Principal Component Analysis

Mathematics for Machine Learning

Lecturer: Matthew Wicker

Logistics: Exam Review

Lecture notes errors will be corrected in green

Practice exam and equation sheet released this week

Review lecture Friday + two problem review sessions

* Computational Complexity
* Concentration Inequalities/Expectation IDs
* Multivariate calculus
* Optimization properties

Doubling back: Bias-Variance

$$\mathbb{E}_{\mathbf{x},y,\mathcal{D}}\left[\left(f^{\theta}_{\mathcal{D}}(\mathbf{x}) - y\right)^2\right]$$

We decomposed the error into three terms:

$$\underbrace{\cancel{\mathbb{E}_{\mathbf{x},y}[(\hat{y} - y)^2]}}_{\text{Noise}} + \underbrace{\mathbb{E}_{\mathbf{x}}\left[(\hat{f}^{\theta}(\mathbf{x}) - \hat{y})^2\right]}_{\text{Squared bias}} + \underbrace{\mathbb{E}_{\mathbf{x},\mathcal{D}}[(f^{\theta}_{\mathcal{D}}(\mathbf{x}) - \hat{f}^{\theta}(\mathbf{x}))^2]}_{\text{Variance}}$$

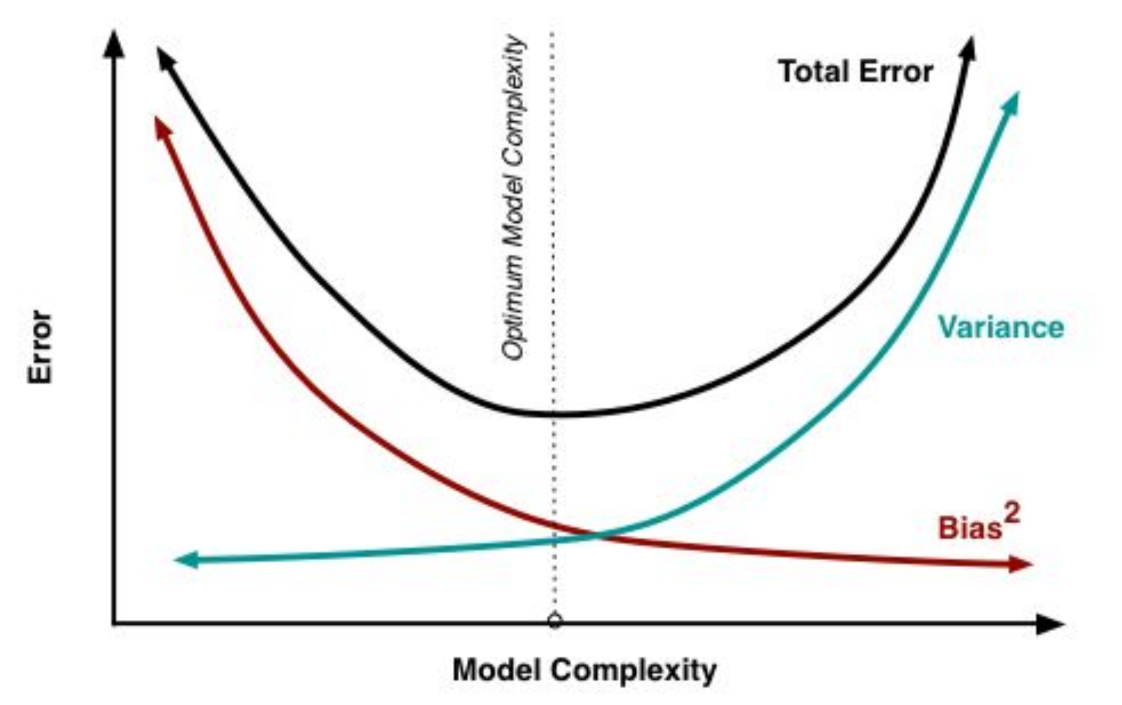
Doubling back: Bias-Variance

$$\theta^{\text{Ridge}} = (\sigma^2 \lambda \mathbf{I} + \boxed{\mathbf{X}^\top \mathbf{X}})^{-1} \boxed{\mathbf{X}^{-1} \mathbf{y}} \qquad \mathbb{E}_{\mathbf{x}, y, \boxed{\mathcal{D}}}\left[(f^\theta_{\mathcal{D}}(\mathbf{x}) - y)^2\right]$$

(Considering this on the board)

$$\cancel{\underbrace{\mathbb{E}_{\mathbf{x},y}[(\hat{y} - y)^2]}_{\text{Noise}}} + \underbrace{\mathbb{E}_{\mathbf{x}}\ [(\hat{f}^\theta(\mathbf{x}) - \hat{y})^2]}_{\text{Squared bias}} + \underbrace{\mathbb{E}_{\mathbf{x},\mathcal{D}}[(f^\theta_{\mathcal{D}}(\mathbf{x}) - \hat{f}^\theta(\mathbf{x}))^2]}_{\text{Variance}}$$

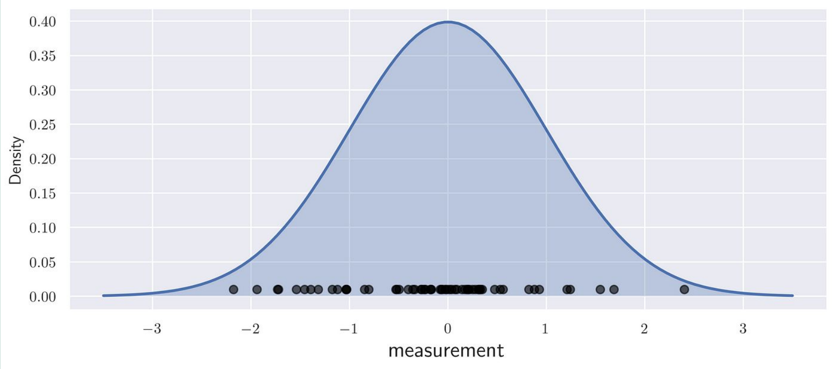
Bias-Variance Trade-off



Today: Unsupervised learning

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

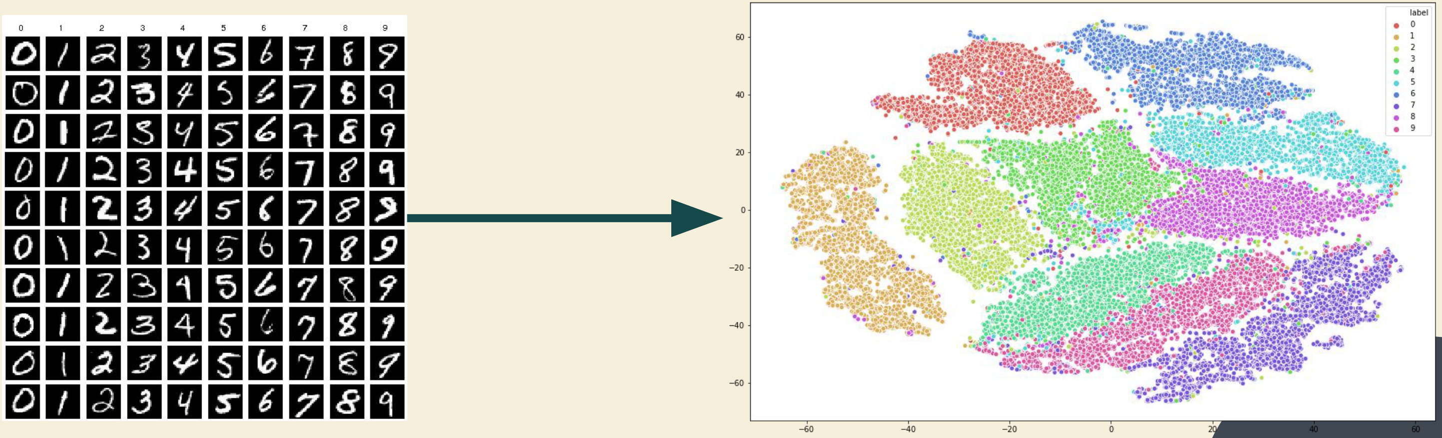
Recall: unsupervised learning or knowledge discovery is an important ML problem setting. We have seen density estimation, today we will look at dimensionality reduction



Today: Unsupervised learning

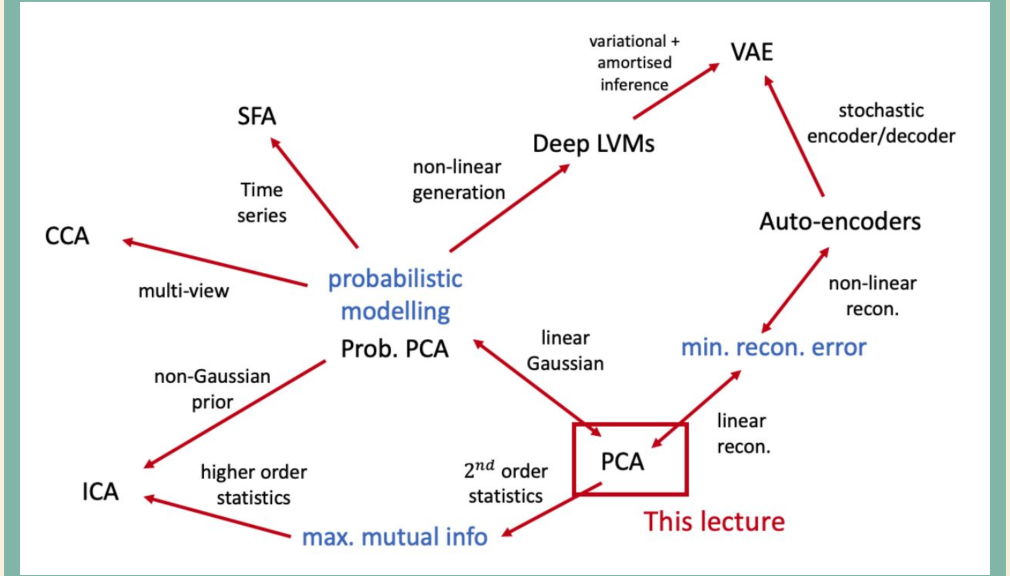
$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

Recall: unsupervised learning or knowledge discovery is an important ML problem setting. We have seen density estimation, today we will look at dimensionality reduction



Today: Unsupervised learning

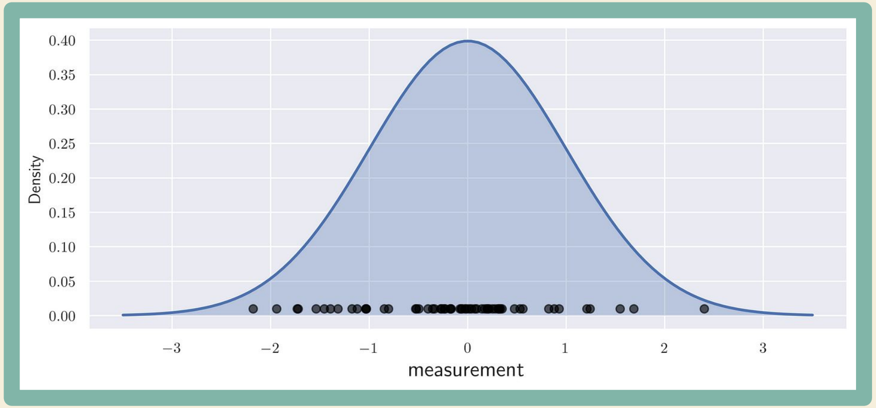
Dimensionality reduction is a rich and interesting area of research that has developed many interesting and powerful models

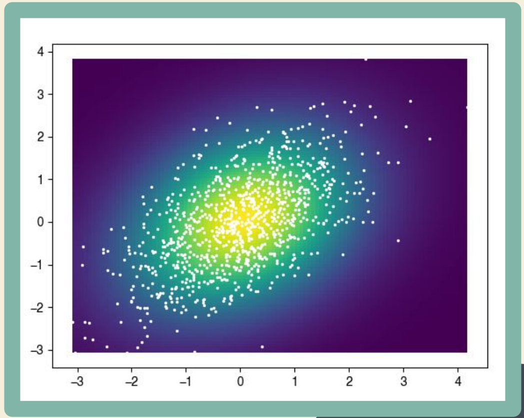


Dealing with high dimensional data

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

Where d = 1 or d = 2 we have seen how density estimation gives us a good framework for thinking about the structure of our dataset:

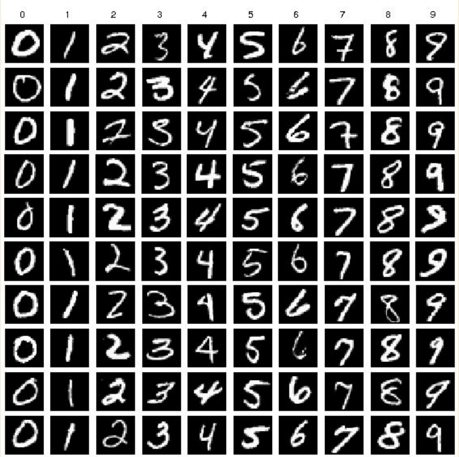




Dealing with high dimensional data

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

When d=784?



Dealing with high dimensional data

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

When d=784? Key idea: Understand the structure of the data in order to project it to a lower dimensional space

$$\mathcal{D} = \{X^{(1)} = x^{(1)}, X^{(2)} = x^{(2)}, \ldots, X^{(n)} = x^{(n)}\}$$

Looking with Probability/Statistics

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

We have seen one such estimator of dataset structure: the empirical mean, or the center of the data

$$\mathcal{D} = \{X^{(1)} = x^{(1)}, X^{(2)} = x^{(2)}, \ldots, X^{(n)} = x^{(n)}\}$$

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} x^{(i)}$$

Looking with Probability/Statistics

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

When we speak of "structure" it is not clear that the mean is relevant and so we often standardize our data such that it is centered at zero

$$\mathcal{D} = \{X^{(1)} = x^{(1)}, X^{(2)} = x^{(2)}, \ldots, X^{(n)} = x^{(n)}\}$$

$$x^{(i)} - \hat{\mu}_n$$

Looking with Probability/Statistics

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

Other structure we know in our data? The empirical covariance

$$\hat{\Sigma}_n = S = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mu} - \mathbf{x}^{(i)})(\hat{\mu} - \mathbf{x}^{(i)})^\top$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}\mathbf{x}^{(i)\top}$$

μ at 0?

Do we think this is a biased estimator?

Studying S with linear algebra

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

Other structure we know in our data? The empirical covariance

$$\hat{\Sigma}_n = S \in \mathbb{R}^{d \times d}$$

What do properties of S tell us about our data?

- Determinant - Spread of the data
- Rank - The dimensionality of the dataset
- Eigendecomposition

Studying S with linear algebra

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

Other structure we know in our data? The empirical covariance

$$\hat{\Sigma}_n = S \in \mathbb{R}^{d \times d}$$

We used the mean to standardize our data, can we do the same with the covariance matrix? Notice the covariance matrix is a normal matrix thus is decomposition can be written as:

$$S = P \Lambda P^\top$$

Studying S with linear algebra

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \qquad \mathbf{x}^{(i)} \in \mathbb{R}^d$$

We used the mean to standardize our data, can we do the same with the covariance matrix?

$$S = P\Lambda P^\top \qquad \mathbf{y}^{(i)} = P^\top \mathbf{x}^{(i)}$$

Consider the above transformation, how does it impact the mean?

Studying S with linear algebra

$$S = P\Lambda P^\top \qquad\qquad \mathbf{y}^{(i)} = P^\top \mathbf{x}^{(i)}$$

How does it impact the covariance structure?

$$S' = \frac{1}{n}\sum_{i=1}^{n} P^\top \mathbf{x}^{(i)} (P^\top \mathbf{x}^{(i)})^\top$$

$= y\, y^T$ ← new covar matrix

Studying S with linear algebra

$$S = P \Lambda P^\top \qquad \mathbf{y}^{(i)} = P^\top \mathbf{x}^{(i)}$$

How does it impact the covariance structure?

$$S' = \frac{1}{n} \sum_{i=1}^{n} P^\top \mathbf{x}^{(i)} (P^\top \mathbf{x}^{(i)})^\top$$

$$= \frac{1}{n} \sum_{i=1}^{n} P^\top \mathbf{x}^{(i)} \mathbf{x}^{(i)\top} P$$

Studying S with linear algebra

$$S = P \Lambda P^\top \qquad \mathbf{y}^{(i)} = P^\top \mathbf{x}^{(i)}$$

How does it impact the covariance structure?

$$\begin{aligned} S' &= \frac{1}{n} \sum_{i=1}^{n} P^\top \mathbf{x}^{(i)} (P^\top \mathbf{x}^{(i)})^\top \\ &= \frac{1}{n} \sum_{i=1}^{n} P^\top \mathbf{x}^{(i)} \mathbf{x}^{(i)\top} P \\ &= P^\top S P \end{aligned}$$

$y^{(i)} = P^\top x^{(i)}$

$\downarrow$

$S' = P^\top S P$, S: sample cov of non altered x

Studying S with linear algebra

How does it impact the covariance structure?

$$S' = \frac{1}{n}\sum_{i=1}^{n} P^\top \mathbf{x}^{(i)}(P^\top \mathbf{x}^{(i)})^\top$$

$$= \frac{1}{n}\sum_{i=1}^{n} P^\top \mathbf{x}^{(i)} \mathbf{x}^{(i)\top} P$$

$$= P^\top S P$$

$$P^\top (P \Lambda P^\top) P = \Lambda$$

Studying S with linear algebra

How does it impact the covariance structure?

$$S' = \frac{1}{n}\sum_{i=1}^{n} P^\top \mathbf{x}^{(i)}(P^\top \mathbf{x}^{(i)})^\top$$

$$= \frac{1}{n}\sum_{i=1}^{n} P\mathbf{x}^{(i)}\mathbf{x}^{(i)\top} P$$

$$= P^\top S P$$

$$P^\top (P\Lambda P^\top) P = \Lambda$$

as $P^\top = P^{-1}$

since cov is symmetric

Now, we have a covariance structure with zero entries everywhere except along the diagonal. This seems very nice to work with!

Formulating the problem we want to solve

Given that we have standardized our data, we can also use linear algebra to begin to formulate the problem we want to solve

$$Q \in \mathbb{R}^{m \times d}$$

We know matrices transform d dimensional space into m dimensional space (assume m < d). But we need the *best* such matrix

0 mean, then $x^{(i)} \to P^T x^{(i)}$, P^T from eigendecomp. of empirical covar $= \frac{1}{N} \sum (\hat{\mu} - x^{(i)})(\mu - x^{(i)})^T$

Then new covar: Λ from ↗

Formulating the problem we want to solve

We need the *best* such matrix. We want to preserve the information content in our initial design matrix

A matrix that preserves the most covariance structure even in the lower dimensional space

$$Q \in \mathbb{R}^{m \times d}$$

Consider the case of m = 1

$$\max_{\mathbf{q}} \mathbb{V}[\mathbf{q}^\top \mathbf{x}], \qquad \text{s.t.,} \quad ||\mathbf{q}||_2^2 = 1$$

Formulating the problem we want to solve

Consider the case of m = 1

$$\max_{\mathbf{q}} \mathbb{V}[\mathbf{q}^\top \mathbf{x}], \qquad \text{s.t.,} \quad ||\mathbf{q}||_2^2 = 1$$

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{q}^\top \mathbf{x}^{(i)})^2$$

Optimization perspective

Consider the case of m = 1

$$\max_{\mathbf{q}} \mathbb{V}[\mathbf{q}^\top \mathbf{x}], \qquad \text{s.t.,} \quad ||\mathbf{q}||_2^2 = 1$$

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{q}^\top \mathbf{x}^{(i)})^2$$

Simplifying the optimization problem

Consider the case of m = 1

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{q}^\top \mathbf{x}^{(i)})^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathbf{q}^\top \mathbf{x}^{(i)}\mathbf{x}^{(i)\top}\mathbf{q}$$

$$= \mathbf{q}^\top \left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}\mathbf{x}^{(i)\top}\right)\mathbf{q}$$

Simplifying the optimization problem

Consider the case of m = 1

$$
\begin{aligned}
\mathbb{V}[\mathbf{q}^\top \mathbf{x}] &= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{q}^\top \mathbf{x}^{(i)})^2 \\
&= \mathbf{q}^\top \left(S\right) \mathbf{q} \\
&= \mathbf{q}^\top (P^\top \Lambda P) \mathbf{q}
\end{aligned}
$$

Simplifying the optimization problem

Consider the case of m = 1

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{q}^\top \mathbf{x}^{(i)})^2$$

$$= \mathbf{q}^\top \left(S\right) \mathbf{q}$$

$$= \mathbf{q}^\top (P^\top \Lambda P)\mathbf{q}$$

$$= \sum_{i=1}^{d}(\lambda_i \beta_i)^2$$

Letting:

$$\beta = P\mathbf{q}$$

Converting our constraints to beta

Consider the case of m = 1

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \sum_{i=1}^{d} (\lambda_i \beta_i)^2$$

Letting:

$$\beta = P\mathbf{q}$$

Recall that we need **q** to be a unit norm. But what are the constraints on beta?

Converting our constraints to beta

Consider the case of m = 1

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \sum_{i=1}^{d} (\lambda_i \beta_i)^2$$

Letting:

$$\beta = P\mathbf{q}$$

Recall that we need **q** to be a unit norm. But what are the constraints on beta?

$$||\mathbf{q}||_2^2 := \mathbf{q}^\top \mathbf{q}$$

Converting our constraints to beta

Consider the case of m = 1

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \sum_{i=1}^{d} (\lambda_i \beta_i)^2$$

Letting:

$$\beta = P\mathbf{q}$$

Recall that we need **q** to be a unit norm. But what are the constraints on beta?

$$||\mathbf{q}||_2^2 := \mathbf{q}^\top \mathbf{q} = \mathbf{q}^\top P^\top P \mathbf{q} = \beta^\top \beta = ||\beta||_2^2$$

P is an orthonormal matrix

Solving the optimization problem

Consider the case of m = 1

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \sum_{i=1}^{d} (\lambda_i \beta_i)^2$$

Letting:

$$\beta = P\mathbf{q}$$

How do you select beta to maximize the above equation? Consider the condition that:

$$\lambda_i \geq \lambda_j \forall i < j$$

Solving the optimization problem

Consider the case of m = 1

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \sum_{i=1}^{d} (\lambda_i \beta_i)^2$$

Letting:

$$\beta = P\mathbf{q}$$

How do you select beta to maximize the above equation? Select beta such that:

$$\beta = [1, 0, \ldots, 0]^\top$$

Solving the optimization problem

Consider the case of m = 1

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \sum_{i=1}^{d} (\lambda_i \beta_i)^2$$

Letting:

$$\beta = P\mathbf{q}$$

How do we realize a q such that beta takes this form?

$$\beta = [1, 0, \ldots, 0]^\top$$

Solving the optimization problem

Consider the case of m = 1

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \sum_{i=1}^{d} (\lambda_i \beta_i)^2$$

Letting:

$$\beta = P\mathbf{q}$$

How do we realize a q such that beta takes this form?

$$\beta = [1, 0, \ldots, 0]^\top$$

Take q equal to the first eigenvector.

Solving the optimization problem

From our derivation of the variance formula:

$$\mathbb{V}[\mathbf{q}^\top \mathbf{x}] = \mathbf{q}^\top \left(S\right) \mathbf{q}$$

By definition of eigenvector/values:

$$S\mathbf{p}_1 = \lambda_1 \mathbf{p}_1$$

We have that setting q to the first eigenvalue:

$$\mathbf{p}_1^\top S \mathbf{p}_1 = \mathbf{p}_1^\top \lambda_1 \mathbf{p}_1 = \lambda_1 \mathbf{p}_1^\top \mathbf{p}_1 = \lambda_1$$

The general algorithm

Expanding beyond the m=1 case, the same exact logic follows by assuming we have taken the first column of our transformation to be the eigenvector corresponding to the maximum eigenvalue

Algorithm 1 PCA

Input: $\boldsymbol{X}$ - Feature/Design matrix, K - Number of components

$\hat{S}_n \leftarrow \sum_{i=1}^{n} \boldsymbol{x}^{(i)} \boldsymbol{x}^{(i)\top}$ data covar. matrix

Compute eigendecomposition $S = P \Lambda P^\top$

Ensure $\Lambda = \mathrm{diag}(\lambda)$ with $\lambda_i \geq \lambda_j \forall i < j$

$B = P_{[:,:k]}$ keep first k eigenvectors (most significant k)

return $\{B\boldsymbol{x}^{(i)}\}_{i=1}^{n}$ project data onto new k-dim. subspace

The general algorithm

Algorithm 1 PCA

Input: $\boldsymbol{X}$ - Feature/Design matrix, K - Number of components

$\hat{S}_n \leftarrow \sum_{i=1}^{n} \boldsymbol{x}^{(i)} \boldsymbol{x}^{(i)\top}$

Compute eigendecomposition $S = P \Lambda P^\top$

Ensure $\Lambda = \text{diag}(\lambda)$ with $\lambda_i \geq \lambda_j \forall i < j$

$B = P_{[:,:k]}$

return $\{B\boldsymbol{x}^{(i)}\}_{i=1}^{n}$

How do we pick K?
(Board)

Non-linear dimensionality reduction

Algorithm 1 PCA

Input: $\boldsymbol{X}$ - Feature/Design matrix, K - Number of components

$\hat{S}_n \leftarrow \sum_{i=1}^{n} \boldsymbol{x}^{(i)} \boldsymbol{x}^{(i)\top}$

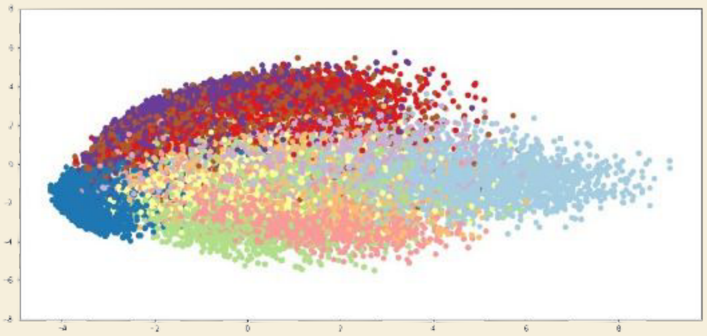
Compute eigendecomposition $S = P \Lambda P^\top$

Ensure $\Lambda = \mathrm{diag}(\lambda)$ with $\lambda_i \geq \lambda_j \forall i < j$

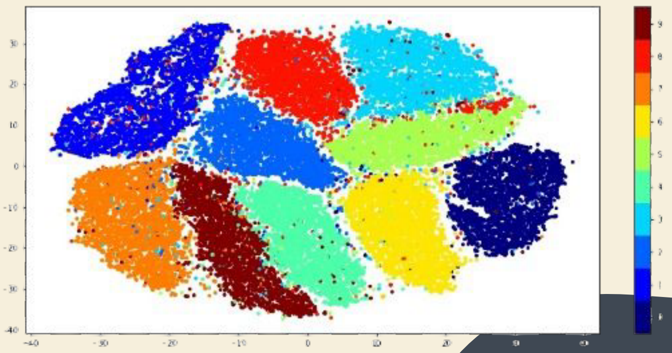
$B = P_{[:,:k]}$

return $\{B\boldsymbol{x}^{(i)}\}_{i=1}^{n}$

MNIST - PCA



MNIST - TSNE



Minimum reconstruction error (PCA)

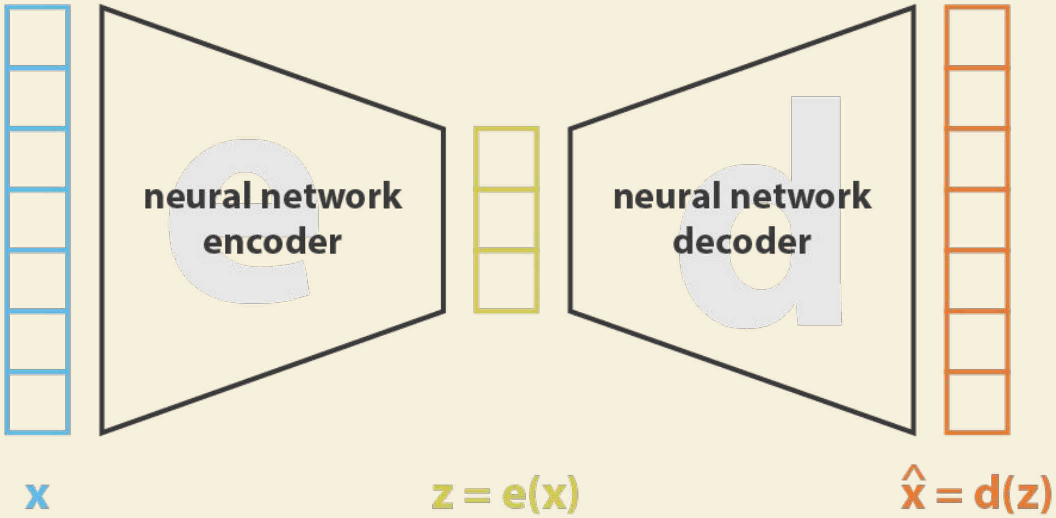
$$\text{Encoder:} \quad \boldsymbol{z} = enc(\boldsymbol{x}) = \mathbf{B}\boldsymbol{x}, \ \mathbf{B} \in \mathbb{R}^{M \times D}$$

$$\text{Decoder:} \quad \hat{\boldsymbol{x}} = dec(\boldsymbol{z}) = \mathbf{A}\boldsymbol{z}, \ \mathbf{A} \in \mathbb{R}^{D \times M}$$

Minimize reconstruction loss/error:

$$\min_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}), \quad L(\mathbf{A}, \mathbf{B}) := \frac{1}{N} \sum_{n=1}^{N} ||\boldsymbol{x}_n - \mathbf{A}\mathbf{B}\boldsymbol{x}_n||_2^2.$$

Non-linear autoencoders



$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

Next lecture:
Overview and Exam Review