

Assignment Project Exam Help

Machine learning lecture slides

COMS 4771 Fall 2020

<https://powcoder.com>

Add WeChat powcoder

Multivariate Gaussians and PCA

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

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

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Multivariate Gaussians: Isotropic Gaussians

- ▶ Start with $X = (X_1, \dots, X_d) \sim N(0, I)$, i.e., X_1, \dots, X_d are iid $N(0, 1)$ random variables.

- ▶ Probability density function is product of (univariate) Gaussian densities

- ▶ $\mathbb{E}(X_i) = 0$

- ▶ $\text{var}(X_i) = \text{cov}(X_i, X_i) = 1, \text{cov}(X_i, X_j) = 0$ for $i \neq j$

- ▶ Arrange in mean vector $\mathbb{E}(X) = 0$, covariance matrix $\text{cov}(X) = I$

Add WeChat powcoder

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

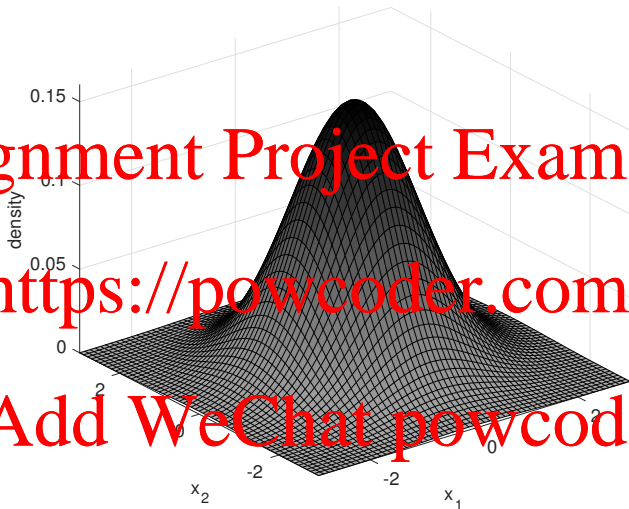


Figure 1: Density function for isotropic Gaussian in \mathbb{R}^2

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

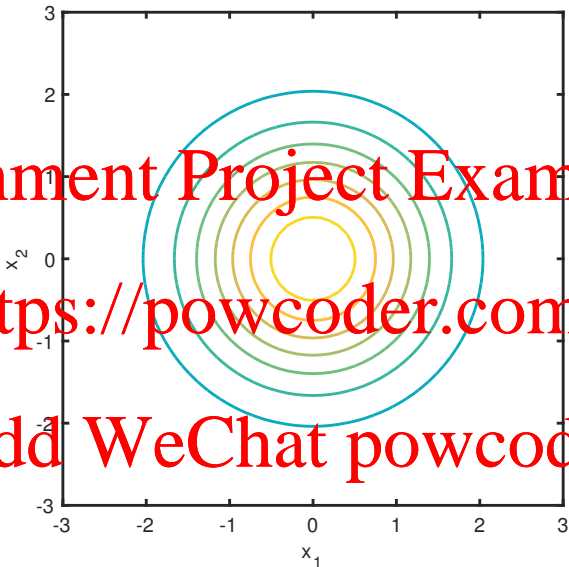


Figure 2: Density function level sets for isotropic Gaussian in \mathbb{R}^2

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: $\mathbb{E}(X) = M\mathbb{E}(Z) + \mu$, $\text{cov}(X) = M\text{cov}(Z)M^T$

▶ E.g., let $u \in \mathbb{R}^d$ be a unit vector ($\|u\|_2 = 1$), and $X := u^T Z$ (projection of X along direction u). Then $\mathbb{E}(X) = u^T \mathbb{E}(Z)$, and $\text{var}(X) = u^T \text{cov}(Z)u$.

- ▶ 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 $Z \sim \mathcal{N}(0, I)$ and $X = MZ + \mu$, we have $\mathbb{E}(X) = \mu$ and $\text{cov}(X) = MM^\top$

▶ Assume $M \in \mathbb{R}^{d \times d}$ is invertible (else we get a degenerate Gaussian distribution).

▶ We say $X \sim \mathcal{N}(\mu, MM^\top)$

▶ Density function given by

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

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

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

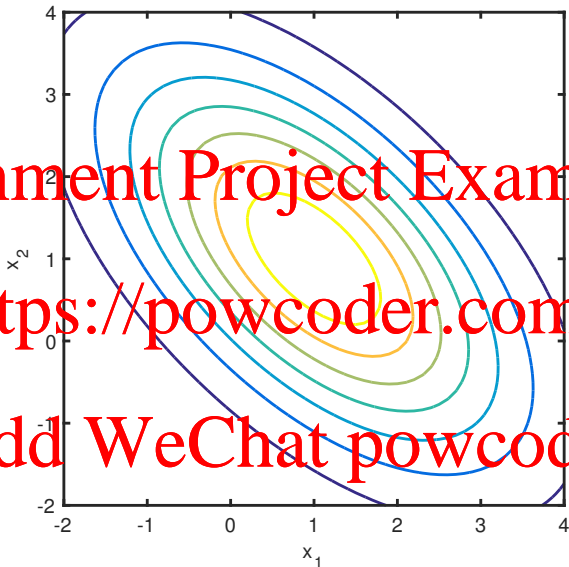


Figure 3: Density function level sets for anisotropic Gaussian in \mathbb{R}^2

Inference with multivariate Gaussians (2)

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

Assignment Project Exam Help

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

- ▶ What is the distribution of X_2 ?

$\rightarrow \mathcal{N}(\mu_2, \Sigma_{2,2})$

<https://powcoder.com>

- ▶ 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

Add WeChat powcoder

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:

<https://powcoder.com>

where

$$\hat{m} = \frac{\text{cov}(X_1, X_2)}{\text{var}(X_1)} = \frac{\Sigma_{1,2}}{\Sigma_{1,1}},$$

Add WeChat powcoder

$$\hat{\theta} = \mathbb{E}(X_2) - \hat{m} \mathbb{E}(X_1) = \mu_2 - \hat{m} \mu_1.$$

- ▶ 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

$$\begin{aligned}\text{var}(X_2 \mid X_1 = x_1) &= \mathbb{E}[\text{var}(X_2 \mid X_1)] \\ &= \text{var}(X_2) - \text{var}(\mathbb{E}[X_2 \mid X_1])\end{aligned}$$

$$= \Sigma_{2,2} - \text{var}(\hat{m}X_1 + \hat{\theta})$$

$$= \Sigma_{2,2} - \hat{m}^2 \text{var}(X_1)$$

$$= \Sigma_{2,2} - \frac{\Sigma_{1,2}^2}{\Sigma_{1,1}}$$

$$= \Sigma_{2,2} - \frac{\Sigma_{1,2}^2}{\Sigma_{1,1}}.$$

Inference with multivariate Gaussians (4)

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

Assignment Project Exam Help

$$\mathbb{E}[X_2 | X_1 = x_1] = \mu_2 + \Sigma_{2,1}\Sigma_{1,1}^{-1}(x_1 - \mu_1)$$

$$\text{cov}(X_2 | X_1 = x_1) = \Sigma_{2,2} - \Sigma_{2,1}\Sigma_{1,1}^{-1}\Sigma_{1,2}$$

<https://powcoder.com>

Add WeChat powcoder

Eigendecomposition (1)

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

Assignment Project Exam Help

$$\lambda_1 \geq \dots \geq \lambda_d$$

- ▶ Can choose corresponding orthonormal eigenvectors

<https://powcoder.com>

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

- ▶ This means

Add WeChat powcoder

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

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

Eigendecomposition (2)

- ▶ Arrange v_1, \dots, v_d in an orthogonal matrix $V := [v_1 | \dots | v_d]$

- ▶ $V^T V = I$ and $V V^T = \sum_{i=1}^d v_i v_i^T = I$

- ▶ Therefore,

$$M = M V V^T$$

$$= \sum_{i=1}^d M v_i v_i^T$$

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

- ▶ This is our preferred way to express the eigendecomposition

- ▶ Also called spectral decomposition

- ▶ Can also write $M = V \Lambda V^T$, where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_d)$

- ▶ The matrix V diagonalizes M :

$$V^T M V = \Lambda$$

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Covariance matrix (1)

- ▶ $A \in \mathbb{R}^{n \times d}$ is data matrix
- ▶ $\Sigma := A^T A = \frac{1}{n} \sum_{i=1}^n x_i x_i^T$ 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 $u \in \mathbb{R}^d$,

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

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. $u = (1, 0, \dots, 0)$)
- Probably can/should ignore these!

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

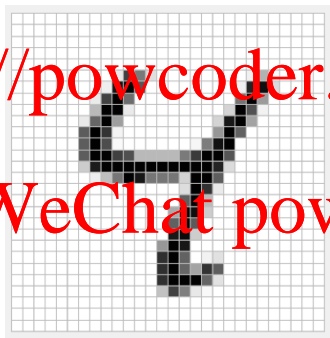


Figure 4: Which pixels are likely to have very little variance?

Top eigenvector

- ▶ Σ is symmetric, so can write eigendecomposition

Assignment Project Exam Help

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

- ▶ 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 :

Add WeChat powcoder

$$v_1^\top \Sigma v_1 = \max_{u \in \mathbb{R}^d, \|u\|_2=1} u^\top \Sigma u = \lambda_1.$$

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

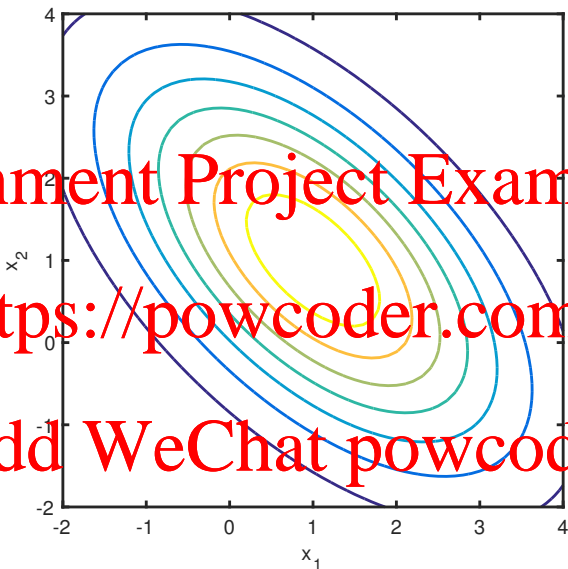


Figure 5: What is the direction of the top eigenvector for the covariance of this Gaussian?

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 L are non-negative!)
- ▶ For any k , $V_k := [v_1 | \cdots | v_k]$ satisfies

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

(the top k eigenvectors)

Add WeChat powcoder

Principal component analysis

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

Assignment Project Exam Help

where $V_k = [v_1 | \dots | v_k] \in \mathbb{R}^{d \times k}$

- ▶ (Only really makes sense when $\lambda_k > 0$.)
- ▶ This is a form of dimensionality reduction when $k < d$.

Add WeChat powcoder

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

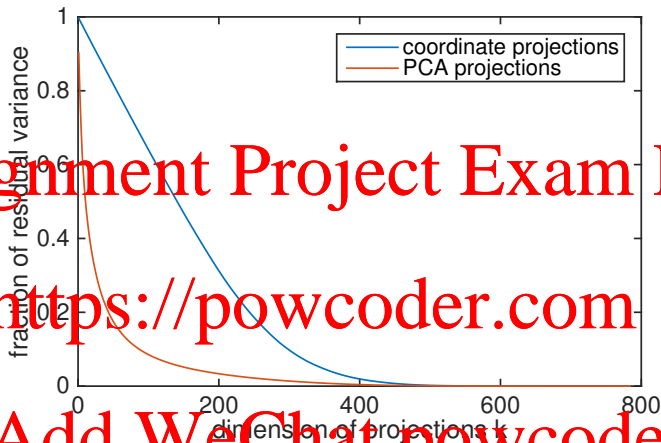


Figure 6: Fraction of residual variance from projections of varying dimension

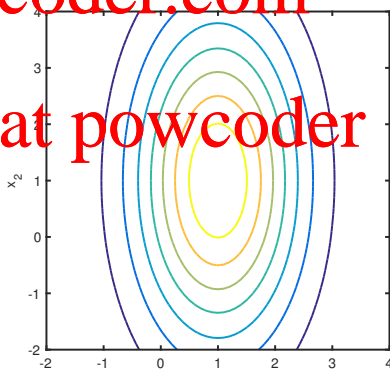
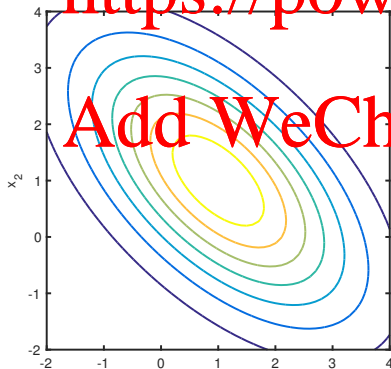
Covariance of data upon PCA mapping

- Covariance of data upon PCA mapping:

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

where Λ_k is diagonal matrix with $\lambda_1, \dots, \lambda_k$ along diagonal.

- In particular, coordinates in $\varphi(x)$ -representation are uncorrelated.



- ▶ Use k -dimensional PCA mapping $\varphi(x) = V_k^\top x$ with ordinary least squares

(Assume rank of A is at least k , so $A^\top A$ has $\lambda_k > 0$)

- ▶ Data matrix is

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

- ▶ Therefore OLS solution is

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

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

Principal component regression

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

$$\begin{aligned} \hat{\mu}(x) &= (V_k^\top x)^\top \Lambda_k^{-1} V_k^\top A^\top b \\ &= x^\top (V_k \Lambda_k^{-1} V_k^\top) (A^\top b) \end{aligned}$$

- So “effective” weight vector (that acts directly on x rather than $\phi(x)$) is given by

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

- 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 $g: \mathbb{R} \rightarrow \mathbb{R}$, write $g(M)$ to mean

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

where M has eigendecomposition $M = \sum_{i=1}^d \lambda_i v_i v_i^\top$.

- ▶ I.e., g is applied to eigenvalues of M
- ▶ Generalizes effect of polynomials: e.g., $g(z) = z^2$

$$M^2 = (V \Lambda V^\top)(V \Lambda V^\top) = V \Lambda^2 V^\top.$$

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

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

for appropriate function g (depending on λ).

Comparing ridge regression and PCR

- ▶ $\hat{w} = g(A^T A)A^T b$

- ▶ 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

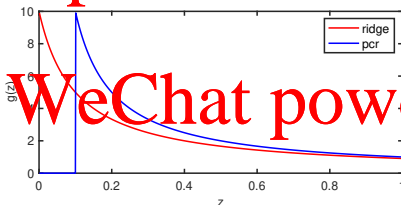


Figure 7: Comparison of ridge regression and PCR

- ▶ Let $A = \begin{bmatrix} \leftarrow & x_1^T & \rightarrow \\ & \vdots & \\ \leftarrow & x_n^T & \rightarrow \end{bmatrix} \in \mathbb{R}^{n \times d}$ (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

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

Assignment Project Exam Help

where

▶ $\sigma_1 \geq \dots \geq \sigma_r > 0$ (singular values)

▶ $u_1, \dots, u_r \in \mathbb{R}^n$ (orthonormal left singular vectors)

▶ $v_1, \dots, v_r \in \mathbb{R}^d$ (orthonormal right singular vectors)

- ▶ Can also write as

$$A = U S V^T$$

Add WeChat powcoder

where

▶ $U = [u_1 | \dots | u_r] \in \mathbb{R}^{n \times r}$, satisfies $U^T U = I$

▶ $S = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$

▶ $V = [v_1 | \dots | v_r] \in \mathbb{R}^{d \times r}$, satisfies $V^T V = I$

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

Assignment Project Exam Help

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

- ▶ Can write as $A_k := U_k S_k V_k^\top$ where

- ▶ $U_k = [u_1 | \dots | u_k] \in \mathbb{R}^{n \times k}$, satisfies $U^\top U = I$

- ▶ $S_k = \text{diag}(\sigma_1, \dots, \sigma_k) \in \mathbb{R}^{k \times k}$

- ▶ $V_k = [v_1 | \dots | v_k] \in \mathbb{R}^{d \times k}$, satisfies $V^\top V = I$

- ▶ **Theorem** (Schmidt/Eckart-Young):

Add WeChat powcoder

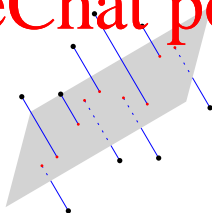
$$\|A - A_k\|_F^2 = \min_{M: \text{rank}(M)=k} \|A - M\|_F^2 = \sum_{i=k+1}^r \sigma_i^2$$

Encoder/decoder interpretation (1)

- ▶ Encoder: $x \mapsto \varphi(x) = V_k^\top x \in \mathbb{R}^k$
 - ▶ Encoding rows of A : $AV_k = U_k S_k$
- ▶ Decoder: $z \mapsto V_k z \in \mathbb{R}^d$
 - ▶ Decoding rows of $U_k S_k$: $U_k S_k V_k^\top = A_k$
- ▶ Same as k -dimensional PCA mapping!
 - ▶ $A^\top A = V S^2 V^\top$, so eigenvectors of $A^\top A$ 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.

<https://powcoder.com>

Add WeChat powcoder



Encoder/decoder interpretation (2)

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

Assignment Project Exam Help

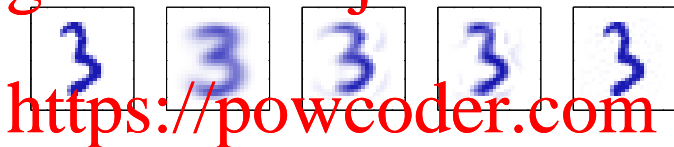


Figure 9: PCA compression of MNIST digit

Add WeChat powcoder

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	...
doc 1	3	0	0	...
doc 2	7	0	4	...
doc 3	2	4	0	...
\vdots	\vdots	\vdots	\vdots	

Add WeChat powcoder

Application: Topic modeling (2)

- ▶ Rank k SVD provides an approximate factorization

Assignment Project Exam Help

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

Add WeChat powcoder

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

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

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

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

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?

<https://powcoder.com>

- ▶ Some nearest-neighbor pairs ($c_j, \text{NN}(c_j)$):
 - ▶ 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)