Statistics for Chemical Engineers:
From Data to Models to Decisions

Victor M. Zavala

Department of Chemical and Biological Engineering
University of Wisconsin-Madison

victor.zavala@wisc.edu

Chapter 6: Statistical Data Analysis and Learning (Part I)

Statistical Data Analysis and Learning

- Data science and machine learning (ML) are fields that provide tools for:
  - Data Analysis (e.g., dimension reduction, time-series, clustering, computer vision)
  - Predictive Modeling (e.g., neural nets, kriging, classification)
  - Artificial Intelligence (e.g., data collection, experimentation, learning, control)
- Some tools used are derived from *statistical principles* while others are derived from other mathematical principles (e.g., geometry, optimization, linear algebra).
- Our focus here is discussing general statistical principles behind such tools.

Statistical Data Analysis and Learning

- Data is at the core of modeling and decision-making tasks.

- Data is what allows us to characterize the behavior of systems and appears in different forms such as spatial fields, time series, images, video, and text.

- Ultimately, our goal is to extract knowledge from such data (in the form of a model) to characterize and analyze the dominant trends.

Statistical Data Analysis and Learning

- We will discuss advanced techniques to extract knowledge from data and to use this knowledge to make predictions and decisions.
- We will explore techniques to extract interesting features of the data.
- We will see how to use this info to analyze and compare complex systems, identify abnormal behavior, build predictive models, and make decisions.
- We will discuss predictive modeling techniques that do not require parameters (called non-parametric) to make predictions.
- We will explore parametric modeling techniques that are universal (can use data to predict virtually any type of behavior seen in practice).
- Techniques mimic how brain learns from data collected from our sensory systems.

Principal Component Analysis (PCA)

- Consider questions:
  - How can I interpret and extract knowledge (e.g., trends) from high-dimensional data?
  - How can I reduce (compress) data to facilitate analysis, visualization, storage, and use?
- We analyze data that is stationary (as opposed to time-dependent).
- Stationary data involves obs that are independent over time.
- Time-dependent involves obs that are correlated over time (more difficult).

Principal Component Analysis

- Have RVs $X = (X_1, X_2, ..., X_n)$ and want to create a *mixture* of the form:

$$T = \sum_{i=1}^{n} w_i X_i = w^T X$$

  where $w_i \in \mathbb{R}$ are mixture proportions, collected in vector $w = (w_1, w_2, ..., w_n)$.

- Consider questions:
  - What proportions w give product T that contains maximum information about X?
  - What proportions w give product T that contains second most information about X?
  - What proportions w give product T that contains minimum information about X?
- Think about analogy of this statistical mixing process with that of physical mixing.
- In physical process, want to mix flows of different quality so that product is valuable.
- Statistical mixing problem can be solved using PCA.

Principal Component Analysis

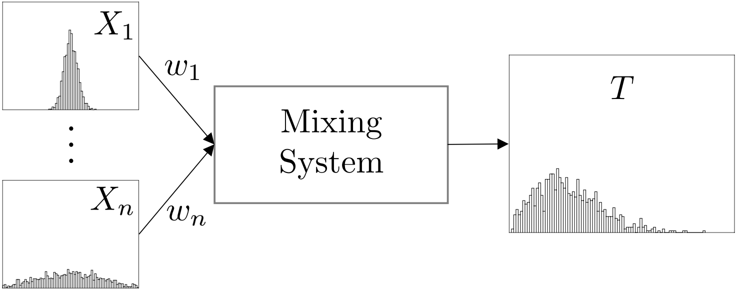


Figure: Statistical mixing problem to be studied using principal component analysis (PCA).

Principal Component Analysis

- Collect obs $x_\omega \in \mathbb{R}^n$, $\omega \in \mathcal{S}$ for $X = (X_1, X_2, ..., X_n)$.
- Make implicit assumption that obs are i.i.d. samples of X.
- Store obs in $S \times n$ data matrix:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{S,1} & x_{S,2} & \cdots & x_{S,n} \end{bmatrix}$$

- Normalize (re-scale) columns of matrix $\mathbf{X}$ in such a way that

$$\frac{1}{S}\sum_{\omega=1}^{S} \mathbf{X}_{\omega,j} = 0, \quad j = 1, ..., n$$

  (e.g., by substracting means of each variable).
- One can show that sample covariance matrix of X is:

$$\hat{\mathrm{Cov}}(X) = \mathbf{X}^T\mathbf{X}.$$

- Matrix $\mathbf{X}^T\mathbf{X}$ is known as *kernel* matrix and we denote this as $\mathbf{\Sigma}$.

Principal Component Analysis

- Kernel contains all info of X; variance is measure of *info content*.
- Kernel is closely related to cov matrix of parameter estimates for structural models.
- Mixture $T = w^T X$ is an RV and one can show that $\hat{\mathbb{V}}[T] = w^T \mathbf{\Sigma} w$.
- Sample variance of T is thus related to sample covariance of X ($\mathbf{\Sigma}$).
- Proportions that contain max information of X are those that max variance of T:

$$\max_w \; w^T \mathbf{\Sigma} w \text{ s.t. } \|w\| = 1$$

- Solution w_1 is eigenvec of $\mathbf{\Sigma}$ associated with largest eigv λ_1 (i.e., $\mathbf{\Sigma} w_1 = \lambda_1 w_1$).
- Eigv is $\lambda_1 = w_1^T \Sigma w_1$ (is measure of info content).
- Optimal mixture associated with eigenvec is given by $T_1 = w_1^T X$.

Principal Component Analysis

- Can identify mixture that contains 2nd most info by solving:

$$\max_{w} \; w^T \mathbf{\Sigma} w \text{ s.t. } \|w\| = 1, \; w^T w_1 = 0.$$

- Solution w_2 is eigenvec associated with 2nd largest eigenv $\lambda_2 = w_2^T \mathbf{\Sigma} w_2$.
- Constraint $w_2^T w_1 = 0$ (a.k.a. orthogonality constraint) ensures that mixture $T_2 = w_2^T X$ is not correlated to mixture $T_1 = w_1^T X$.
- This guarantees that T_2 and T_1 provide complementary (non-redundant) info.
- Can continue procedure to obtain all eigenvecs w_k, $k = 1, ..., n$ of $\mathbf{\Sigma}$.
- Each eigenvec w_k has mixture $T_k = w_k^T X$ with info content $\lambda_k = w^T \Sigma w$.
- Mixtures are ranked in order of info content $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n$.

Principal Component Analysis

- Set of eigenvecs and eigenv can be used to decompose kernel as:

$$\mathbf{\Sigma} = \lambda_1 w_1 w_1^T + \lambda_2 w_2 w_2^T + \cdots + \lambda_n w_n w_n^T$$

- Truncating series (drop smallest eigenvs) enables compression of $\mathbf{\Sigma}$.

- Eigendecomposition of $\mathbf{\Sigma}$ can be written in compact form:

$$\mathbf{\Sigma} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T$$

  where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is:

$$\mathbf{W} = [w_1 | w_2 | \dots | w_n].$$

- Matrix $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is diagonal with entries $\mathbf{\Lambda}_{k,k} = \lambda_k$.

- Eigenvalues are usually visualized using spectrum, which summarizes information content.

Principal Component Analysis

- Eigenvalues are usually visualized using spectrum, which summarizes % of total information content.

$$\alpha_k = 100 \cdot \frac{\lambda_k}{\sum_{j=1}^{n} \lambda_j}, \; k = 1, ..., n.$$

- Spectrum that decays rapidly indicates that matrix has a high degree of redundancy (can be compressed more easily).

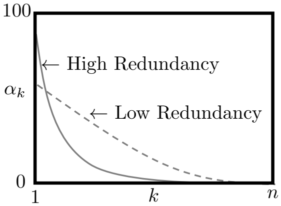


Figure: Illustration of spectrum of a couple of covariance matrices small amount of redundancy (solid) and high amount of redundancy (dashed).

Principal Component Analysis

- Mixtures $T_1, T_2, ...T_n$ (a.k.a. principal components) contain all info about X.
- Eigenvec matrix $\mathbf{W}$ *projects* data $\mathbf{X}$ into space of PCs as:

$$\mathbf{T} = \mathbf{X}\mathbf{W}$$

  where $\mathbf{T} \in \mathbb{R}^{S\times n}$ is matrix with entries $\mathbf{T}_{i,j}$, $i = 1, ..., S$, $j = 1, ..., n$.

- Here, $\mathbf{T}_{i,j}$ is the i-th observation of PC T_j.
- PCA can be seen as a projection of the data from a physical space to an information space (a.k.a. as latent space).
- Can project back from latent space to physical space using $\mathbf{X} = \mathbf{T}\mathbf{W}^T$.

Principal Component Analysis

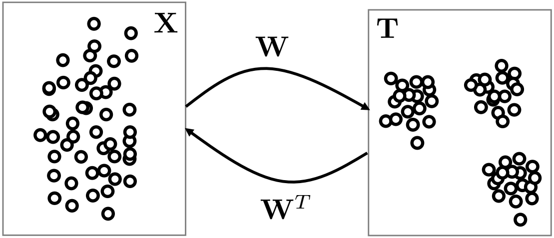


Figure: Projection of data from physical space to information/latent space (and back) using the eigenvectors $\mathbf{W}$.

Principal Component Analysis

- From projection $\mathbf{T} = \mathbf{X}\mathbf{W}$:

$$\begin{aligned}
\hat{\text{Cov}}(T) &= \mathbf{T}^T\mathbf{T} \\
&= \mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W} \\
&= \mathbf{W}^T\mathbf{W}\mathbf{\Lambda}\mathbf{W}\mathbf{W} \\
&= \mathbf{\Lambda},
\end{aligned}$$

- Here, we use property that $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ (eigenvecs are orthogonal).
- PCs are thus uncorrelated and all info of T (variance) is contained in eigenvalues.
- Projection $\mathbf{T} = \mathbf{X}\mathbf{W}$ transforms data from physical space (under which $\mathbf{X}$ lives) into an information space (under which $\mathbf{T}$ lives).
- **Note:** it is not necessary to normalize data to apply PCA.
- Normalization is required if one seeks to interpret $\mathbf{X}^T\mathbf{X}$ as covariance of X.

Example: PCA Analysis for Gibbs Reactor Data `ch6_gibbs_pca.m`

- Dataset with measurements of pressure, temperature, and conversion (P, T, C).
- If visualize data in 3D (P, T, C) space, separates in clusters (failure/normal mode).
- Humans can only visualize data in a small number of dimensions (typically 2D plots).
- Imagine we visualize our data in 2D by ignoring temperature (P, C).
- Visualizing data in this reduced space hides clusters in the data.
- Common issue with visualizing data in many dimensions (hundreds to thousands).
- Specifically, reduction might *hide* key aspects of data.

Example: PCA Analysis for Gibbs Reactor Data `ch6_gibbs_pca.m`

- We visualize data in 2D but in info space of PCs (T_1, T_2) and of (T_2, T_3).
- Visualization clearly reveals the clusters.

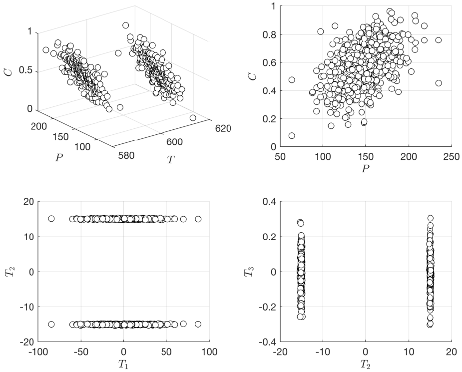


Figure: 3D visual in physical space of P, T, C (top-left). 2D visualization in physical space of C, P (top-right). 2D visual in info space of T_1, T_2 (bottom-left) and of T_2, T_3 (bottom-right).

Principal Component Estimation

- Previously explored strategies to estimate params for models $\mathbf{y} = \mathbf{X}\theta$.

- Discussed how data redundancy leads to rows or columns in $\mathbf{X}$ that are linear dependent and how this can lead to high param variance (low precision).

- We can use PCA to identify and eliminate data redundancies.

- Kernel $\mathbf{X}^T\mathbf{X}$ is related to covariance of estimates $\hat{\theta}$ as:

$$\begin{aligned}
\mathrm{Cov}(\hat{\theta}) &= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} \\
&= \sigma^2(\mathbf{W}\mathbf{\Lambda}\mathbf{W}^T)^{-1} \\
&= \sigma^2\mathbf{W}\mathbf{\Lambda}^{-1}\mathbf{W}^T \\
&= \sigma^2(\lambda_1^{-1}w_1w_1^T + \lambda_2^{-1}w_2w_2^T + \cdots + \lambda_n^{-1}w_nw_n^T).
\end{aligned}$$

- Small eigs of $\mathbf{X}^T\mathbf{X}$ lead to large variances of estimates. This is often a sign of strong correlations or collinearities (linear dependencies).

Principal Component Estimation

- It makes sense to build $\mathbf{y} = \mathbf{X}\theta$ by *pre-processing* (cleaning) data $\mathbf{X}$.
- Specifically, remove effect of small eigs associated with non-informative data.
- This is done by partitioning eigenvector matrix:

$$\begin{aligned}\mathbf{W} &= [\mathbf{W}_1 | \mathbf{W}_2] \\ &= [\underbrace{w_1, w_2, ..., w_{n_1}}_{\text{low information}} \mid \underbrace{w_{n_1+1}, w_{n_1+2}, ..., w_n}_{\text{high information}}].\end{aligned}$$

- This induces a partitioning of eigs matrix:

$$\mathbf{\Lambda} = \left[\begin{array}{c|c} \mathbf{\Lambda}_1 & \\ \hline & \mathbf{\Lambda}_2 \end{array}\right],$$

- We project (reduce) input data into information space as $\mathbf{T}_1 = \mathbf{X}\mathbf{W}_1$.
- Build a reduced model of form $\mathbf{y} = \mathbf{T}_1\gamma$ where $\gamma \in \mathbb{R}^{n_1}$ are the parameters.
- Projection reduces number of params (from n in θ to n_1 in γ) .

Principal Component Estimation

- Estimate params γ by solving:

$$\hat{\gamma} \in \underset{\gamma}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{y} - \mathbf{T}_1\gamma\|_2^2.$$

- Covariance of estimate is:

$$\operatorname{Cov}(\hat{\gamma}) = \sigma^2(\mathbf{T}_1^T\mathbf{T}_1)^{-1}.$$

- Estimated params $\hat{\gamma}$ live in info space that differs from physical space of θ.
- Estimate of params in original physical space is $\hat{\theta} = \mathbf{W}_1\hat{\gamma}$.
- One can show that:

$$\begin{aligned}\operatorname{Cov}(\hat{\theta}) &= \sigma^2\mathbf{W}_1\mathbf{\Lambda}_1^{-1}\mathbf{W}_1^T \\ &= \sigma^2(\lambda_1^{-1}w_1w_1^T + \lambda_2^{-1}w_2w_2^T + \cdots + \lambda_{n_1}^{-1}w_{n_1}w_{n_1}^T)\end{aligned}$$

- As desired, data pre-processing removes effect of small eigs.

Example: PCA Estimation for Gibbs Reactor `ch6_gibbs_pca_estimation.m`

- Develop a model that predicts conversion as a function of different vars:

$$Y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 + \theta_4 X_4 + \theta_5 X_5 + \epsilon$$

- X_1 is press, X_2 is CO flow, X_3 is H_2 flow, X_4 is CH_3OH flow, and X_5 is temp.
- Have total of $n + 1 = 6$ params and $S = 250$ obs.
- Some of the data is redundant (variables are strongly correlated).
- However, it is difficult to know which set of variables can predict conversion or if there is a minimum set of variables that we should consider.

Example: PCA Estimation for Gibbs Reactor `ch6_gibbs_pca_estimation.m`

- Standard MLE gives estimates $\hat{\theta}$ with good SSE of 2.44×10^{-3}.
- However, we find that covariance matrix of these parameters is singular.
- This is because the eigs of kernel matrix $\mathbf{X}^T\mathbf{X}$ are:

$$\lambda = (1.45 \times 10^{8}, 2.36 \times 10^{5}, 1.97 \times 10^{3}, -2.99 \times 10^{-8}, 1.47 \times 10^{-10}, -1.72 \times 10^{-11}).$$

- Three parameters cannot be estimated reliably.

Example: PCA Estimation for Gibbs Reactor `ch6_gibbs_pca_estimation.m`

- To eliminate effect, we obtain eigenvecs $\mathbf{W}$ and eliminate columns corresponding to three smallest eigs, to give $\mathbf{W}_1$.
- We reduce data $\mathbf{T}_1 = \mathbf{X}\mathbf{W}_1$ and estimate $\hat{\gamma}$ and $\hat{\theta} = \mathbf{W}_1\hat{\gamma}$.
- SSE value remains the same 2.44×10^{-3} but variances are:

$$\mathbb{V}(\hat{\theta}) = (7.34 \times 10^{-13}, 3.19 \times 10^{-7}, 1.77 \times 10^{-7}, 2.49 \times 10^{-7}, 2.56 \times 10^{-7}, 2.49 \times 10^{-7}).$$

- Parameters can be estimated reliably with data reduction.

Example: PCA Estimation for Gibbs Reactor `ch6_gibbs_pca_estimation.m`

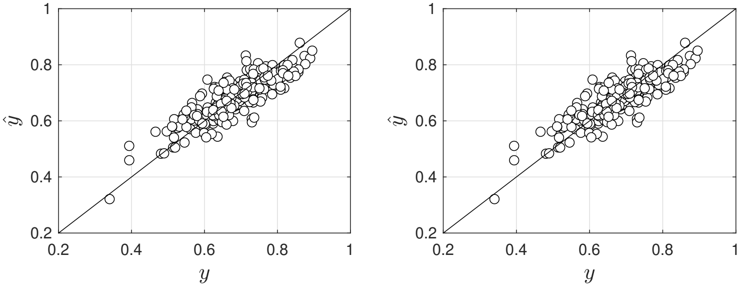


Figure: Fit of linear model under standard estimation (left) and PCA estimation (right).

Sparse PCA

- Eigenvector w contains proportions of mixture $T = \sum_{j=1}^{n} w_j X_j$.
- Magnitudes $|w_j|$ reveal key (dominant) input variables (ingredients) in our mixture.
- Number of variables can be quite large (e.g., hundreds) and we want to find a handful of key variables.

Sparse PCA

- Sparse PCA sparsifies eigenvec w_j by solving regularized problem:

$$\tilde{w}_j \in \operatorname*{argmin}_{w} \ \frac{1}{2}\|\mathbf{X}w_j - \mathbf{X}w\|_2^2 + \kappa\|w\|_1$$

- Finds vector w that minimizes distance to data projection obtained with w_j and minimizes $\|w\|_1 = \sum_{i=1}^{n} |w_i|$.

- $\kappa \in \mathbb{R}_+$ is param that trades-off sparsity of eigenvec $\tilde{w}_j$ and proximity to w_j.

- Sparsification is applied to eigenvecs $w_j, \ j = 1, ..., n$ to obtain a sparse $\tilde{\mathbf{W}}$.

- If κ is large, sparse eigenvec $\tilde{w}_j$ will contain a single variable.

Example: Sparse PCA for Gibbs Reactor `ch6_gibbs_pca_estimation_sparse.m`

- Applying PCA to $\mathbf{X}^T\mathbf{X}$ we obtain:

$$\mathbf{W} = \begin{bmatrix} -0.00 & -0.00 & -0.00 & -0.00 & 0.00 & -0.00 \\ -0.19 & -0.84 & 0.50 & -0.20 & 0.01 & -0.01 \\ -0.04 & 0.23 & 0.38 & -0.03 & -0.43 & -0.79 \\ -0.61 & 0.40 & 0.45 & -0.59 & 0.53 & 0.06 \\ -0.09 & -0.25 & -0.45 & -0.04 & 0.59 & -0.62 \\ -0.76 & -0.09 & -0.45 & -0.78 & -0.42 & 0.02 \end{bmatrix}.$$

- Last 3 eigenvecs responsible for small eigs but do not indicate variable driving behavior.

- To identify problematic variables, we apply sparse PCA and obtain:

$$\tilde{\mathbf{W}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.00 & 0 \\ 0 & 0 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.00 \\ 1.00 & 0 & 0 & 1.00 & 0 & 0 \end{bmatrix}.$$

- Suggests to eliminate X_5 (temperature), X_2 (CO flow), and X_4 (CH_3OH flow).

Example: Sparse PCA for Gibbs Reactor `ch6_gibbs_pca_estimation_sparse.m`

- Elimination of X_5, X_2, X_4 does not affect model fit and provides precise estimates.

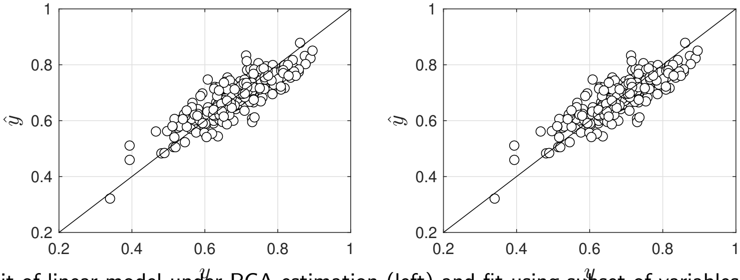


Figure: Fit of linear model under PCA estimation (left) and fit using subset of variables identified with sparse PCA (right).

Singular Value Decomposition (SVD)

- PCA relies on eigendecomp $\mathbf{\Sigma} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T$.
- SVD is a generalization of eigendecomp.
- Eigendecomp can only be applied to square matrices (e.g., $\mathbf{\Sigma}$) while SVD can be applied to rectangular matrices (e.g.,$\mathbf{X}$).
- SVD is broadly applicable for reduction of different forms of data (e.g., images).

Singular Value Decomposition (SVD)

- To explain how SVD works, it is convenient to think about this in context of PCA.
- Consider data matrix $\mathbf{X} \in \mathbb{R}^{S \times n}$ (assume $n \leq S$) and its decomposition:

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

- $\mathbf{U} \in \mathbb{R}^{S \times n}$ is left eigenvec matrix, $\mathbf{V} \in \mathbb{R}^{n \times n}$ is right eigenvec matrix.
- $\mathbf{S} \in \mathbb{R}^{S \times n}$ is singular value (SV) matrix.
- Matrix $\mathbf{S}$ contains SVs $s_j \in \mathbb{R},\ j = 1, ..., n$ in diagonal.
- Singular values are ordered in magnitude $s_1 \geq s_2 \geq \cdots \geq s_n$.

Singular Value Decomposition (SVD)

- Number of non-zero singular values reveals *rank* of $\mathbf{X}$.
- Rank is max number of linearly independent rows (same as number of linearly independent columns).
- Since we have assumed that $n \leq S$ (most common case in applications), $\mathbf{X}$ is regular if rank is n.
- If matrix is non-regular it means that there are redundant rows or columns.
- In PCA, a non-regular $\mathbf{X}$ means that there are redundant observations or variables.
- SVs can thus help us detect and eliminate data redundancies.

Singular Value Decomposition (SVD)

- Matrices $\mathbf{U}, \mathbf{V}$ are orthogonal and thus satisfy $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

- Consequently, we can observe that:

$$\begin{aligned}\mathbf{X}^T\mathbf{X} &= (\mathbf{U}\mathbf{S}\mathbf{V}^T)^T(\mathbf{U}\mathbf{S}\mathbf{V}^T)\\ &= \mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T.\end{aligned}$$

- By defining $\mathbf{W} = \mathbf{V}$ and $\mathbf{\Lambda} = \mathbf{S}^T\mathbf{S}$, we obtain standard PCA.

- Eigs of $\mathbf{X}^T\mathbf{X}$ and singular values of $\mathbf{X}$ are related as $\lambda_i = s_i^2,\ i = 1, ..., n$.

Singular Value Decomposition (SVD)

- SVD of $\mathbf{X}$ can be written as an expansion:

$$\mathbf{X} = \sum_{j=1}^{n} s_j u_j v_j^T$$

- $s_j = \mathbf{S}_{j,j}$ and u_j and v_j are j-th columns of $\mathbf{U}$ and $\mathbf{V}$, respectively.

- One can truncate expansion to obtain a compressed representation of the matrix $\mathbf{X}$.

Example: SVD for Gibbs Reactor `ch6_gibbs_pca_estimation_svd.m`

- Perform SVD on $\mathbf{X}$ (this has $S = 250$ rows and $n = 6$ columns) to obtain:

$$\mathrm{diag}(\mathbf{S}) = (12059.49, 486.03, 44.43, 0.00, 0.00, 0.00).$$

- Matrix has a rank of 3 (there are 3 non-zero singular values), matrix is irregular.

- By inspecting data matrix (few rows):

$$\mathbf{X} = \begin{bmatrix} 1.00 & 166.98 & 25.56 & 451.12 & 74.44 & 583.15 \\ 1.00 & 170.71 & 24.75 & 449.51 & 75.25 & 583.15 \\ 1.00 & 141.17 & 32.22 & 464.44 & 67.78 & 583.15 \\ 1.00 & 146.63 & 30.64 & 461.28 & 69.36 & 583.15 \\ 1.00 & 164.67 & 26.08 & 452.16 & 73.92 & 583.15 \\ 1.00 & 157.43 & 27.79 & 455.59 & 72.21 & 583.15 \end{bmatrix}.$$

- 1st and 6th columns are linearly dependent but rest of dependencies are less obvious.

Example: SVD for Gibbs Reactor `ch6_gibbs_pca_estimation_svd.m`

- 1st and 6th columns are linearly dependent but rest of dependencies are less obvious.

- Redundancies are identified as:

$$
\begin{aligned}
\mathrm{rank}(\mathbf{X}[:,1]) &= 1\\
\mathrm{rank}(\mathbf{X}[:,1,2]) &= 2\\
\mathrm{rank}(\mathbf{X}[:,1,2,3]) &= 3\\
\mathrm{rank}(\mathbf{X}[:,1,2,3,4]) &= 3\\
\mathrm{rank}(\mathbf{X}[:,1,2,3,4,5]) &= 3\\
\mathrm{rank}(\mathbf{X}[:,1,2,3,4,5,6]) &= 3
\end{aligned}
$$

- 4th, 5th, and 6th columns are dependent on the 1st, 2nd, and 3rd columns.

- These dependencies arise due to mass balances (conservation); sum of some flows results in another flow.

Example: Images as Matrices `ch6_svd_image_field.m`

- Any matrix $\mathbf{X}$ can be represented as grayscale image; each entry corresponds to a pixel and number at such entry is intensity (i.e., how dark pixel is).
- Similarly, any grayscale image can be represented as a matrix $\mathbf{X}$ and this is the basic property that enables image processing (e.g., compression and filtering).
- To illustrate this, consider matrix (notice symmetry and decaying magnitudes):

$\mathbf{X} =$

0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	2	4	4	2	1	0	0
0	1	2	6	9	9	6	2	1	0
0	1	4	9	14	14	9	4	1	0
0	1	4	9	14	14	9	4	1	0
0	1	2	6	9	9	6	2	1	0
0	0	1	2	4	4	2	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

- What is image corresponding to this matrix? What if this is corrupted with noise?

Example: Images as Matrices `ch6_svd_image_field.m`

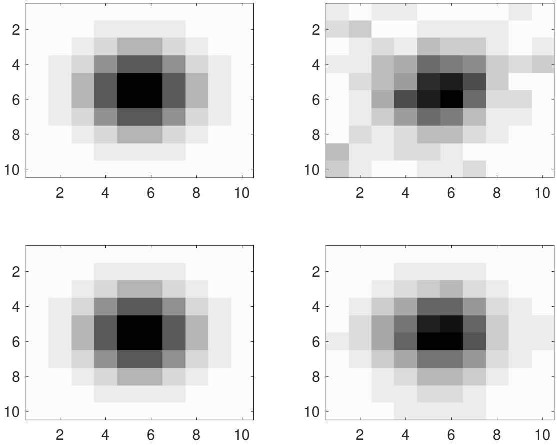


Figure: Image corresponding to matrix $\mathbf{X}$ (top-left) and to perturbed matrix $\mathbf{X}_\epsilon$ (top-right). Image corresponding to compression of matrix $\mathbf{X}$ (bottom-left) and of perturbed matrix $\mathbf{X}_\epsilon$ (bottom-right) using first singular value.

Example: Images as Matrices `ch6_svd_image_field.m`

- Images contain $10 \times 10 =$100 pixels and capture spatial pattern of matrices.
- To human eye, it is easier to visualize patterns using images.
- However, matrix representation enables analysis, storage, and manipulation.
- Perform SVD of image $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ and write as expansion:

$$\mathbf{X} = \sum_{i=1}^{10} s_i \mathbf{u}_i \mathbf{v}_i^T$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are vectors of dimension 10.

Example: Images as Matrices `ch6_svd_image_field.m`

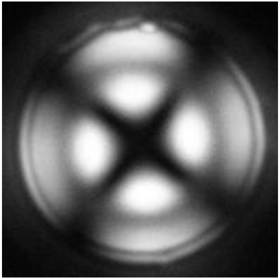
- From SVD we find that image has 4 non-zero singular values:

$$(41.8629, 1.5625, 0.8498, 0.5757).$$

- 1st SV dominates, image can be approximated as $\mathbf{X} \approx s_1 \mathbf{u}_1 \mathbf{v}_1^T$.
- We have reduced the amount of data that we need to store by 80%.
- SVD applied to noisy image can be used to eliminate noise.
- Compression possible because of strong redundancies in the image (note that it is symmetric in all directions).

Example: Image Compression using SVD `ch6_svd_reconstruction.m`

- Decompose complex image $\mathbf{X} = \sum_{k=1}^{n} s_k \mathbf{u}_k \mathbf{v}_k^T$.
- Large components tend to contain broad (global) features while small components contain granular (local) features such as noise.
- By truncating expansion to compress image, we ignore (filter out) granular behavior.
- Image is matrix $\mathbf{X}$ of dimension 605$\times$ 605 and has total of 366,025 pixels.



Example: Image Compression using SVD `ch6_svd_reconstruction.m`

- Upon applying SVD, we find that matrix is full rank (regular).
- Obtain approx images by truncating series with $n = 1, 3, 10, 100$.
- Coarse features develop quickly ($n = 3$) while granular features develop slowly.
- Elements $s_j u_j v_j^T$ of SVD series encode different "features" of image.
- Observe that a nearly perfect image is obtained with $n = 100$.
- Truncating series filters out granular features (*imperceptible to human eye*).

Example: Image Compression using SVD `ch6_svd_reconstruction.m`

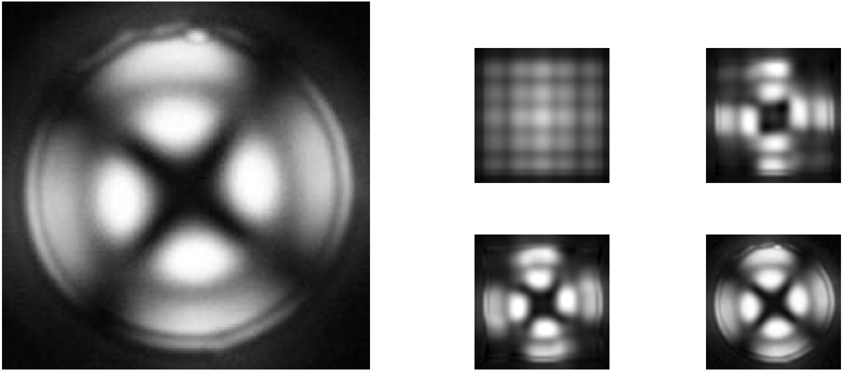


Figure: Original image (left) and compressions using $n = 1, 3, 10, 100$ singular values (right).