Machine Learning

PATRICK HALL

DEPARTMENT OF DECISION SCIENCE

Matrix Factorization

- Dimension Reduction:
  - **Principle Component Analysis**
  - **Singular Value Decomposition**
  - **Non-Negative Matrix Factorization**
  - **Generalized Low Rank Models**
  - **Factorization Machines**

Principal Components

Principal Component Analysis

Principal Components are a sequence of projections of the data - mutually correlated and ordered in variance.

- Linear manifolds approximating a set of N points $x_i = \mathbb{R}^p$
- Nonlinear generalization of principal curves and surfaces

Principal Components

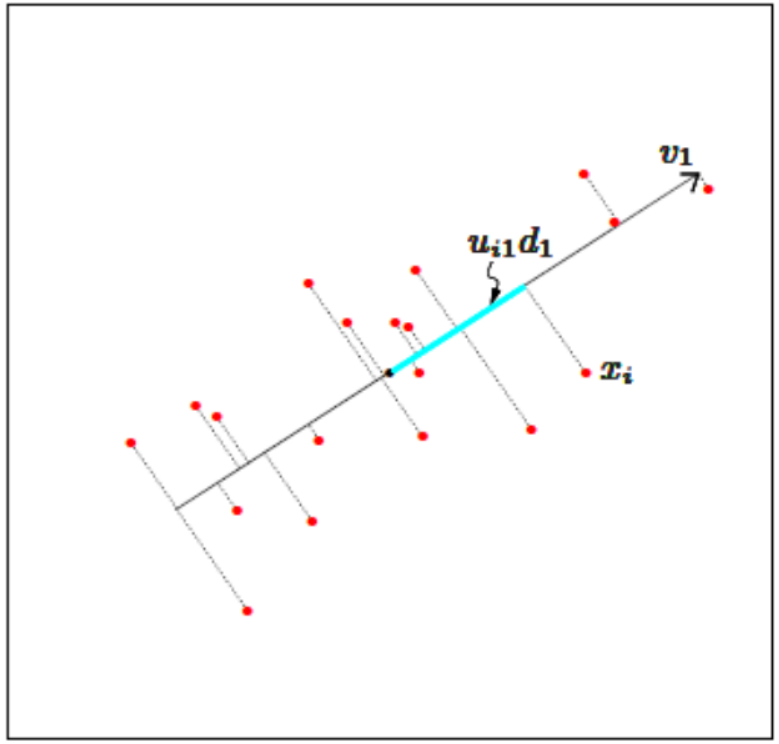

FIGURE 14.20. *The first linear principal component of a set of data. The line minimizes the total squared distance from each point to its orthogonal projection onto the line.*

Elements of Statistical Learning (pg.534) –

The one-dimensional principal component line in the $\mathbb{R}^2$ with $q = 1$. For each data point x_i, there is a closest point on the line given by $u_{i1}d_1v_1$, where v_1 is the direction of the line and $\hat{\lambda}_i = u_{i1}d_1$ measures the distance along the line from the origin.

Principal Components

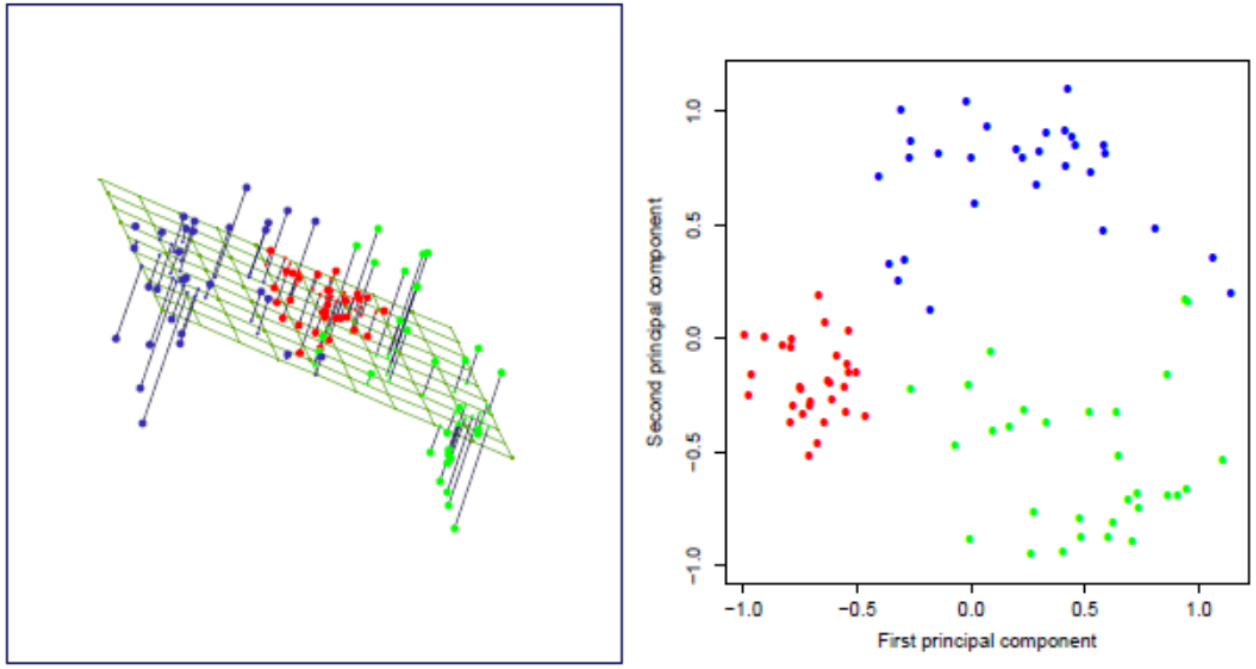

FIGURE 14.21. *The best rank-two linear approximation to the half-sphere data. The right panel shows the projected points with coordinates given by* $\mathbf{U}_2\mathbf{D}_2$, *the first two principal components of the data.*

Elements of Statistical Learning (pg.536)

The two-dimensional principal component surface with $q = 2$ to fit the half-sphere data.

Principal Components

Handwritten Digits
Principal components are a useful tools for dimension reduction and compression.
Consider sample of 130 handwritten 3's - each a digitized 16x16 grayscale image
as shown in Figure 14.22.

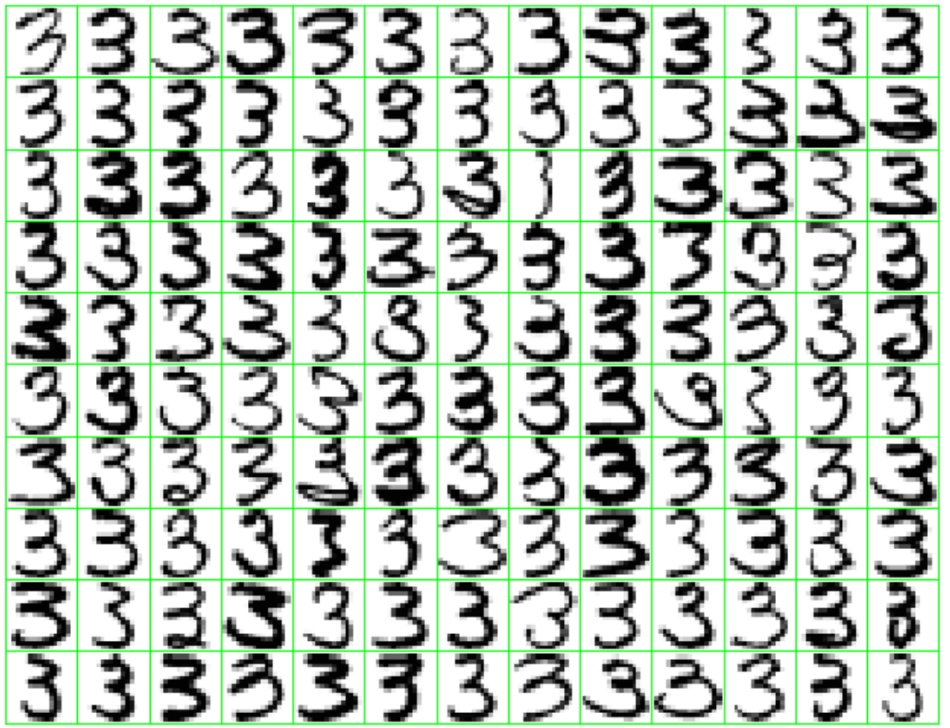

FIGURE 14.22. *A sample of* 130 *handwritten 3's shows a variety of writing styles.*

Principal Components

The principal components are computed via SVD and Figure 14.23 shows the first two principal components of these data. In the parametrized model of the form, equation(4), the two-component model is defined as:

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

$$= \bar{x} + \lambda_1 \cdot v_1 + \lambda_2 \cdot v_2 .$$

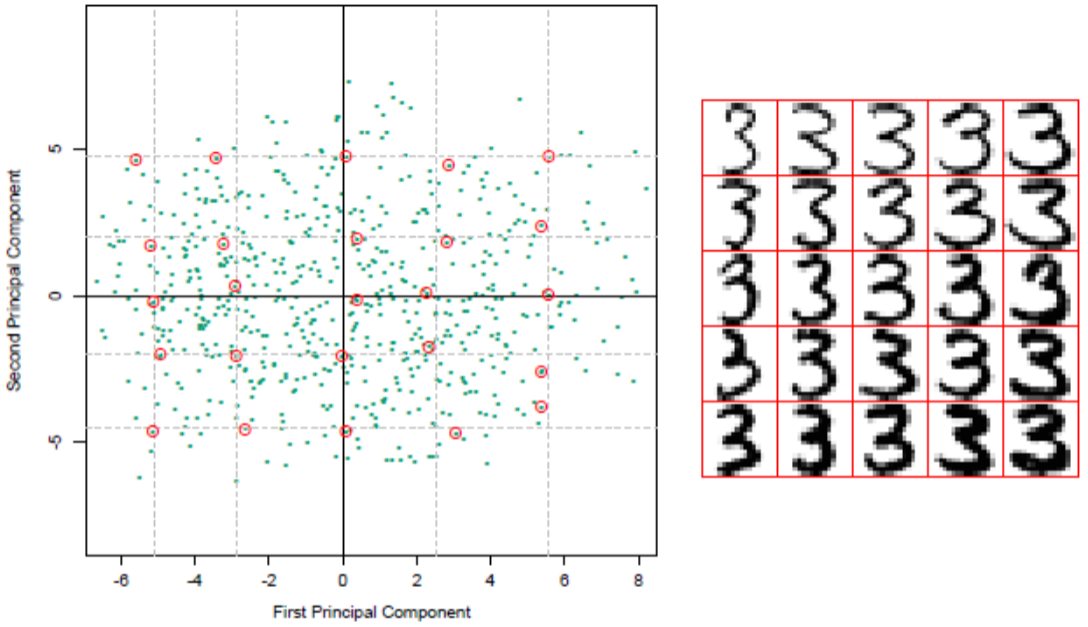

FIGURE 14.23. *(Left panel:) the first two principal components of the hand-written threes. The circled points are the closest projected images to the vertices of a grid, defined by the marginal quantiles of the principal components. (Right panel:) The images corresponding to the circled points. These show the nature of the first two principal components.*

Principal Components

Compare the singular values to those obtained for equivalent uncorrelated data by randomly scrambling each column of $\boldsymbol{X}$. The pixels in a digitized image are inherently correlated, and since these are all the same digit the correlations are even stronger.
This example well illustrates where a relatively small subset of the principal components serve as excellent lower-dimensional features for representing the high-dimensional data.

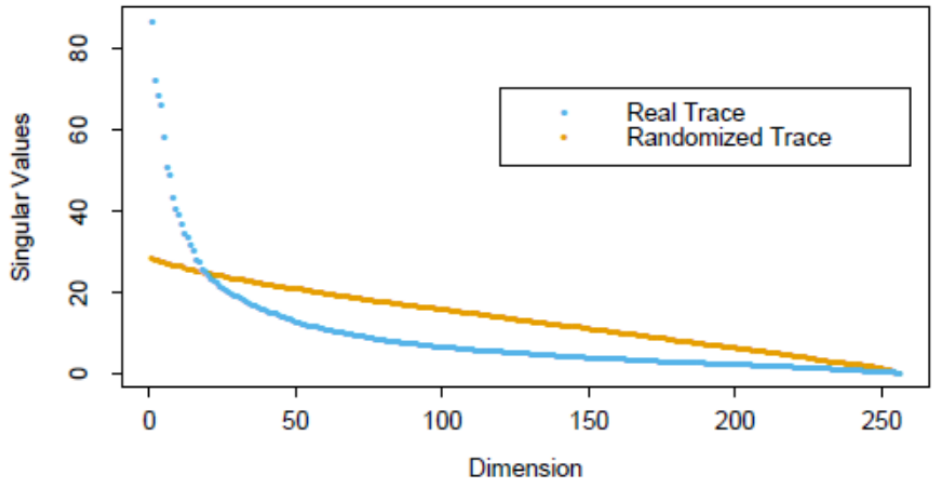

FIGURE 14.24. *The 256 singular values for the digitized threes, compared to those for a randomized version of the data (each column of* $\mathbf{X}$ *was scrambled).*

Principal Components

Principal Components

The Principal components of a set of data in $x_i = \mathbb{R}^p$ provide a sequence of best linear approximation to that data with all ranks $q \leq p$. If we denote the set of observations by $x_1, x_2, ..., x_N$ and consider the $rank - q$ linear model:

$$f(x) = \mu + \boldsymbol{V}_q \lambda \tag{4}$$

where λ is a location vector in $x_i = \mathbb{R}^p$, $\boldsymbol{V}_q$ is pxq matrix with q orthogonal unit vectors as columns, and λ is a q vector of parameters.

Principal Components

Fitting a model to the data by least squares amounts to minimizing the *reconstruction error*

$$\min_{\mu,\{\lambda_i\},\boldsymbol{V}_q} \sum_{i=1}^{N} \|x_i - \mu - \boldsymbol{V}_q \lambda_i\|^2 \tag{5}$$

We can partially optimize for μ and λ_i to obtain:

$$\hat{\mu} = \bar{x}, \tag{6}$$

$$\hat{\lambda}_i = \boldsymbol{V}_q^T (x_i - \bar{x}) \tag{7}$$

Principal Components

Which leads to find the orthogonal matrix $\boldsymbol{V}_q$:

$$\min_{\boldsymbol{V}_q} = \sum_{i=1}^{N} \|(x_i - \bar{x}) - \boldsymbol{V}_q \boldsymbol{V}_q^T (x_i - \bar{x})\|^2 \qquad (8)$$

Here, $\boldsymbol{H}_q = \boldsymbol{V}_q \boldsymbol{V}_q^T$,(8), is a *pxp projection matrix* that maps each point x_i onto its rank-q reconstruction $\boldsymbol{H}_q x_i$, a orthogonal projection of x_i onto the subspace spanned by the columns of $\boldsymbol{V}_q$.

Principal Components

The solution can be expressed as the stacked (centered) observations into the rows of an *Nxp* matrix $\boldsymbol{X}$ and we can construct the *singular value decomposition* of $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\boldsymbol{T}} \tag{9}$$

where $\boldsymbol{U}$ is *Nxp* orthogonal matrix whose columns $\boldsymbol{u}_j$ are called the *left singular vectors*; and $\boldsymbol{V}$ is *pxp* orthogonal matrix whose columns $\boldsymbol{v}_j$ are called the *right singular vectors*; and $\boldsymbol{D}$ is *pxp* diagonal matrix with diagonal elements $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ known as the *singular values*. The columns of $\boldsymbol{UD}$ are called the principal components of $\boldsymbol{X}$.

Singular Value Decomposition

Matrix Factorization

Diagonalization: In many cases, the eigenvalues-eigenvetor of matrix $\boldsymbol{A}$ can be descriptively portrayed in a useful factorization of the form $\boldsymbol{A} = \boldsymbol{PDP}^{-1}$ where $\boldsymbol{D}$ is a diagonal matrix. Moreover, diagonal entries of $\boldsymbol{D}$ are the eigenvalues of $\boldsymbol{A}$ that corresponds, respectively, to the eigenvectors of $\boldsymbol{P}$. Unfortunately, not all matrices can be factored as $\boldsymbol{A} = \boldsymbol{PDP}^{-1}$ with $\boldsymbol{D}$ diagonal matrix. However, a factorization of a form $\boldsymbol{A} = \boldsymbol{QDP}^{-1}$ is possible for any *mxn* matrix $\boldsymbol{A}$. A special factorization of this type, $\boldsymbol{A} = \boldsymbol{QDP}^{-1}$, is called the *singular value decomposition*, and is one of the most useful matrix factorization in applied linear algebra and its related applications.

Singular Value Decomposition

Singular Values of an *mxn* Matrix

Let $\boldsymbol{A}$ be an *mxn* matrix. Then $\boldsymbol{A^T A}$ is symmetric and can be orthogonally diagonalized. Let $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_n\}$ be an orthonormal basis for $\mathbb{R}^n$ consisting of eigenvectors of $\boldsymbol{A^T A}$, and $\lambda_1, ..., \lambda_n$ be associated eigenvalues of $\boldsymbol{A^T A}$. Then, for $1 \leq i \leq n$,

$$\begin{aligned} \|A\boldsymbol{v}_i\|^2 &= (A\boldsymbol{v}_i)^T A\boldsymbol{v}_i = \boldsymbol{v}_i^T A^T A\boldsymbol{v}_i \\ &= \boldsymbol{v}_i^T(\lambda_i \boldsymbol{v}_i) \\ &= \lambda_i \end{aligned} \tag{1}$$

with, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. The **singular values** of $\boldsymbol{A}$ are the square roots of the eigenvalues of $\boldsymbol{A^T A}$, $\sigma_1, ..., \sigma_n$.

Singular Value Decomposition

Now, suppose that $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_n\}$ is an orthonormal basis of $\mathbb{R}^n$ consisting of eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$, arranged so that the corresponding eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$ satisfy $\lambda_1 \geq \cdots \geq \lambda_n$, and suppose that $\boldsymbol{A}$ has r nonzero singular values. Then, $\{A\boldsymbol{v}_1, ..., A\boldsymbol{v}_n\}$ is an orthogonal biasis for $ColA$ and $rankA = r$

Singular Value Decomposition

The Singular Value Decomposition

The "Diagonal" Matrix Σ: The decomposition of A involves an mxn Σ of the form

$$\Sigma = \begin{bmatrix} \boldsymbol{D} & 0 \\ 0 & 0 \end{bmatrix} \tag{2}$$

where $\boldsymbol{D}$ is an rxr diagonal matrix for some $r \leq min(m, n)$ with $m - r$ rows and $n - r$ columns of 0 entries.

Singular Value Decomposition

Now, we can define:

$$\boldsymbol{A = U\Sigma V^T} \tag{3}$$

where $\boldsymbol{A}$ is an mxn matrix with $rankA = r$. Then there exists an mxn matrix Σ as defined in (2), where the diagonal entries in $\boldsymbol{D}$ are the first r singular values of $\boldsymbol{A}$, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, and there exists an mxm orthogonal matrix $\boldsymbol{U}$ and an nxn orthogonal matrix $\boldsymbol{V}$. ANY factorization of the form $\boldsymbol{A = U\Sigma V^T}$ with positive diagonal entries in $\boldsymbol{D}$ is called a **singular value decomposition** of $\boldsymbol{A}$.

Singular Value Decomposition

Example:

Find a singular value decomposition of $A = \begin{bmatrix} 1 & -1 \\ -2 & 2 \\ 2 & -2 \end{bmatrix}$

$$\begin{bmatrix} 1 & -1 \\ -2 & 2 \\ 2 & -2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{2}{\sqrt{5}} & \frac{-2}{\sqrt{45}} \\ \frac{-2}{3} & \frac{1}{\sqrt{5}} & \frac{4}{\sqrt{45}} \\ \frac{2}{3} & 0 & \frac{5}{\sqrt{45}} \end{bmatrix} \cdot \begin{bmatrix} 3\sqrt{2} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Non-negative Matrix Factorization

Non-negative Matrix factorization - a recent alternative approach to PCA in which the data and components are assumed to be non-negative. It is useful for modeling non-negative dataset such as images.
The Nxp data matrix $\boldsymbol{X}$ is approximated by

$$\boldsymbol{X} \approx \boldsymbol{WH} \tag{10}$$

where $\boldsymbol{W}$ is Nxp and $\boldsymbol{H}$ is $rxp, r \leq max(N, p)$. We assume that $x_{ij}, w_{ik}, H_{kj} \geq 0$.

Non-negative Matrix Factorization

Elements of Statistical Learning (pg.534) – Figure 14.33
Compare the NMF, VQ, and PCA.
All three learning methods were applied to a database of N=2429 facial images, each with 19x19 pixels.
Note that, unlike VQ and PCA, NMF learns to represent faces with set of basis images resembling parts of faces

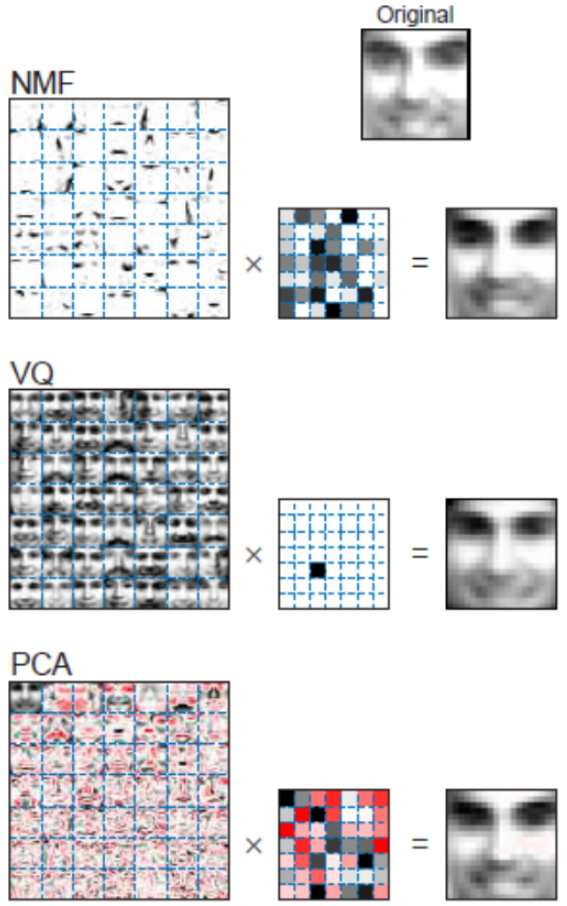

FIGURE 14.33. *Non-negative matrix factorization (NMF), vector quantization (VQ, equivalent to k-means clustering) and principal components analysis (PCA) applied to a database of facial images. Details are given in the text. Unlike VQ and PCA, NMF learns to represent faces with a set of basis images resembling parts of faces.*

Non-negative Matrix Factorization

The matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ are found by maximizing

$$L(\boldsymbol{L}, \boldsymbol{H}) = \sum_{i=}^{N} \sum_{j=1}^{p} [x_{ij} \log (\boldsymbol{WH})_{ij} - (\boldsymbol{WH}_{ij})] \qquad (11)$$

where above (11) is the log-likelihood from a model in which x_{ij} has Poisson distribution with mean $(\boldsymbol{WH})_{ij}$ - a reasonable for a positive data.

Non-negative Matrix Factorization

The following algorithm converges to a local maximum of $L(\boldsymbol{W}, \boldsymbol{H})$:

$$w_{ik} \leftarrow w_{ik} \frac{\sum_{j=1}^{p} h_{kj} x_{ij} / (\boldsymbol{WH})_{ij}}{\sum_{j=1}^{p} h_{kj}} \tag{12}$$

$$h_{kj} \leftarrow h_{kj} \frac{\sum_{i=1}^{N} w_{ik} x_{ij} / (\boldsymbol{WH})_{ij}}{\sum_{i=1}^{N} w_{ik}} \tag{13}$$

Non-negative Matrix Factorization

Archetypal Analysis approximates each data point by a convex combination of a collection of prototypes that lie on the "convex hull o f the data cloud." In this sens, the prototypes are "pure" or "archetypal".
Recall, the Nxp matrix $\boldsymbol{X}$ is modeled as

$$\boldsymbol{X} \approx \boldsymbol{WH} \tag{14}$$

where $\boldsymbol{W}$ is Nxr and $\boldsymbol{H}$ is rxp. Here we assume that $w_{ik} \geq 0$ and $\sum_{k=1}^{r} = 1 \forall i$ Hence, the N data points (row of $\boldsymbol{X}$) in $p - dimensional$ space are represented by convex combination of the r archetypes (rows of $\boldsymbol{H}$).

Non-negative Matrix Factorization

We also assume that

$$\boldsymbol{H} = \boldsymbol{BX} \tag{15}$$

where $\boldsymbol{B}$ is rxN with $b_{ki} \geq 0$ and $\sum i = 1^{N} b_{ki} = 1 \forall k$. Thus the archetypes are convex combinations of the data points. Using both equation(14) and equation(15), we can minimize

$$\begin{aligned} J(\boldsymbol{W}, \boldsymbol{B}) &= \|\boldsymbol{X} - \boldsymbol{WH}\|^2 \\ &= \|\boldsymbol{X} - \boldsymbol{WBX}\|^2 \end{aligned} \tag{16}$$

over the weights $\boldsymbol{W}$ and $\boldsymbol{B}$, and the algorithm converges to a local minimum of the criterion.

Generalized Low Rank Model

GLRM - extension of PCA where:

- Appropriate loss function instead of least-square approach as in PCA
- Can add regularization on the low dimensional factors to improve generalization error and avoid overfitting
- Can impose some structure in the low dimensional factors - such as sparsity

In essence, the term *Generalized Low Rank Model (GLRM)* refer to any low rank approximation of a dataset obtained by minimizing a loss function on the approximation error together with regularization of the low dimensional factors.

Generalized Low Rank Model

Note, in general, low rank approximations problems cannot be solved globally and efficiently. However, can be solved locally by methods that alternate between updating the two factors in the low rank approximation.

That is, each step involves solving a convex or non-convex problem that can be solved exactly. While these alternating methods do not find the globally best low rank approximation, they are very useful and effective for the original data analysis.

Generalized Low Rank Model

Suppose now that our data $\boldsymbol{A}$ is a database consisting of m examples and n features, with entries $\boldsymbol{A}_{ij}$ drawn from a feature set $\mathcal{F}_j$. We observe only entries $\boldsymbol{A}_{ij}$ for $(i,j) \in \Omega \subseteq \{i,...,m\}x\{1,...,n\}$ from the matrix $\boldsymbol{A}$, so the other entries are unknown. Now the user provides a loss function $L_{ij} : \mathbb{R}x\mathcal{F}_j \to \mathbb{R}$. The loss $L_{ij}(u,a)$ describes the approximation error incurred when we represent a feature value $a \in \mathcal{F}_i$ by the number $u \in \mathbb{R}$. The user also provides regularizers $r_i : \mathbb{R}^k \to \mathbb{R} \bigcup \{\infty\}$ for $i = 1,...,n$ and $j = 1,...,m$. These regularizers prevent overfitting.

Generalized Low Rank Model

Now, we can define a GLRM on the database $\boldsymbol{A}$ as:

$$\textit{minimize} \quad \sum_{(i,j)\in\Omega} L_{ij}(x_i y_j, \boldsymbol{A}_{ij}) + \sum_{i=1}^{m} r_i(x_i) + \sum_{j=1}^{n} \tilde{r}_j(y_j) \qquad (23)$$

with variable $X \in \mathbb{R}^{nxk}$ and $Y \in \mathbb{R}^{kxm}$, and with loss L_{ij} and regularizers $r_i(x_i) : \mathbb{R}^{1xk} \to \mathbb{R}$ and $\tilde{r}_j(y_j) : \mathbb{R}^{kx1} \to \mathbb{R}$

Generalized Low Rank Model

Few examples of GLRM and $L_{ij}(u, a)$ are illustrated.

Model	$\mathbf{L_{ij}(u, a)}$	$\mathbf{r(x)}$	$\mathbf{\tilde{r}(y)}$
PCA	$(u-a)^2$	0	0
Quadratically regularized PCA	$(u-a)^2$	$\|x\|_2^2$	$\|y\|_2^2$
nonnegative matrix factorization	$(u-a)^2$	$I_+(x)$	$I_+(y)$
Sparse PCA	$(u-a)^2$	$\|x\|_1$	$\|y\|_1$
k-means	$(u-a)^2$	$I_1(x)$	0
Robust PCA	$\|u-a\|$	$\|x\|_2^2$	$\|y\|_2^2$
Boolean PCA (MMMF)	$(1-au)_+$	$\|x\|_2^2$	$\|y\|_2^2$
Logistic PCA	$\log(1+\exp(-au))$	$\|x\|_2^2$	$\|y\|_2^2$
Poisson PCA	$\exp(u) - au + a\log a - a$	$\|x\|_2^2$	$\|y\|_2^2$
Ordinal PCA	$\sum_{a'<a}(1-u+a')_+ + \sum_{a'>a}(1+u-a')_+$	$\|x\|_2^2$	$\|y\|_2^2$

Table 1: A few examples of GLRMs. Here I_+ is the indicator of the nonnegative orthant, and I_1 is the indicator of the 1-sparse unit vectors.

Factorization Machines

Factorization Machines(FM) can estimate reliable parameters under very high sparsity. The factorization machine models all nested variable interactions, but uses a factorized parametrization, and can be computed in linear time and that it depends only on a linear number of parameters. This allows direct optimization and storage of model parameters without the need of storing any training data for prediction.
The advantages of **FM** are:

- FM allows parameter estimation under very spare data
- FM have linear complexity
- FM are a general predictor that can work with any real valued feature vector

Factorization Machines

FM Model Equation: FM of degree 2 is defined as

$$\hat{y}(x) = w_0 + \sum_{i-1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j \qquad (20)$$

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \boldsymbol{w} \in \mathbb{R}^{\mathrm{n}}, \mathrm{and}\, \boldsymbol{V} \in \mathbb{R}^{\mathrm{nxk}}$$

Factorization Machines

And $< \cdot, \cdot >$ is the dot product of two vectors of size k:

$$< v_i, v_j >= \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \tag{21}$$

A row $\boldsymbol{v}_i$ within $\boldsymbol{V}$ is the $i-$th variable with k factors and $k \in \mathbb{N}_0^+$ is the hyperparameter that defines the dimensionality of the factorization.

Factorization Machines

FM can be applied in variety of prediction tasks such as:

- Regression: $\hat{y}(x)$ can be used directly as the predictor and the optimization criterion is the minimum least square error in D.
- Binary Classification: the sign of $\hat{y}(x)$ is used and the parameters are optimized for hinge loss or digit loss
- Ranking: the vectors $\boldsymbol{x}$ are ordered by the scores of $\hat{y}(x)\mathrm{W}$ and optimization is done over pairs of instance vectors $< x^{(a)}, x^{(b)} > \in D$ with pairwise classification loss

Multidimensional Scaling

Multidimensional Scaling has a similar goal of PCA and NMF (lower-dimensional manifold), but approaches the problem in different way.

Lets suppose we have observations $x_1, x_2, ..., x_N \in \mathbb{R}^p$ and let d_{ij} be the distance between i and$j$. We often choose the Euclidean distance, $d_{ij} = \|x_i - x_j\|$, but other distances may be used such as *dissimilarity* measure d_{ij}. Multidimensional scaling seeks values $z_1, z_2, ..., z_N \in \mathbb{R}^k$ to minimize the stress function

$$S_M(z_1, z_2, ..., z_N) = \sum_{i \neq i'} (d_{ii'} - \|z_i - z_{i'}\|)^2 \qquad (17)$$

This is known as *least squares* or *Kruskal-Shephard* scaling that preserves the pairwise distance as well as possible.

Multidimensional Scaling

A variation on *least squares* scaling is - *Sammon mapping* that seeks to minimizes

$$S_{Sm}(z_1, z_2, ..., z_N) = \sum_{i \neq \acute{i}} \frac{(d_{i\acute{i}} - \|z_i - z_{\acute{i}}\|)^2}{d_{i\acute{i}}} \qquad (18)$$

with an emphasis on preserving smaller pairwise distance.
In *classical scaling*, we use the centered inner product $s_{i\acute{i}} =< x_i - \bar{x}, x_{\acute{i}} - \bar{x} >$ and minimize

$$S_C(z_1, z_2, ..., z_N) = \sum_{i,\acute{i}} (s_{i\acute{i}} - < z_i - \bar{z}, z_{i\acute{i}} - \bar{z} >)^2 \qquad (19)$$

over the $z_i, z_2, ..., z_N \in \mathbb{R}^k$