# Kernel-Based Feature Extraction For Collaborative Filtering

**Saket Sathe, Charu C. Aggarwal**
**IBM T.J. Watson Research Center**

**Xiangnan Kong, Xinyue Liu**
**Worcester Polytechnic Institute, USA**

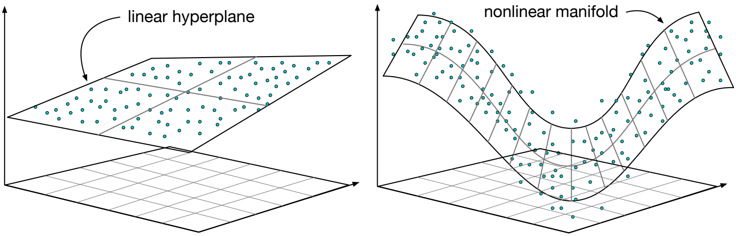
# Motivation – Fitting Ratings to Nonlinear Manifolds

- **Collaborating Filtering:** Given an incompletely filled matrix, predict the missing entries

- **Example:** Recommender systems. Only some users rate some items. Predict how a user would rate an unrated item.

- **Matrix Factorization (MF):** Established linear models for collaborative filtering

- **Limitation of MF:**




**(a)** Suitable for linear MF models **(b)** Unsuitable for linear MF models

# Matrix Factorization

- **Notation:**
  - $R = [r_{ij}]$ be an $m \times n$ $(users \times items)$ incompletely specified ratings matrix
  - $E$ be the set of specified entries $E = \{(i,j) : r_{ij}$ is specified$\}$
- **SVD Factorization** factorizes $R \approx UV^T$, such that $U = [u_{ij}]$ is an $m \times k$ and $V = [v_{ij}]$ is $n \times k$. $k$ is known as the rank of the factorization.
- **Prediction rule:** $U$ and $V$ are used to predict the $(i,j)$ th rating as

$$\hat{r}_{ij} = \sum_{s=1}^{k} u_{is} v_{js}$$

- **Optimization Problem:** Minimizes the squared prediction error on the observed entries:

$$\text{Minimize } J = \frac{1}{2} \sum_{(i,j) \in E} \left( r_{ij} - \sum_{s=1}^{k} u_{is} \cdot v_{js} \right)^2 + \frac{\lambda_f}{2} \left( \sum_{i,s} u_{is}^2 + \sum_{j,s} v_{js}^2 \right).$$

Here $\lambda_f$ is a regularization parameter.

# Broad Contours of the Approach

- Traditional matrix factorization derives $U$ and $V$ simultaneously.
- Therefore, starigtforward application of kernels is non-trivial, because they can only be used on the users or items, but not on both

**Our Approach:**

- We decouple the process of determining the item and user factors
- Extract a $k$-dimensional embedding $V_0$ for the items using Kernel PCA. This results in an $n \times k$ item factor matrix $V_0$.
- Factorize the original ratings matrix as $R \approx UV_0$. Note that $V_0$ is fixed in this factorization and only $U$ is learned.
- $V_0$ can be substituted for the item factors in many MF models. Therefore, the approach is relatively general.

# Kernel Latent Factor Models - Item Factor Learning

1. **Bias estimation** is done using the following prediction rule $\hat{r}_{ij} = b_i^{user} + b_j^{item}$.

2. **Bias Removal:** User- and item-specific biases are removed from each entry of $R$ and the unbiased ratings matrix is denoted as $R_u$

3. **Similarity Matrix:** Generate $n \times n$ Gaussian kernel similarity matrix $S$ from $R_u$ by computing similarity between pairs of columns $\overline{rc_i}, \overline{rc_j}$ of $R_u$ as:

$$K(\overline{rc_i}, \overline{rc_j}) = \exp\left(-\frac{||\overline{rc_i} - \overline{rc_j}||^2}{2\sigma^2}\right)$$

   Unobserved entries are set to 0 after bias removal.

4. **Mean Centering:** $S$ is then mean-centered as $S \leftarrow (I - O/n)S(I - O/n)$. $O$ is an $n \times n$ matrix of all 1s, and $I$ is an $n \times n$ identity matrix.

5. **Embedding:** Extract rank-$k$ embedding $V_0$ from $S$ by using top-$k$ eigenvectors of $S$ as $V_0 = Q\Sigma$

# Kernelizing Matrix Factorization Models

## Kernel Biased Matrix Factorization

- **Principle:** Users and items have inherent biases that should be used for adjusting the predictions
- **Prediction Rule:** $\hat{r}_{ij} = b_i^{user} + b_j^{item} + \sum_{s=1}^{k} u_{is} v_{js}^0$.
- **Notation:** Original $\Longrightarrow$ BMF, Kernel $\Longrightarrow$ K-BMF

[1] Y. Koren. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. KDD, 2008.

# Kernelizing Matrix Factorization Models

## Kernel Biased Matrix Factorization

- **Principle:** Users and items have inherent biases that should be used for adjusting the predictions
- **Prediction Rule:** $\hat{r}_{ij} = b_i^{user} + b_j^{item} + \sum_{s=1}^{k} u_{is} v_{js}^0$.
- **Notation:** Original $\implies$ BMF, Kernel $\implies$ K-BMF

## Kernel Implicit Feedback Factorization

- **Principle:** The act of rating an item has tremendous predictive value, irrespective of the actual value of the rating.
- **Prediction Rule:** $\hat{r}_{ij} = b_i^{user} + b_j^{item} + \sum_{s=1}^{k} \left( u_{is} + \frac{\sum_{t \in N(u)} y_{jt}}{\sqrt{|N(u)|}} \right) v_{js}^0$,

  where $N(u)$ is the set of items rated by user $u$ and $y_{jt}$ are implicit feedback affinities
- **Notation:** Original $\implies$ SVD++, Kernel $\implies$ K-SVD++

[1] Y. Koren. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. KDD, 2008.

# Experimental Evaluation

## Data Sets:

| Data set | Users | Items | Ratings | Density (%) |
|---|---|---|---|---|
| Filmtrust | 1,508 | 2,071 | 35,497 | 1.13 |
| ML100k | 943 | 1,682 | 100,000 | 6.30 |
| Ciao | 7,257 | 10,000 | 141,984 | 0.20 |
| Epinions | 21,427 | 10,000 | 385,358 | 0.18 |
| Jester | 63,978 | 150 | 1,761,439 | 18.35 |

**Accuracy Metric:** RMSE is used to measure accuracy. Lower values are better.

| | Data set | | | | |
|---|---|---|---|---|---|
| Methods | Filmtrust | ML100k | Ciao | Epinions | Jester |
| BMF | 0.8120 | 0.9467 | 0.9835 | 1.0704 | 5.0720 |
| K-BMF | **0.7988** | **0.9312** | **0.9663** | **1.0462** | **4.2139** |
| SVD++ | 0.8133 | 0.9352 | 1.0001 | 1.0827 | 5.2920 |
| K-SVD++ | **0.7993** | **0.9305** | **0.9671** | **1.0465** | **4.2111** |

**Table:** Accuracy comparison $(\lambda_b = 0.005, \lambda_f = 0.015)$.

**Maximum accuracy gains of 17% for K-BMF and 20% for K-SVD++**

# Summary and Conclusion

- We showed a general technique for improving many off-the-shelf collaborative filtering methods with the use of kernel features.
- Multi-step training methods have tremendously helped in other domains. For e.g., pre-training in deep neural networks
- Our experiments show consistent performance improvements in accuracy