Collaborative filtering in Hilbert spaces with spectral regularization

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

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

Collaborative Filtering (CF)

The problem

- Given a set of $n_x$ “movies” $x \in X$ and a set of $n_y$ “customers” $y \in Y$,
- predict the “rating” $z(x, y) \in Z$ of customer $y$ for movie $x$,
- Training data: large $n_x \times n_y$ incomplete matrix $Z$ that describes the known ratings of some customers for some movies,
- Goal: complete the matrix.
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**?
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
Fitting low-rank models (Srebro et al, 2004)

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

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

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

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

where $\ell(z, z')$ is a convex loss function.

- This is an SDP if $\ell$ is SDP-representable.
Fitting low-rank models (Srebro et al, 2004)

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

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

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

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

where $\ell(z, z')$ is a convex loss function.

- This is an SDP if $\ell$ is SDP-representable
The problem

- Often we have **additional attributes**:
  - gender, age of customers; 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 customers.

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
CF with attributes

The problem

- Often we have **additional attributes**:
  - gender, age of customers; 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 **customers**.

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 idea

Basic facts

- $n_X$ movies and $n_Y$ customers
- The known rating $z(x_i, y_j)$ of customer $y_j$ for movie $x_i$ is stored in the $(i, j)$-th entry of a matrix $M$ (of size $n_X \times n_Y$).
- $M$ represents a linear application / bilinear form:

$$ M : \mathbb{R}^{n_Y} \rightarrow \mathbb{R}^{n_X} $$

defined by:

$$ e_i^\top M f_j = M_{i,j} $$

- Rank / trace norm are spectral properties of the linear application
Reformulations

- Represent the $i$-th movie $\mathbf{x}_i \in \mathcal{X}$ (resp. $j$-th customer $\mathbf{y}_j \in \mathcal{Y}$) by the $i$-th basis vector $\mathbf{e}_i \in \mathbb{R}^{n_X}$ (resp. $\mathbf{f}_j \in \mathbb{R}^{n_Y}$):

$$\phi_X(\mathbf{x}_i) = \mathbf{e}_i, \quad \phi_Y(\mathbf{y}_j) = \mathbf{f}_j.$$

- Approximate the rating function by a bilinear form:

$$\forall (\mathbf{x}_i, \mathbf{y}_j) \in \mathcal{X} \times \mathcal{Y}, \quad G_M(\mathbf{x}_i, \mathbf{y}_j) = \phi_X(\mathbf{x}_i)^\top \mathbf{M} \phi_Y(\mathbf{y}_j),$$

by constraining a spectral property of $\mathbf{M} : \mathbb{R}^{n_X} \mapsto \mathbb{R}^{n_X}$.

An idea

If we have additional attributes about movies / customer, why not include them in $\phi_X(\mathbf{x})$ and $\phi_Y(\mathbf{y})$?
The idea

Reformulations

- Represent the $i$-th movie $x_i \in X$ (resp. $j$-th customer $y_j \in Y$) by the $i$-th basis vector $e_i \in \mathbb{R}^{n_X}$ (resp. $f_j \in \mathbb{R}^{n_Y}$):

$$\phi_X(x_i) = e_i, \quad \phi_Y(y_j) = f_j.$$

- Approximate the rating function by a bilinear form:

$$\forall (x_i, y_j) \in X \times Y, \quad G_M(x_i, y_j) = \phi_X(x_i)^\top M \phi_Y(y_j),$$

by constraining a spectral property of $M : \mathbb{R}^{n_X} \mapsto \mathbb{R}^{n_X}$.

An idea

If we have additional attributes about movies / customer, why not include them in $\phi_X(x)$ and $\phi_Y(y)$?
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”).
Classical results

- For \((x, y)\) in \(\mathcal{X} \times \mathcal{Y}\) the tensor product \(x \otimes y\) is the operator

\[
\forall h \in \mathcal{Y}, \quad (x \otimes y) h = \langle y, h \rangle_{\mathcal{Y}} x.
\]

- Any compact operator \(F : \mathcal{Y} \to \mathcal{X}\) 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.
Spectra of compact operators

Classical results

- For \((x, y)\) in \(\mathcal{X} \times \mathcal{Y}\) the tensor product \(x \otimes y\) is the operator
  \[
  \forall h \in \mathcal{Y}, \quad (x \otimes y)h = \langle y, h \rangle_{\mathcal{Y}}x.
  \]

- 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.

JP Vert (ParisTech)
Spectra of compact operators

Classical results

- For \((x, y)\) in \(\mathcal{X} \times \mathcal{Y}\) the tensor product \(x \otimes y\) is the operator
  \[\forall h \in \mathcal{Y}, \quad (x \otimes y)h = \langle y, h \rangle_{\mathcal{Y}} x.\]

- Any compact operator \(F : \mathcal{Y} \to \mathcal{X}\) 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.
Useful classes for operators

Operators of finite rank

- The rank of an operator is the number of strictly positive singular values.
- Hence operators of rank smaller or equal to \( k \) are characterized by:
  \[
  \sigma_{k+1}(F) = 0. 
  \]

Trace-class operators

The trace-class operators are the compact operators \( F \) that satisfy:

\[
\| F \|_* := \sum_{i=1}^{\infty} \sigma_i(F) < \infty. 
\]

\( \| F \|_* \) is a norm over the trace-class operators, called the trace norm.
Useful classes for operators

Operators of finite rank

- The **rank** of an operator is the number of strictly positive singular values.
- Hence operators of rank smaller or equal to $k$ are characterized by:

\[ \sigma_{k+1}(F) = 0. \]

Trace-class operators

The **trace-class** operators are the compact operators $F$ that satisfy:

\[ \| F \|_* := \sum_{i=1}^{\infty} \sigma_i(F) < \infty. \]

\( \| F \|_* \) is a norm over the trace-class operators, called the **trace norm**.
Hilbert-Schmidt operators

The **Hilbert-Schmidt operators** are compact operators $F$ that satisfy:

$$\| F \|^2_{\text{Fro}} := \sum_{i=1}^{\infty} \sigma_i(F)^2 < \infty.$$ 

They form a **Hilbert space** with inner product:

$$\langle x \otimes y, x' \otimes y' \rangle_{X \otimes Y} = \langle x, x' \rangle_X \langle y, y' \rangle_Y.$$
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
  \[
  \Omega(F) = \| F \|_{\text{Fro}}^2.
  \]
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

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

### Setting
- **Training set**: \((x_i, y_i, t_i)_{i=1,\ldots,N}\) a set of (movie, customer, 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

\[
\min_{F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X}), \Omega(F) < \infty} \{ R_N(F) + \lambda \Omega(F) \}.
\]
**Setting**

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

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

**Learning an operator**

\[
\min_{F \in B_0(\mathcal{Y},\mathcal{X}), \Omega(F) < \infty} \left\{ R_N(F) + \lambda \Omega(F) \right\}.
\]
Questions

Theory

Is it a "good" algorithm in theory?
- To be investigated...
- See Srebro et al. (2004), Bach (2007) for preliminary results with the trace norm

Practice

Can we implement it? Does it work on real data?
- Optimization problem in the space of compact operators... but we show later that it boils down to a finite-dimensional optimization problem
- Promising results on real data
Questions

Theory

Is it a "good" algorithm in theory?
- To be investigated...
- See Srebro et al. (2004), Bach (2007) for preliminary results with the trace norm

Practice

Can we implement it? Does it work on real data?
- Optimization problem in the space of compact operators... but we show later that it boils down to a finite-dimensional optimization problem
- Promising results on real data
Questions

Theory

Is it a "good" algorithm in theory?
- To be investigated...
- See Srebro et al. (2004), Bach (2007) for preliminary results with the trace norm

Practice

Can we implement it? Does it work on real data?
- Optimization problem in the space of compact operators... but we show later that it boils down to a finite-dimensional optimization problem
- Promising results on real data
Questions

Theory

Is it a "good" algorithm in theory?
- To be investigated...
- See Srebro et al. (2004), Bach (2007) for preliminary results with the trace norm

Practice

Can we implement it? Does it work on real data?
- Optimization problem in the space of compact operators... but we show later that it boils down to a finite-dimensional optimization problem
- Promising results on real data
A classical representer theorem

Theorem

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

$$
\min_{F \in \mathcal{B}_2(\mathcal{Y},\mathcal{X})} \left\{ R_N(F) + \lambda \sum_{i=1}^{\infty} \sigma_i(F)^2 \right\},
$$

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

JP Vert (ParisTech)
Collaborative filtering
STAR 2009
Proof sketch

- $B_2(\mathcal{Y}, \mathcal{X})$ is isomorphic to the RKHS of the tensor product kernel:

$$k_\otimes((x, y), (x', y')) = \langle x, x' \rangle_\mathcal{X} \langle y, y' \rangle_\mathcal{Y},$$

by $f(x, y) = \langle x, F y \rangle_\mathcal{X}$. In particular,

$$\| f \|_{\mathcal{H}_\otimes}^2 = \| F \|^2 = \Omega(F).$$

- The problem is therefore a classical kernel method:

$$\min_{f \in \mathcal{H}_\otimes} \left\{ R_N(f) + \lambda \| f \|_{\otimes}^2 \right\},$$

so the classical representer theorem can be used. \qed
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.
Proof sketch

For any operator $F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X})$, let

$$G = \Pi_{\mathcal{X}_N} F \Pi_{\mathcal{Y}_N},$$

where $\Pi_U$ is the orthogonal projection onto $U$.

Lemma: we can show that for all $i \geq 0$:

$$\sigma_i(G) \leq \sigma_i(F).$$

Therefore $\Omega(G) \leq \Omega(F)$.

On the other hand $R_N(G) = R_N(F)$.

Consequently for any solution $F$ we have another solution $G \in \mathcal{X}_N \otimes \mathcal{Y}_N$. □
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:

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

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 $K = XX^T$ and $G = YY^T$ for the two Gram matrices $K$ and $G$ of $\mathcal{X}_N$ and $\mathcal{Y}_N$. 

JP Vert (ParisTech)  Collaborative filtering  STAR 2009 21 / 27
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 Gram matrices (aka kernel matrices)

Both kernels and spectral regularization can be used to constrain the solution
A family of kernels

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

\[
\begin{align*}
K &= \eta K^x_{\text{Attribute}} + (1 - \eta) K^x_{\text{Dirac}}, \\
G &= \zeta K^y_{\text{Attribute}} + (1 - \zeta) K^y_{\text{Dirac}},
\end{align*}
\]

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
  \[
  z = x^\top B y + \varepsilon
  \]

- 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)
Movies

- MovieLens 100k database, ratings with attributes
- Experiments with 943 movies and 1,642 customers, 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 choice of the kernel

Reference