Collaborative filtering with attributes

Jacob Abernethy[1] Francis Bach[2]
Theodoros Evgeniou[3] Jean-Philippe Vert[4]

[1] UC Berkeley

[2] INRIA / Ecole normale superieure de Paris

[3] INSEAD

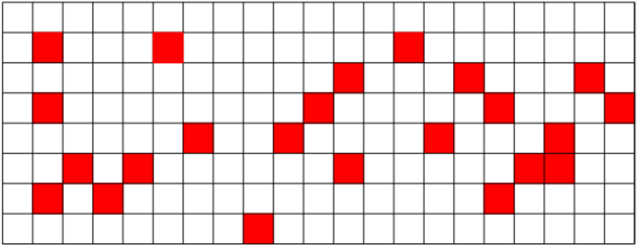
[4] ParisTech / Institut Curie / INSERM

Xerox Research Center Europe, Grenoble, France, May 23, 2008.

Collaborative Filtering (CF)

The problem

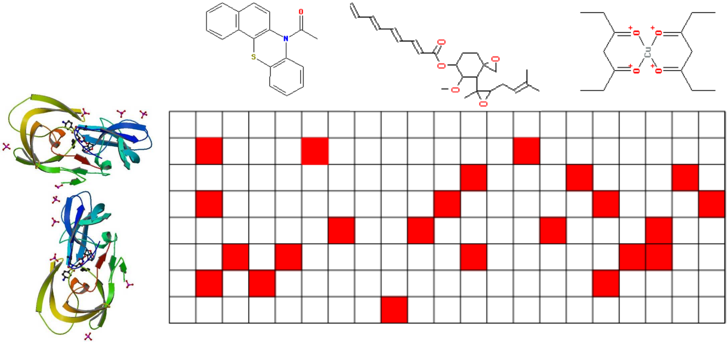
- Given a set of $n_{\mathcal{X}}$ "movies" $\mathbf{x} \in \mathcal{X}$ and a set of $n_{\mathcal{Y}}$ "customers" $\mathbf{y} \in \mathcal{Y}$,
- predict the "rating" $z(\mathbf{x}, \mathbf{y}) \in \mathcal{Z}$ of customer $\mathbf{y}$ for movie $\mathbf{x}$
- Training data: large $n_{\mathcal{X}}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



Another CF example

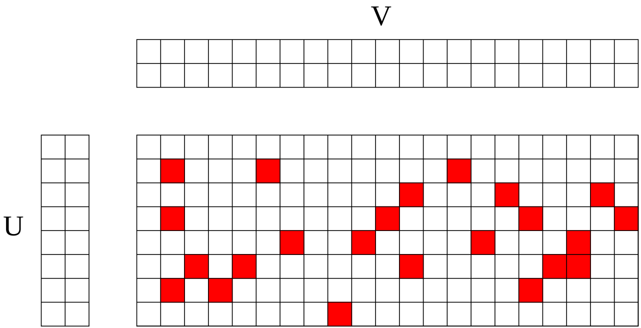
Drug design

- 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?



CF by low-rank matrix approximation

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



CF by low-rank matrix approximation example

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,

$$\mathrm{rank} Z = \sum_i \mathbf{1}_{\sigma_i(Z)>0} \qquad \|Z\|_* = \sum_i \sigma_i(Z) .$$

- n observations z_u corresponding to $\mathbf{x}_{i(u)}$ and $\mathbf{y}_{j(u)}$, $u = 1, \ldots, n$:

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

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

- This is an SDP if ℓ is SDP-representable

CF by low-rank matrix approximation example

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,

$$\mathrm{rank} Z = \sum_i 1_{\sigma_i(Z)>0} \qquad \|Z\|_* = \sum_i \sigma_i(Z)\,.$$

- n observations z_u corresponding to $\mathbf{x}_{i(u)}$ and $\mathbf{y}_{j(u)}$, $u = 1, \ldots, n$:

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

  where $\ell(z, z')$ is a convex loss function.
- This is an SDP if ℓ is SDP-representable

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

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

- $n_\mathcal{X}$ movies and $n_\mathcal{Y}$ customers
- The known rating $z(i,j)$ of customer j for movie i is stored in the (i,j)-th entry of a matrix M (of size $n_\mathcal{X} \times n_\mathcal{Y}$).
- M represents a linear application / bilinear form:

$$M : \mathbb{R}^{n_\mathcal{Y}} \to \mathbb{R}^{n_\mathcal{X}}$$

  defined by:

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

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

The idea

Reformulations

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

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

- Approximate the rating function by a bilinear form:

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

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

An idea

If we have additional attributes about movies / customer, why not include them in $\phi(x)$ and $\phi(y)$?

The idea

Reformulations

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

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

- Approximate the rating function by a bilinear form:

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

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

An idea

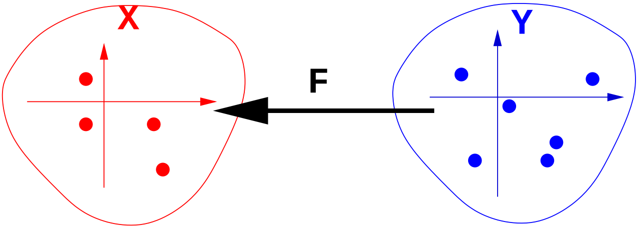
If we have additional attributes about movies / customer, why not include them in $\phi(x)$ and $\phi(y)$?

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:

$$f(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, F\mathbf{y} \rangle_{\mathcal{X}} \ ,$$

where $F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X})$ is a compact linear operator (i.e., a "matrix").



Spectra of compact operators

Classical results

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

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

- Any compact operator $F : \mathcal{Y} \to \mathcal{X}$ admits a spectral decomposition:

$$F = \sum_{i=1}^{\infty} \sigma_i \mathbf{u}_i \otimes \mathbf{v}_i\,.$$

  where the $\sigma_i \geq 0$ are the singular values and $(\mathbf{u}_i)_{i \in \mathbb{N}}$ and $(\mathbf{v}_i)_{i \in \mathbb{N}}$ 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.

Useful classes for operators (cont.)

Hilbert-Schmidt operators

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

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

- They form a Hilbert space with inner product:

$$\left\langle \mathbf{x} \otimes \mathbf{y}, \mathbf{x}' \otimes \mathbf{y}' \right\rangle_{\mathcal{X} \otimes \mathcal{Y}} = \left\langle \mathbf{x}, \mathbf{x}' \right\rangle_{\mathcal{X}} \left\langle \mathbf{y}, \mathbf{y}' \right\rangle_{\mathcal{Y}} \,.$$

Spectral penalty function

Definition

A function $\Omega : \mathcal{B}_0(\mathcal{Y}, \mathcal{X}) \mapsto \mathbb{R} \cup \{+\infty\}$ 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 $s_i(0) = 0$.

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

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

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

$$\Omega(F) = \parallel F \parallel_* .$$

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

$$\Omega(F) = \parallel F \parallel_{Fro}^2 .$$

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

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

- 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

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

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

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

- 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

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

Learning operator with spectral regularization

Setting

- Training set: $(\mathbf{x}_i, \mathbf{y}_i, t_i)_{i=1,\dots,N}$ 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:

$$R_N(F) = \frac{1}{N} \sum_{i=1}^{N} l\left(\langle \mathbf{x}_i, F\mathbf{y}_i \rangle_{\mathcal{X}}, t_i\right) .$$

Learning an operator

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

Learning operator with spectral regularization

Setting

- Training set: $(\mathbf{x}_i, \mathbf{y}_i, t_i)_{i=1,\ldots,N}$ 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:

$$R_N(F) = \frac{1}{N} \sum_{i=1}^{N} l\left(\langle \mathbf{x}_i, F\mathbf{y}_i \rangle_{\mathcal{X}}, t_i\right) .$$

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 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 $\{\mathbf{x}_i \otimes \mathbf{y}_i \,:\, i = 1, \ldots, N\}$, i.e., it can be written as:

$$\hat{F} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \otimes \mathbf{y}_i \,,$$

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

Proof sketch

- $\mathcal{B}_2(\mathcal{Y}, \mathcal{X})$ is isomorphic to the RKHS of the tensor product kernel:

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

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

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

- The problem is therefore a classical kernel method:

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

  so the classical representer theorem can be used. $\square$

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 $\alpha \in \mathbb{R}^{m_\mathcal{X} \times m_\mathcal{Y}}$ such that:

$$F = \sum_{i=1}^{m_\mathcal{X}} \sum_{j=1}^{m_\mathcal{Y}} \alpha_{ij} \mathbf{u}_i \otimes \mathbf{v}_j ,$$

where $(\mathbf{u}_1, \ldots, \mathbf{u}_{m_\mathcal{X}})$ and $\left(\mathbf{v}_1, \ldots, \mathbf{v}_{m_\mathcal{Y}}\right)$ 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 Π_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 $G \in \mathcal{X}_N \otimes \mathcal{Y}_N$. $\square$

Practical consequence

Theorem (cont.)

The coefficients α that define the solution by

$$F = \sum_{i=1}^{m_{\mathcal{X}}} \sum_{j=1}^{m_{\mathcal{Y}}} \alpha_{ij} \mathbf{u}_i \otimes \mathbf{v}_j \,,$$

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

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

where $\Omega(\alpha)$ refers to the spectral penalty function applied to the matrix α seen as an operator from $\mathbb{R}^{m_{\mathcal{Y}}}$ to $\mathbb{R}^{m_{\mathcal{X}}}$, 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 $\mathcal{Y}_N$.

Summary

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 customers, spectral regularization (rank, trace norm) = multi-task learning (rank constraints enforces sharing the weights between customers).

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 customers, spectral regularization (rank, trace norm) = multi-task learning (rank constraints enforces sharing the weights between customers).

Examples

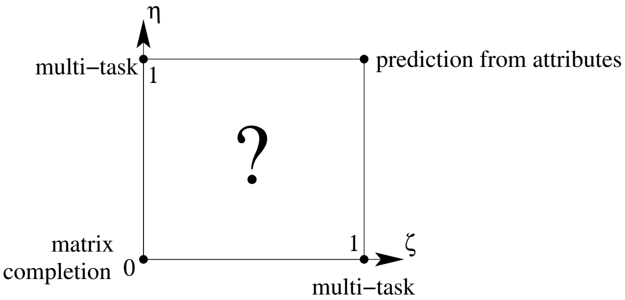
- Dirac kernel + spectral constraint (rank, trace norm) = matrix completion
- Attribute kernels + Hilbert-Schmidt regularization = kernel methods for pairs with tensor product kernel
- Attribute kernel on movies, Dirac on customers, spectral regularization (rank, trace norm) = multi-task learning (rank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

A family of kernels

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

$$\begin{cases} K = \eta K^x_{Attribute} + (1-\eta) K^x_{Dirac}, \\ G = \zeta K^y_{Attribute} + (1-\zeta) K^y_{Dirac}, \end{cases}$$

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



Simulated data

Experiment

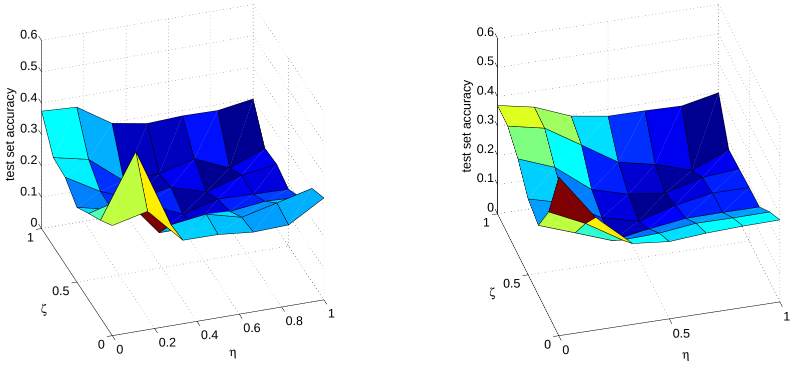
- Generate data $(\mathbf{x}, \mathbf{y}, z) \in \mathbb{R}^{f_X} \times \mathbb{R}^{f_Y} \times \mathbb{R}$ according to

$$z = \mathbf{x}^\top B\mathbf{y} + \varepsilon$$

- 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)

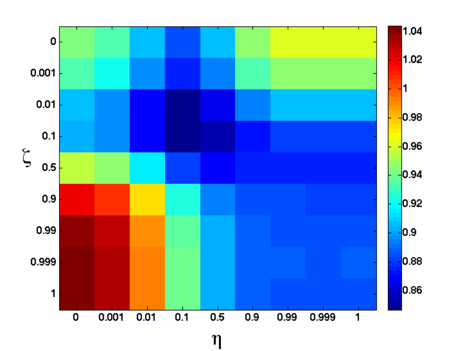
Simulated data: results

- 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 kernel optimization

Reference

J. Abernethy, F. Bach, T. Evgeniou and J.-P. Vert, “A new approach to collaborative filtering: operator estimation with spectral constraint”, *technical report arXiv 0802-1430*, 2008.

Annex: algorithms

The problem

- Let $\psi_i(t) = l(t, t_i)$, supposed to be convex.
- Suppose
$$\Omega(A) = \sum_{i \geq 1} s(\sigma_i(A)),$$
where s is a convex even function s.t. $s(0) = 0$.
- The problem we wish to solve is:
$$\min_{\alpha \in \mathbb{R}^{m_X \times m_Y}} \sum_{i=1}^{n} \psi_i((X\alpha Y^\top)_{ii}) + \lambda \Omega(\alpha)$$
- One may directly solve this primal problem.

Fenchel conjugacy

- Let ψ_i^* denote the Fenchel conjugate of ϕ_i:

$$\psi_i^*(\alpha_i) = \max_{v_i \in \mathbb{R}} \alpha_i v_i - \psi_i(v_i) \,.$$

- Let Ω^* denote the Fenchel conjugate of Ω:

$$\Omega^*(\beta) = \max_{\in \mathbb{R}^{m_x \times m_y}} Tr(\alpha^\top \beta) - \Omega(\alpha) \,.$$

- In fact Ω^* is a spectral function corresponding to s^*:

$$\Omega^*(\beta) = \sum_{i \geq 1} s^*(\sigma_i(\beta)) \,.$$

Dual problem

- Primal:

$$\min_{\alpha \in \mathbb{R}^{m_x \times m_y}} \sum_{i=1}^{N} \psi_i((X\alpha Y^\top)_{ii}) + \lambda \Omega(\alpha)$$

- Dual (strong duality):

$$\max_{\beta \in \mathbb{R}^N} - \sum_{i=1}^{N} \psi_i^*(\beta_i) - \lambda \Omega^* \left(-\frac{1}{\lambda} X^\top \mathrm{Diag}(\beta) Y \right) .$$

- The solution α of the primal is among the Fenchel duals of $-\frac{1}{\lambda} X^\top \mathrm{Diag}(\beta) Y$ (closed form if s if differentiable).
- Choosing the primal or dual formulation depends on the number of training patterns N compared to $m_x \times m_y$.

Example: trace norm constraint

- Primal:

$$\min_{\alpha \in \mathbb{R}^{m_X \times m_Y}} \sum_{i=1}^{N} \psi_i((X \alpha Y^\top)_{ii}) + \lambda \| \alpha \|_*$$

- Large convex, non-smooth problem (can be cast as a SDP).
- Dual:

$$\max_{\beta \in \mathbb{R}^N} - \sum_{i=1}^{N} \psi_i^*(\beta_i) \text{ such that } \max_i \sigma_i \left(-X^\top \mathrm{Diag}(\beta) Y \right) \leq \lambda \,.$$

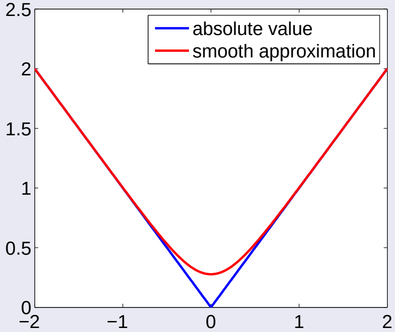
- Two tricks to (approximately) solve this problem:
  - Make it smooth
  - Make it low-rank

Smoothing the problem

Make the problem smooth by approximating the non smooth functions:

- loss: (depends on the loss)
- trace norm:

$$f_\varepsilon(b) = \varepsilon \log(1 + e^{\sigma/\varepsilon}) + \varepsilon \log(1 + e^{-\sigma/\varepsilon}) \,.$$



Making the problem low-rank

Trick

- Let $G(M)$ be a convex twice differentiable function to optimize over $\mathbb{R}^{p \times q}$.
- If the global minimum of G has rank r, then G restricted to matrices of rank $r+1$ have no local minimum apart from the global minimum.

Algorithm

1. Start with small r.
2. Find local minimum with Quasi-Newton.
3. If solution is rank-defficient then we have the global optimum; otherwise increase r and start again in 2.

Making the problem low-rank

Trick

- Let $G(M)$ be a convex twice differentiable function to optimize over $\mathbb{R}^{p \times q}$.
- If the global minimum of G has rank r, then G restricted to matrices of rank $r+1$ have no local minimum apart from the global minimum.

Algorithm

1. Start with small r.
2. Find local minimum with Quasi-Newton.
3. If solution is rank-defficient then we have the global optimum; otherwise increase r and start again in 2.