Machine learning lecture slides

COMS 4771 Fall 2020

Multivariate Gaussians and PCA

Outline

- Multivariate Gaussians
- Eigendecompositions and covariance matrices
- Principal component analysis
- Principal component regression and spectral regularization
- Singular value decomposition
- Examples: topic modeling and matrix completion

Multivariate Gaussians: Isotropic Gaussians

- Start with $X = (X_1, \ldots, X_d) \sim \mathrm{N}(0, I)$, i.e., $X_1, \ldots, X_d$ are iid $\mathrm{N}(0, 1)$ random variables.
  - Probability density function is product of (univariate) Gaussian densities
  - $\mathbb{E}(X_i) = 0$
  - $\mathrm{var}(X_i) = \mathrm{cov}(X_i, X_i) = 1$, $\mathrm{cov}(X_i, X_j) = 0$ for $i \neq j$
  - Arrange in mean vector $\mathbb{E}(X) = 0$, covariance matrix $\mathrm{cov}(X) = I$

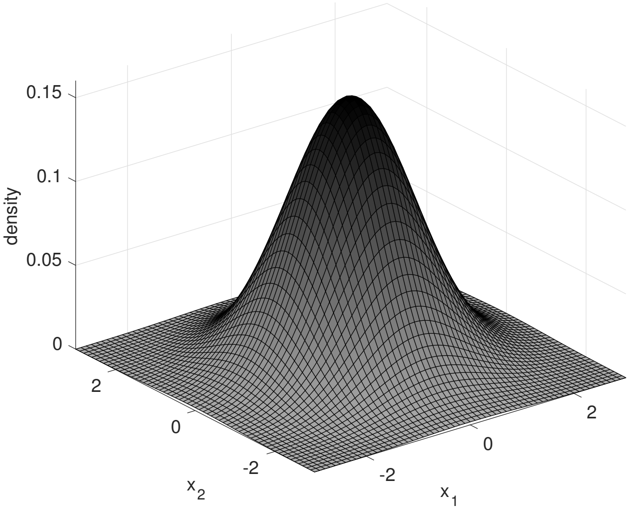


Figure 1: Density function for isotropic Gaussian in $\mathbb{R}^2$

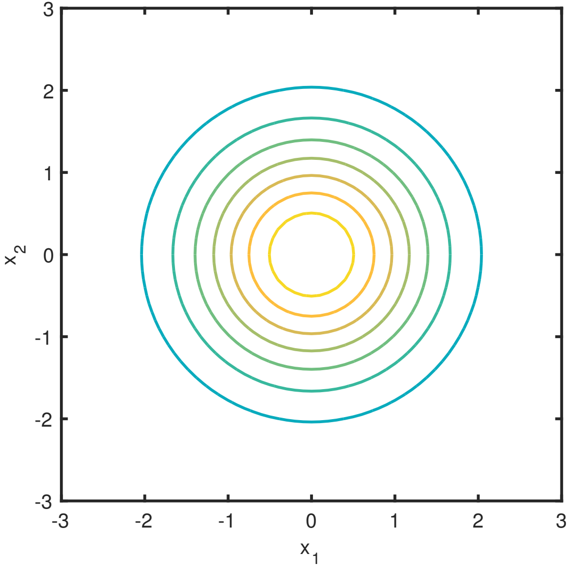


Figure 2: Density function level sets for isotropic Gaussian in $\mathbb{R}^2$

Affine transformations of random vectors

- Start with any random vector Z, then apply linear transformation, followed by translation
  - $X := MZ + \mu$, for $M \in \mathbb{R}^{k \times d}$ and $\mu \in \mathbb{R}^k$
  - Fact: $\mathbb{E}(X) = M\mathbb{E}(Z) + \mu$, $\operatorname{cov}(X) = M \operatorname{cov}(Z) M^\intercal$
  - E.g., let $u \in \mathbb{R}^d$ be a unit vector ($\|u\|_2 = 1$), and $X := u^\intercal Z$ (projection of X along direction u). Then $\mathbb{E}(X) = u^\intercal \mathbb{E}(Z)$, and $\operatorname{var}(X) = u^\intercal \operatorname{cov}(Z) u$.
- Note: These transformations work for random vectors with any distribution, not just Gaussian distributions.
  - However, it is convenient to illustrate the effect of these transformations on Gaussian distributions, since the "shape" of the Gaussian pdf is easy to understand.

Multivariate Gaussians: General Gaussians

- If $Z \sim \mathrm{N}(0, I)$ and $X = MZ + \mu$, we have $\mathbb{E}(X) = \mu$ and $\mathrm{cov}(X) = MM^\intercal$
  - Assume $M \in \mathbb{R}^{d \times d}$ is invertible (else we get a degenerate Gaussian distribution).
  - We say $X \sim \mathrm{N}(\mu, MM^\intercal)$
  - Density function given by

$$p(x) = \frac{1}{(2\pi)^{d/2}|MM^\intercal|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\intercal (MM^\intercal)^{-1}(x-\mu)\right).$$

- Note: every non-singular covariance matrix Σ can be written as $MM^\intercal$ for some non-singular matrix M. (We'll see why soon.)

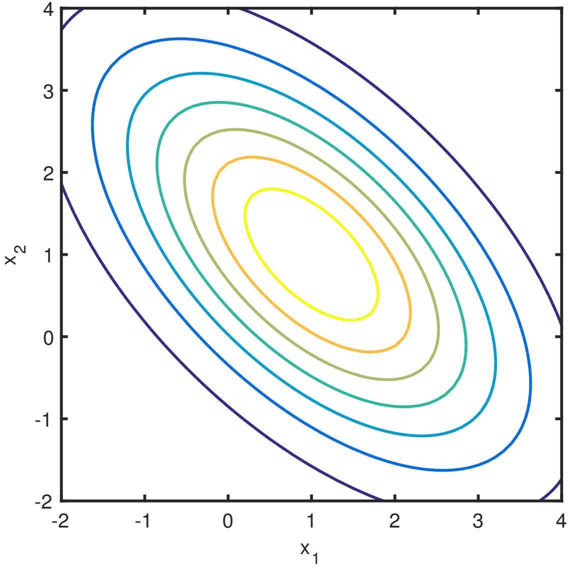


Figure 3: Density function level sets for anisotropic Gaussian in $\mathbb{R}^2$

Inference with multivariate Gaussians (2)

- Bivariate case: $(X_1, X_2) \sim \mathrm{N}(\mu, \Sigma)$ in $\mathbb{R}^2$

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2} \end{bmatrix}$$

- What is the distribution of X_2?
  - $\mathrm{N}(\mu_2, \Sigma_{2,2})$
- What is the distribution of $X_2 \mid X_1 = x_1$?
  - Miracle 1: it is a Gaussian distribution
  - Miracle 2: mean provided by linear prediction of X_2 from X_1 with smallest MSE
  - Miracle 3: variance doesn't depend on x_1

Inference with multivariate Gaussians (2)

- What is the distribution of $X_2 \mid X_1 = x_1$?
  - Miracle 1: it is a Gaussian distribution
  - Miracle 2: mean provided by linear prediction of X_2 from X_1 with smallest MSE
  - Miracle 3: variance doesn't depend on x_1
  - OLS with X_1 as input variable and X_2 as output variable:

$$x_1 \mapsto \hat{m}x_1 + \hat{\theta}$$

where

$$\hat{m} = \frac{\operatorname{cov}(X_1, X_2)}{\operatorname{var}(X_1)} = \frac{\Sigma_{1,2}}{\Sigma_{1,1}},$$
$$\hat{\theta} = \mathbb{E}(X_2) - \hat{m}\mathbb{E}(X_1) = \mu_2 - \hat{m}\mu_1.$$

  - Therefore:

$$\begin{aligned}\mathbb{E}[X_2 \mid X_1 = x_1] &= \hat{m}x_1 + \hat{\theta} \\ &= \mu_2 + \hat{m}(x_1 - \mu_1) \\ &= \mu_2 + \frac{\Sigma_{1,2}}{\Sigma_{1,1}}(x_1 - \mu_1)\end{aligned}$$

Inference with multivariate Gaussians (3)

- What is the distribution of $X_2 \mid X_1 = x_1$?
  - Miracle 1: it is a Gaussian distribution
  - Miracle 2: mean provided by linear prediction of X_2 from X_1 with smallest MSE
  - Miracle 3: variance doesn't depend on x_1

$$\begin{aligned}
\operatorname{var}(X_2 \mid X_1 = x_1) &= \mathbb{E}[\operatorname{var}(X_2 \mid X_1)] \\
&= \operatorname{var}(X_2) - \operatorname{var}(\mathbb{E}[X_2 \mid X_1]) \\
&= \Sigma_{2,2} - \operatorname{var}(\hat{m}X_1 + \hat{\theta}) \\
&= \Sigma_{2,2} - \hat{m}^2 \operatorname{var}(X_1) \\
&= \Sigma_{2,2} - \frac{\Sigma_{1,2}^2}{\Sigma_{1,1}^2}\Sigma_{1,1} \\
&= \Sigma_{2,2} - \frac{\Sigma_{1,2}^2}{\Sigma_{1,1}}.
\end{aligned}$$

Inference with multivariate Gaussians (4)

- Beyond bivariate Gaussians: same as above, but just writing things properly using matrix notations

$$\begin{aligned} \mathbb{E}[X_2 \mid X_1 = x_1] &= \mu_2 + \Sigma_{2,1}\Sigma_{1,1}^{-1}(x_1 - \mu_1) \\ \operatorname{cov}(X_2 \mid X_1 = x_1) &= \Sigma_{2,2} - \Sigma_{2,1}\Sigma_{1,1}^{-1}\Sigma_{1,2} \end{aligned}$$

Eigendecomposition (1)

- Every symmetric matrix $M \in \mathbb{R}^{d \times d}$ has d real *eigenvalues*, which we arrange as

$$\lambda_1 \geq \cdots \geq \lambda_d$$

- Can choose corresponding orthonormal *eigenvectors*

$$v_1, \ldots, v_d \in \mathbb{R}^d$$

  - This means

$$M v_i = \lambda_i v_i$$

    for each $i = 1, \ldots, d$, and

$$v_i^{\top} v_j = \mathbf{1}_{\{i=j\}}$$

Eigendecomposition (2)

- Arrange $v_1, \ldots, v_d$ in an *orthogonal matrix* $V := [v_1 | \cdots | v_d]$
  - $V^\intercal V = I$ and $V V^\intercal = \sum_{i=1}^{d} v_i v_i^\intercal = I$
  - Therefore,

$$\begin{aligned} M &= M V V^\intercal \\ &= \sum_{i=1}^{d} M v_i v_i^\intercal \\ &= \sum_{i=1}^{d} \lambda_i v_i v_i^\intercal \end{aligned}$$

- This is our preferred way to express the *eigendecomposition*
  - Also called *spectral decomposition*
  - Can also write $M = V \Lambda V^\intercal$, where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$
  - The matrix V *diagonalizes* M:

$$V^\intercal M V = \Lambda$$

Covariance matrix (1)

- $A \in \mathbb{R}^{n \times d}$ is data matrix
- $\Sigma := A^\mathsf{T} A = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^\mathsf{T}$ is *(empirical) second-moment matrix*
  - If $\frac{1}{n}\sum_{i=1}^{n} x_i = 0$ (data are "centered"), this is the *(empirical) covariance matrix*
  - For purpose of exposition, just say/write "(co)variance" even though "second-moment" is technically correct
- For any unit vector $u \in \mathbb{R}^d$,

$$u^\mathsf{T} \Sigma u = \frac{1}{n}\sum_{i=1}^{n}(u^\mathsf{T} x_i)^2$$

is (empirical) variance of data along direction u

Covariance matrix (2)

- Note: some pixels in OCR data have very little (or zero!) variation
  - These are "coordinate directions" (e.g., $u = (1, 0, \ldots, 0)$)
  - Probably can/should ignore these!

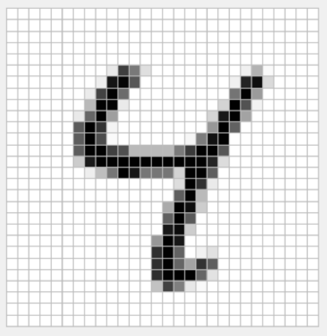


Figure 4: Which pixels are likely to have very little variance?

Top eigenvector

- Σ is symmetric, so can write eigendecomposition

$$\Sigma = \sum_{i=1}^{d} \lambda_i v_i v_i^{\mathsf{T}}$$

- In which direction is variance maximized?
- Answer: v_1, corresponding to largest eigenvalue λ_1
  - Called the *top eigenvector*
  - This follows from the following characterization of v_1:

$$v_1^{\mathsf{T}} \Sigma v_1 = \max_{u \in \mathbb{R}^d : \|u\|_2 = 1} u^{\mathsf{T}} \Sigma u = \lambda_1 .$$

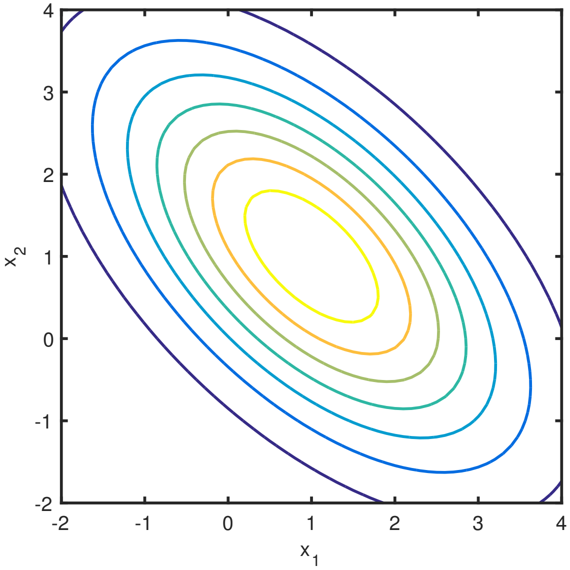


Figure 5: What is the direction of the top eigenvector for the covariance of this Gaussian?

Top k eigenvectors

- What about among directions orthogonal to v_1?
  - Answer: v_2, corresponding to second largest eigenvalue λ_2
- (Note: all eigenvalues of Σ are non-negative!)
- For any k, $V_k := [v_1 | \cdots | v_k]$ satisfies

$$\sum_{i=1}^{k} v_i^\top \Sigma v_i = \operatorname{tr}(V_k^\top \Sigma V_k) = \max_{U \in \mathbb{R}^{d \times k} : U^\top U = I} \operatorname{tr}(U^\top \Sigma U) = \sum_{i=1}^{k} \lambda_i$$

(the *top k eigenvectors*)

Principal component analysis

- k-dimensional *principal components analysis (PCA)* mapping:

$$\varphi(x) = (x^\mathsf{T} v_1, \dots, x^\mathsf{T} v_k) = V_k^\mathsf{T} x \in \mathbb{R}^k$$

  where $V_k = [v_1 | \cdots | v_k] \in \mathbb{R}^{d \times k}$
- (Only really makes sense when $\lambda_k > 0$.)
- This is a form of *dimensionality reduction* when $k < d$.

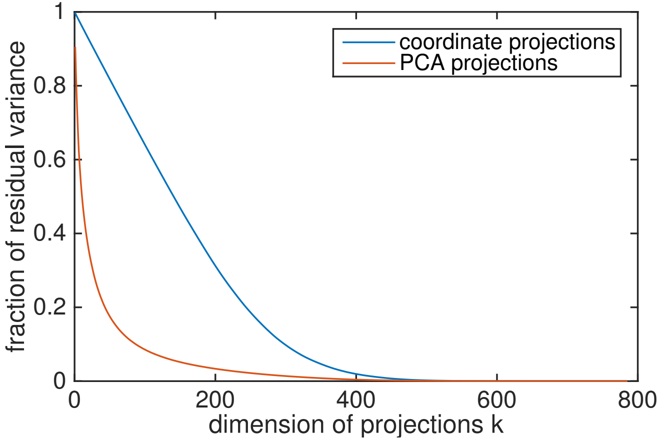


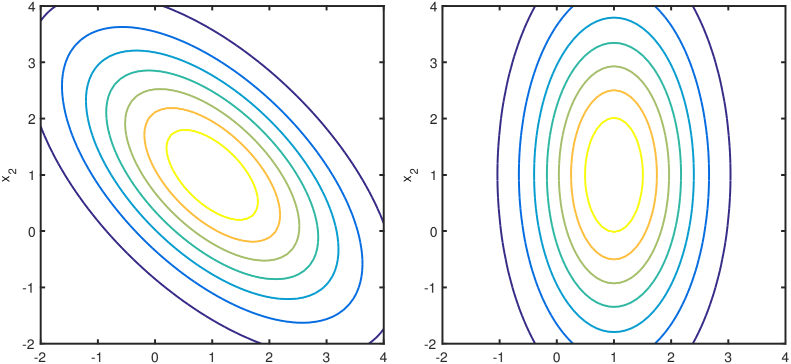
Figure 6: Fraction of residual variance from projections of varying dimension

Covariance of data upon PCA mapping

- Covariance of data upon PCA mapping:

$$\frac{1}{n}\sum_{i=1}^{n}\varphi(x_i)\varphi(x_i)^\intercal = \frac{1}{n}\sum_{i=1}^{n} V_k^\intercal x_i x_i^\intercal V_k = V_k^\intercal \Sigma V_k = \Lambda_k$$

  where Λ_k is diagonal matrix with $\lambda_1, \ldots, \lambda_k$ along diagonal.
  - In particular, coordinates in $\varphi(x)$-representation are uncorrelated.



PCA and linear regression

- Use k-dimensional PCA mapping $\varphi(x) = V_k^\mathsf{T} x$ with ordinary least squares
- (Assume rank of A is at least k, so $A^\mathsf{T} A$ has $\lambda_k > 0$)
- Data matrix is

$$\frac{1}{\sqrt{n}} \begin{bmatrix} \leftarrow & \varphi(x_1)^\mathsf{T} & \rightarrow \\ & \vdots & \\ \leftarrow & \varphi(x_n)^\mathsf{T} & \rightarrow \end{bmatrix} = \frac{1}{\sqrt{n}} \begin{bmatrix} \leftarrow & x_1^\mathsf{T} V_k & \rightarrow \\ & \vdots & \\ \leftarrow & x_n^\mathsf{T} V_k & \rightarrow \end{bmatrix} = AV_k \in \mathbb{R}^{n \times k}$$

- Therefore, OLS solution is

$$\begin{aligned} \hat{\beta} &= (V_k^\mathsf{T} A^\mathsf{T} A V_k)^{-1} (AV_k)^\mathsf{T} b \\ &= \Lambda_k^{-1} V_k^\mathsf{T} A^\mathsf{T} b \end{aligned}$$

(Note: here $\hat{\beta} \in \mathbb{R}^k$.)

Principal component regression

- Use $\hat{\beta} = \Lambda_k^{-1} V_k^\mathsf{T} A^\mathsf{T} b$ to predict on new $x \in \mathbb{R}^d$:

$$\begin{aligned} \varphi(x)^\mathsf{T} \hat{\beta} &= (V_k^\mathsf{T} x)^\mathsf{T} \Lambda_k^{-1} V_k^\mathsf{T} A^\mathsf{T} b \\ &= x^\mathsf{T} (V_k \Lambda_k^{-1} V_k^\mathsf{T})(A^\mathsf{T} b) \end{aligned}$$

- So "effective" weight vector (that acts directly on x rather than $\varphi(x)$) is given by

$$\hat{w} := (V_k \Lambda_k^{-1} V_k^\mathsf{T})(A^\mathsf{T} b).$$

- This is called *principal component regression (PCR)* (here, k is hyperparameter)
- Alternative hyper-parameterization: $\lambda > 0$; same as before but using the largest k such that $\lambda_k \geq \lambda$.

Spectral regularization

- PCR and ridge regression are examples of *spectral regularization*.
- For a function $g\colon \mathbb{R} \to \mathbb{R}$, write $g(M)$ to mean

$$g(M) = \sum_{i=1}^{d} g(\lambda_i) v_i v_i^{\mathsf{T}}$$

where M has eigendecomposition $M = \sum_{i=1}^{d} \lambda_i v_i v_i^{\mathsf{T}}$.
  - I.e., g is applied to eigenvalues of M
  - Generalizes effect of polynomials: e.g., $g(z) = z^2$

$$M^2 = (V\Lambda V^{\mathsf{T}})(V\Lambda V^{\mathsf{T}}) = V\Lambda^2 V^{\mathsf{T}}.$$

- **Claim**: Can write each of PCR and ridge regression as

$$\hat{w} = g(A^{\mathsf{T}}A)A^{\mathsf{T}}b$$

for appropriate function g (depending on λ).

Comparing ridge regression and PCR

- $\hat{w} = g(A^\intercal A)A^\intercal b$
- Ridge regression (with parameter λ): $g(z) = \frac{1}{z+\lambda}$
- PCR (with parameter λ): $g(z) = \mathbf{1}_{\{z \geq \lambda\}} \cdot \frac{1}{z}$
- Interpretation:
  - PCR uses directions with sufficient variability; ignores the rest
  - Ridge artificially inflates the variance in all directions

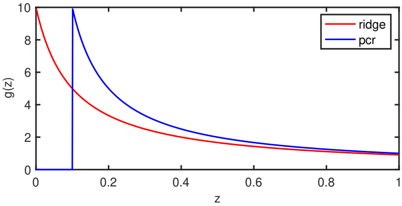


Figure 7: Comparison of ridge regression and PCR

Matrix factorization

- Let $A = \begin{bmatrix} \leftarrow & x_1^\intercal & \rightarrow \\ & \vdots & \\ \leftarrow & x_n^\intercal & \rightarrow \end{bmatrix} \in \mathbb{R}^{n \times d}$ (forget the $1/\sqrt{n}$ scaling)
- Try to approximate A with BC, where $B \in \mathbb{R}^{n \times k}$ and $C \in \mathbb{R}^{k \times d}$, to minimize $\|A - BC\|_F^2$.
  - Here, $\|\cdot\|_F$ is a matrix norm called *Frobenius norm*, which treats the $n \times d$ matrix as a vector in nd-dimensional Euclidean space
  - Think of B as the encodings of the data in A
  - "Dimension reduction" when $k < d$
- **Theorem** (Schmidt, 1907; Eckart-Young, 1936): Optimal solution is given by truncating the *singular value decomposition (SVD)* of A

Singular value decomposition

- Every matrix $A \in \mathbb{R}^{n \times d}$—say, with rank r—can be written as

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^\top$$

  where
  - $\sigma_1 \geq \cdots \geq \sigma_r > 0$ (*singular values*)
  - $u_1, \ldots, u_r \in \mathbb{R}^n$ (orthonormal *left singular vectors*)
  - $v_1, \ldots, v_r \in \mathbb{R}^d$ (orthonormal *right singular vectors*)
- Can also write as

$$A = USV^\top$$

  where
  - $U = [u_1 | \cdots | u_r] \in \mathbb{R}^{n \times r}$, satisfies $U^\top U = I$
  - $S = \mathrm{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r \times r}$
  - $V = [v_1 | \cdots | v_r] \in \mathbb{R}^{d \times r}$, satisfies $V^\top V = I$

Truncated SVD

- Let A have SVD $A = \sum_{i=1}^{r} \sigma_i u_i v_i^\intercal$ (rank of A is r)
- Truncate at rank k (for any $k \leq r$): *rank-k SVD*

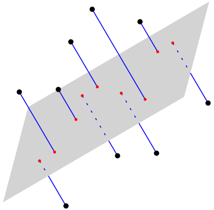
$$A_k := \sum_{i=1}^{k} \sigma_i u_i v_i^\intercal$$

- Can write as $A_k := U_k S_k V_k^\intercal$, where
  - $U_k = [u_1 | \cdots | u_k] \in \mathbb{R}^{n \times k}$, satisfies $U^\intercal U = I$
  - $S_k = \mathrm{diag}(\sigma_1, \ldots, \sigma_k) \in \mathbb{R}^{k \times k}$
  - $V_k = [v_1 | \cdots | v_k] \in \mathbb{R}^{d \times r}$, satisfies $V^\intercal V = I$
- **Theorem** (Schmidt/Eckart-Young):

$$\|A - A_k\|_F^2 = \min_{M : \mathrm{rank}(M) = k} \|A - M\|_F^2 = \sum_{i=k+1}^{r} \sigma_i^2$$

Encoder/decoder interpretation (1)

- Encoder: $x \mapsto \varphi(x) = V_k^\mathsf{T} x \in \mathbb{R}^k$
  - Encoding rows of A: $AV_k = U_k S_k$
- Decoder: $z \mapsto V_k z \in \mathbb{R}^d$
  - Decoding rows of $U_k S_k$: $U_k S_k V_k^\mathsf{T} = A_k$
- Same as k-dimensional PCA mapping!
  - $A^\mathsf{T} A = V S^2 V^\mathsf{T}$, so eigenvectors of $A^\mathsf{T} A$ are right singular vectors of A, non-zero eigenvalues are squares of the singular values
  - PCA/SVD finds k-dimensional subspace of smallest sum of squared distances to data points.



- Example: OCR data, compare original image to decoding of k-dimensional PCA encoding ($k \in \{1, 10, 50, 200\}$)

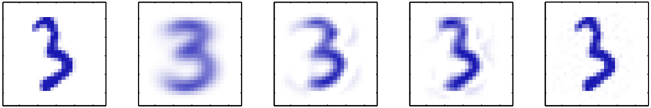


Figure 9: PCA compression of MNIST digit

Application: Topic modeling (1)

- Start with n documents, represent using "bag-of-words" count vectors
- Arrange in matrix $A \in \mathbb{R}^{n \times d}$, where d is vocabulary size

	aardvark	abacus	abalone	$\cdots$
doc 1	3	0	0	$\cdots$
doc 2	7	0	4	$\cdots$
doc 3	2	4	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Application: Topic modeling (2)

- Rank k SVD provides an approximate factorization

$$A \approx BC$$

  where $B \in \mathbb{R}^{n \times k}$ and $C \in \mathbb{R}^{k \times d}$
- This use of SVD is called *Latent Semantic Analysis (LSA)*
- Interpret rows of C as "topics"
- $B_{i,t}$ is "weight" of document i on topic t
- If rows of C were probability distributions, could interpret as $C_{t,w}$ as probability that word w appears in topic t

Application: Matrix completion (1)

- Start with ratings of movies given by users
- Arrange in a matrix $A \in \mathbb{R}^{n \times d}$, where $A_{i,j}$ is rating given by user i for movie j.
  - Netflix: $n = 480000$, $d = 18000$; on average, each user rates 200 movies
  - Most entries of A are unknown
- Idea: Approximate A with low-rank matrix, i.e., find

$$B = \begin{bmatrix} \leftarrow & b_1^\intercal & \rightarrow \\ & \vdots & \\ \leftarrow & b_n^\intercal & \rightarrow \end{bmatrix} \in \mathbb{R}^{n \times k}, \qquad C = \begin{bmatrix} \uparrow & & \uparrow \\ c_1 & \cdots & c_d \\ \downarrow & & \downarrow \end{bmatrix} \in \mathbb{R}^{k \times d}$$

  with goal of minimizing $\|A - BC\|_F^2$
- Note: If all entries of A were observed, we could do this with truncated SVD.

Application: Matrix completion (2)

- Need to find a low-rank approximation without all of A: *(low-rank) matrix completion*
  - Lots of ways to do this
  - Popular way (used in Netflix competition): based on "stochastic gradient descent" (discussed later)
  - Another way: fill in missing entries with plug-in estimates (based on a statistical model), then compute truncated SVD as usual

Feature representations from matrix completion

- MovieLens data set ($n = 6040$ users, $d = 3952$ movies, $|\Omega| = 800000$ ratings)
- Fit B and C by using a standard matrix completion method (based on SGD, discussed later)
- Are $c_1, \dots, c_d \in \mathbb{R}^k$ useful feature vectors for movies?

- Some nearest-neighbor pairs $(c_j, \mathrm{NN}(c_j))$:
  - Toy Story (1995), Toy Story 2 (1999)
  - Sense and Sensibility (1995), Emma (1996)
  - Heat (1995), Carlito's Way (1993)
  - The Crow (1994), Blade (1998)
  - Forrest Gump (1994), Dances with Wolves (1990)
  - Mrs. Doubtfire (1993), The Bodyguard (1992)