CSE446: Dimensionality Reduction and PCA

Spring 2017

Slides adapted from Carlos Guestrin and Luke Zettlemoyer

Dimensionality reduction

- Input data may have thousands or millions of dimensions!
  - e.g., text data has ???, images have ???
- **Dimensionality reduction**: represent data with fewer dimensions
  - easier learning – fewer parameters
  - visualization – hard to visualize more than 3D or 4D
  - discover "intrinsic dimensionality" of data
    - high dimensional data that is truly lower dimensional

Feature selection

- Want to learn $f: \mathbf{X} \rightarrow Y$
  - $\mathbf{X} = <X_1, \ldots, X_n>$
  - but some features are more important than others

- **Approach**: select subset of features to be used by learning algorithm
  - **Score** each feature (or sets of features)
  - **Select** set of features with best score

Greedy **forward** feature selection algorithm

- Pick a dictionary of features
  - e.g., polynomials for linear regression
- Greedy: Start from empty (or simple) set of features $F_0 = \emptyset$
  - Run learning algorithm for current set of features F_t
    - Obtain h_t
  - Select **next best feature X_i**
    - e.g., X_j that results in lowest held out error when learning with $F_t \cup \{X_j\}$
  - $F_{t+1} \leftarrow F_t \cup \{X_i\}$
  - Repeat

Greedy **backward** feature selection algorithm

- Pick a dictionary of features
  - e.g., polynomials for linear regression
- Greedy: Start with all features $F_0 = F$
  - Run learning algorithm for current set of features F_t
    - Obtain h_t
  - Select **next worst feature X_i**
    - e.g., X_j that results in lowest held out error learner when learning with F_t - $\{X_j\}$
  - $F_{t+1} \leftarrow F_t - \{X_i\}$
  - Repeat

Impact of feature selection on classification of fMRI data [Pereira et al. '05]

Accuracy classifying category of word read by subject

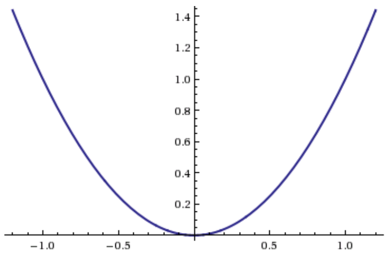
#voxels	mean	subjects 233B	329B	332B	424B	474B	496B	77B	86B
50	0.735	0.783	0.817	0.55	0.783	0.75	0.8	0.65	0.75
100	0.742	0.767	0.8	0.533	0.817	0.85	0.783	0.6	0.783
200	0.737	0.783	0.783	0.517	0.817	0.883	0.75	0.583	0.783
300	**0.75**	**0.8**	**0.817**	**0.567**	**0.833**	**0.883**	**0.75**	**0.583**	**0.767**
400	0.742	0.8	0.783	0.583	0.85	0.833	0.75	0.583	0.75
800	0.735	0.833	0.817	0.567	0.833	0.833	0.7	0.55	0.75
1600	0.698	0.8	0.817	0.45	0.783	0.833	0.633	0.5	0.75
all (~2500)	0.638	0.767	0.767	0.25	0.75	0.833	0.567	0.433	0.733

Table 1: **Average accuracy across all pairs of categories, restricting the procedure to use a certain number of voxels for each subject.** The highlighted line corresponds to the best mean accuracy, obtained using 300 voxels.

Feature Selection through Regularization

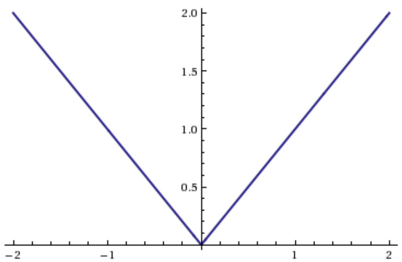
- Previously, we discussed regularization with a squared norm:

$$\hat{\theta} = \arg\min_{\theta} Loss(\theta; \mathcal{D}) + \lambda \sum_i \theta_i^2$$



- What if we used an L_1 norm instead?

$$\hat{\theta} = \arg\min_{\theta} Loss(\theta; \mathcal{D}) + \lambda \sum_i |\theta_i|$$



- What about L_∞?
- These norms work, but are harder to optimize! And, it can be tricky to set λ!!!

Lower dimensional projections

- Rather than picking a subset of the features, we can make new ones by combining existing features x_1 … x_n

$$z_1 = w_0^{(1)} + \sum_i w_i^{(1)} x_i$$

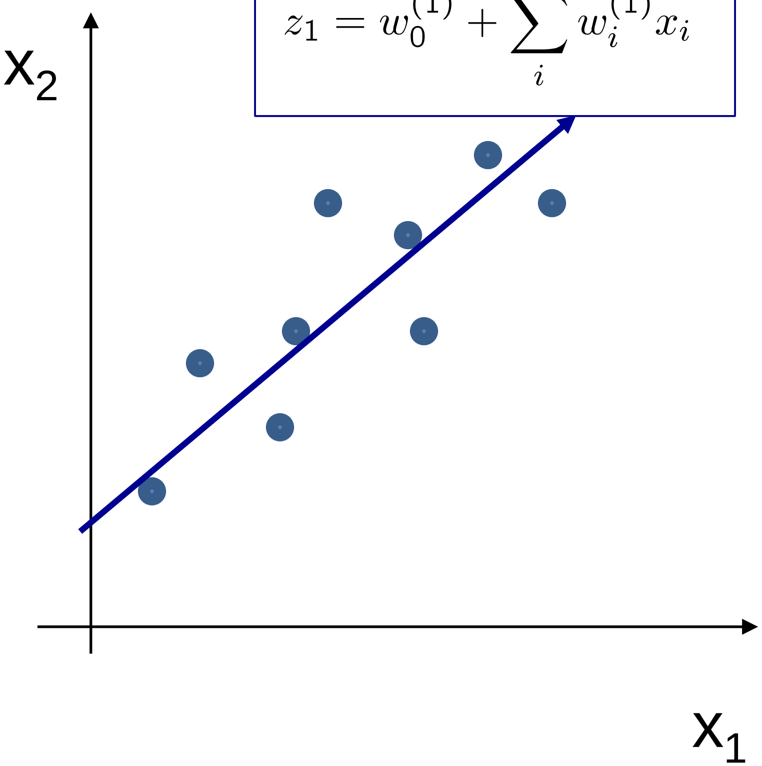
$$\cdots$$

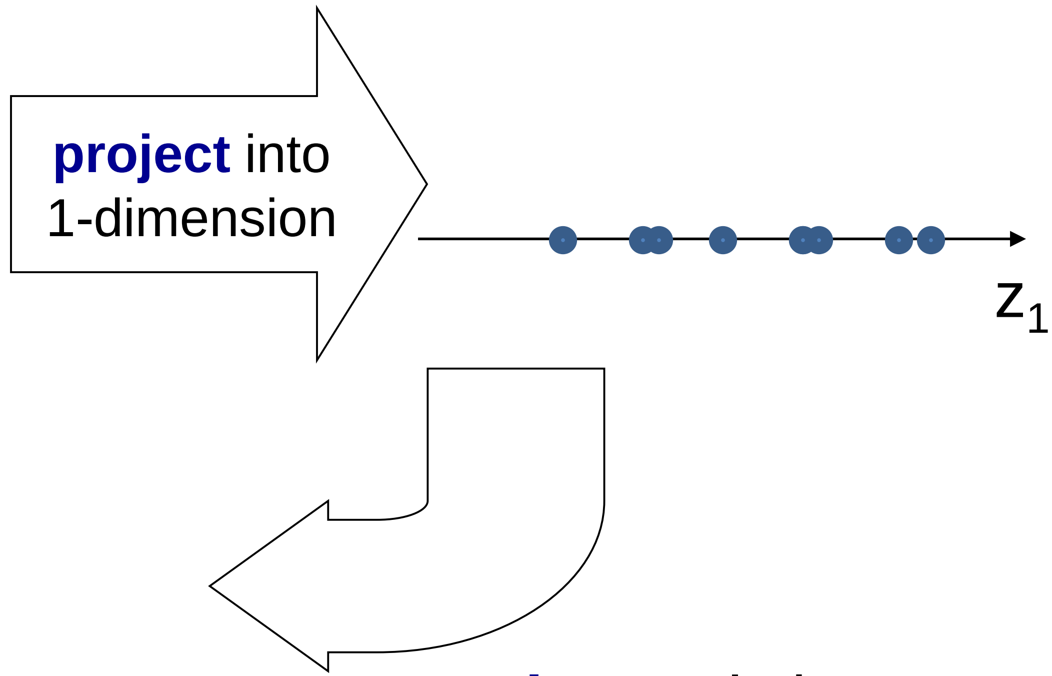
$$z_k = w_0^{(k)} + \sum_i w_i^{(k)} x_i$$

- New features are linear combinations of old ones
- Reduces dimension when k<n
- Let's see this in the unsupervised setting
  - just **X**, but no Y

Linear projection and reconstruction

$$z_1 = w_0^{(1)} + \sum_i w_i^{(1)} x_i$$





reconstruction: only know z_1, and projection weights, what was (x_1, x_2)

Principal component analysis – basic idea

- Project n-dimensional data into k-dimensional space while preserving information:
  - e.g., project space of 10000 words into 3-dimensions
  - e.g., project 3-d into 2-d

- Choose projection with minimum reconstruction error

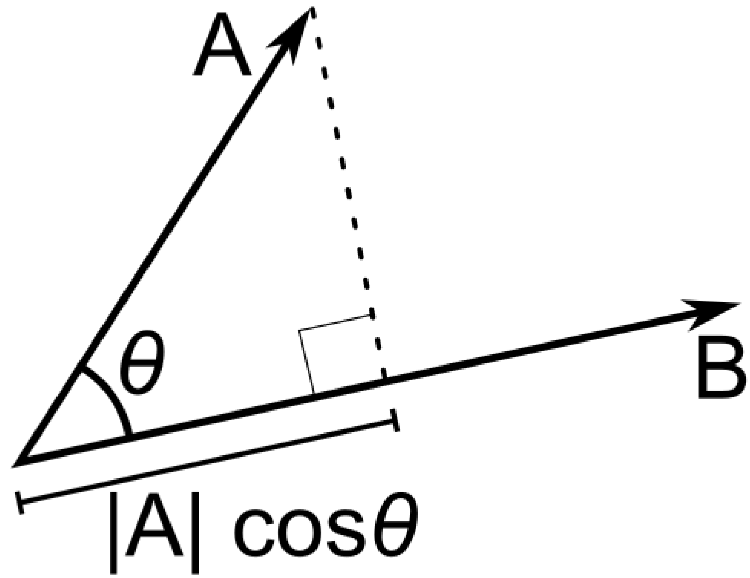
Linear projections, a review

$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

- Project a point into a (lower dimensional) space:
  - **point**: $\mathbf{x} = (x_1,\ldots,x_n)$
  - **select a basis** – set of unit (length 1) basis vectors $(\mathbf{u}_1,\ldots,\mathbf{u}_k)$
    - we consider orthonormal basis:
      - $\mathbf{u}_i \bullet \mathbf{u}_i = 1$, and $\mathbf{u}_i \bullet \mathbf{u}_j = 0$ for $i \neq j$
  - **select a center** – $\bar{\mathbf{x}}$, defines offset of space
  - **best coordinates** in lower dimensional space defined by dot-products: $(z_1,\ldots,z_k)$, $z_i = (\mathbf{x}-\bar{\mathbf{x}}) \bullet \mathbf{u}_i$

Reminder: Vector Projections

- Basic definitions:
  - $A.B = |A||B| \cos \theta$
  - $\cos \theta = |adj|/|hyp|$



- Assume $|B|=1$ (unit vector)
  - $A.B = |A| \cos \theta$
  - So, dot product is length of projection!!!

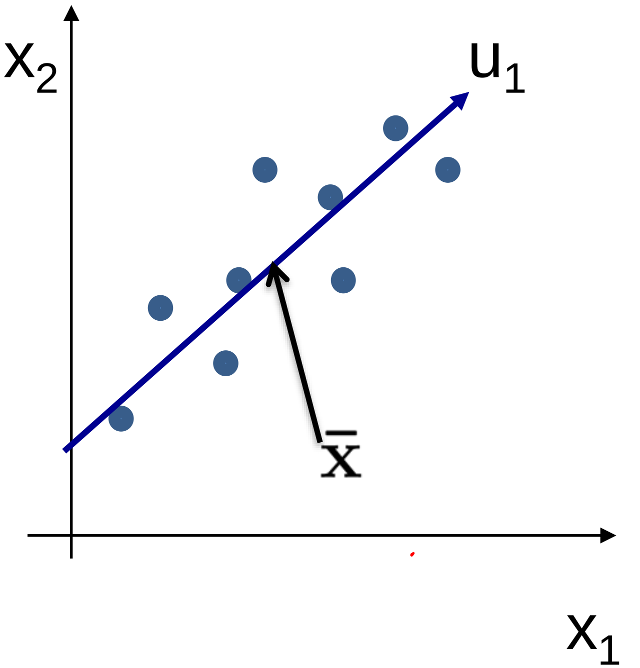
PCA finds projection that minimizes reconstruction error

- Given m data points: $\mathbf{x}^i = (x_1^i,\ldots,x_n^i)$, i=1…m
- Will represent each point as a projection:

$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}^i$$

$$z_j^i = (\mathbf{x}^i - \bar{\mathbf{x}}) \cdot \mathbf{u}_j$$



- PCA:
  – Given k<n, find ($\mathbf{u}_1,\ldots,\mathbf{u}_k$) minimizing reconstruction error:

$$error_k = \sum_{i=1}^{m} (\mathbf{x}^i - \hat{\mathbf{x}}^i)^2$$

Understanding the reconstruction error

$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

$$z_j^i = (\mathbf{x}^i - \bar{\mathbf{x}}) \cdot \mathbf{u}_j$$

- Given k<n, find ($\mathbf{u}_1$,…,$\mathbf{u}_k$) minimizing reconstruction error:

$$error_k = \sum_{i=1}^{m} (\mathbf{x}^i - \hat{\mathbf{x}}^i)^2$$

- Note that $\mathbf{x}^i$ can be represented exactly by n-dimensional projection:

$$\mathbf{x}^i = \bar{\mathbf{x}} + \sum_{j=1}^{n} z_j^i \mathbf{u}_j$$

- Rewriting error:

$$error_k = \sum_{i=1}^{m} \left(x^i - \left[\bar{x} + \sum_{j=1}^{k} z_j^i u_j \right] \right)^2 = \sum_{i=1}^{m} \left(\left[\bar{x} + \sum_{j=1}^{n} z_j^i u_j \right] - \left[\bar{x} + \sum_{j=1}^{k} z_j^i u_j \right] \right)^2$$

$$= \sum_{i=1}^{m} \left(\sum_{j=k+1}^{n} z_j^i u_j \right)^2$$

… $u_i u_j$ is 1 if i==j, and zero otherwise, because us are an orntho-normal basis …

$$= \sum_{i=1}^{m} \sum_{j=k+1}^{n} (z_j^i)^2$$

Error is sum of squared weights that would have be used for dimensions that are cut!!!!

$$error_k = \sum_{i=1}^{m} \sum_{j=k+1}^{n} [\mathbf{u}_j \cdot (\mathbf{x}^i - \bar{\mathbf{x}})]^2$$

Reconstruction error and covariance matrix

$$\boxed{error_k = \sum_{i=1}^{m} \sum_{j=k+1}^{n} [\mathbf{u}_j \cdot (\mathbf{x}^i - \bar{\mathbf{x}})]^2}$$

$$= \sum_{i=1}^{m} \sum_{j=k+1}^{n} u_j^T (x^i - \bar{x})(x^i - \bar{x})^T u_j$$

$$= \sum_{j=k+1}^{n} u_j^T \left[\sum_{i=1}^{m} (x^i - \bar{x})(x^i - \bar{x})^T \right] u_j$$

$$\boxed{error_k = m \sum_{j=k+1}^{n} u_j^T \Sigma u_j}$$

$$\boxed{\Sigma = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{x}^i - \bar{\mathbf{x}})(\mathbf{x}^i - \bar{\mathbf{x}})^T}$$

Now, to find the u_j, we minimize:

$$u^T \Sigma u + \lambda(1 - u^T u)$$

Lagrange multiplier to ensure orthonormal

Take derivative, set equal to 0, …, solutions are eigenvectors

$$\boxed{\Sigma u_i = \lambda_i u_i}$$

Minimizing reconstruction error and eigen vectors

- Minimizing reconstruction error equivalent to picking orthonormal basis ($\mathbf{u}_1,\ldots,\mathbf{u}_n$) minimizing:

$$error_k = m \sum_{j=k+1}^{n} u_j^T \Sigma u_j$$

- Solutions: eigen vectors

$$\Sigma u_i = \lambda_i u_i$$

- So, minimizing reconstruction error equivalent to picking ($\mathbf{u}_{k+1},\ldots,\mathbf{u}_n$) to be eigen vectors with smallest eigen values
- And, our projection should be onto the ($\mathbf{u}_1,\ldots,\mathbf{u}_k$) with the largest values

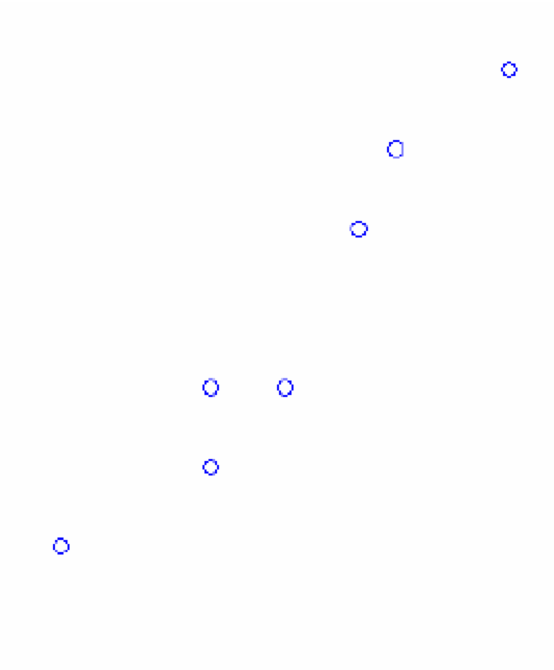
Basic PCA algorithm

- Start from m by n data matrix **X**
- **Recenter**: subtract mean from each row of **X**
  - $\mathbf{X_c} \leftarrow \mathbf{X} - \overline{\mathbf{X}}$
- **Compute covariance** matrix:
  - $\Sigma \leftarrow 1/m\ \mathbf{X_c}^{\mathbf{T}}\ \mathbf{X_c}$
- Find **eigen vectors and values** of Σ
- **Principal components:** k eigen vectors with highest eigen values

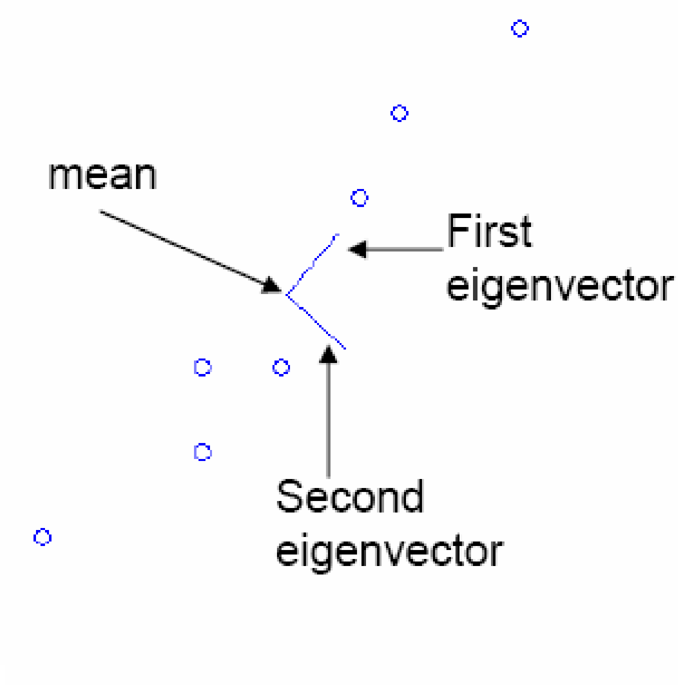
PCA example

$$\hat{\mathbf{x}}^i = \bar{\mathbf{x}} + \sum_{j=1}^{k} z_j^i \mathbf{u}_j$$

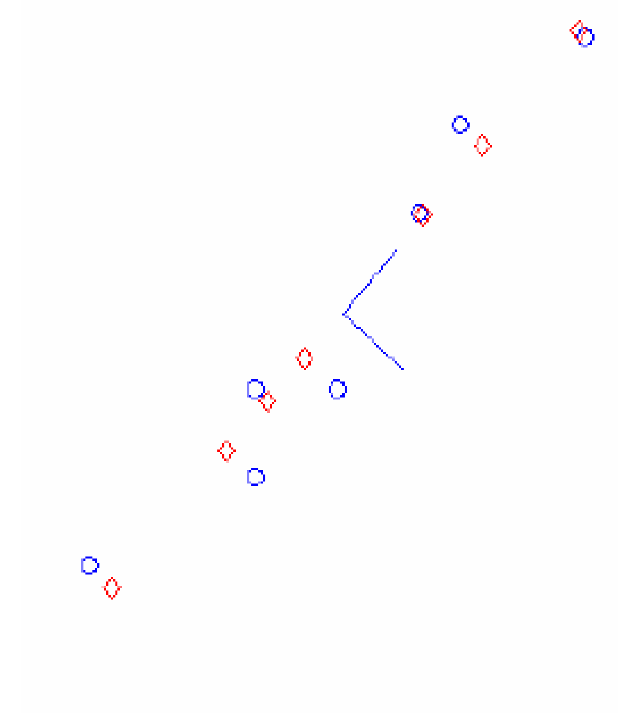
Data:



Projection:



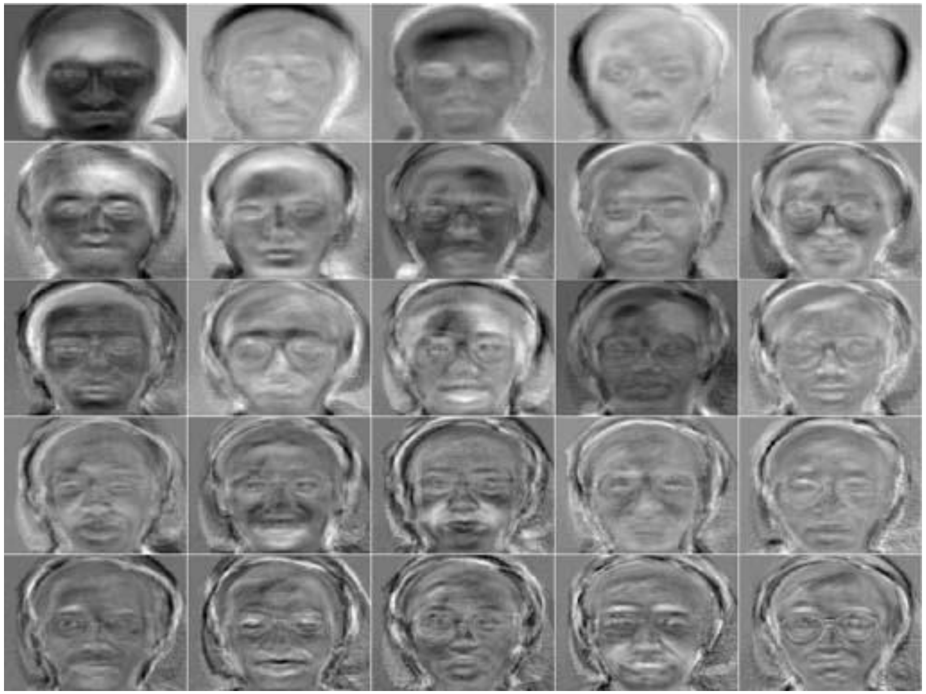
Reconstruction:



Eigenfaces [Turk, Pentland ’91]

- Input images:

- Principal components:



Eigenfaces reconstruction

- Each image corresponds to adding together the principal components:

What you need to know

- Dimensionality reduction
  – why and when it's important
- Simple feature selection
- Regularization as a type of feature selection
- Principal component analysis
  – minimizing reconstruction error
  – relationship to covariance matrix and eigenvectors