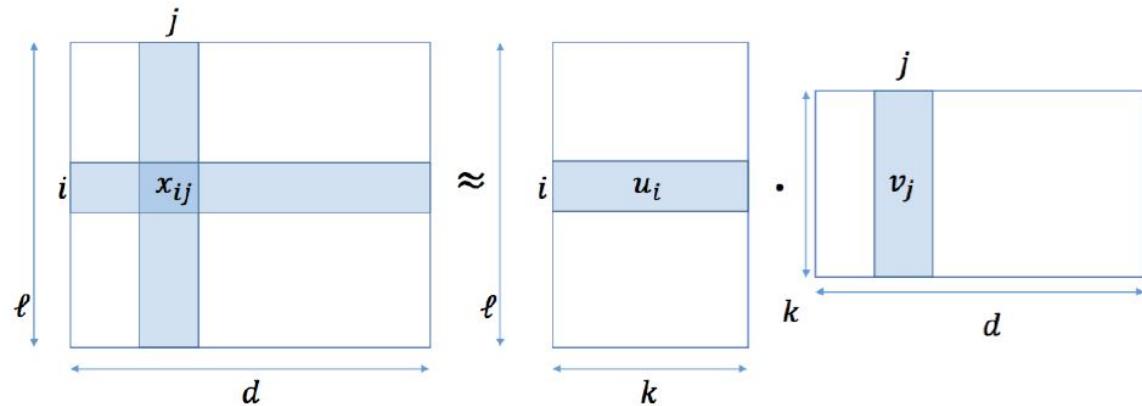
- In ML we often work with high-dimensional data
 - Hundreds or thousands of features
- Hard to visualize
- Slow training
- Some models perform worse on high-dimensional sparse input

Dimensionality reduction

- Factorization into smaller-rank matrices

$$X_{l,d} \approx U_{l,k} \cdot V_{k,d}^T$$

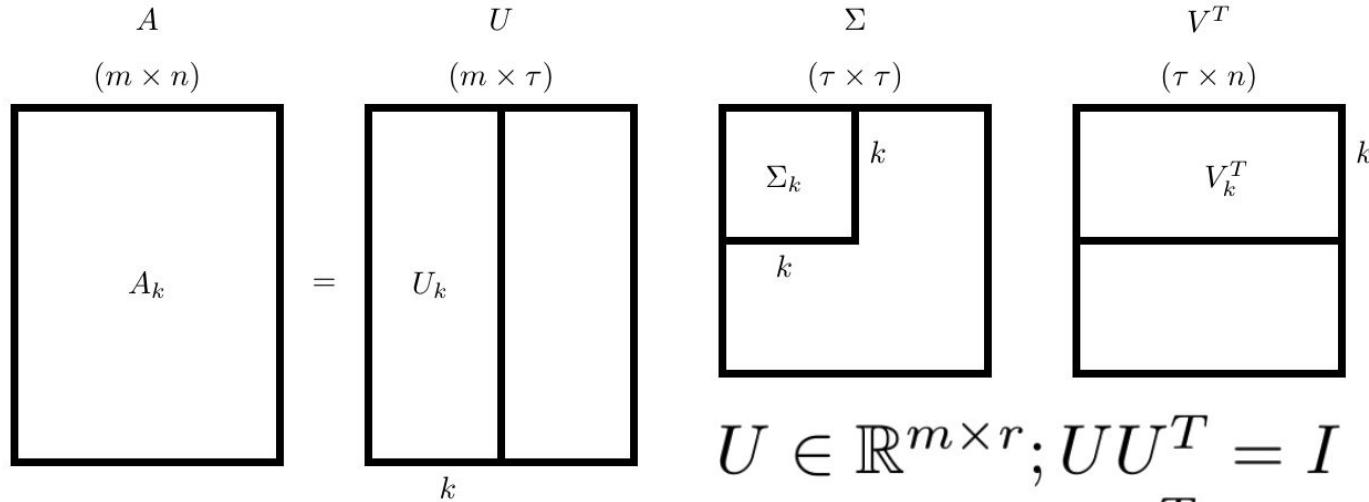
$$\|X - UV^T\| \rightarrow \min$$



Dimensionality reduction with SVD

$$A = U\Sigma V^T$$

$$A_k = U_k \Sigma_k V_k^T = (U_k \Sigma_k) V_k^T = U_k (\Sigma_k V_k^T)$$



$$U \in \mathbb{R}^{m \times r}; UU^T = I$$

$$V \in \mathbb{R}^{n \times r}; VV^T = I$$

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r); r = \text{rank}(M)$$

Theorem (Eckart-Young)

- Truncated SVD gives best low-rank approximation for a given matrix A
- More formally,

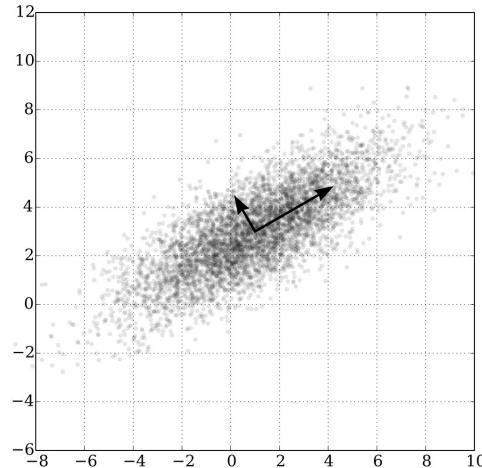
$$A_k = U_k \Sigma_k V_k^T$$

$$\forall B_k : \text{rank}(B_k) = k$$

$$\|A - B_k\|_F \geq \|A - A_k\|_F$$

PCA: projection into a subspace

- Project all data points into a smaller dimension subspace
- Maximize *variance* along new basis vectors

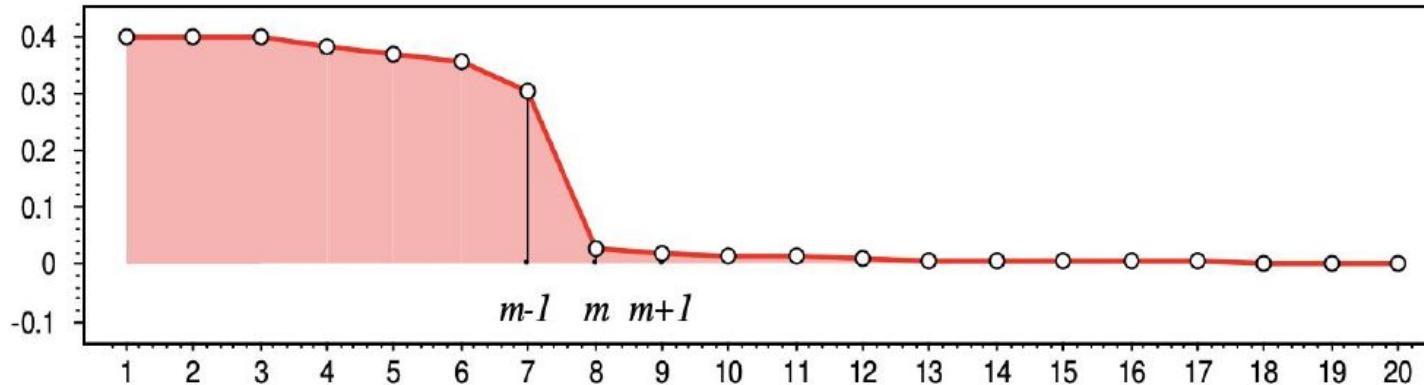


PCA: projection into a subspace

- Consider columns of matrix V new basis vectors: **principal directions**
 - Columns of matrix US are called **principal components** of the data
 - Singular values are sorted: truncated SVD gives the best projection of $\text{dim } K$

PCA: effective dimensionality

- Often data is noisy and has non-informative features
- Get rid of low-variance components in PCA



$$E_m = \frac{\|GU^\top - F\|^2}{\|F\|^2} = \frac{\lambda_{m+1} + \dots + \lambda_n}{\lambda_1 + \dots + \lambda_n} \leq \varepsilon.$$

PCA in practice

- Above said is correct only if X is centered
 - **Normalize data before PCA!**
- Dimensionality reduction:
- Reconstruction:

$$X_k = U_k \Sigma_k$$

$$\overline{X} = U_k \Sigma_k V_k^T$$

PCA in practice: examples

- Word embeddings visualization

**Let's walk through
space...**

PCA in practice: examples

- Eigenfaces: image examples



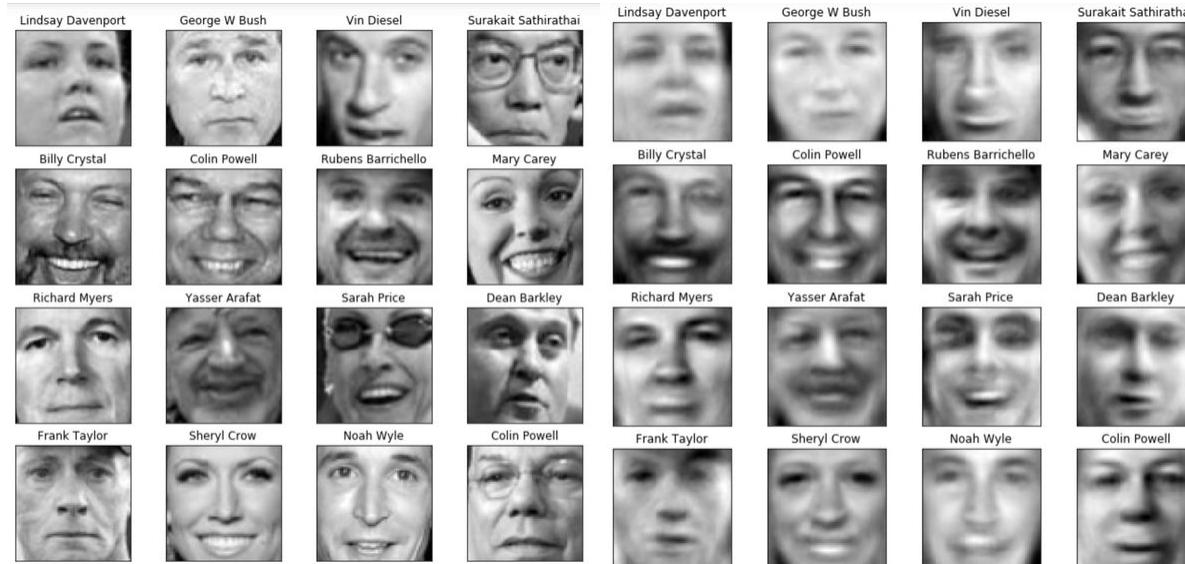
PCA in practice: examples

- Eigenfaces: top-16 components



PCA in practice: examples

- Eigenfaces: reconstruction with n=50





girafe
ai

Outro

Karush-Kuhn-Tucker conditions for SVM

- Dual optimization task for SVM

$$\begin{cases} -\mathcal{L}(\lambda) = -\sum_{i=1}^{\ell} \lambda_i + \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \lambda_i \lambda_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \rightarrow \min_{\lambda}; \\ 0 \leq \lambda_i \leq C, \quad i = 1, \dots, \ell; \\ \sum_{i=1}^{\ell} \lambda_i y_i = 0. \end{cases}$$

SVM: general solution

- Use the dual optimization problem solution

$$\begin{cases} w = \sum_{i=1}^l \lambda_i y_i x_i; \\ w_0 = \langle w, x_i \rangle - y_i, i : \lambda_i > 0; M_i = 1. \end{cases}$$

- SVM classifier general form

$$a(x) = \text{sign}\left(\sum_{i=1}^l \lambda_i y_i \langle \mathbf{x}, \mathbf{x}_i \rangle - w_0\right).$$