GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
LECTURE 12: DIMENSIONALITY REDUCTION

Paul Burkard

12/09/2015

LAST TIME:

- CLUSTER ANALYSIS
- K-MEANS CLUSTERING
- OTHER CLUSTERING ALGORITHMS
- CLUSTERING EVALUATION

QUESTIONS?

I. DIMENSIONALITY REDUCTION
II. FEATURE SELECTION
III. FEATURE EXTRACTION - PCA AND SVD
HANDS-ON: FEATURE EXTRACTION WITH PCA

- What is Dimensionality Reduction?
  - Why might we want to do it?
  - What are the broad types?
- What are feature selection and feature extraction?
- What is the goal of PCA? SVD?

I. DIMENSIONALITY REDUCTION

	continuous	categorical
supervised	regression	classification
unsupervised	dim reduction	clustering

Q: *What is* **dimensionality reduction***?*

A: *A set of techniques for* **reducing the size** *(in terms of features, records, and/or bytes)* **of the dataset** *under examination.*

In general, the idea is to regard the dataset as a matrix and to **decompose the matrix** *into simpler, meaningful pieces.*

Dimensionality reduction is frequently performed as a **pre-processing step** *before another learning algorithm is applied.*

Q: *What are the motivations for dimensionality reduction?*

A: *The number of features in our dataset can be difficult to manage, or even misleading (eg, if the relationships are actually simpler than they appear).*

Q: *What is the goal of dimensionality reduction?*

A:

- *reduce computational expense*
- *reduce susceptibility to overfitting*
- *reduce noise in the dataset*
- *enhance our intuition*
- *reduce multicollinearity*

Q: *What are some applications of dimensionality reduction?*

A:

- *topic models (document clustering)*
- *image recognition/computer vision*
- *recommender systems*

Q: *How is dimensionality reduction performed?*

A: *There are two approaches:* **feature selection** *and* **feature extraction.**

feature selection – *selecting a subset of features using an external criterion (filter) or the learning algorithm accuracy itself (wrapper)*

feature extraction – *mapping the features to a lower dimensional space*

The goal of feature selection is to select out the best possible subset of features for model-building from the original available features.

Feature selection is important, but typically when people say dimensionality reduction, they are referring to feature extraction.

The goal of feature extraction is to create a new set of coordinates that simplify the representation of the data.

II. FEATURE SELECTION

The goal of feature selection is to select out the best possible subset of features for model-building from the original available features.

This problem can be thought of as a search through the space of all possible feature subsets for the optimal subset.

For even a moderate number of features, this comprehensive search becomes impossible, so we need a heuristic approach.

Q: *How do we perform* **feature selection***?*

A: *By making use of* **wrappers, filters,** *or* **embedded methods**

wrappers *– potential feature subsets are compared based on the success of* **built models** *projected via* **cross-validation**

filters *– feature subsets are determined based on some simple prescribed metric over the features*

embedded *– feature selection happens within the model-building itself*

Wrappers use some criteria for trying out various feature subsets and build models of the desired type with each subset.

The models' performance are all estimated via cross-validation and the feature subset chosen that yields the greatest performance.

Ex: Stepwise Regression - Starting from zero features, test out adding each feature alone and training the model. Keep the feature that leads to the best model. Continue adding features until no more improvement

Filters use a prescribed general metric for determining which features to include in the model.

Filters are often far less computationally expensive than wrappers as the inclusion criteria for features is easy to compute.

Ex:

- *Information gain*
- *Correlation Coefficient, etc.*

Embedded methods have the model-building itself incorporate the feature selection.

These methods often strike a good balance between the strengths and weaknesses of wrappers and filters.

Ex: A good example is the LASSO Regression, in which the L1 regularized regression tends to zero out coefficients and implicitly choose features to be excluded.

III. FEATURE EXTRACTION

The goal of **feature extraction** *is to create a new set of coordinates that simplify the representation of the data.*

Typically we do this by using **matrix factorizations** *to map the features to a lower-dimensional space that minimizes information loss.*

Two prominent examples of such matrix factorization methods are **Principal Component Analysis (PCA)** *and* **Singular Value Decomposition (SVD)**

PCA

Principal component analysis is a dimension reduction technique that can be used on a matrix of any dimensions.

This procedure produces a new basis, each of whose components retain as much variance from the original data as possible.

The PCA of a matrix A *boils down to the* **eigenvalue decomposition** *of the* **covariance matrix** *of* A.

What is variance?

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

Variance is the average of the squared distance between the mean of a dataset and a point in the dataset.

In other words, it is a measure of the spread in the dataset. Recall that the square root of the variance is the standard deviation.

What is covariance?

A measure of how much 2 random variables change together.

Variance:

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)} \qquad var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

Covariance: $$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

The covariance matrix C of a matrix A is always square:

$$\mathrm{C} = \begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}.$$

off-diagonal elements C_{ij} give the covariance between X_i, X_j $(i \neq j)$

diagonal elements C_{ii} give the variance of X_i

The eigenvalue decomposition of a square matrix A is given by:

$$A = Q \Lambda Q^{-1}$$

The columns of Q are the **eigenvectors** *of A, and the values in Λ are the associated* **eigenvalues** *of A.*

For an eigenvector v of A and its eigenvalue λ, we have the important relation:

$$Av = \lambda v$$

NOTE

This relationship *defines* what it means to be an eigenvector of A.

The eigenvectors form a basis of the vector space on which A acts (eg, they are orthogonal).

*Furthermore the basis elements are ordered by their eigenvalues (from largest to smallest), and these **eigenvalues represent the amount of variance explained by each basis element**.*

Q: *So what comes out of a PCA?*

A: *Eigenvectors and eigenvalues.*

- *Eigenvectors are linear combinations of the original feature vectors*
- *Each eigenvector represents a feature in our new transformed feature space*
- *The eigenvalues represent a measure of the amount of variance explained by each corresponding eigenvector ("new feature")*
- *We can choose only the first k (whatever we like) of our "new features" from the eigenvector space and work with them as our new data knowing we'll have minimal data loss for a feature space of that size*

source: http://glowingpython.blogspot.it/2011/07/pca-and-image-compression-with-numpy.html

SVD

Consider a matrix M with m rows and n features.

The **singular value decomposition** *of M is given by:*

$$\underset{(m \times n)}{M} = \underset{(m \times r)}{U} \; \underset{(r \times r)}{\Sigma} \; \underset{(r \times n)}{V^T}$$

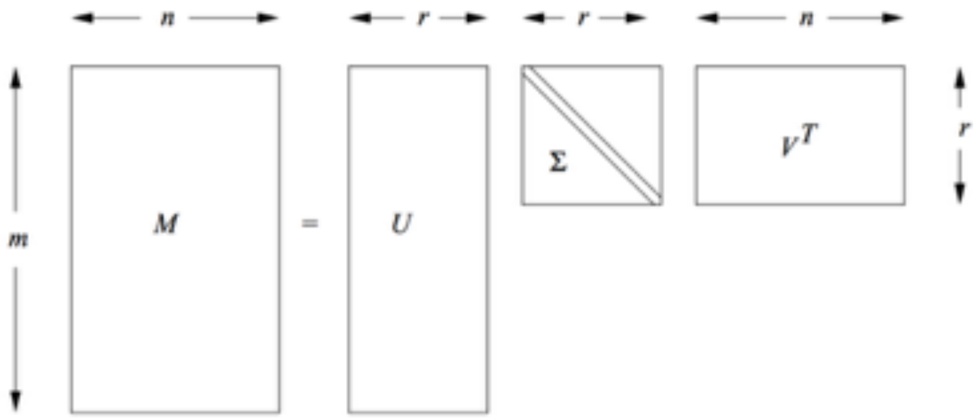
st. U, V are **orthogonal** *matrices and Σ is a* **diagonal** *matrix.*

$\rightarrow \quad UU^T = I_m,\ VV^T = I_n \ \rightarrow\ \Sigma_{ij} = 0 \ \ (i \neq j)$

The **singular value decomposition** *of M is given by:*

$$M = U \Sigma V^T$$

(m x n) (m x r) (r x r) (r x n)



The nonzero entries of Σ are the **singular values** *of A. These are real, nonnegative, and rank-ordered (decreasing from left to right).*

NOTE

The number of singular values is equal to the *rank* of A.

The rank of a matrix measures its *non-degeneracy.*

Ratings of movies by users:

	Matrix	Alien	Star Wars	Casablanca	Titanic
Joe	1	1	1	0	0
Jim	3	3	3	0	0
John	4	4	4	0	0
Jack	5	5	5	0	0
Jill	0	0	0	4	4
Jenny	0	0	0	5	5
Jane	0	0	0	2	2

there are two "concepts" underlying the movies:

science-fiction and romance

Ratings of movies by users:

$$\underset{M}{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix}} = \underset{U}{\begin{bmatrix} .14 & 0 \\ .42 & 0 \\ .56 & 0 \\ .70 & 0 \\ 0 & .60 \\ 0 & .75 \\ 0 & .30 \end{bmatrix}} \underset{\Sigma}{\begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix}} \underset{V^{\mathrm{T}}}{\begin{bmatrix} .58 & .58 & .58 & 0 & 0 \\ 0 & 0 & 0 & .71 & .71 \end{bmatrix}}$$

SINGULAR VALUE DECOMPOSITION – EXAMPLE

	Matrix	Alien	Star Wars	Casablanca	Titanic
Joe	1	1	1	0	0
Jim	3	3	3	0	0
John	4	4	4	0	0
Jack	5	5	5	0	0
Jill	0	0	0	4	4
Jenny	0	0	0	5	5
Jane	0	0	0	2	2

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix}}_{M} = \underbrace{\begin{bmatrix} .14 & 0 \\ .42 & 0 \\ .56 & 0 \\ .70 & 0 \\ 0 & .60 \\ 0 & .75 \\ 0 & .30 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} .58 & .58 & .58 & 0 & 0 \\ 0 & 0 & 0 & .71 & .71 \end{bmatrix}}_{V^{\mathrm{T}}}$$

M: people -> movies

U: people -> concepts

V: concepts -> movies

Σ: the strength of each of the concepts

SINGULAR VALUE DECOMPOSITION – A MORE REALISTIC EXAMPLE

	Matrix	Alien	Star Wars	Casablanca	Titanic
Joe	1	1	1	0	0
Jim	3	3	3	0	0
John	4	4	4	0	0
Jack	5	5	5	0	0
Jill	0	0	0	4	4
Jenny	0	0	0	5	5
Jane	0	0	0	2	2

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix}}_{M'} =$$

$$\underbrace{\begin{bmatrix} .13 & .02 & -.01 \\ .41 & .07 & -.03 \\ .55 & .09 & -.04 \\ .68 & .11 & -.05 \\ .15 & -.59 & .65 \\ .07 & -.73 & -.67 \\ .07 & -.29 & .32 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \\ .40 & -.80 & .40 & .09 & .09 \end{bmatrix}}_{V^{\mathrm{T}}}$$

How to reduce dimensions?

<u>Drop Low Singular Values</u> -> eliminate corresponding rows of U and V

$$\underset{M'}{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix}} =$$

$$\underset{U}{\begin{bmatrix} .13 & .02 & -.01 \\ .41 & .07 & -.03 \\ .55 & .09 & -.04 \\ .68 & .11 & -.05 \\ .15 & -.59 & .65 \\ .07 & -.73 & -.67 \\ .07 & -.29 & .32 \end{bmatrix}} \underset{\Sigma}{\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}} \underset{V^T}{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \\ .40 & -.80 & .40 & .09 & .09 \end{bmatrix}}$$

$$\underset{\Sigma}{\begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix}}$$

SINGULAR VALUE DECOMPOSITION – A MORE REALISTIC EXAMPLE

How to reduce dimensions?

__Drop Low Singular Values__

$$\underset{M'}{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix}} =$$

$$\underset{U}{\begin{bmatrix} .13 & .02 & -.01 \\ .41 & .07 & -.03 \\ .55 & .09 & -.04 \\ .68 & .11 & -.05 \\ .15 & -.59 & .65 \\ .07 & -.73 & -.67 \\ .07 & -.29 & .32 \end{bmatrix}} \underset{\Sigma}{\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}} \underset{V^T}{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \\ .40 & -.80 & .40 & .09 & .09 \end{bmatrix}}$$

$$\begin{bmatrix} .13 & .02 \\ .41 & .07 \\ .55 & .09 \\ .68 & .11 \\ .15 & -.59 \\ .07 & -.73 \\ .07 & -.29 \end{bmatrix} \begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix} \begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \end{bmatrix}$$

$$= \begin{bmatrix} 0.93 & 0.95 & 0.93 & .014 & .014 \\ 2.93 & 2.99 & 2.93 & .000 & .000 \\ 3.92 & 4.01 & 3.92 & .026 & .026 \\ 4.84 & 4.96 & 4.84 & .040 & .040 \\ 0.37 & 1.21 & 0.37 & 4.04 & 4.04 \\ 0.35 & 0.65 & 0.35 & 4.87 & 4.87 \\ 0.16 & 0.57 & 0.16 & 1.98 & 1.98 \end{bmatrix}$$

HANDS-ON: PCA