Collaborative filtering with attributes

Jacob Abernethy\textsuperscript{1} \quad Francis Bach\textsuperscript{2}
Theodoros Evgeniou\textsuperscript{3} \quad Jean-Philippe Vert\textsuperscript{4}

\textsuperscript{1}UC Berkeley
\textsuperscript{2}INRIA / Ecole normale superieure de Paris
\textsuperscript{3}INSEAD
\textsuperscript{4}ParisTech / Institut Curie / INSERM

The Snowbird Learning Workshop, Snowbird, USA, April 1st, 2008.
Collaborative Filtering (CF)

The problem

- Given a set of $n_{\mathcal{X}}$ “movies” $\mathbf{x} \in \mathcal{X}$ and a set of $n_{\mathcal{Y}}$ “people” $\mathbf{y} \in \mathcal{Y}$,
- predict the “rating” $z(\mathbf{x}, \mathbf{y}) \in \mathcal{Z}$ of person $\mathbf{x}$ for film $\mathbf{y}$
- Training data: large $n_{\mathcal{X}} \times n_{\mathcal{Y}}$ incomplete matrix $\mathcal{Z}$ that describes the known ratings of some persons for some movies
- Goal: complete the matrix.

---

Abernethy et al. ()

CF with attributes

Snowbird 2008
Another CF example

Drug design

- Given a family of proteins of therapeutic interest (e.g., GPCR’s)
- Given all known small molecules that bind to these proteins
- Can we predict unknown interactions?
CF by low-rank matrix approximation

- A common strategy for CF
- $Z$ has rank less than $k \iff Z = UV^\top, U \in \mathbb{R}^{n_x \times k}, V \in \mathbb{R}^{n_y \times k}$
- Examples: PLSA (Hoffmann, 2001), MMMF (Srebro et al, 2004)
- Numerical and statistical efficiency

![Matrix Representation]

\[ Z = UV^\top \]
Choose a **convex loss function** $\ell(z, z')$ (hinge, square, etc...)

Relax the (non-convex) rank of $Z$ into the (convex) trace norm of $Z$: if $\sigma_i(Z)$ are the singular values of $Z$,

$$\text{rank} Z = \sum_i 1_{\sigma_i(Z) > 0} \quad \|Z\|_\ast = \sum_i \sigma_i(Z) .$$

$n$ observations $z_u$ corresponding to $x_{i(u)}$ and $y_{j(u)}$, $u = 1, \ldots, n$:

$$\min_{Z \in \mathbb{R}^{nx \times ny}} \sum_{u=1}^{n} \ell(z_u, Z_{i(u),j(u)}) + \lambda \|Z\|_\ast$$

This is an SDP if $\ell$ is SDP-representable
CF with attributes

The problem

- Often we have **additional attributes**: 
  - gender, age of people; type, actors of movies.. 
  - 3D structures of proteins and ligands for protein-ligand interaction prediction

- **How to include attributes in CF?**

- **Expected gains**: increase **performance**, allow predictions on **new movie and/or people**.

Our contributions

- **A general framework** for CF with or without attributes, using **kernels** to describe attributes (“kernel-CF”)

- **A family of algorithms** for CF in this setting
The problem

- Often we have additional attributes:
  - gender, age of people; type, actors of movies.
  - 3D structures of proteins and ligands for protein-ligand interaction prediction

- How to include attributes in CF?

- Expected gains: increase performance, allow predictions on new movie and/or people.

Our contributions

- A general framework for CF with or without attributes, using kernels to describe attributes ("kernel-CF")
- A family of algorithms for CF in this setting
Movies: points in a Hilbert space $\mathcal{X}$
Customers: points in a Hilbert space $\mathcal{Y}$
We model the preference of customer $y$ for a movie $x$ by a bilinear form:
\[
f(x, y) = \langle x, Fy \rangle_{\mathcal{X}},
\]
where $F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X})$ is a compact linear operator (i.e., a "matrix").
Any compact operator $F : \mathcal{Y} \rightarrow \mathcal{X}$ admits a spectral decomposition:

$$F = \sum_{i=1}^{\infty} \sigma_i u_i \otimes v_i.$$ 

where the $\sigma_i \geq 0$ are the **singular values** and $(u_i)_{i \in \mathbb{N}}$ and $(v_i)_{i \in \mathbb{N}}$ are orthonormal families in $\mathcal{X}$ and $\mathcal{Y}$.

The **spectrum of $F$** is the set of singular values sorted in decreasing order: $\sigma_1(F) \geq \sigma_2(F) \geq \ldots \geq 0$.

This is the natural generalization of singular values for matrices.
A function \( \Omega : B_0 (\mathcal{Y}, \mathcal{X}) \mapsto \mathbb{R} \cup \{+\infty\} \) is called a spectral penalty function if it can be written as:

\[
\Omega(F) = \sum_{i=1}^{\infty} s_i (\sigma_i(F)),
\]

where for any \( i \geq 1 \), \( s_i : \mathbb{R}^+ \mapsto \mathbb{R}^+ \cup \{+\infty\} \) is a non-decreasing penalty function satisfying \( s_i(0) = 0 \).
Examples

- **Rank constraint**: take $s_{k+1}(0) = 0$ and $s_{k+1}(u) = +\infty$ for $u > 0$, and $s_i = 0$ for $i \geq k$. Then

$$\Omega(F) = \begin{cases} 0 & \text{if } \text{rank}(F) \leq k, \\ +\infty & \text{if } \text{rank}(F) > k. \end{cases}$$

- **Trace norm**: take $s_i(u) = u$ for all $i$, then:

$$\Omega(F) = \| F \|_*.$$

- **Hilbert-Schmidt norm**: take $s_i(u) = u^2$ for all $i$, then

$$\Omega(F) = \| F \|_{\text{Fro}}^2.$$
Spectral penalty function

Examples

- **Rank constraint**: take $s_{k+1}(0) = 0$ and $s_{k+1}(u) = +\infty$ for $u > 0$, and $s_i = 0$ for $i \geq k$. Then

$$
\Omega(F) = \begin{cases} 
0 & \text{if } \text{rank}(F) \leq k, \\
+\infty & \text{if } \text{rank}(F) > k. 
\end{cases}
$$

- **Trace norm**: take $s_i(u) = u$ for all $i$, then:

$$
\Omega(F) = \| F \|_*.
$$

- **Hilbert-Schmidt norm**: take $s_i(u) = u^2$ for all $i$, then

$$
\Omega(F) = \| F \|_{\text{Fro}}^2.
$$
Examples

- **Rank constraint**: take \( s_{k+1}(0) = 0 \) and \( s_{k+1}(u) = +\infty \) for \( u > 0 \), and \( s_i = 0 \) for \( i \geq k \). Then

\[
\Omega(F) = \begin{cases} 
0 & \text{if } \text{rank}(F) \leq k, \\
+\infty & \text{if } \text{rank}(F) > k.
\end{cases}
\]

- **Trace norm**: take \( s_i(u) = u \) for all \( i \), then:

\[
\Omega(F) = \| F \|_*.
\]

- **Hilbert-Schmidt norm**: take \( s_i(u) = u^2 \) for all \( i \), then

\[
\Omega(F) = \| F \|_{\text{Fro}}^2.
\]
# Learning operator with spectral regularization

## Setting

- **Training set**: \((x_i, y_i, t_i)_{i=1,\ldots,N}\) a set of (movie, people, preference).
- **Loss function** \(l(t, t')\): cost of predicting preference \(t\) instead of \(t'\).
- **Empirical risk** of an operator \(F\):

\[
R_N(F) = \frac{1}{N} \sum_{i=1}^{N} l(\langle x_i, Fy_i \rangle_{\mathcal{X}}, t_i) .
\]

## Learning an operator

\[
\min_{F \in \mathcal{B}_0(Y, X), \Omega(F) < \infty} \left\{ R_N(F) + \lambda \Omega(F) \right\} .
\]
Learning operator with spectral regularization

Setting

- Training set: \((x_i, y_i, t_i)_{i=1,...,N}\) a set of (movie, people, preference).
- Loss function \(l(t, t')\): cost of predicting preference \(t\) instead of \(t'\).
- Empirical risk of an operator \(F\):

\[
R_N(F) = \frac{1}{N} \sum_{i=1}^{N} l(\langle x_i, Fy_i \rangle, t_i).
\]

Learning an operator

\[
\min_{F \in \mathcal{B}_0(\mathcal{Y},\mathcal{X}), \Omega(F)<\infty} \left\{ R_N(F) + \lambda \Omega(F) \right\}.
\]
A classical representer theorem

Theorem

If $\hat{F}$ is a solution to the problem:

$$\min_{F \in B_2(\mathcal{Y},\mathcal{X})} \left\{ R_N(F) + \lambda \sum_{i=1}^{\infty} \sigma_i(F)^2 \right\},$$

then it is necessarily in the linear span of $\{x_i \otimes y_i : i = 1, \ldots, N\}$, i.e., it can be written as:

$$\hat{F} = \sum_{i=1}^{N} \alpha_i x_i \otimes y_i,$$

for some $\alpha \in \mathbb{R}^N$.

Proof

This is just the classical representer theorem for tensor product kernels.
A generalized representer theorem

Theorem

For any spectral penalty function $\Omega : \mathcal{B}_0(\mathcal{Y}, \mathcal{X}) \mapsto \mathbb{R}$, let the optimization problem:

$$\min_{F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X}), \Omega(F) < \infty} \left\{ R_N(F) + \lambda \Omega(F) \right\}.$$

If the set of solutions is not empty, then there is a solution $F$ in $\mathcal{X}_N \otimes \mathcal{Y}_N$, i.e., there exists $\alpha \in \mathbb{R}^{m_X \times m_Y}$ such that:

$$F = \sum_{i=1}^{m_X} \sum_{j=1}^{m_Y} \alpha_{ij} u_i \otimes v_j,$$

where $(u_1, \ldots, u_{m_X})$ and $(v_1, \ldots, v_{m_Y})$ form orthonormal bases of $\mathcal{X}_N$ and $\mathcal{Y}_N$, respectively.
The coefficients $\alpha$ that define the solution by

$$F = \sum_{i=1}^{m_X} \sum_{j=1}^{m_Y} \alpha_{ij} u_i \otimes v_j,$$

can be found by solving the following finite-dimensional optimization problem:

$$\min_{\alpha \in \mathbb{R}^{m_X \times m_Y}, \Omega(\alpha) < \infty} R_N \left( \text{diag} \left( X \alpha Y^\top \right) \right) + \lambda \Omega(\alpha),$$

where $\Omega(\alpha)$ refers to the spectral penalty function applied to the matrix $\alpha$ seen as an operator from $\mathbb{R}^{m_Y}$ to $\mathbb{R}^{m_X}$, and $X$ and $Y$ denote any matrices that satisfy $K = XX^\top$ and $G = YY^\top$ for the two Gram matrices $K$ and $G$ of $\mathcal{X}_N$ and $\mathcal{Y}_N$. 
We obtain various algorithms by choosing:

1. A loss function (depends on the application)
2. A spectral regularization (that is amenable to optimization)
3. Two kernels.

Both kernels and spectral regularization can be used to constrain the solution.
Examples

- Dirac kernel + spectral constraint (rank, trace norm) = matrix completion
- Attribute kernels + Hilbert-Schmidt regularization = kernel methods for pairs with tensor product kernel
- Attribute kernel on movies, Dirac on people, spectral regularization (rank, trace norm) = multi-task learning (rank constraints enforces sharing the weights between people).
A family of kernels

Taken $K_{\otimes} = K \times G$ with

$$\begin{cases} 
K = \eta K_{\text{Attribute}}^x + (1 - \eta) K_{\text{Dirac}}^x, \\
G = \zeta K_{\text{Attribute}}^y + (1 - \zeta) K_{\text{Dirac}}^y,
\end{cases}$$

for $0 \leq \eta \leq 1$ and $0 \leq \zeta \leq 1$
Simulated data

Experiment

- Generate data \((x, y, z) \in \mathbb{R}^{f_X} \times \mathbb{R}^{f_Y} \times \mathbb{R}\) according to
  \[
  z = x^T By + \varepsilon
  \]
- Observe only \(n_X < f_X\) and \(n_Y < f_Y\) features
  - Low-rank assumption will find the missing features
  - Observed attributes will help the low-rank formulation to concentrate mostly on the unknown features
- Comparison of
  - Low-rank constraint without tracenorm (note that it requires regularization)
  - Trace-norm formulation (regularization is implicit)
Simulated data: results

- Compare MSE
- Left: rank constraint (best: 0.1540), right: trace norm (best: 0.1522)
- MovieLens 100k database, ratings with attributes
- Experiments with 943 movies and 1,642 people, 100,000 rankings in \( \{1, \ldots, 5\} \)
- Train on a subset of the ratings, test on the rest
- Error measured with MSE (best constant prediction: 1.26)
Conclusion

What we saw

- A general framework for CF with or without attributes
- A generalized representation theorem valid for any spectral penalty function
- A family of new methods;

Future work

- The bottleneck is often practical optimization. Online version possible.
- Automatic kernel optimization

Reference