PGE 383 Subsurface Machine Learning

Lecture 8: Dimensionality Reduction

Lecture outline:

- **Curse of Dimensionality**
- **Dimensionality Reduction**
- **Principal Component Analysis**
- **Principal Component Analysis Examples**
- **Principal Component Analysis Hands-on**

© 2019 Michael Pyrcz

Motivation

We work with highly multivariate datasets

- **Feature Projection** to a lower dimension may improve interpretability, and modeling accuracy by:
  - avoiding overfit and multicollinearity

Recall Feature Engineering

Feature selection, transformation, and projection

- Working with features that have more information to improve machine learning performance, i.e., prediction accuracy

© 2019 Michael Pyrcz

PGE 383 Subsurface Machine Learning

Lecture 8: Dimensionality Reduction

Lecture outline:

- **Curse of Dimensionality**

© 2019 Michael Pyrcz

Curse of Dimensionality

The challenges of working with many features (i.e., high dimensional models)

- impossible to visualize in high dimensionality
- insufficient sampling
- low coverage of feature space
- distorted feature space
- multicollinearity between features

© 2019 Michael Pyrcz

Curse of Dimensionality Visualization

Consider this simple model:

- 1 predictor feature
- 1 response feature

How's our model performing?

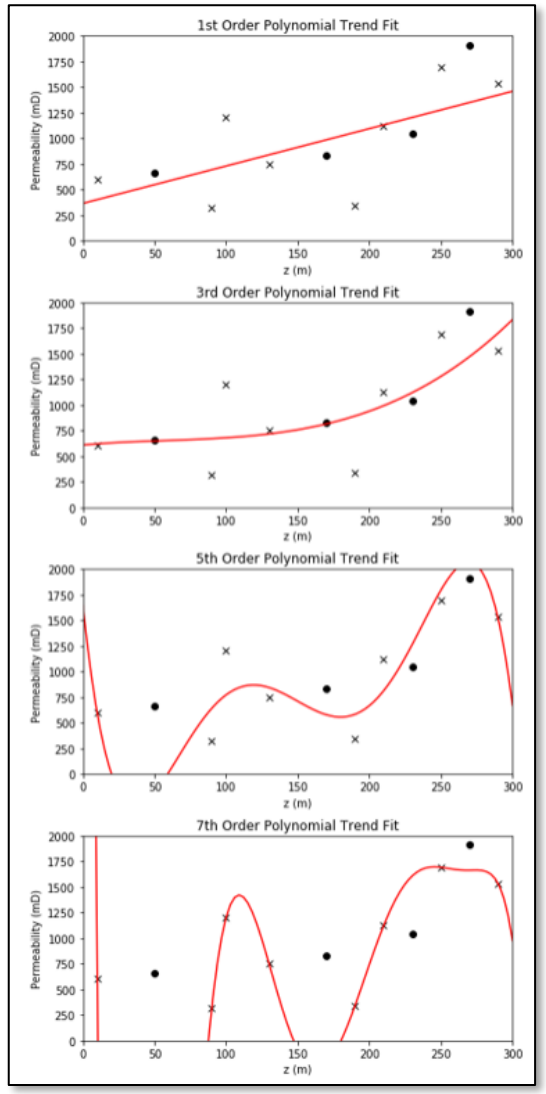
- Accuracy in training and testing

Range of Applicability?

- Are we extrapolating?

Overfit

- Is the model defendable given the data?



Simple to complicated prediction models, from MachineLearning_model_tuning chapter of Applied Machine Learning in Python e-book.

© 2019 Michael Pyrcz

Curse of Dimensionality Visualization

Consider this simple model:

- 2 predictor feature
- 1 response feature

How's our model performing?

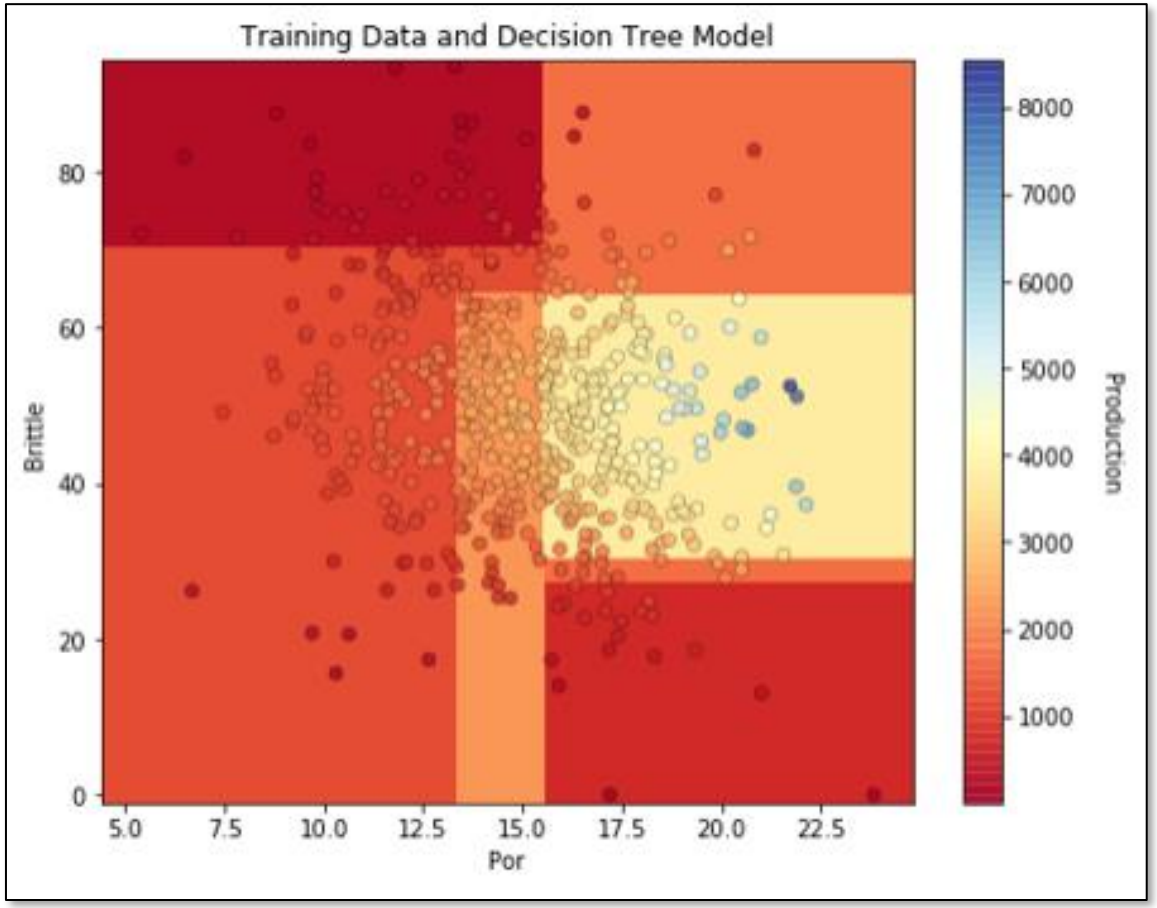
- Accuracy in training and testing

Range of Applicability?

- Are we extrapolating?

Overfit

- Is the model defendable given the data?



Two predictor feature decision tree prediction model, from MachineLearning_decision_tree chapter of Applied Machine Learning in Python e-book.

© 2019 Michael Pyrcz

Curse of Dimensionality Visualization

Consider this simple model:

- 4 predictor feature
- 1 response feature

How's our model performing?

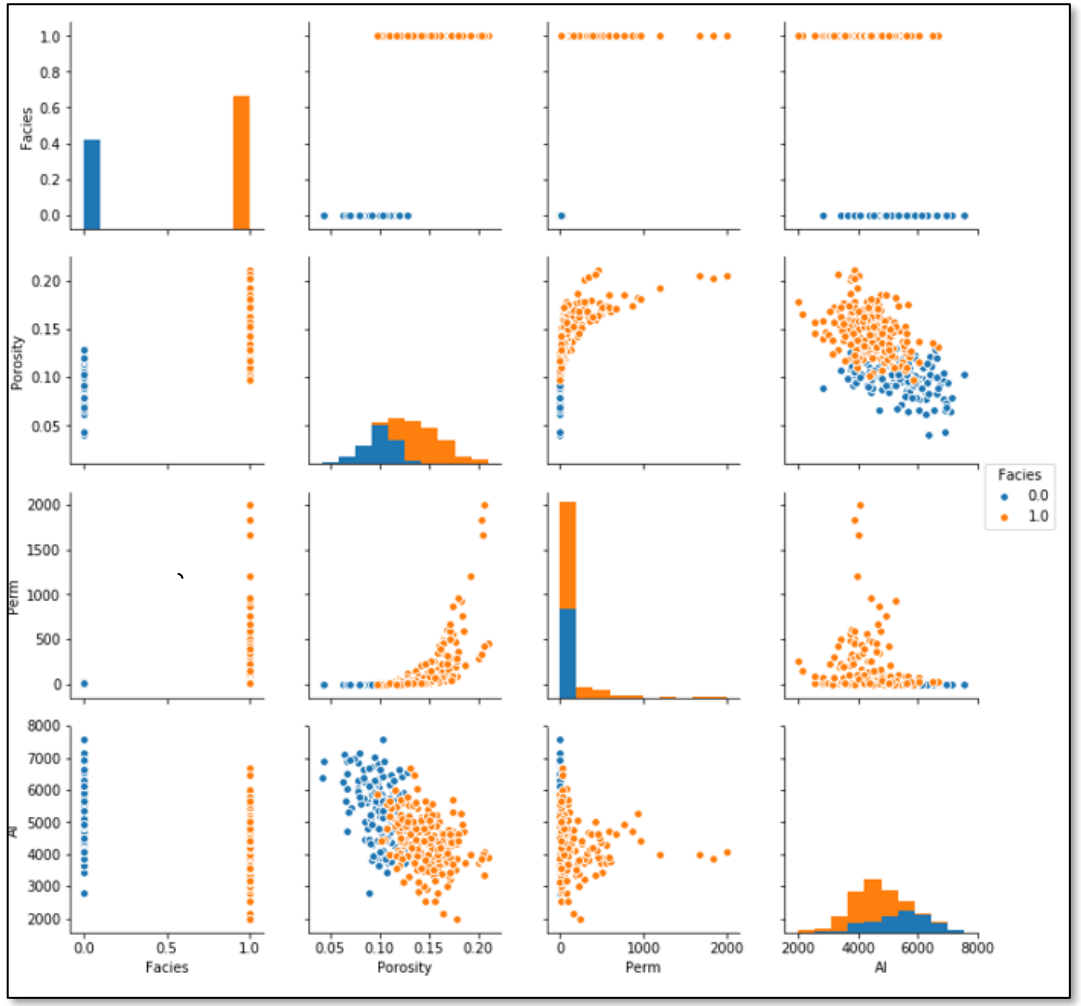
- Accuracy in training and testing

Range of Applicability?

- Are we extrapolating?

Overfit

- Is the model defendable given the data?



Matrix scatter plot, from Bivariate Analysis chapter of course e-book.

© 2019 Michael Pyrcz

Curse of Dimensionality
Sampling

Recall the calculation of a histogram / normalized histogram.

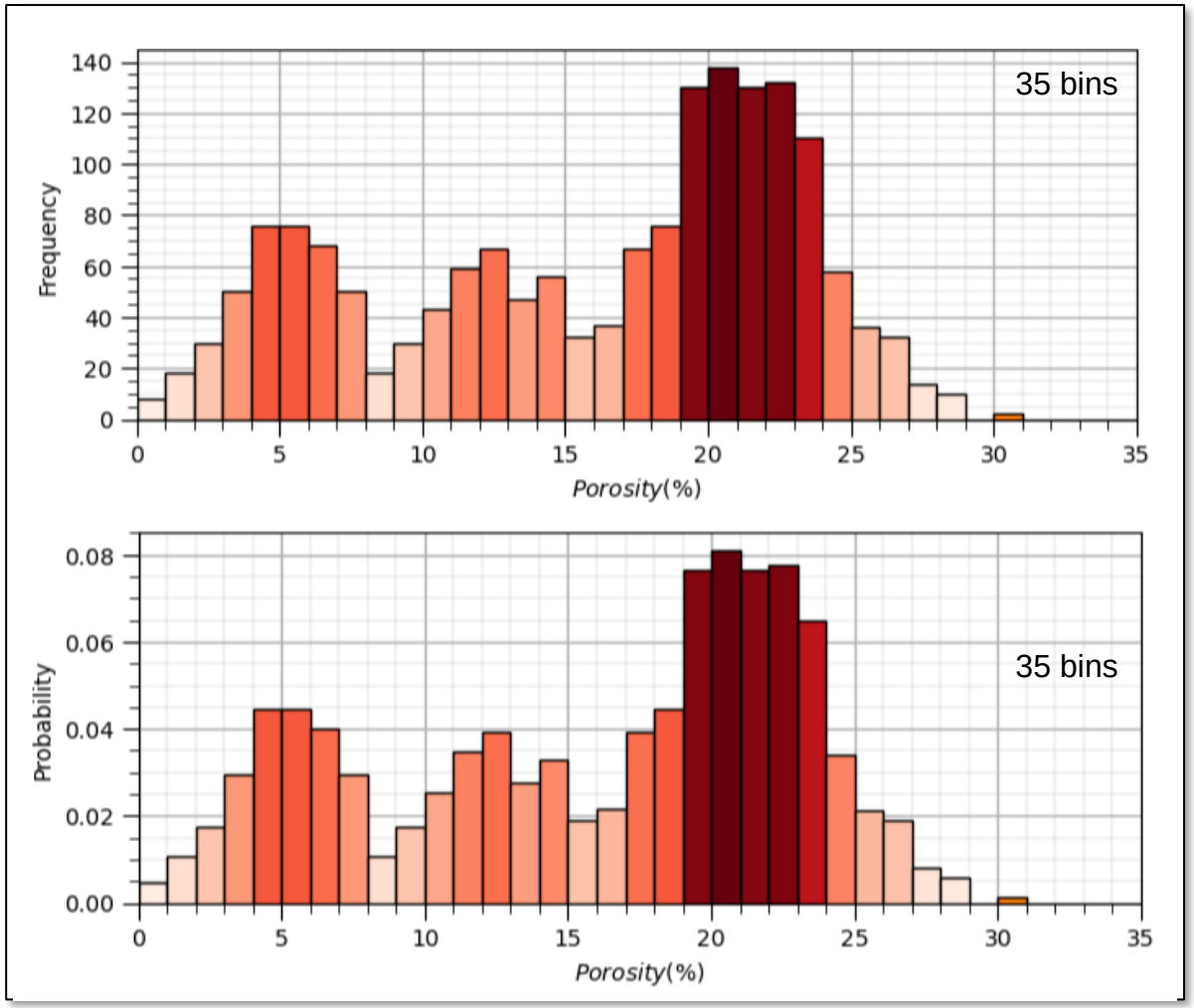
1. We establish ‘bins’, discretize the range of each feature.

2. We calculate probabilities with a ratio and enough samples/bin.

Number of samples needed, approx.,

$$n = n_{s/bin} \cdot n_{bins}$$

$$n = 10 \cdot 35 = 350 \text{ data}$$



Histogram and normalized histogram, from PythonDataBasics_Bivariate_Visualization.ipynb.

$$n(X_1^i \leq X \leq X_1^{i+1})$$

Samples to infer this probability for each bin.

$$P(X_1^i \leq X \leq X_1^{i+1}) = \frac{n(X_1^i \leq X \leq X_1^{i+1})}{n}$$

© 2019 Michael Pyrcz

Curse of Dimensionality Sampling

Calculating the Joint Probability

Consider any joint probability:

$$P\left(X_1 \in bin_{i_1} \cap, \ldots, \cap X_m \in bin_{i_m}\right)$$

- E.g. $P(10\% < \varphi < 11\%, 600 < k < 640) = n(10\% < \varphi < 11\%, 600 < k < 640)/n$

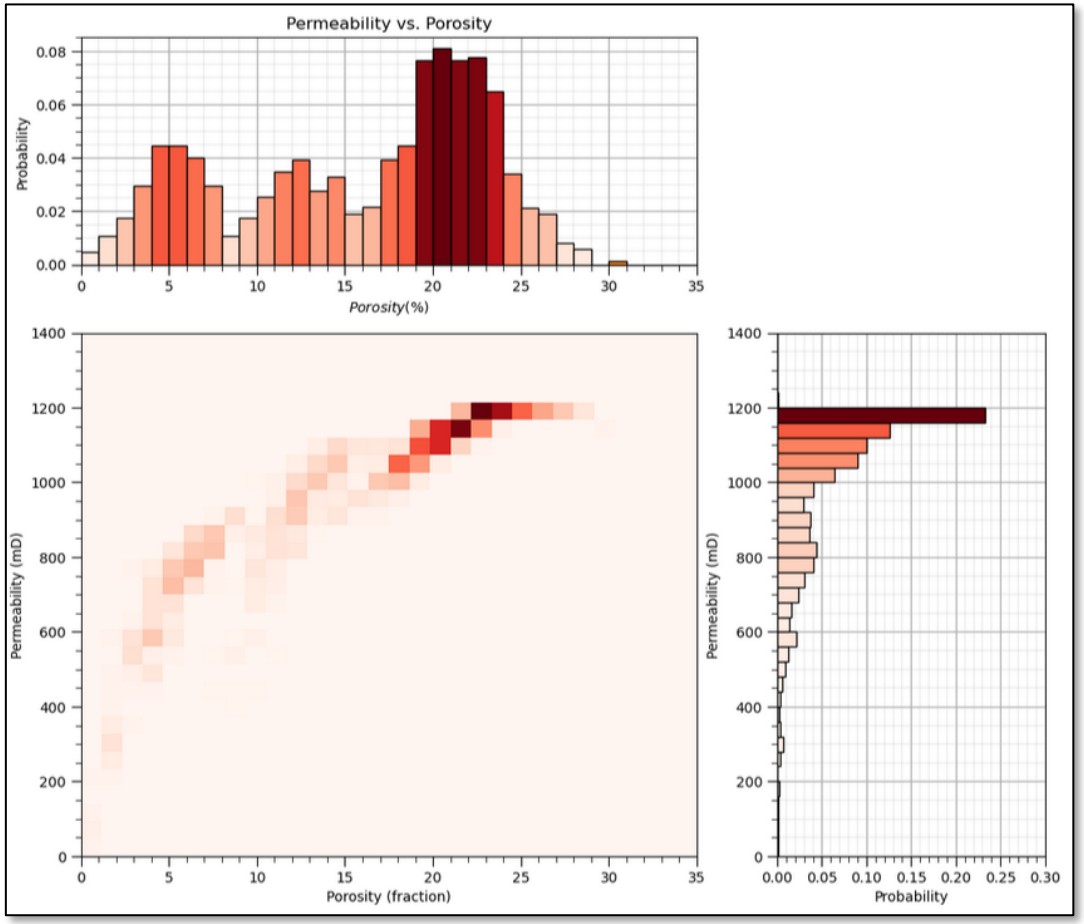
where n is the total number of samples.

- We need enough samples, n , replicates of all possible combinations to go from frequency to probability.
- Where, $n_{s/bin}$, is the nominal number of samples per combination.

$$n = n_{s/bin} \cdot {n_{bins}}^m$$

- Note: this is optimistic, as it assumes uniform sampling

  For this example, $n = n_{s/bin} \cdot {n_{bins}}^m = 10 \cdot 35^2$,
  given we need 10 nominally samples per bin,
  we need 12,250 data.



Joint PDF with normalized histograms, from PythonDataBasics_Bivariate_Visualization.ipynb.

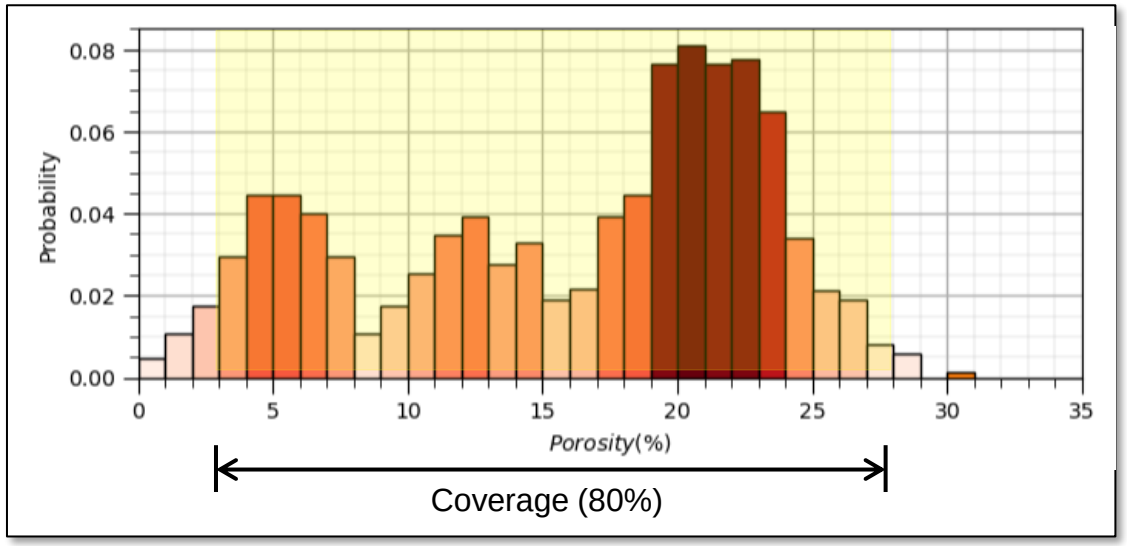
© 2019 Michael Pyrcz

Curse of Dimensionality Coverage

Sample Coverage

The range of the sample values

- The fraction of the possible solution space that is sampled
- Let's return to 1 feature and assume 80% coverage!
- That's pretty good right?
- Remember, we usually, directly sample only $\frac{1}{10^7}$ of the volume of the subsurface
- Yes, the concept of coverage is subjective, how much data to cover? What about gaps? etc.



Histogram and normalized histogram, from PythonDataBasics_Bivariate_Visualization.ipynb.

© 2019 Michael Pyrcz

Curse of Dimensionality
Coverage

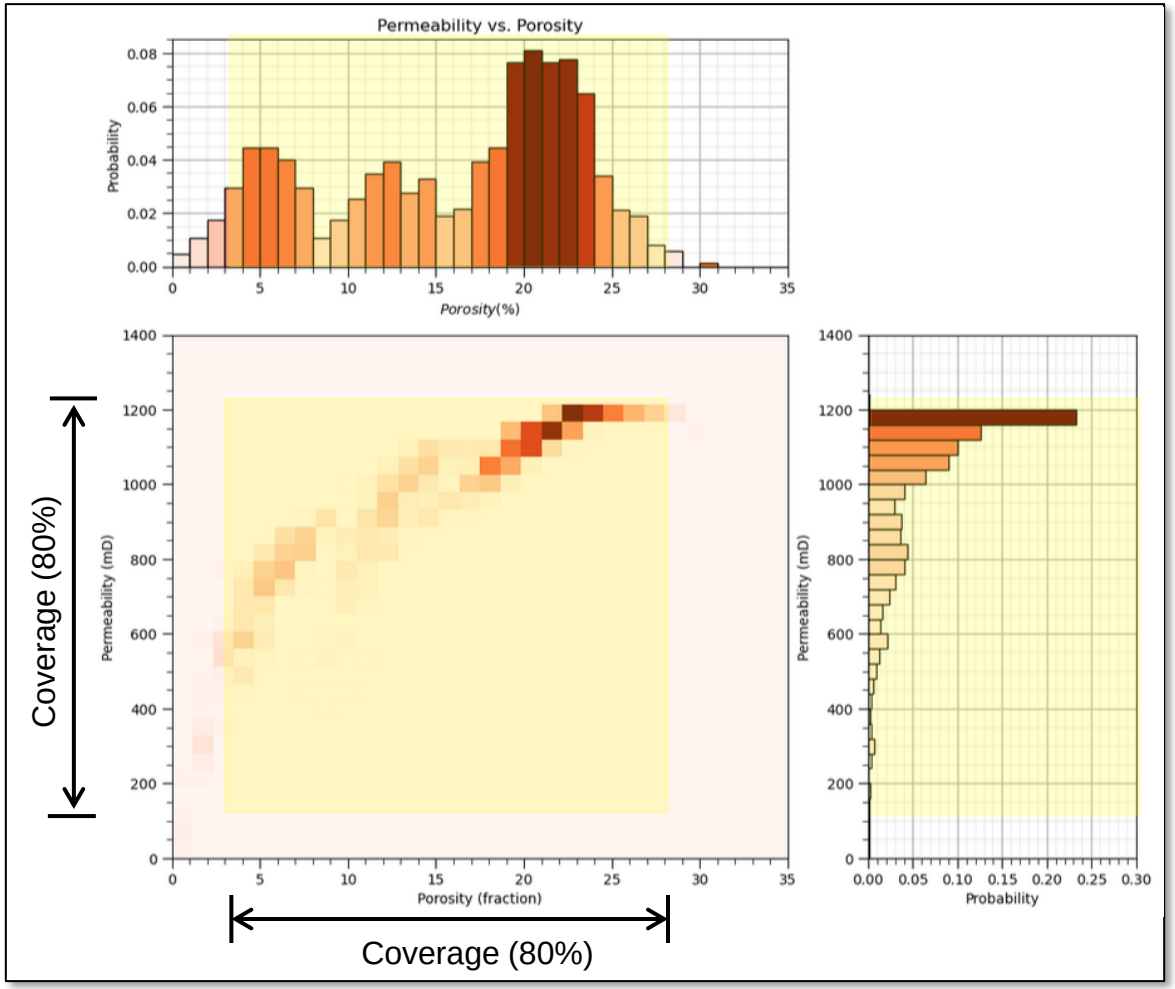
Sample Coverage

It is common not to have samples that cover the entire predictor feature space

- Consider coverage over each feature, c_1
- How much of the solution space is covered?

$$c = c_1^m$$

- coverage is decreasing, exponential decay with decay constant, $\lambda = 1$, as we increase the number of features, m!



Coverage for this example is, $c = c_1^m = 0.8^2 = 0.64$, image from PythonDataBasics_Bivariate_Visualization.ipynb.

© 2019 Michael Pyrcz

Curse of Dimensionality
Distorted Space

Distances in High Dimensional Space

Hyperdimensional space is distorted,

- Take the ratio of the volume of an inscribed hypersphere in a hypercube.

$$\frac{\pi^{m/2}}{m2^{m-1}\Gamma(m/2)} \rightarrow 0 \text{ as } m \rightarrow \infty$$

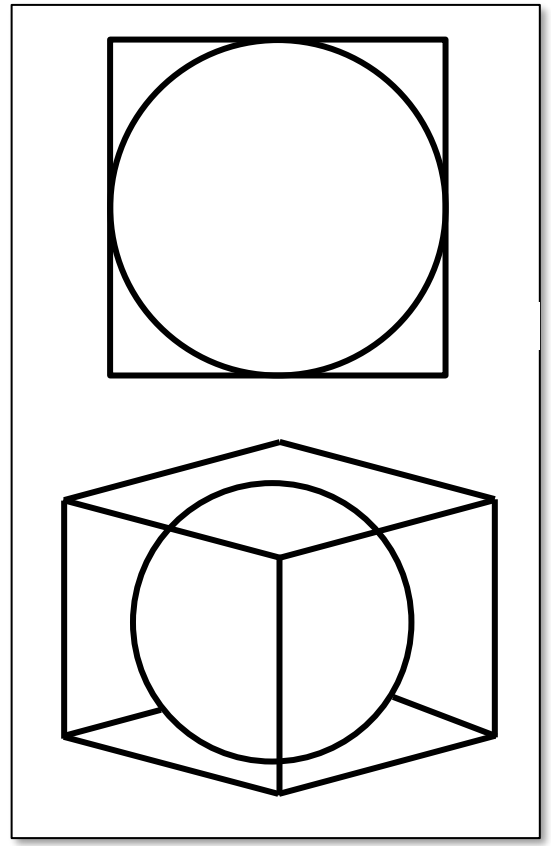
Recall, $\Gamma(n) = (n-1)!$

- High dimensional space is all corners and no 'middle' and most of high dimensional space is far from the middle (all corners!).

Distance in hyperdimensional space loses variance,

$$\lim_{m \rightarrow \infty} E\{dist_{max}(m) - dist_{min}(m)\} \rightarrow 0$$

- The limit of the expectation of the range of pairwise distances over random points in hyperdimensional space tends to zero.
  - Distances are almost all the same, Euclidian distance is no longer meaningful



Ratio of (hyper)sphere inscribed in (hyper)cube.

© 2019 Michael Pyrcz

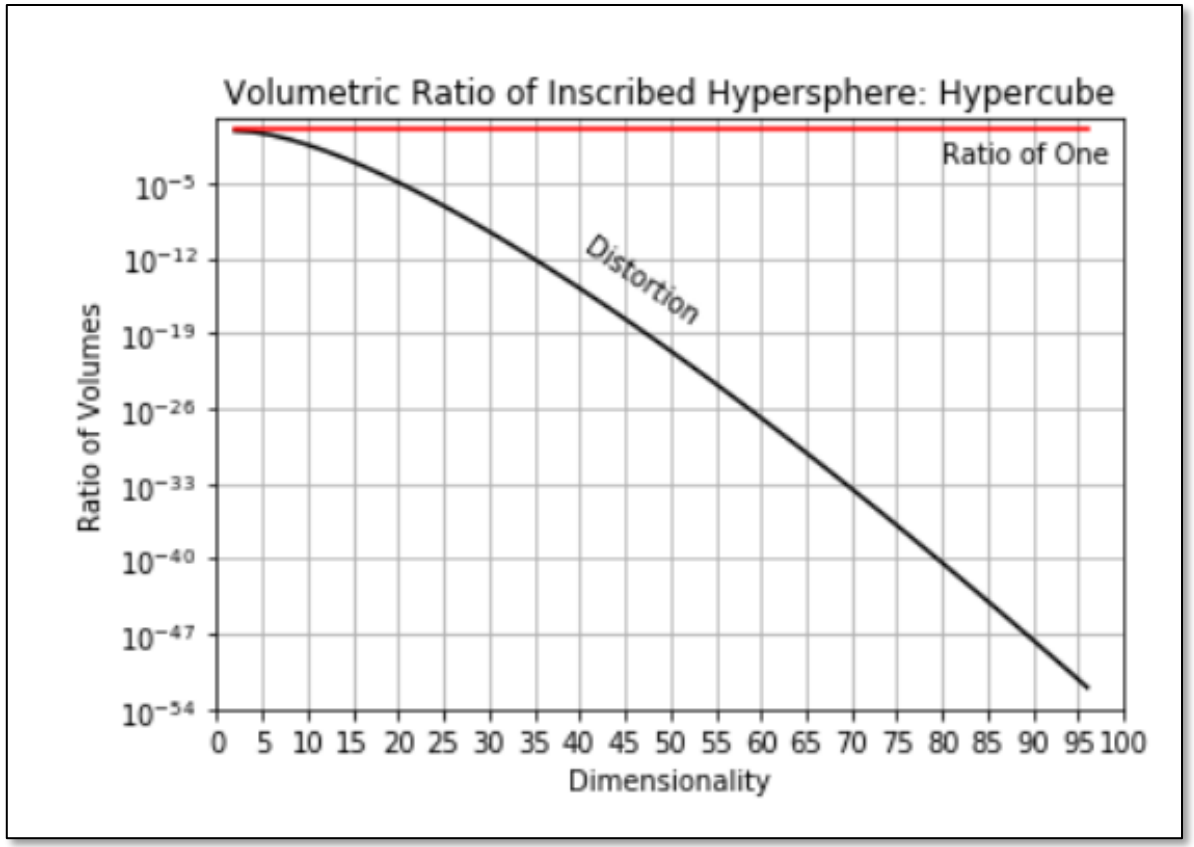
Curse of Dimensionality
Distorted Space

Distortion of High Dimensional Space

How many dimensions, features to observe the curse of dimensionality?

- Here's the ratio of center to corners from the previous equation.
- 10 or more features, the distortion is severe,

m	mD / 2D
2	1.0
5	0.28
10	0.003
20	0.00000003



Volumetric ratio of an inscribed hypersphere (n-sphere) in a hypercube (n-cube), note y-axis is log-scale.

© 2019 Michael Pyrcz

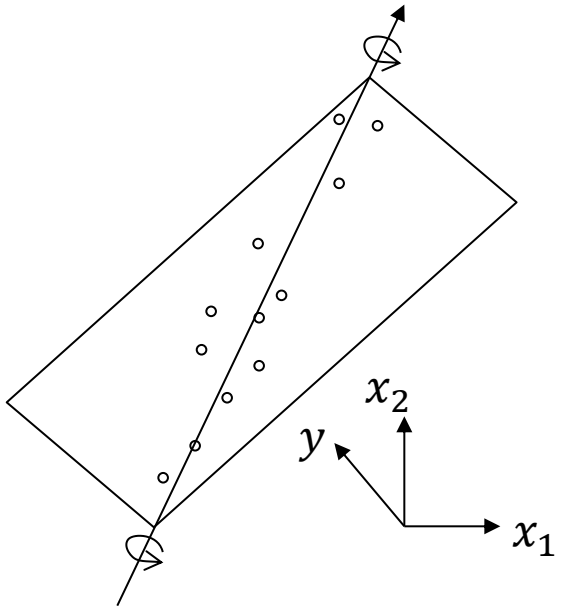
Curse of Dimensionality
Multicolinearity

"*The existence of such a high degree of correlation between supposedly independent variables being used to estimate a dependent variable that the contribution of each independent variable to variation in the dependent variable cannot be determined*"

- Merriam-Webster Online Dictionary

"*In statistics, multicollinearity (also collinearity) is a phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy.*"

- Wikipedia



Colinearity of x_1 and x_2 when modeling y = f(x_1, x_2), is like fitting a plane to a line!

© 2019 Michael Pyrcz

Curse of Dimensionality

Working with More Features is More Difficult

1. More difficult to visualize data and model
2. More data are required to infer the joint probabilities
3. Less data coverage of feature space
4. More difficult to interrogate / check the model
5. More likely redundant features resulting in model instability
6. More computational effort, more computational resources and longer run times
7. More complicated model is more likely overfit
8. More professional time for model construction

© 2019 Michael Pyrcz

Motivation for Dimensionality Reduction

Feature Projection to Address the Curse of Dimensionality

We get a better model with fewer, informative features than

‘Throwing everything and the kitchen sink into the model!’

Fewer features for models are simpler, faster, easier to visualize and less likely overfit.

Feature projection takes care of multicollinearity!

© 2019 Michael Pyrcz

PGE 383 Subsurface Machine Learning

Lecture 8: Dimensionality Reduction

Lecture outline:

- **Dimensionality Reduction**

© 2019 Michael Pyrcz

Dimensionality Reduction

- Motivated by the curse of dimensionality and multicollinearity
- Known as dimension reduction or dimensionality reduction
- Applied in statistics, machine learning and information theory
- Multiple strategies:

1. Features Selection – find the subset of original features that are most important for the problem (big hitters).

We already covered this

2. Feature Projection – transform the data from a higher to lower dimensional space

© 2019 Michael Pyrcz

Feature Selection Recall

Consider a wide-array approach to assess variable importance.

- Here's the general types of metrics that we will consider for feature ranking:

1. Visual Inspection of Data Distributions and Scatter Plots
2. Statistical Summaries
3. Model-based
4. Recursive Feature Elimination

© 2019 Michael Pyrcz

Feature Selection Recall

Expert Knowledge

Also, we should not neglect expert knowledge.

- If additional information is known about physical processes, causation, reliability and availability of features this should be integrated into assigning feature ranks.

- We should be learning as we perform our analysis, testing new hypotheses.

© 2019 Michael Pyrcz

Feature Projection

- Dimensionality reduction by feature projection transforms the data to a lower dimension

- Given features, $X_1, \ldots, X_m$ we would require $\binom{m}{2} = m(m-1)/2$ scatter plots to visualize just the two-dimensional scatter plots.

- These many 2D representations do not capture $> 2D$ structures due to marginalization!

- Once we have 4 or more variables understanding our data gets very challenging.
  - Recall the curse of dimensionality. It extends to visualization, not just sampling!

© 2019 Michael Pyrcz

Feature Projection

Find a good lower dimensional, p, representation of the original dimensions m.

$$p \ll m$$

The Benefits:

- Data storage, computational time, and visualization
- Modeling with $p \ll m$ features takes care of multicollinearity because new features are linearly independent after projection

You're New Features are Independent

- The Limitations:
  - It may be more difficult to understand the model
  - The new features, p, are combinations of the original features, m, they lose their physical meaning!

© 2019 Michael Pyrcz

Feature Projection

Variety of feature projecton methods:

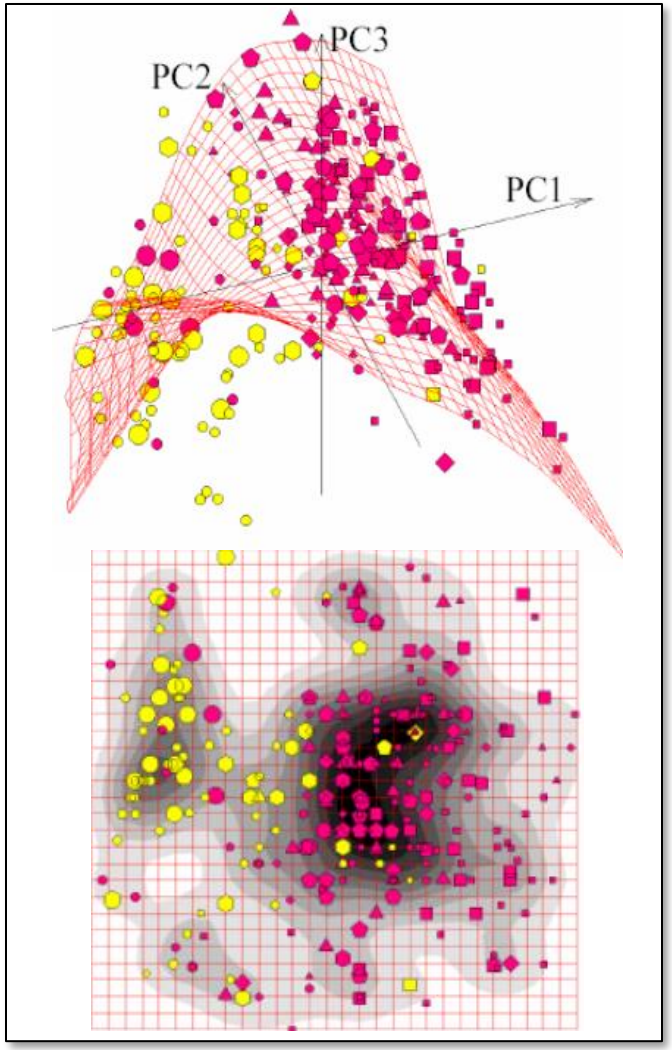
Principal Component Analysis (PCA) (will cover this)

- **Linear mapping** of the data to lower dimensional space
- **Maximizes the variance explained** by the reduced subset of features

Nonlinear Principal Component Analysis (Kernel PCA)

- **Nonlinear mapping** of the data to higher dimensional space with the **kernel trick**
- **Kernel Trick** – use of a kernel function to operate in higher dimensional feature space with only the ‘similarity’ between the data samples
- We will cover the kernel trick with support vector machines

More later about this with Support Vector Machines



Nonlinear PCA manifold for 3D to 2D, image from Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques, Olivas E.S. et al Eds. Information Science Reference, IGI Global: Hershey, PA, USA, 2009. 28–59.

© 2019 Michael Pyrcz

Feature Projection

Variety of feature projecton methods:

Factor Analysis

- Like PCA, linear combinations of the features
- Explain relationships between predictor features by grouping into fewer “factors” / latent variables

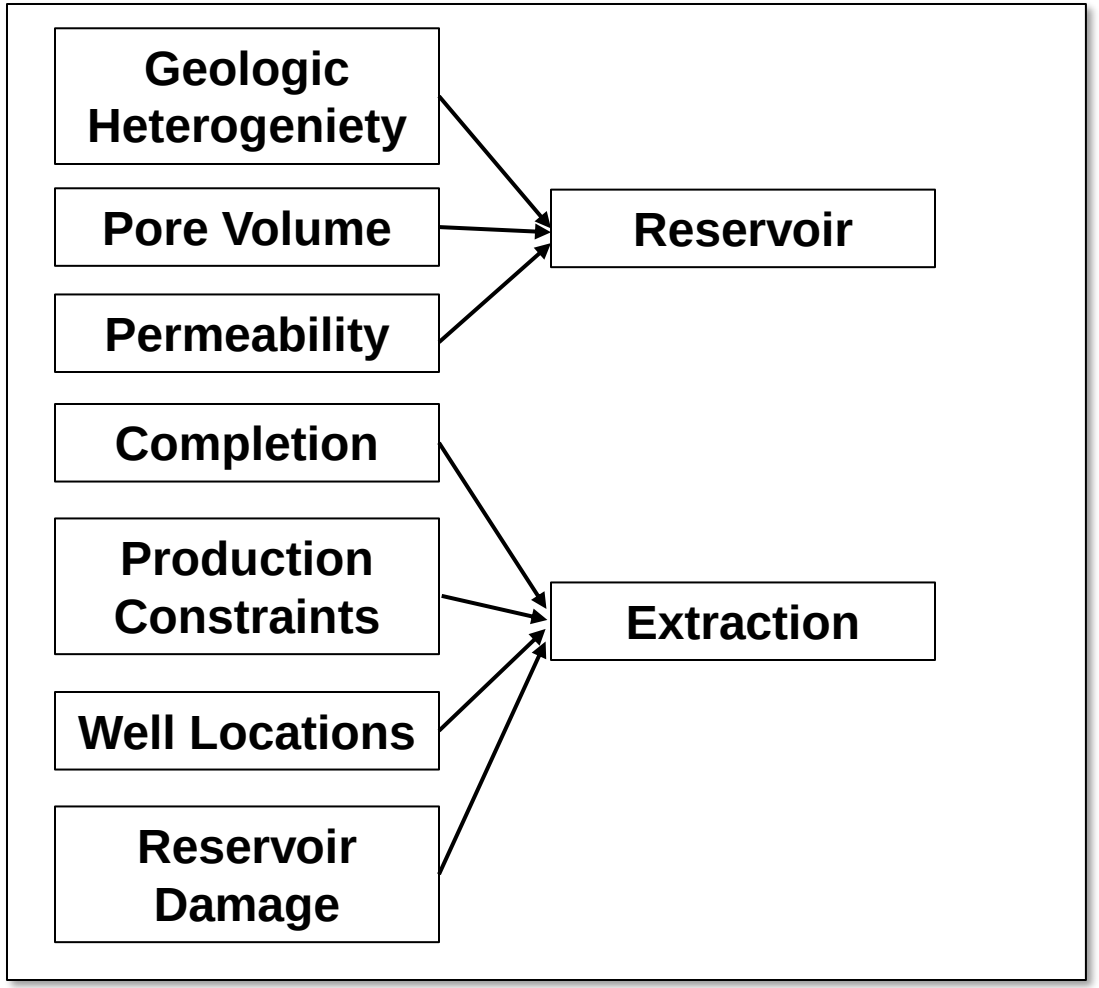


Illustration of factor analysis for dimensionality reduction.

© 2019 Michael Pyrcz

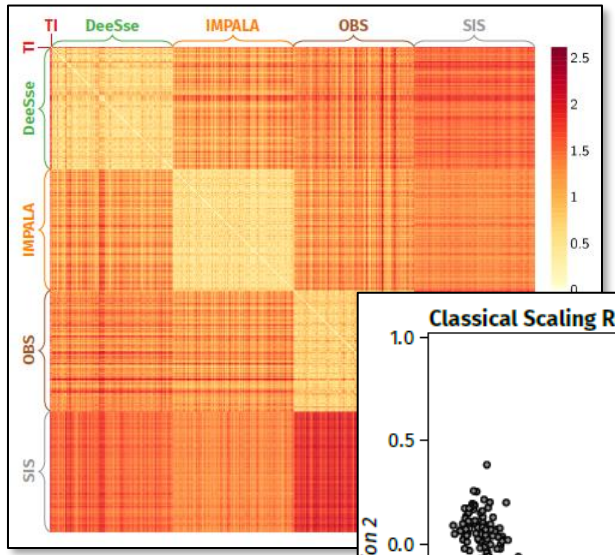
Feature Projection

Variety of feature projection methods:

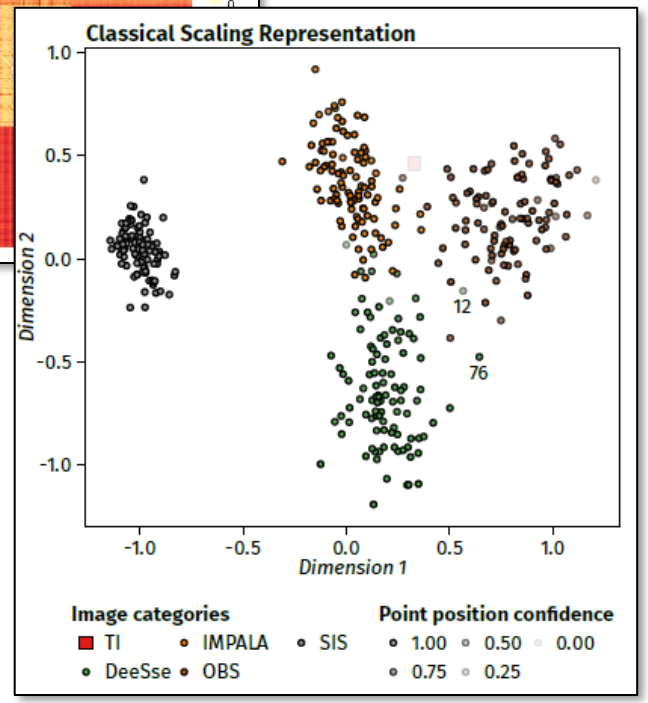
Multidimensional Scaling (MDS)

- Ordination technique for information visualization
- Non-linear dimensional reduction
- Given a matrix of pairwise distances between all data, project to lower dimensional space, p, such that the between sample distance is preserved as well as possible.

We will cover this in our next lecture.



Dissimilarity based on combination of metrics: proportions, transitions, connectivity, shape, networks



MDS to Visualize Model Uncertainty Space Sampled with Scenarios and Realizations, image from from Rongier, G. Ph.D. thesis.

© 2019 Michael Pyrcz

PGE 383 Subsurface Machine Learning

Lecture 8: Dimensionality Reduction

Lecture outline:

- **Principal Component Analysis**

© 2019 Michael Pyrcz

Principal Component Analysis

Explained 3 ways,

- best fitting interpretation – *for modellers*
- orthogonal rotation to maximize variance – *for visual learners*
- eigenvalues and eigenvectors of covariance matrix – *looking under the hood at the engine*

© 2019 Michael Pyrcz

Principal Component Analysis

'Best Fitting' Interpretation

find V_p orthogonal projection, new basis (1D vector, 2D plane, or >2D hyperplane) that minimizes the orthogonal projection / error

$$min \sum_{i-1}^{n} \left((X_i - \bar{X}) - (X_i - \bar{X}) V_p V_p^T \right)^2$$

Centered original data — $(X_i - \bar{X})$

Data Projection with p components — $(X_i - \bar{X}) V_p V_p^T$

sum of orthogonal projections

$$(X_i - \bar{X}) V_p = v_i$$ principal component scores

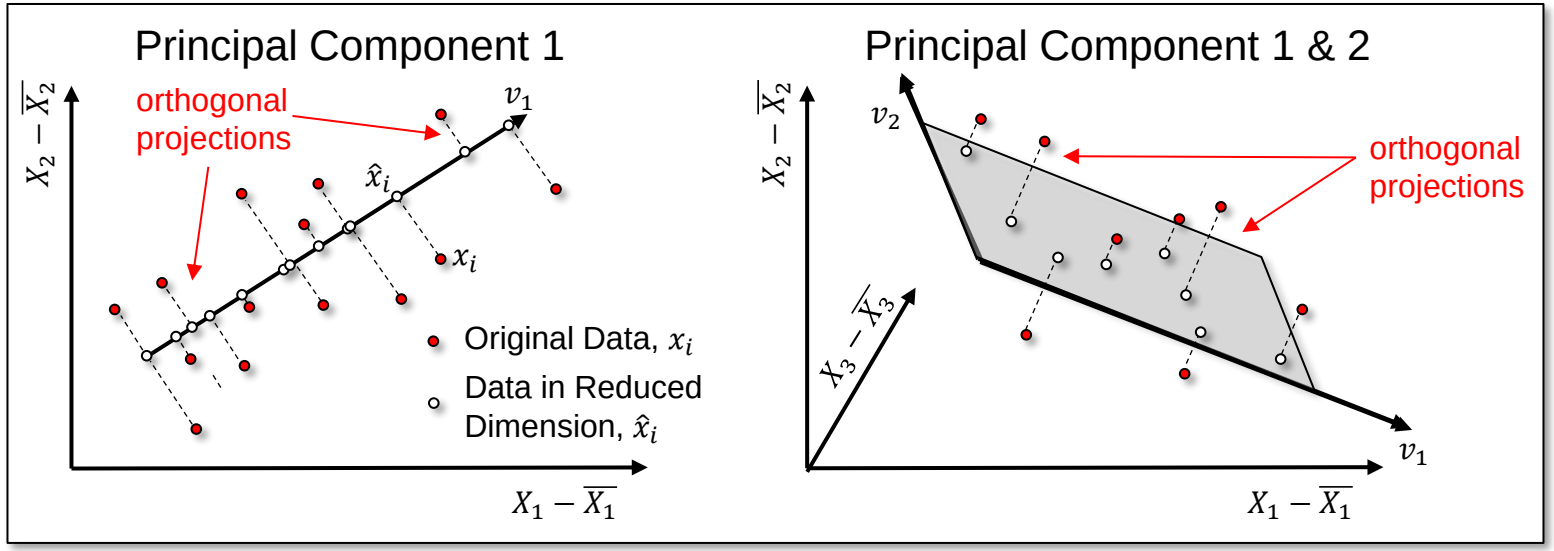
Back transform, reduced dimension, p

$$(X_i - \bar{X}) V_p V_p^T = \hat{X}$$ lower dimensional data

Recall, $V_p^T = V_p^{-1}$ Given orthogonality

where,

- V_p is a matrix of principal component loadings for our first p vectors
- X is a matrix with the original m dimensional features
- $\bar{X}$ is a vector of the means
  v_i are the principal component scores
- $\hat{X}$ is the features in p lower dimensional space



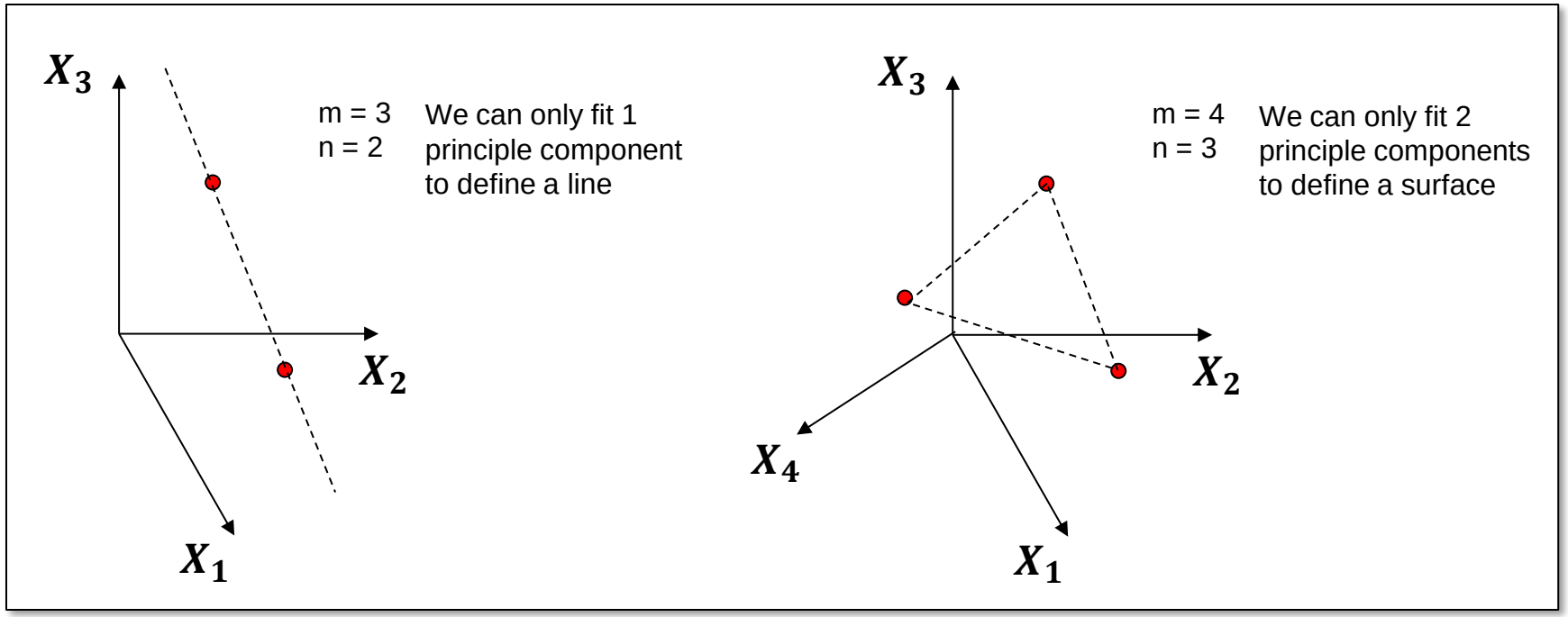
The best-fit line in 2D (left) and plane in 3D (right) that best describes the data (2 different examples).

© 2019 Michael Pyrcz

Principal Component Analysis

Best Fitting, the number of principal components available are $\min(n-1, m)$

Limited by the number of features, m, and the number of data, n



Sample data in 3D (left) and 4D (right) feature space.

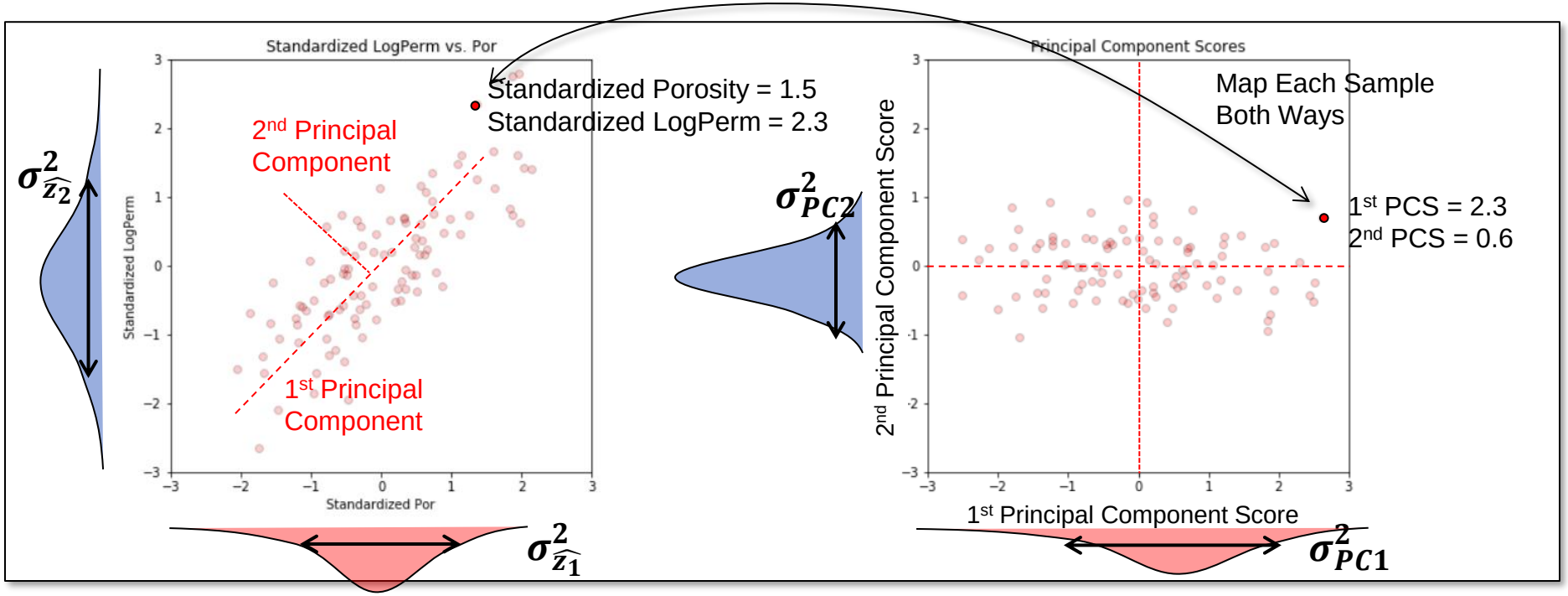
For the remainder of this lecture, we assume n is not a limiter, it would not be a good idea to model with such a short (few samples) and wide (many featues) data table!

© 2019 Michael Pyrcz

Principal Component Analysis

Orthogonal Transformation / Rotation to Maximize Variance

- Convert a set of observations into a set of linearly uncorrelated features known as **principal components**, **linear combinations of the original features.**
- Forms a **new orthogonal basis**, the features are uncorrelated with each other
- The sample values transformed to each principle component are known as **principal component scores**
- **Orthogonal transformations are rotations**, there are no distortions, that maximize variance.



© 2019 Michael Pyrcz

Principal Component Analysis

Orthogonal Transformation / Rotation to Maximize Variance

- Maximize the variance on the 1st principal component, the variance of the 1st principal component scores
- Then maximize the remaining variance on the 2nd principal component, constrained by orthogonality to the 1st principal component
- Then maximize the remaining variance on the 3rd principal component, constrained by orthogonality to the 1st and 2nd principal component
- up to $(\boldsymbol{n-1, m})$ possible principal components

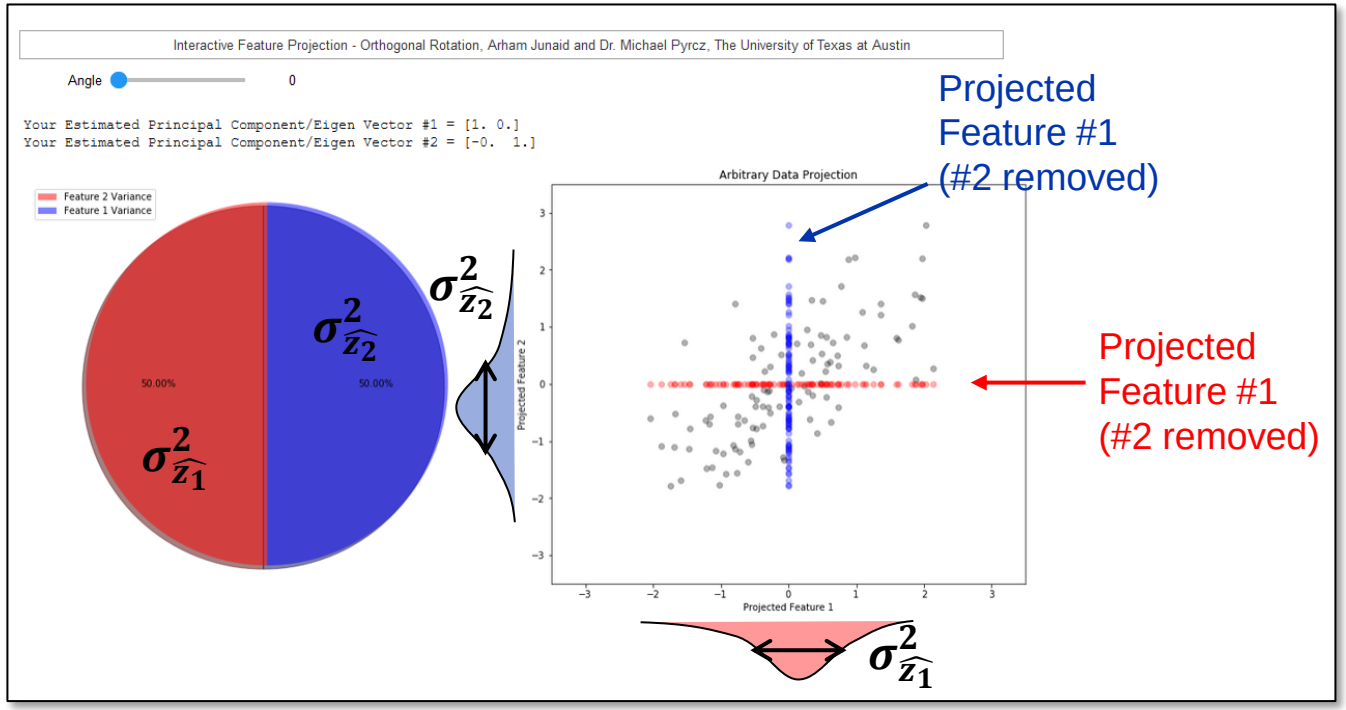
© 2019 Michael Pyrcz

Principal Component Analysis

Now Recall Orthogonal Transformation → Rotation

- Experiential Learning – let's take a dataset and rotate it.

Can we actually maximize the variance seen / explained in a single direction?



$$\begin{bmatrix} \boldsymbol{cos(\theta)} & \boldsymbol{-sin(\theta)} \\ \boldsymbol{sin(\theta)} & \boldsymbol{cos(\theta)} \end{bmatrix}$$

Note, 2D rotation matrix.

Interactive PCA via rotation, file is Interactive_PCA.ipynb.

- That direction, is described by a vector, our '1st principle component' $[\cos(\vartheta), -\sin(\vartheta)]$

The direction orthogonal to this is the second principal component:

- That direction, is described by a vector, our '2nd principle component' $[\sin(\vartheta), \cos(\vartheta),]$

© 2019 Michael Pyrcz

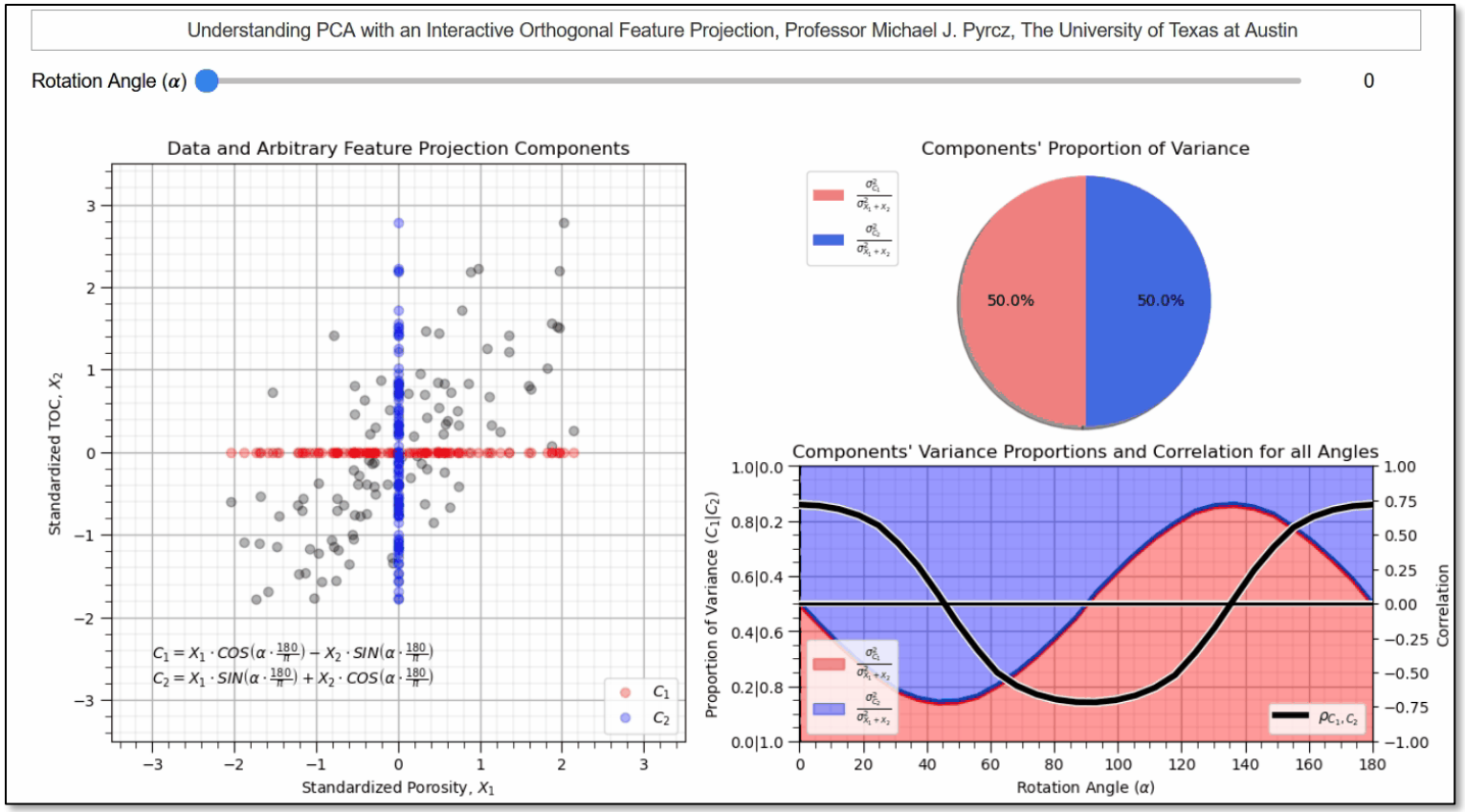
Principal Component Analysis

Orthogonal Transformation → Rotation

- Experiential Learning – let's take a dataset and rotate it.

Can we maximize the variance seen / explained in a single direction?

Yes, and independence also!



Interactive PCA via rotation,
file is Interactive_PCA.ipynb.

© 2019 Michael Pyrcz

Eigenvectors and Eigenvalues for PCA

Eigen interpretation, we solve for the eigenvectors to calculate the principal component loadings and the variance explained by each principle component.

- Eigenvalues / Eigenvectors
  - The **eigenvalues of the data covariance matrix** are the variance explained for each component.
  - The **eigenvectors of the data covariance matrix** are the principal components loadings (linear combinations of the original data)

- Eigenvectors for PCA
  - the main directions over the features that the data is spread along
  - directions of important patterns in the data

- Eigenvalues for PCA
  - how much the data varies along those directions
  - how important is each pattern

© 2019 Michael Pyrcz

Eigen Vectors and Values for PCA

Eigen interpretation, we solve for the eigenvectors to calculate the principal component loadings.

- Components are ordered
  - First component describes the larges possible variance / accounts for as much variability as possible
  - Next component describes the largest possible remaining variance
  - Up to the maximum number of principal components

- We will only mention some eigenvalues and eigenvectors concepts.
  - For an excellent introduction see: https://www.youtube.com/watch?v=PFDu9oVAE-g

- Let's show all the steps of PCA!

© 2019 Michael Pyrcz

Principal Component Analysis

Calculation of Principal Components

1. Standardize the Features

$$X^s = \frac{X - \bar{X}}{\sigma_X}$$

$$X_1, \ldots, X_m \rightarrow X_1^s, \ldots, X_m^s$$

- otherwise, the features with larger variance will dominate the solution.

© 2019 Michael Pyrcz

Principal Component Analysis

Calculation of Principal Components

2. Calculate the standardized feature covariance matrix

$$C_{X_{m_1},X_{m_2}} = \frac{\sum_{i=1}^{n}(x_{m_1} - \bar{x}_{m_1})(x_{m_2} - \bar{x}_{m_2})}{(n-1)}$$

$$C = \begin{bmatrix} C(X_1,X_1) & \dots & C(X_1,X_m) \\ \vdots & \ddots & \vdots \\ C(X_m,X_1) & \dots & C(X_m,X_m) \end{bmatrix}$$

Recall:

$$\rho_{X_{m_1},X_{m_2}} = \frac{C_{X_{m_1},X_{m_2}}}{\sigma_{X_{m_1}} \cdot \sigma_{X_{m_2}}}$$

- given the features are standardized the matrix is a correlation matrix

$$C = \begin{bmatrix} \rho(X_1,X_1) & \dots & \rho(X_1,X_m) \\ \vdots & \ddots & \vdots \\ \rho(X_m,X_1) & \dots & \rho(X_m,X_m) \end{bmatrix}$$

if $\sigma_j^2 = 1 \; \forall \; j = 1, \dots, m$

© 2019 Michael Pyrcz

Principal Component Analysis

Calculation of Principal Components

3. Calculate the eigenvalues and eigenvectors of covariance matrix, C

$$Cv = \lambda v$$

given C is a square matrix $(m \times m)$, v $(m \times 1)$ is a vector and λ is a scaler (1).

we can reorder to,

$$(C - \lambda \cdot I) \cdot v = 0$$

The vectors, v, for which the matrix multiplication of C is equivalent to a rescaling by a scalar, λ.

set the determinant equal to 0,

$$\det(C - \lambda \cdot I) = 0$$

Covariance matrices are symmetric; therefore have real eigenvalues.

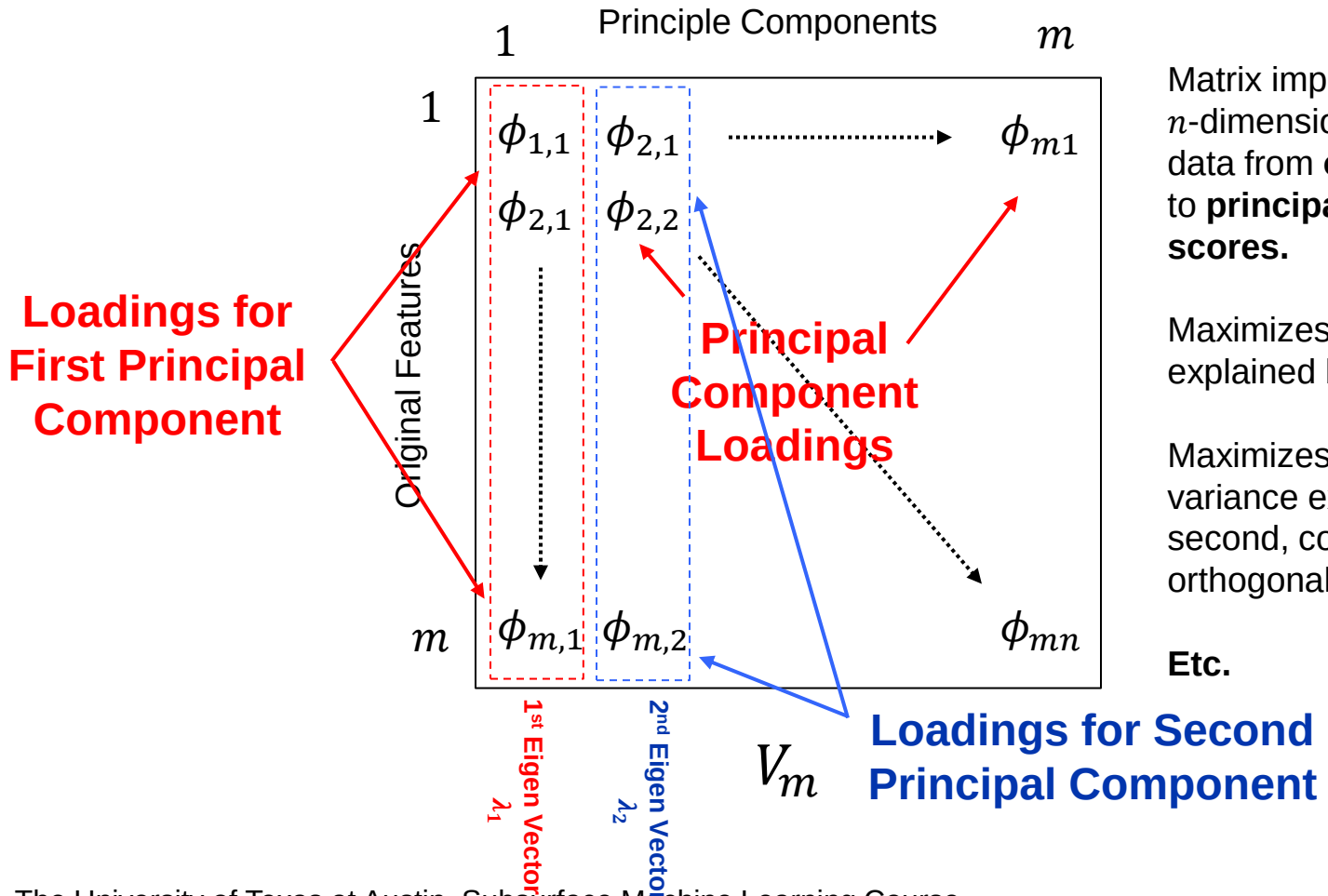
Find the possible eigenvalues, λ_α, and solve for eigenvectors $\boldsymbol{v_\alpha, \alpha = 1, \ldots, m}$

© 2019 Michael Pyrcz

Principal Component Analysis

Calculation of Principal Components

3. The resulting $min(m, n-1)$ eigenvectors in a matrix, V_m.



Matrix imposes a n-dimensional rotation of the data from **original features** to **principal components scores.**

Maximizes variance explained by first component.

Maximizes remaining variance explained by second, constrained by orthogonality to the first.

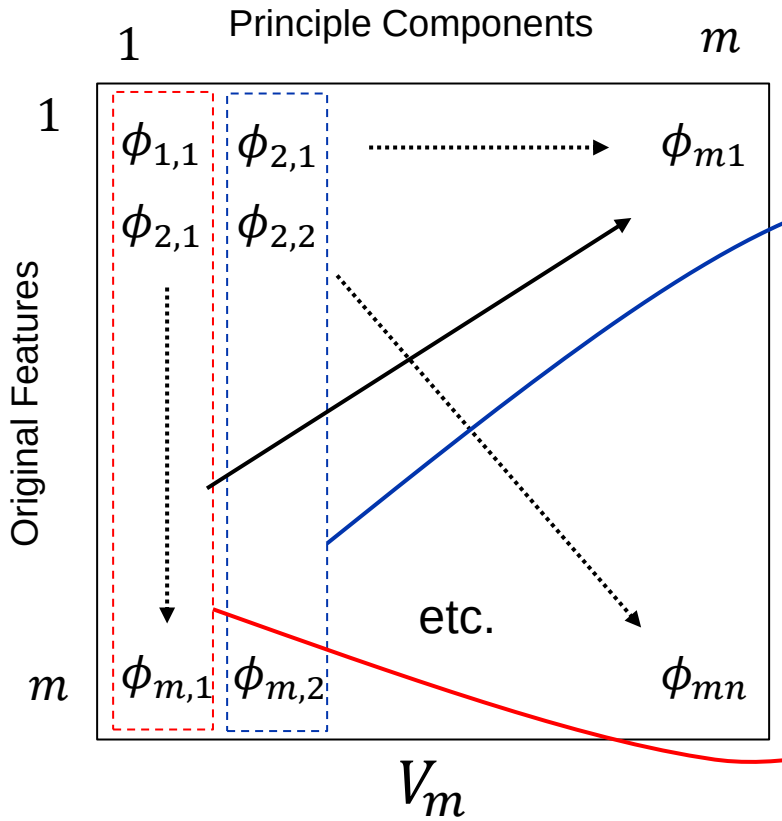
Etc.

© 2019 Michael Pyrcz

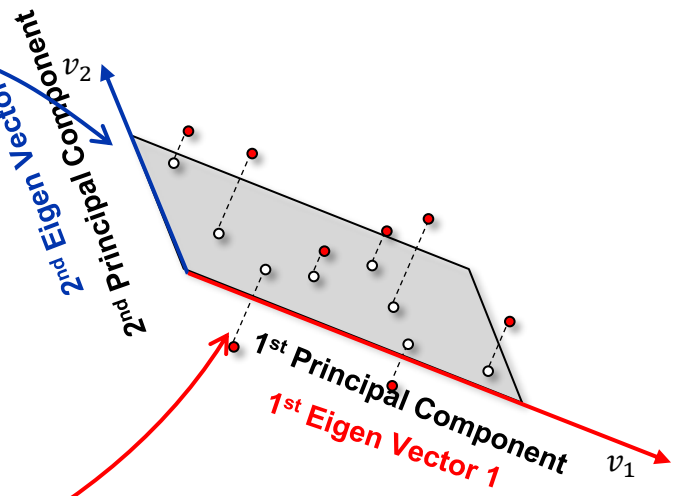
Principal Component Analysis

Calculation of Principal Components

3. The resulting $min(m, n-1)$ eigenvectors in a matrix, V_m.



First and second principal component are two orthogonal vectors, describe a plane.



First principal component is a vector, describes a line.

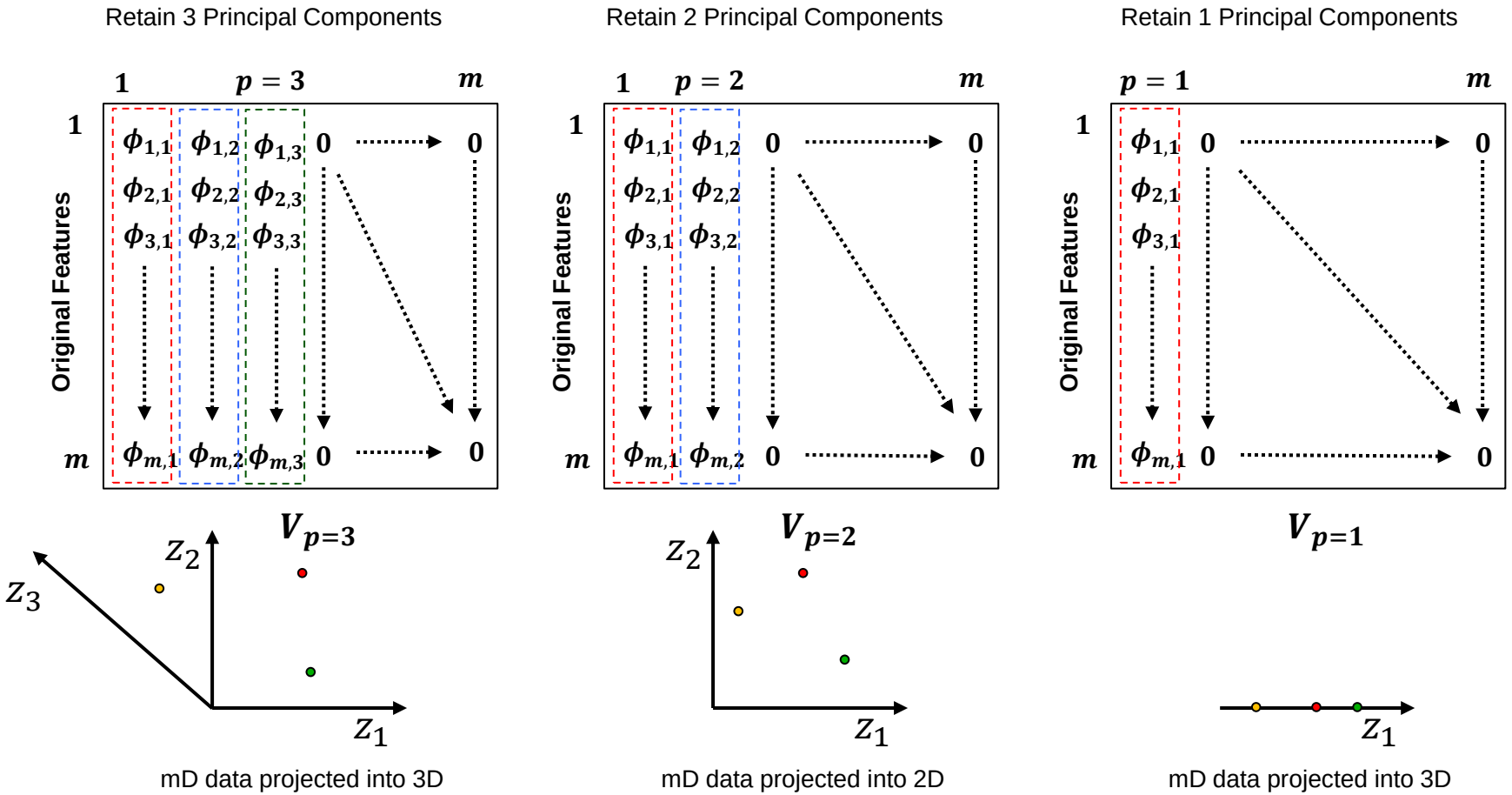
Our principal components are mutually orthogonal, because the input covariance matrix is symmetric. So they form an orthonormal basis, in which each component is linearly uncorrelated.

© 2019 Michael Pyrcz

Principal Component Analysis

Calculation of Principal Components

4. Dimensionality reduction by zero'ing the $m - p$ last eigenvectors.



© 2019 Michael Pyrcz

Principal Component Analysis

Assumptions

- large enough sample set for reliable covariance / correlation calculation

$$n = 5 \times m \text{ and } n > 150$$

- there is correlation between the features

$$\rho_{i,j|i \neq j} \neq 0 \qquad \text{in some cases}$$

- outliers are addressed

$$C_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)}$$

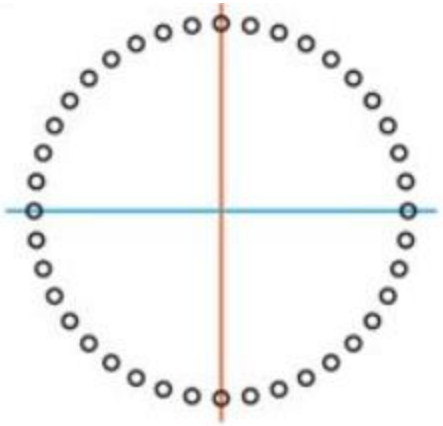
recall the covariance is sensitive to bivariate outliers

© 2019 Michael Pyrcz

Principal Component Analysis

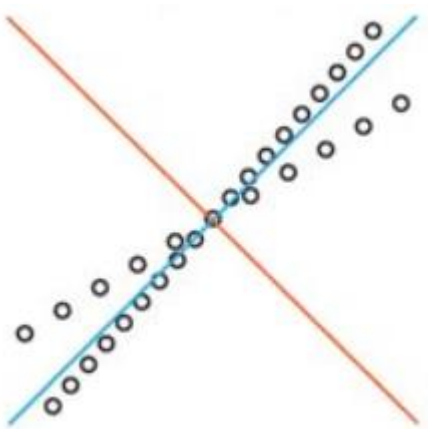
Assumptions, will not perform well in the presence of,

Nonlinear patterns



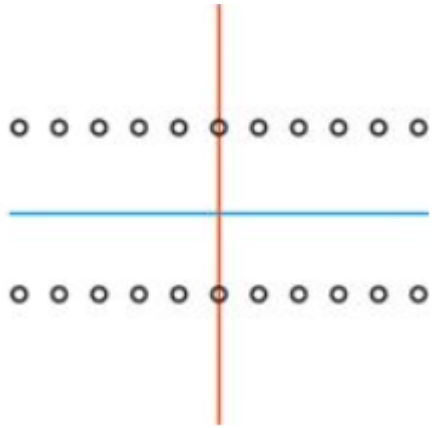
Data structures
Must be linear

Nonorthongonal Patterns



Principal components
must be orthogonal

Obscure Patterns



Goal to is maximize
variance on each projection,
not cluster identification

Image from https://www.nature.com/articles/nmeth.4346.
© 2019 Michael Pyrcz

Principal Component Analysis

Principal Component

The first **principal component** of a set of features, $X_1, \ldots, X_m$, is the centered linear combination of the features:

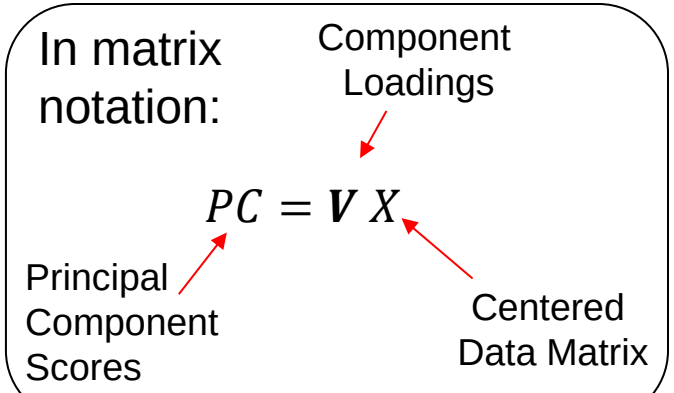
$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{m1}X_m$$

with the largest variance.

- Orthogonality constraint requires: $\sum_{i=1}^{m} \phi_{i1}^2 = 1.0$

as a result, this transform is a rotation that preserves distances.

In matrix notation:



- The values $\mathbf{V} = [\phi_{1k}, \phi_{2k} \ldots \phi_{mk}] \; \forall \; j = 1, \ldots, K$ components, are known as factor or component loadings

- We can calculate the first **principal component scores** (values projected onto this principal coordinate) as:

$$z_{i1} = \phi_{1,1}x_{i,1} + \phi_{2,1}x_{i,2} + \cdots + \phi_{m,1}x_{i,m} \qquad \text{for } i = 1, \ldots n, \text{ data}$$

© 2019 Michael Pyrcz

Principal Component Analysis

Principal Component Loadings

Observes the groups of features that strongly influence each principal coordinate.

$$z_{i,1} = \phi_{1,1}x_{i,1} + \phi_{2,1}X_{i,2} + \cdots + \phi_{m,1}x_{i,m}$$

1st Principal Component Score for i^{th} data

the loadings for PC1 are $\phi_{1,1}, \phi_{2,1} \ldots \phi_{m,1}$ is the 1st principal component.

- Check if specific features strongly influence specific principal components for the remainder. Compare them to each other.

$$z_{i,2} = \phi_{1,2}x_{i,1} + \phi_{2,2}x_{i,2} + \cdots + \phi_{m,2}x_{i,m}$$

$$\downarrow$$

$$z_{i,m} = \phi_{1,p}x_{i,1} + \phi_{2,p}x_{i,2} + \cdots + \phi_{m,p}x_{i,m}$$

2nd Principal Component Score for i^{th} data

$\downarrow$

mth Principal Component Score for i^{th} data

May be able to describe each principal component! E.g., for a reservoir:

- PC1 is the mainly the heterogeneity component
- PC2 is the mainly the completion component

© 2019 Michael Pyrcz

Principal Component Analysis

How do we do Dimensional Reduction?

We have converted our data set, $X_{n\times m}$, to principal component scores, $PC_{n\times m}$

- If we retain all the m components then have not achieved any dimensional reduction. **We just have orthogonal, linear combination of our original features!**
- We gain dimensional reduction by retaining only p principal components or in other words by dropping the last $m - p$ components as they describe very little of the variance.

$$PC = \boldsymbol{V}\, X$$

$$\hat{X} = \boldsymbol{V^{-1}} \cdot \boldsymbol{PC}$$

Back transforming from principal components scores to original values.

- But since the loadings, $\boldsymbol{V}$, are orthogonal then $\boldsymbol{V^{-1}} = \boldsymbol{V^{T}}$

$$\hat{X} = \boldsymbol{V^{T}} \cdot \boldsymbol{PC}$$

$$\hat{x}_{i,j}^{p} \approx \sum_{k=1}^{p} \phi_{j,k}\, z_{i,j}$$

component loadings

principal component scores

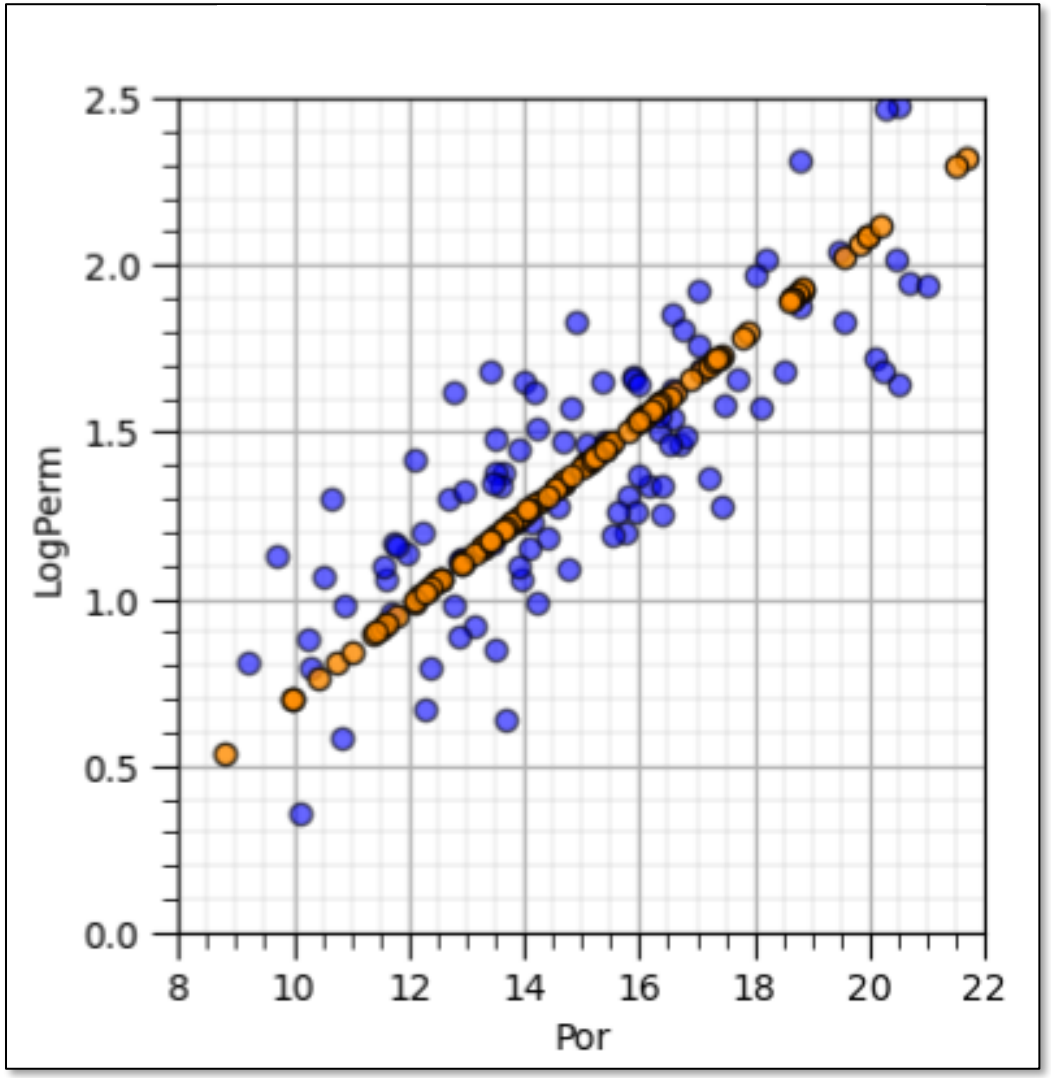
where $i = 1, \ldots, n$ data and j $= 1, \ldots, m$ variables / features, and p principal components (of $j = 1, \ldots, p$) are retained.

© 2019 Michael Pyrcz

Principal Component Analysis

Graphical Representation

- Line is the 1st principal component
- Projection of data onto the line are the 1st principal component scores
- Given the problem is 2D the 2nd principal component is determined from the first (must be orthogonal)
- If we approximated this dataset with just the 1st principal component for dimensional reduction.
- The first principal component maximizes the variance of the projected data.



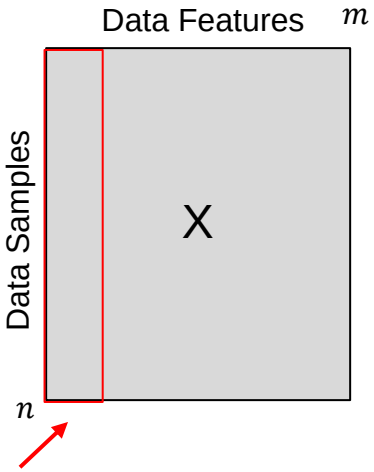
Original data (blue) and projected to 1D data (orange), from MachineLearning PCA chapter of e-book.

© 2019 Michael Pyrcz

Principal Components Analysis Summary

Forward Transformation

Normalized Data

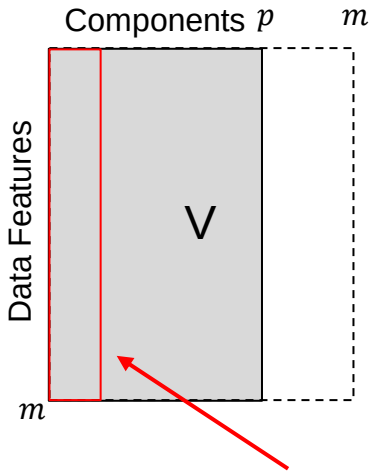


Normalized by-Feature

$\bar{x}_{x_j} = 0.0, \sigma^2_{x_j} = 1.0$

×

Component Loadings

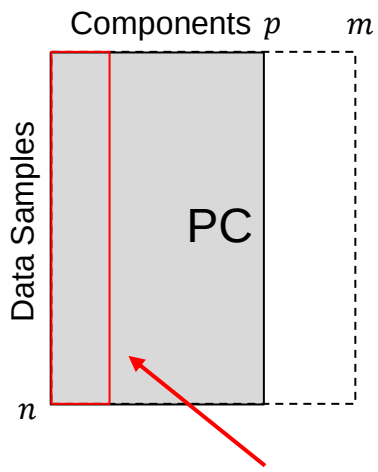


Orthogonal / Rotation

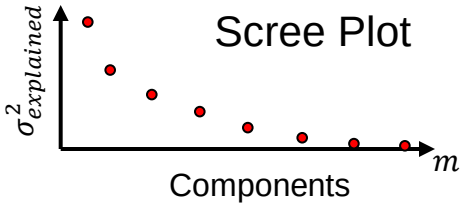
$\sum_{j=1}^{m} \phi^2_{,j} = 1.0$

=

Principal Component Scores



Data Projection / Coordinate in Orientation that Explains the Most Variance

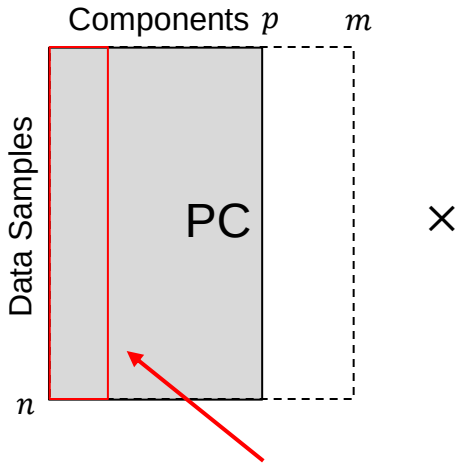


© 2019 Michael Pyrcz

Principal Components Analysis Summary

Reverse Transformation

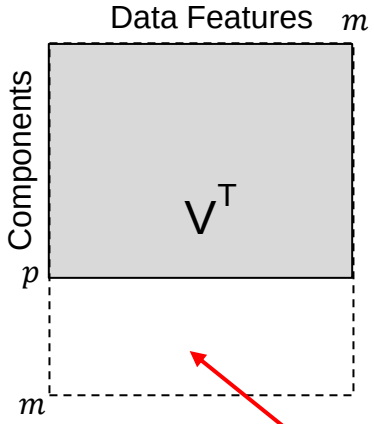
Principal Component Scores



Date Projection / Coordinate In Orientation that Explains the Most Variance

×

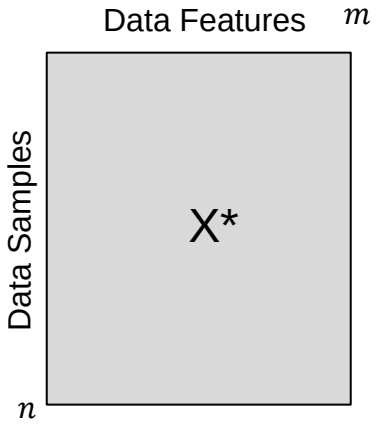
Component Loadings



Remove / Zero Unused Components $m - p$

=

Normalized Data Projection



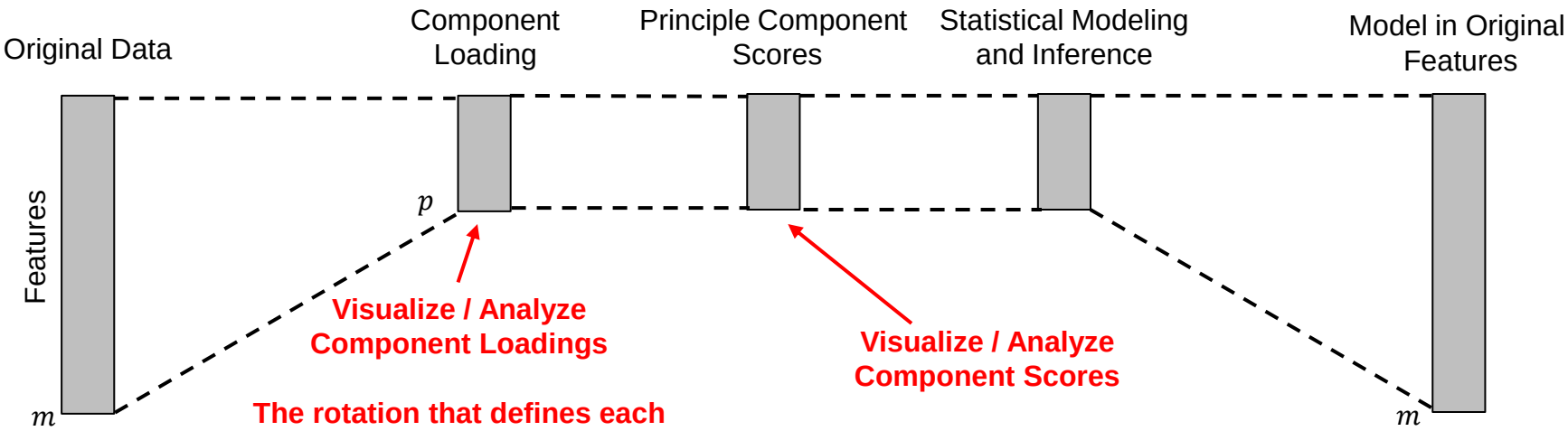
De-normalize to Restore Original Features

Restore Correct Variance and Mean
Affine Correction

© 2019 Michael Pyrcz

Principal Components Analysis Summary

Typical PCA Workflow



Visualize / Analyze / Model with Component Scores

Lower dimensional space, lower risk of overfit and easier to model.

Visualize / Analyze Component Loadings

The rotation that defines each independent component that describe components of the variance

Variable groupings

Important features (images) that describe the observed variability between training

Visualize / Analyze Component Scores

Set of independent Features that describe components of the variance

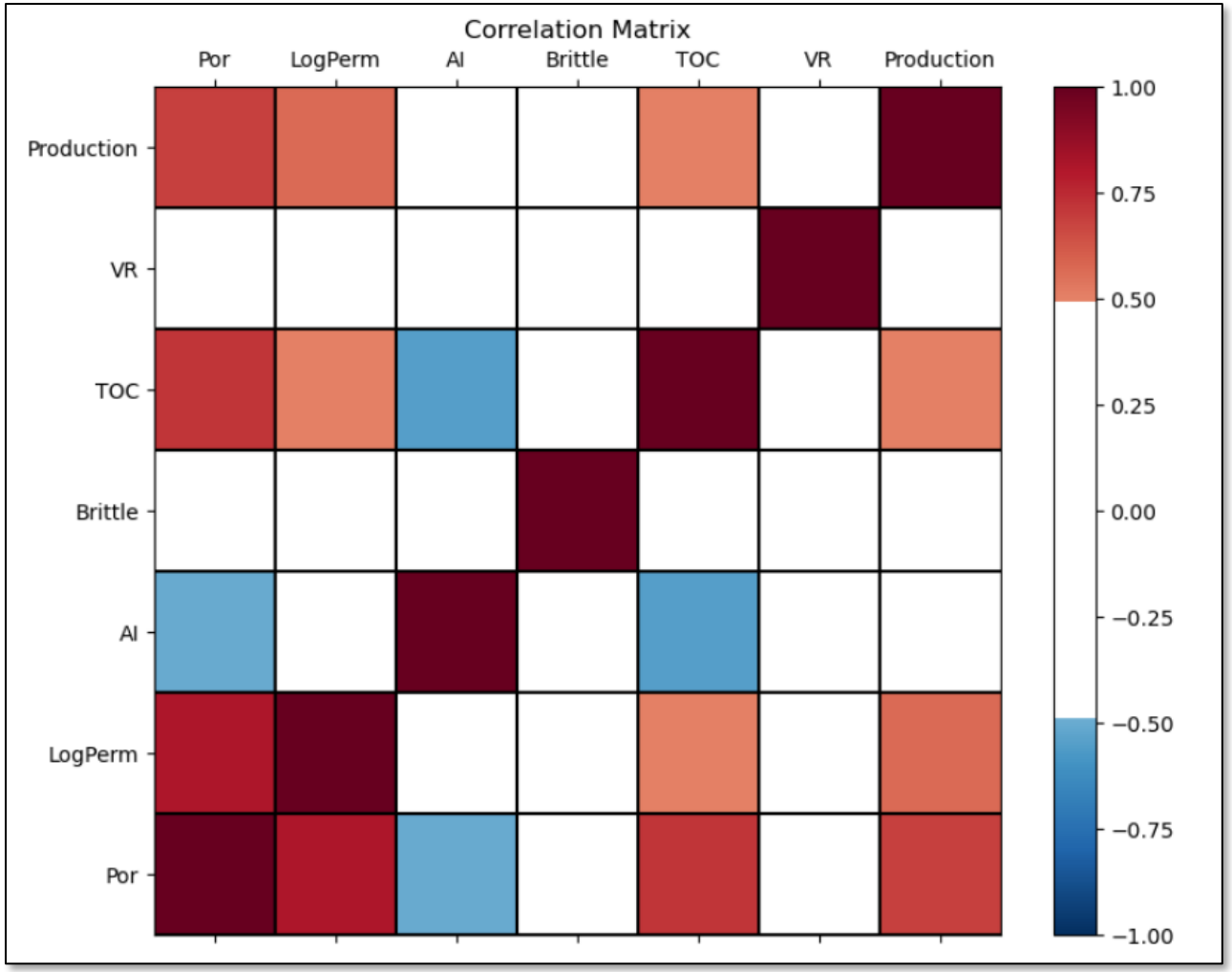
© 2019 Michael Pyrcz

Principal Components Analysis Example

Multiple variate dataset with,

- 1,000 wells
- Porosity (%)
- Log transform of permeability to linearize relationship with porosity
- Acoustic impedance, product of density and velocity
- Brittleness index, percentage based on rock mechanics
- Total organic carbon
- Vitrinite reflectance, reflected light
- Production, initial production average monthly production over first 3 months

Some features are highly correlated, opportunity for dimensionality reduction.



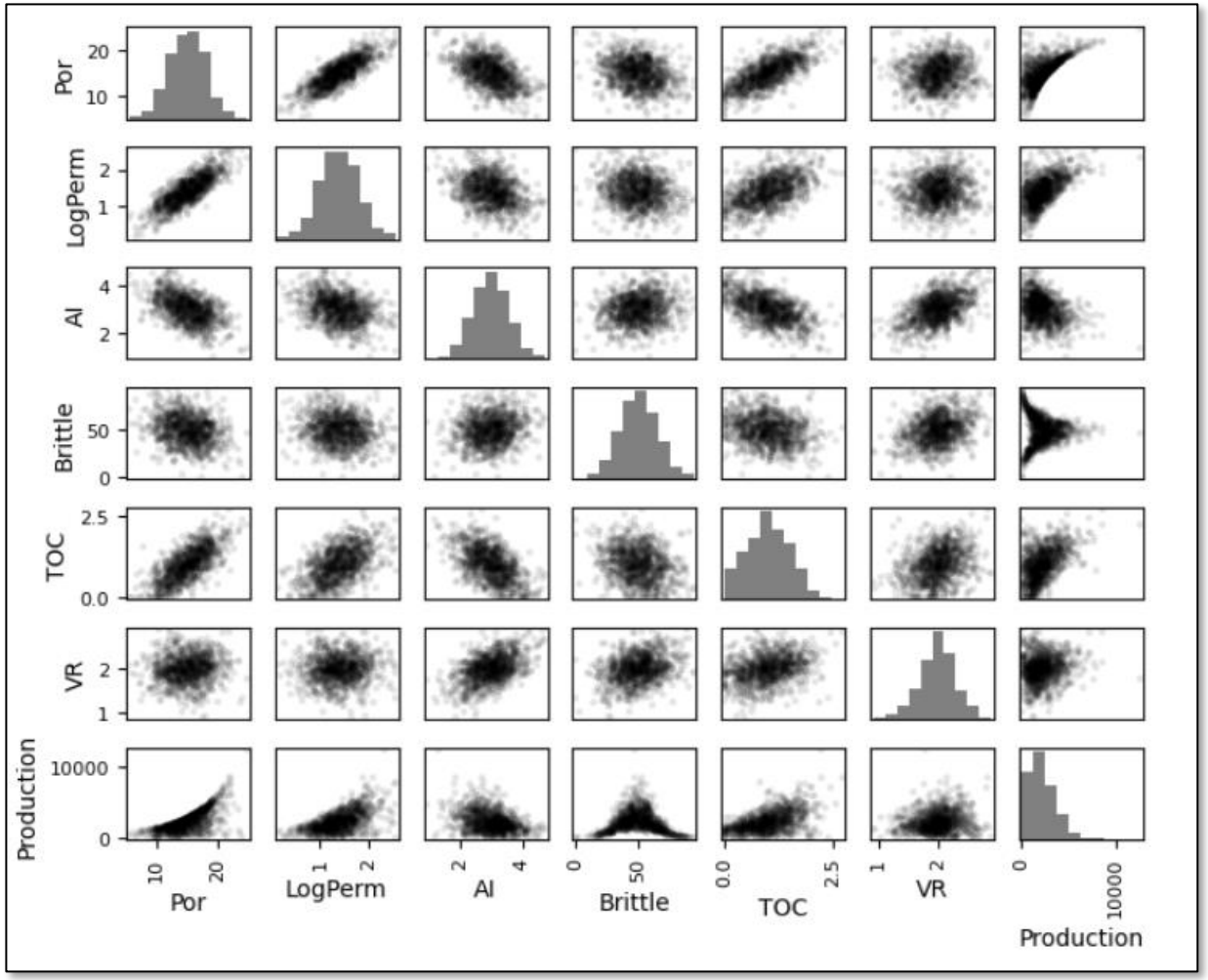
Correlation matrix of our dataset, from MachineLearning PCA chapter of e-book.

© 2019 Michael Pyrcz

Principal Components Analysis Example

Check matrix scatter plot,

- some nonlinearity, constraints but no obvious outliers



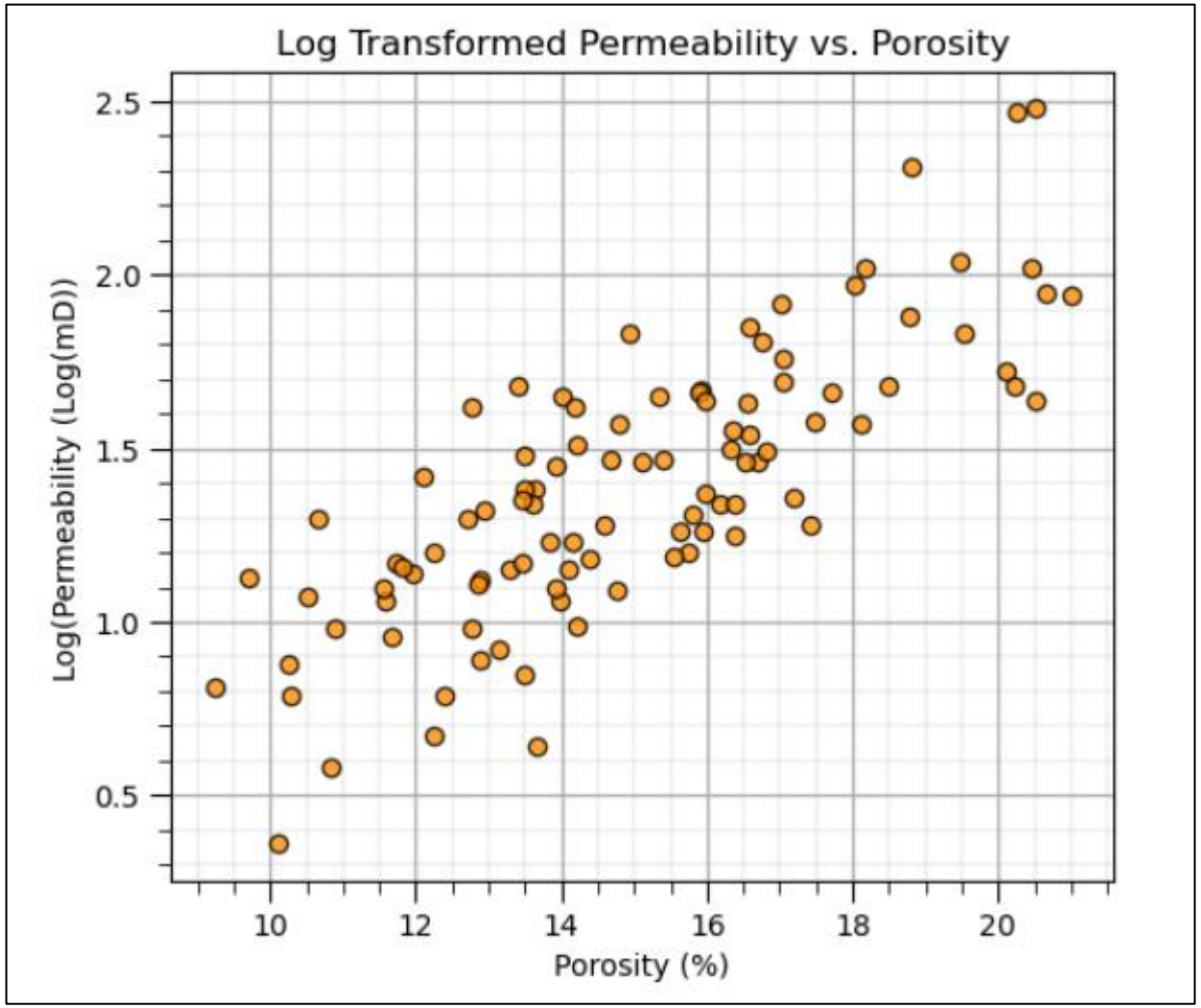
Matrix scatter plot of our dataset, from MachineLearning PCA chapter of e-book.

© 2019 Michael Pyrcz

Principal Components Analysis Example

Let's start simple:

- Only consider porosity and log perm.
- We reduce our problem to bivariate since it is very easy to visualize.

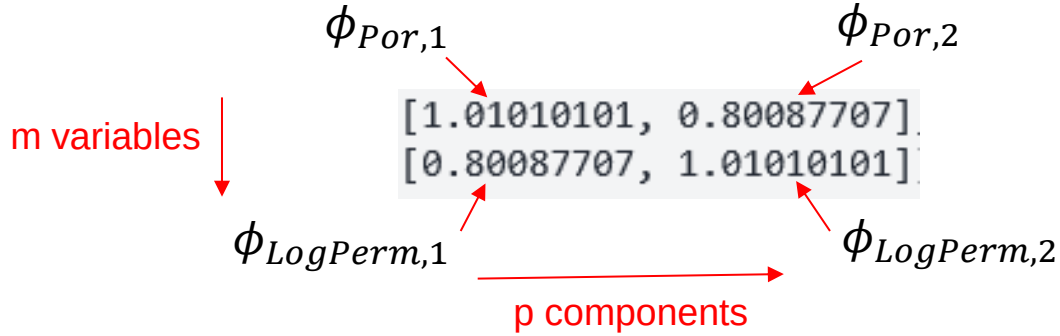


Porosity and log permeability scatter plot, from MachineLearning PCA chapter of e-book.

© 2019 Michael Pyrcz

Principal Components Analysis Example

- Here's the resulting factor loadings:



- We can calculate the principal component scores as:

$$Z_{i,1} = \phi_{Por,1} \cdot (Por_i - \overline{Por}) + \phi_{LogPerm,1} \cdot (LogPerm_i - \overline{LogPerm})$$

$$Z_{i,2} = \phi_{Por,2} \cdot (Por_i - \overline{Por}) + \phi_{LogPerm,2} \cdot (LogPerm_i - \overline{LogPerm})$$

For $i = 1, \dots, n$ number of data, $j = 1, \dots, p$ components

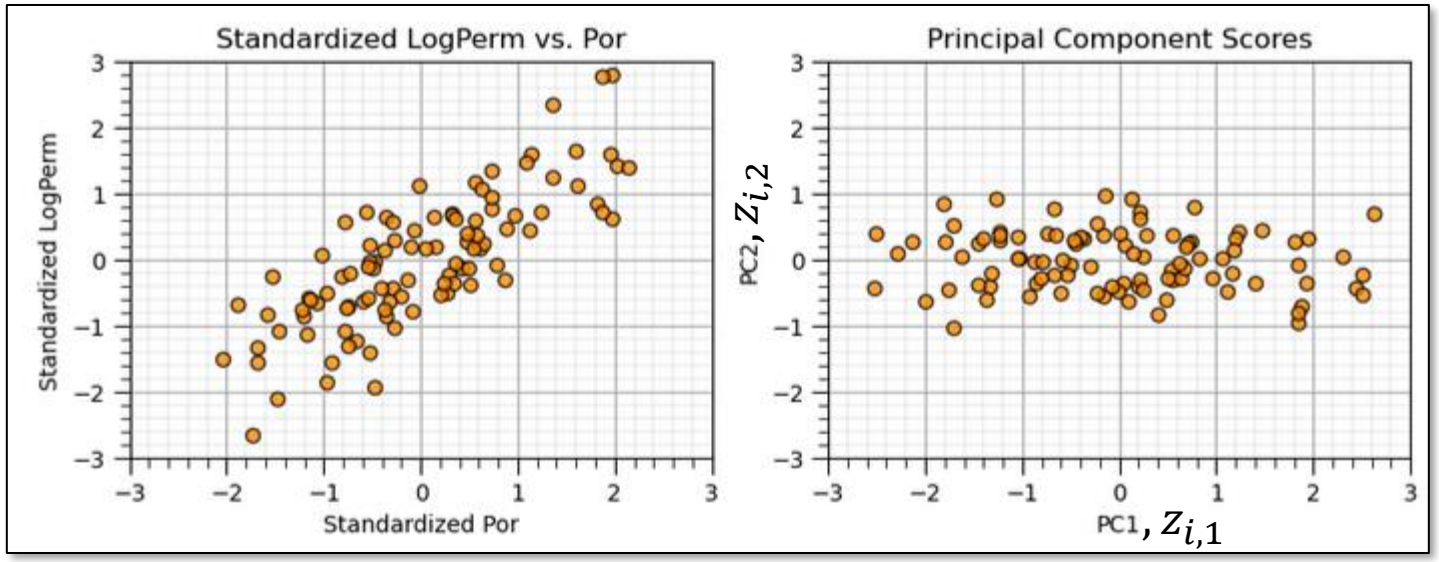
Data **Principle Components**

We have mapped original $Por_i, LogPerm_i \longrightarrow z_{i,1}, z_{i,2}$ where $z_{i,1}$ describes as much variance as possible.

© 2019 Michael Pyrcz

Principal Components Analysis Example

Here we plot the original data and compare it to the plot of the principal component scores, $z_{i,1}$ and $z_{i,1}$ for $i = 1, \ldots, n$ data.



Porosity and log permeability scatter plot (left) and principle component scores, from MachineLearning PCA chapter of e-book.

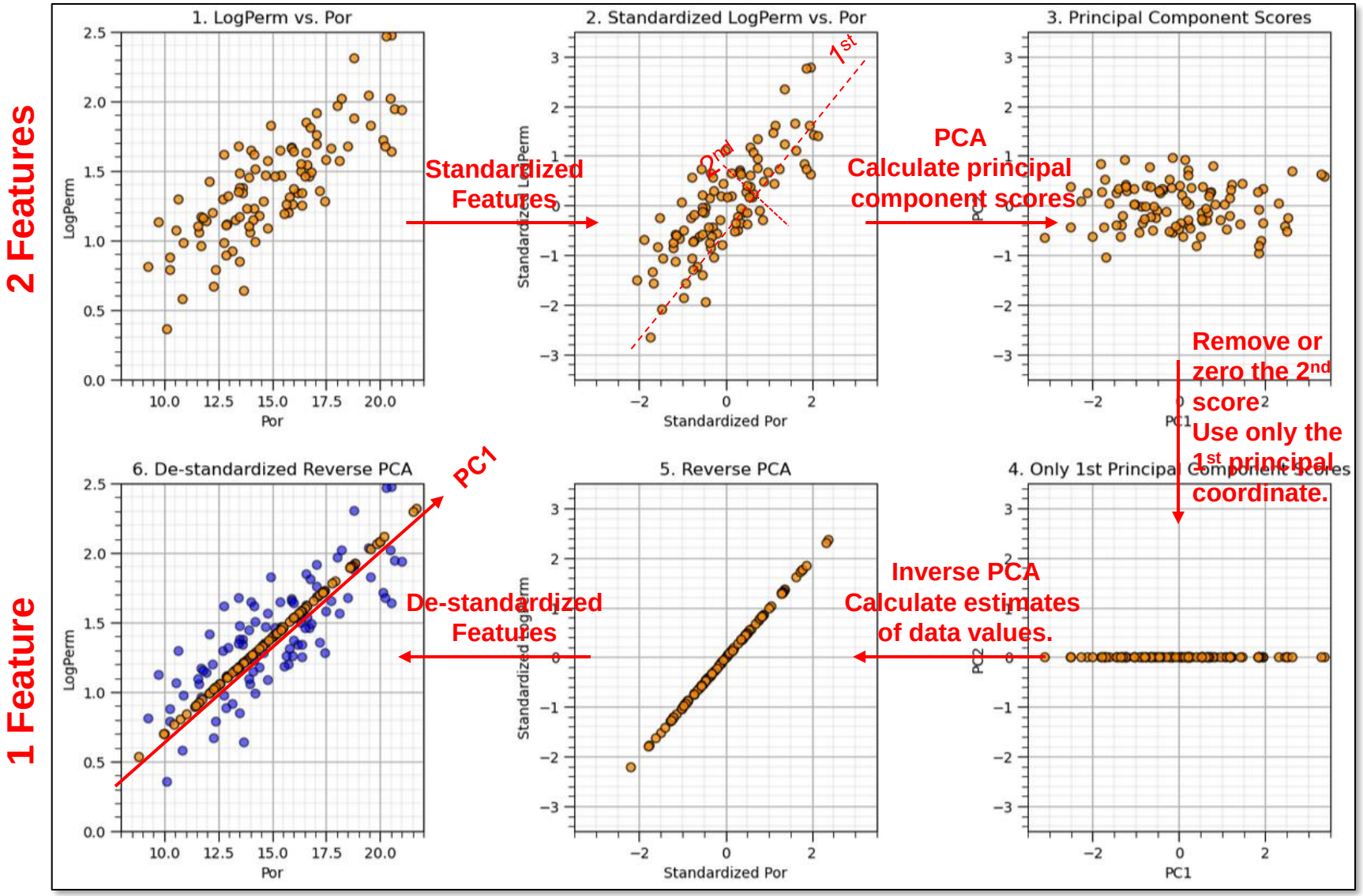
- We could just retain the first principal component score.
- How much information would we loose? How much of the variance would be explained?
- Let's try that.

© 2019 Michael Pyrcz

Principal Components Analysis Example

Now let's walk through dimensionality reduction from 2 features to 1 feature

- 89.6% variance explained by PC1



© 2019 Michael Pyrcz

Principal Components Analysis Example

Variance Described by Each Principal Component / Scree Plots

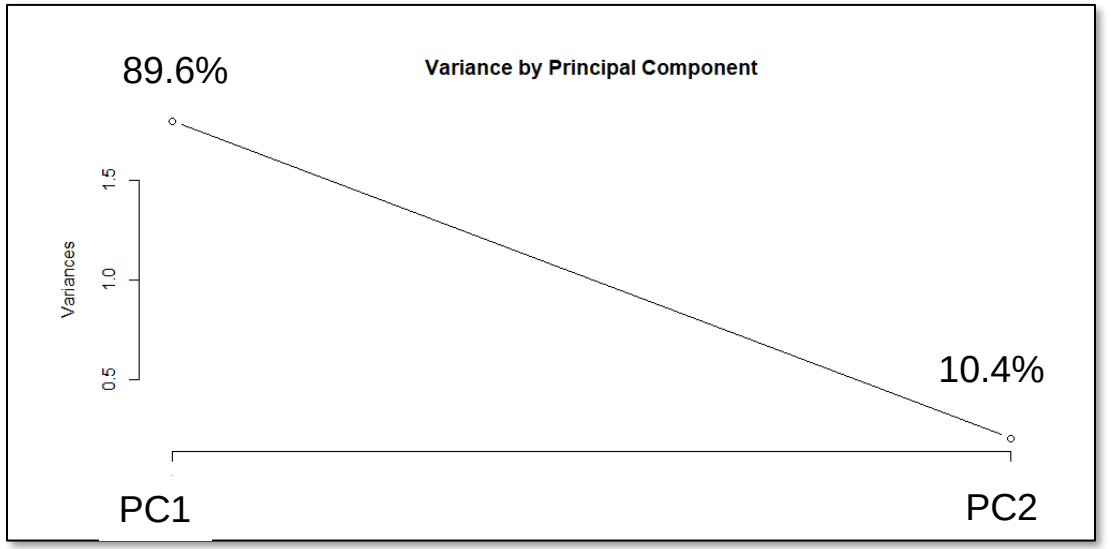
- So, how much variance did we capture with our components? We can calculate the proportion of variance explained by each principal component as:

$$PVE_k = \frac{1}{n}\sum_{i=1}^{n} z_{i,k}^2$$

- Should be monotonically decreasing for $k = 1, \ldots, K$.

- In our example:

Component	$\sigma^2_{explained}$
PC1	89.6%
PC2	10.4%



Bivariate dataset scree plot.

© 2019 Michael Pyrcz

Principal Components Analysis Example

Variance Described by Each Principal Component / Scree Plots

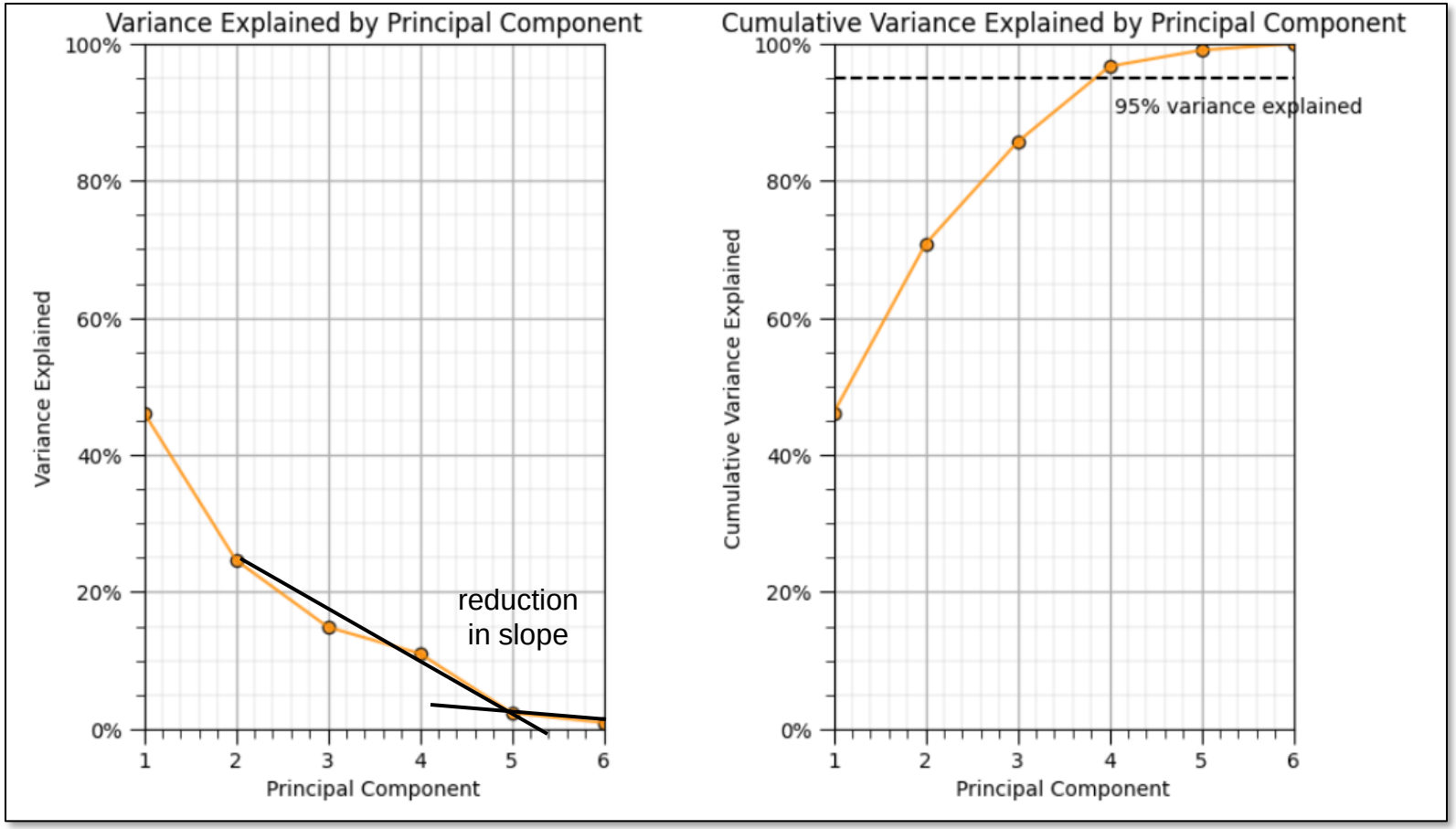
- Example based on all 6 features

Decision Criteria for Number of Principal Components to Retain

- variance explained target, e.g., 95% requires atleast 4 principal components

Diminishing Returns, Slope of Scree Plot

- after 5 principal components



© 2019 Michael Pyrcz

PGE 383 Subsurface Machine Learning

Lecture 8: Dimensionality Reduction

Lecture outline:

- **Principal Component Analysis Examples**

© 2019 Michael Pyrcz

Principal Components Analysis Workflows

Example PCA Workflows

Prediction

- Reduce dimensions, build a model with the principal component scores and then restore to estimates the data values.
  - PCA regression, regression on the most important principal components

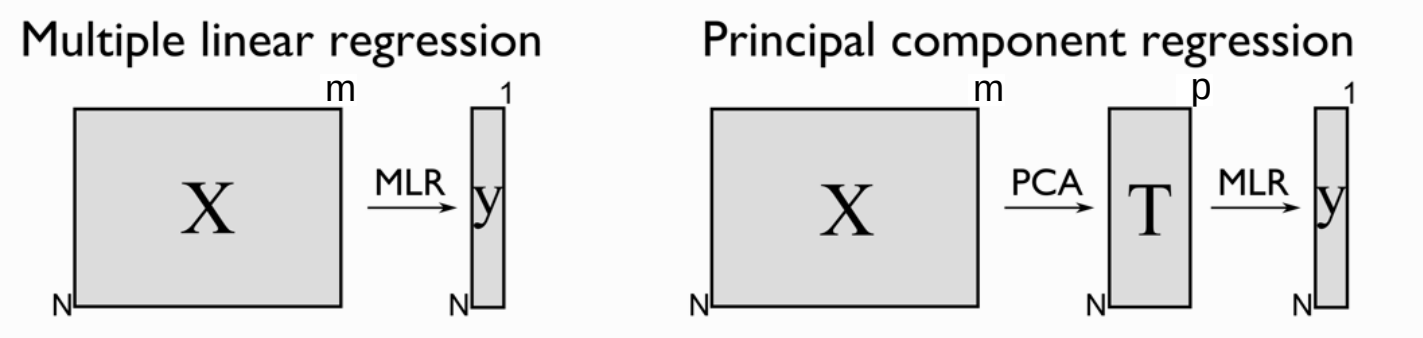


Image from: https://learnche.org/pid/latent-variable-modelling/principal-components-regression

Inference

Understand our variables and how variance is partitioned

- Check for and mitigate multi-collinearity and exclude principal components that have low variance

© 2019 Michael Pyrcz

Principal Components Analysis Image Example

PCA with images:

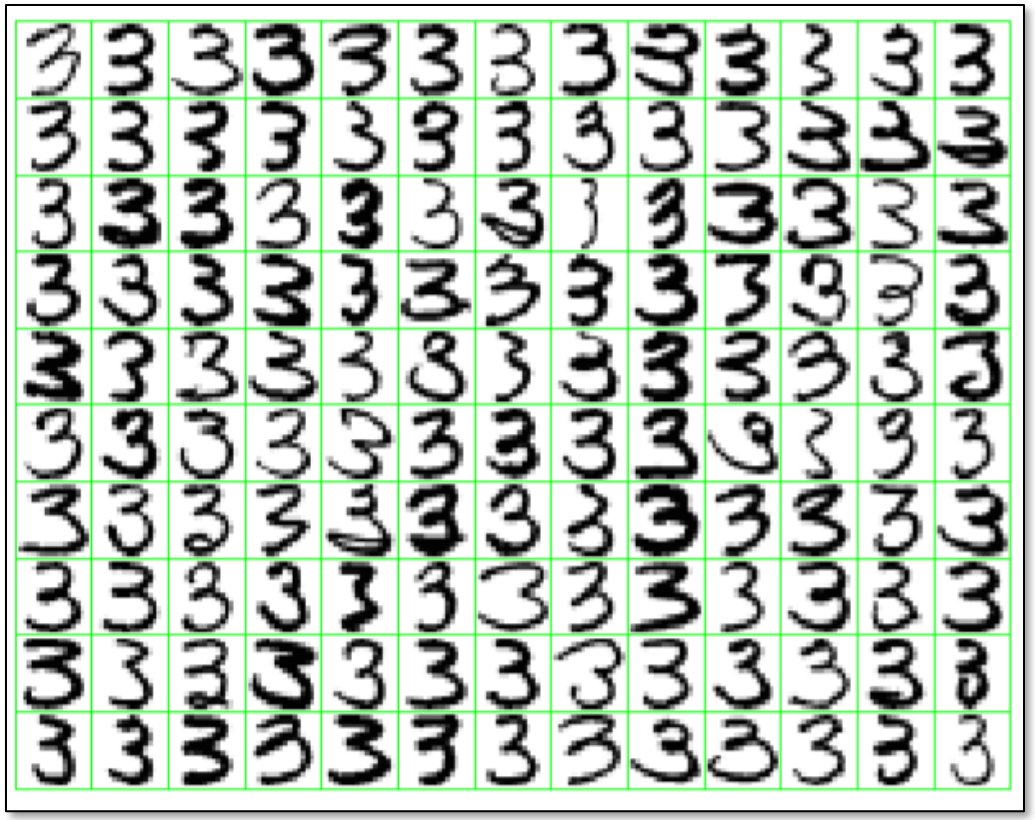
- 130 examples of hand writing
- 16 x 16 grey scale images
- $m = 256$ dimensional

Comments:

- Clearly the images have commonality
- We can describe their variability with fewer than 256 features!

Workflow:

- Calculate the covariance matrix of all pixels with each other
- Results in a 256 x 256 covariance matrix
- Center by removing average of each pixel, then calculate the Eigen values and vectors (Singular value decomposition)



Handwritten 3's, image and example from Hastie et al., 2009.

© 2019 Michael Pyrcz

Principal Components Analysis Image Example

Retain the first 2 principal components:

principal component score ($z_{i,2}$)

16x16 of loading PC #2 ($\phi_{2,k}$)

$$\hat{f}(\lambda) = \bar{x} + \lambda_1 v_1 + \lambda_2 v_2$$

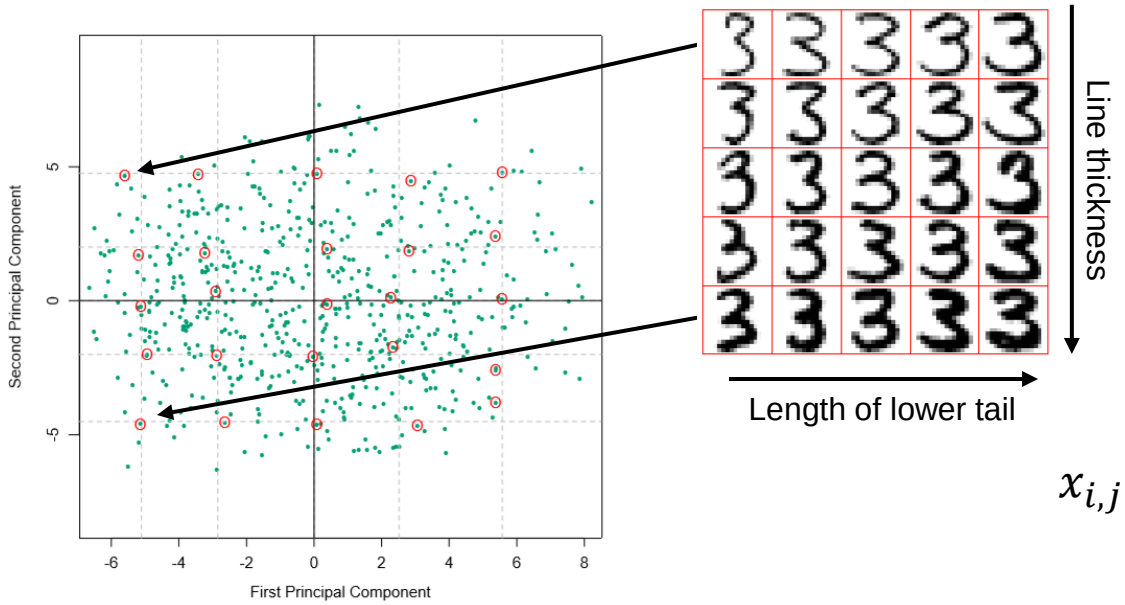
$$= \bar{x} + \lambda_1 \cdot \phi_1 + \lambda_2 \cdot \phi_2.$$

256x256 loadings

Comments:

We can now explore the variability of these handwritten 3's with the first two 2 principle components.

2 dials, instead of 256 to explore the space.



$$x_{i,j} \approx \sum_{k=1}^{p} z_{i,j}\phi_{j,k}$$

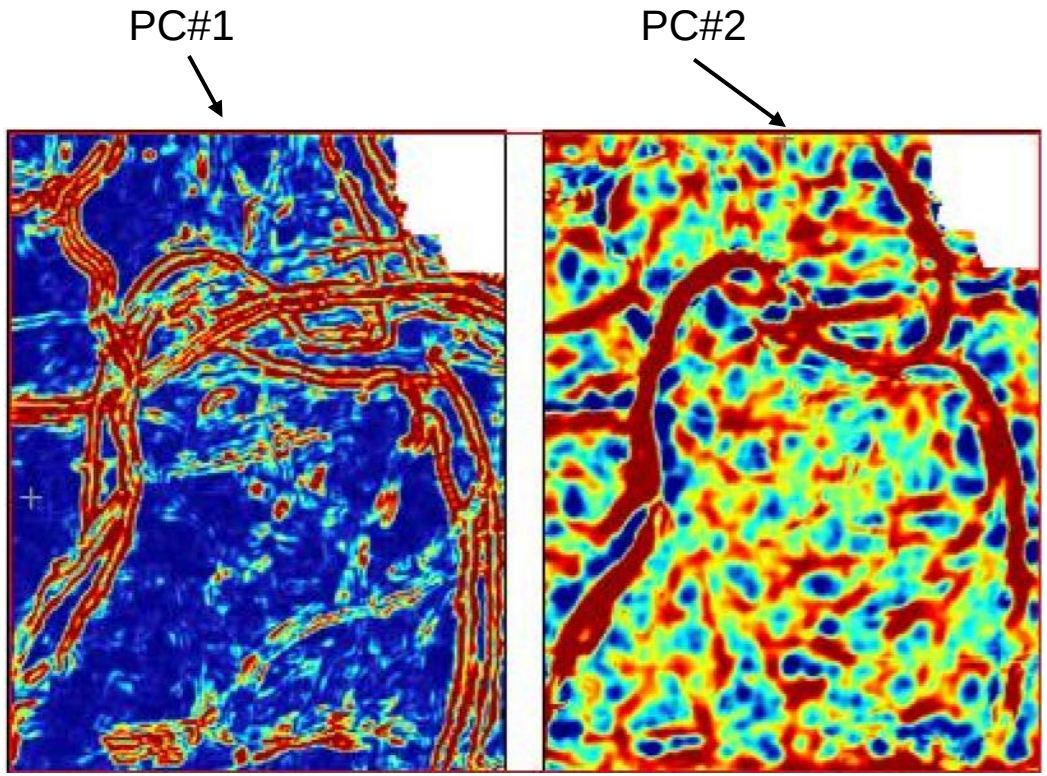
Figures and example from Hastie et al., 2009. © 2019 Michael Pyrcz

Principal Components Analysis Seismic Example

6 Seismic attributes and first 3 capture 97% of variability! Here's 2.

Each principal component describes different aspects of the multivariate seismic.

- Trailing principal components are dominated by noise
- Fit models with less probability of overfit



Principal components 1 (left) and 2 (right) for a 6 seismic attribute dataset, image and example from Chopra and Marfurt (2014).

© 2019 Michael Pyrcz

Principal Components in Reservoir Modeling

Examples of PCA in Subsurface Modeling

- Modeling multivariate relationship while avoiding over fitting, porosity from a set of seismic attributes.
- Image analysis on seismic information, separating multiple attributes into information and noise.
- Analysis of feature grouping, redundancy
- Reducing dimensionality to support simpler workflows, e.g. bivariate, cosimulation methods

© 2019 Michael Pyrcz

PGE 383 Subsurface Machine Learning

Lecture 8: Dimensionality Reduction

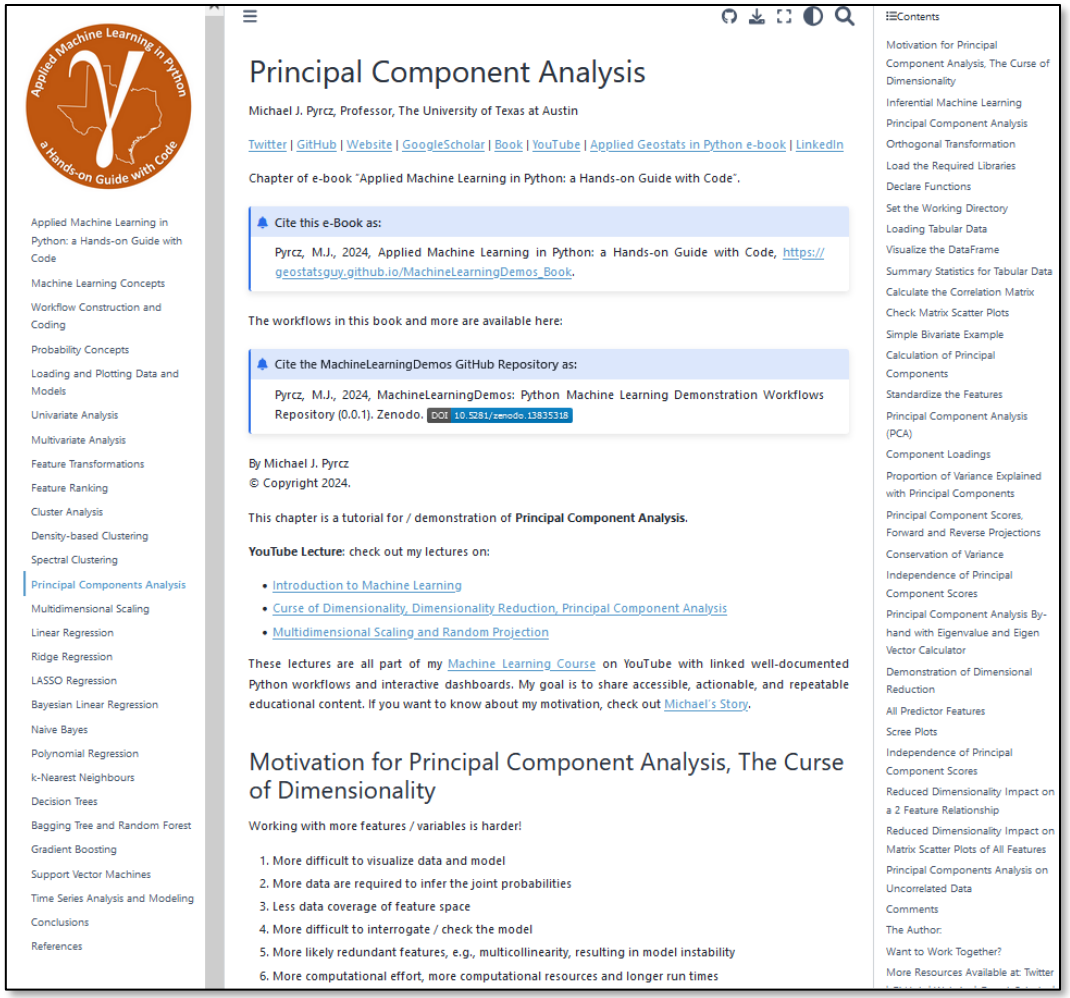
Lecture outline:

- **Principal Component Analysis Hands-on**

© 2019 Michael Pyrcz

PCA Demonstration in Python

Demonstration of PCA with a well-documented workflow.



Applied Machine Learning in Python: a Hands-on Guide with Code

Machine Learning Concepts
Workflow Construction and Coding
Probability Concepts
Loading and Plotting Data and Models
Univariate Analysis
Multivariate Analysis
Feature Transformations
Feature Ranking
Cluster Analysis
Density-based Clustering
Spectral Clustering
Principal Components Analysis
Multidimensional Scaling
Linear Regression
Ridge Regression
LASSO Regression
Bayesian Linear Regression
Naive Bayes
Polynomial Regression
k-Nearest Neighbours
Decision Trees
Bagging Tree and Random Forest
Gradient Boosting
Support Vector Machines
Time Series Analysis and Modeling
Conclusions
References

Principal Component Analysis

Michael J. Pyrcz, Professor, The University of Texas at Austin

Twitter | GitHub | Website | GoogleScholar | Book | YouTube | Applied Geostats in Python e-book | LinkedIn

Chapter of e-book "Applied Machine Learning in Python: a Hands-on Guide with Code".

Cite this e-Book as:

Pyrcz, M.J., 2024, Applied Machine Learning in Python: a Hands-on Guide with Code, https://geostatsguy.github.io/MachineLearningDemos_Book.

The workflows in this book and more are available here:

Cite the MachineLearningDemos GitHub Repository as:

Pyrcz, M.J., 2024, MachineLearningDemos: Python Machine Learning Demonstration Workflows Repository (0.0.1). Zenodo. DOI 10.5281/zenodo.13835318

By Michael J. Pyrcz
© Copyright 2024.

This chapter is a tutorial for / demonstration of **Principal Component Analysis**.

YouTube Lecture: check out my lectures on:

- Introduction to Machine Learning
- Curse of Dimensionality, Dimensionality Reduction, Principal Component Analysis
- Multidimensional Scaling and Random Projection

These lectures are all part of my Machine Learning Course on YouTube with linked well-documented Python workflows and interactive dashboards. My goal is to share accessible, actionable, and repeatable educational content. If you want to know about my motivation, check out Michael's Story.

Motivation for Principal Component Analysis, The Curse of Dimensionality

Working with more features / variables is harder!

1. More difficult to visualize data and model
2. More data are required to infer the joint probabilities
3. Less data coverage of feature space
4. More difficult to interrogate / check the model
5. More likely redundant features, e.g., multicollinearity, resulting in model instability
6. More computational effort, more computational resources and longer run times

Contents

Motivation for Principal Component Analysis, The Curse of Dimensionality
Inferential Machine Learning
Principal Component Analysis
Orthogonal Transformation
Load the Required Libraries
Declare Functions
Set the Working Directory
Loading Tabular Data
Visualize the DataFrame
Summary Statistics for Tabular Data
Calculate the Correlation Matrix
Check Matrix Scatter Plots
Simple Bivariate Example
Calculation of Principal Components
Standardize the Features
Principal Component Analysis (PCA)
Component Loadings
Proportion of Variance Explained with Principal Components
Principal Component Scores, Forward and Reverse Projections
Conservation of Variance
Independence of Principal Component Scores
Principal Component Analysis By-hand with Eigenvalue and Eigen Vector Calculator
Demonstration of Dimensional Reduction
All Predictor Features
Scree Plots
Independence of Principal Component Scores
Reduced Dimensionality Impact on a 2 Feature Relationship
Reduced Dimensionality Impact on Matrix Scatter Plots of All Features
Principal Components Analysis on Uncorrelated Data
Comments
The Author:
Want to Work Together?
More Resources Available at: Twitter

MachineLearning_PCA chapter of e-book.

© 2019 Michael Pyrcz

PGE 383 Subsurface Machine Learning

Lecture 8: Dimensionality Reduction

Lecture outline:

- **Curse of Dimensionality**
- **Dimensionality Reduction**
- **Principal Component Analysis**
- **Principal Component Analysis Examples**
- **Principal Component Analysis Hands-on**

© 2019 Michael Pyrcz