

Output Kernel Learning Methods

Francesco Dinuzzo ¹ Cheng Soon Ong ² Kenji Fukumizu ³

¹Max Planck Institute for Intelligent Systems, Tübingen, Germany

²NICTA, Melbourne

³Institute of Statistical Mathematics, Japan



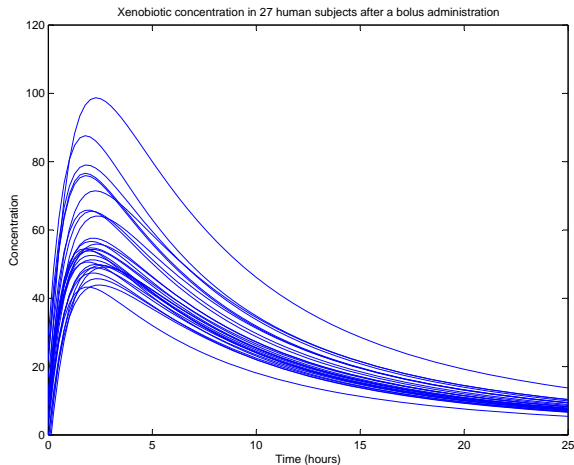
MAX-PLANCK-GESELLSCHAFT



Part I

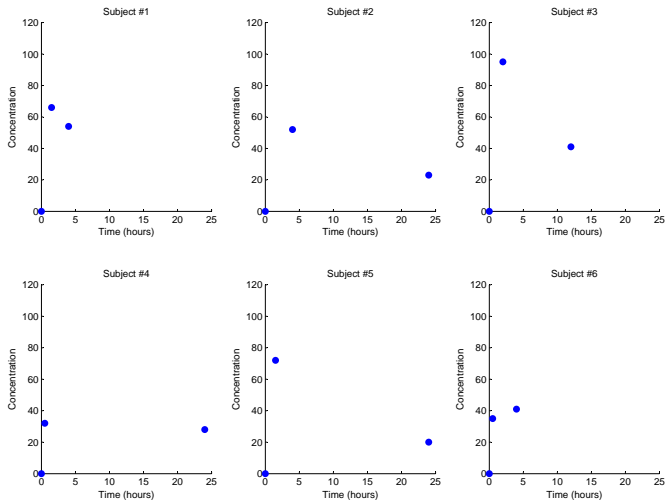
Learning multiple tasks and their relationships

Multiple regression: population pharmacokinetics



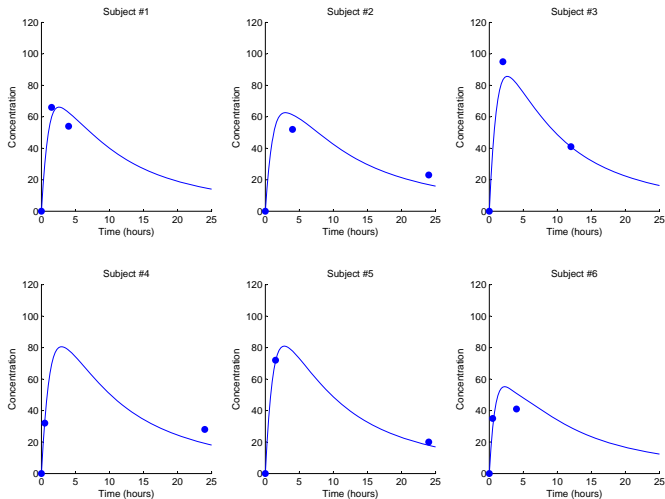
- The response curves have similar shapes.
- However, there is macroscopic inter-individual variability.

Multiple regression: population pharmacokinetics



Few data points per subject with sparse sampling.

Multiple regression: population pharmacokinetics



Can we combine the datasets to better estimate all the curves?

Collaborative filtering and recommender systems

- **Data:** collections of ratings assigned by several users to a set of items.
- **Problem:** estimate the preferences of the every user for all the items.

The screenshot shows the MovieLens website interface. At the top left is the logo "movielens" with the tagline "helping you find the right movies". To the right, a "Welcome" message says "You've rated 10 movies. You're the 17th visitor in the past hour." and includes a "(Log Out)" link. A legend on the right explains the star ratings: 5 stars = Must See, 4 stars = Will Enjoy, 3 stars = It's OK, 2 stars = Fairly Bad, 1 star = Awful.

Below the welcome message, it says "So far you have rated 10 movies. MovieLens needs at least 15 ratings from you to generate predictions for you. Please rate as many movies as you can from the list below." and a "next >" link.

Your Rating	Movie Information
★★★★ 4.0 stars	Last of the Mohicans, The (1992) Action, Romance, War, Western
??? Not seen	Splash (1984) Comedy, Fantasy, Romance
★★½ 2.5 stars	Three Kings (1999) Action, Adventure, Comedy, Drama, War
★★★★ 4.0 stars	Fatal Attraction (1987) Drama, Thriller
??? Not seen	Time to Kill, A (1996) Drama, Thriller
★½ 1.5 stars	First Knight (1995) Action, Drama, Romance

- Preference profiles are different for every user.
- However, similar users have similar preferences.

Collaborative filtering and recommender systems

Additional information:

- Data about the items.
- Data about the users (e.g. gender, age, occupation).
- Data about the ratings themselves (e.g. timestamp, tags).



The screenshot shows the IMDb page for the movie 'Der letzte Mohikaner' (1992). The page includes the IMDb logo, a search bar, and navigation tabs for Movies, TV, News, Trailers, Community, and IMDbPro. The movie's title is displayed in German, along with its original title 'The Last of the Mohicans'. The page features a star rating of 7.8/10 based on 66,510 user ratings and 336 reviews. The plot summary, director (Michael Mann), writers (James Fenimore Cooper, John L. Balderston), and stars (Daniel Day-Lewis, Madeleine Stowe, Russell Crowe) are also visible.

IMDb Find Movies, TV shows, Celebrities and more... All 

Movies ▾ **TV** ▾ **News** ▾ **Trailers** ▾ **Community** ▾ **IMDbPro** ▾

DANIEL DAY LEWIS

Der letzte Mohikaner (1992)
The Last of the Mohicans (*original title*)

 112 min - [Action](#) | [Adventure](#) | [Romance](#) -
14 January 1993 (Germany)

Your rating: ★★★★★★★★★★ - /10
7.8 Ratings: 7.8/10 from 66,510 users
Reviews: 336 user | 66 critic

Three trappers protect a British Colonel's daughters in the midst of the French and Indian War.

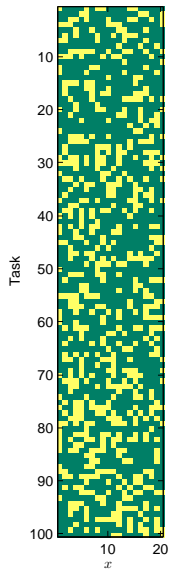
Director: [Michael Mann](#)

Writers: [James Fenimore Cooper](#) (novel), [John L. Balderston](#) (adaptation), [and 5 more credits](#) »

Stars: [Daniel Day-Lewis](#), [Madeleine Stowe](#) and [Russell](#)

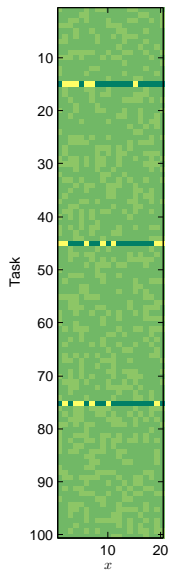
Can we combine all these data to better estimate individual preferences?

Multi-task learning: dataset structure



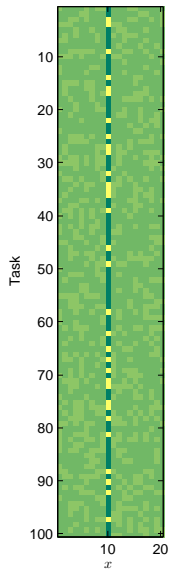
Sampling can be very sparse.

Multi-task learning: dataset structure



Few samples per task...

Multi-task learning: dataset structure



... but each sample is shared by many tasks.

Caltech 256



- 1 Build an object classifier with good generalization performance
- 2 Discover relationships between the different classes

Part II

Output Kernel Learning

Multi-task supervised learning

Synthesizing multiple functions

$$f_j : \mathcal{X} \rightarrow \mathcal{Y}, \quad j = 1, \dots, m$$

from multiple datasets of input-output pairs (x_{ij}, y_{ij}) .

Multi-task supervised learning

Synthesizing multiple functions

$$f_j : \mathcal{X} \rightarrow \mathcal{Y}, \quad j = 1, \dots, m$$

from multiple datasets of input-output pairs (x_{ij}, y_{ij}) .

Multi-task kernels

- For every pair of inputs (x_1, x_2) and every pair of task indices (i, j) , specify a similarity value

$$K((x_1, i), (x_2, j))$$

- Equivalently, specify a matrix valued function H such that

$$[H(x_1, x_2)]_{ij} = K((x_1, i), (x_2, j))$$

Decomposable kernels

$$K((x_1, i), (x_2, j)) = K_X(x_1, x_2)K_Y(i, j),$$

Matrix-valued kernel

$$H(x_1, x_2) = K_X(x_1, x_2) \cdot \mathbf{L}, \quad \mathbf{L}_{ij} = K_Y(i, j)$$

- K_X is the **input kernel**.
- K_Y is the **output kernel** (equivalently, $\mathbf{L} \in \mathbb{S}_+^m$).

Kernel-based regularization

$$\min_{f \in \mathcal{H}_{\mathbf{L}}} \left(\sum_{j=1}^m \sum_{i=1}^{\ell_j} V(y_{ij}, f_j(x_{ij})) + \|f\|_{\mathcal{H}_{\mathbf{L}}}^2 \right)$$

Kernel-based regularization methods

Kernel-based regularization

$$\min_{f \in \mathcal{H}_L} \left(\sum_{j=1}^m \sum_{i=1}^{\ell_j} V(y_{ij}, f_j(x_{ij})) + \|f\|_{\mathcal{H}_L}^2 \right)$$

Representer theorem

$$f_j(x) = \sum_{k=1}^m \mathbf{L}_{jk} \left(\sum_{i=1}^{\ell_k} c_{ij} K_X(x_{ij}, x) \right)$$

Kernel-based regularization methods

Kernel-based regularization

$$\min_{f \in \mathcal{H}_{\mathbf{L}}} \left(\sum_{j=1}^m \sum_{i=1}^{\ell_j} V(y_{ij}, f_j(x_{ij})) + \|f\|_{\mathcal{H}_{\mathbf{L}}}^2 \right)$$

Representer theorem

$$f_j(x) = \sum_{k=1}^m \mathbf{L}_{jk} \left(\sum_{i=1}^{\ell_k} c_{ij} K_X(x_{ij}, x) \right)$$

How to choose the output kernel?

- Independent single-task learning: $\mathbf{L} = \mathbf{I}$.

Kernel-based regularization methods

Kernel-based regularization

$$\min_{f \in \mathcal{H}_{\mathbf{L}}} \left(\sum_{j=1}^m \sum_{i=1}^{\ell_j} V(y_{ij}, f_j(x_{ij})) + \|f\|_{\mathcal{H}_{\mathbf{L}}}^2 \right)$$

Representer theorem

$$f_j(x) = \sum_{k=1}^m \mathbf{L}_{jk} \left(\sum_{i=1}^{\ell_k} c_{ij} K_X(x_{ij}, x) \right)$$

How to choose the output kernel?

- Independent single-task learning: $\mathbf{L} = \mathbf{I}$.
- Pooled single-task learning: $\mathbf{L} = \mathbf{1}$.

Kernel-based regularization methods

Kernel-based regularization

$$\min_{f \in \mathcal{H}_{\mathbf{L}}} \left(\sum_{j=1}^m \sum_{i=1}^{\ell_j} V(y_{ij}, f_j(x_{ij})) + \|f\|_{\mathcal{H}_{\mathbf{L}}}^2 \right)$$

Representer theorem

$$f_j(x) = \sum_{k=1}^m \mathbf{L}_{jk} \left(\sum_{i=1}^{\ell_k} c_{ij} K_X(x_{ij}, x) \right)$$

How to choose the output kernel?

- Independent single-task learning: $\mathbf{L} = \mathbf{I}$.
- Pooled single-task learning: $\mathbf{L} = \mathbf{1}$.
- Design it using prior knowledge.

Kernel-based regularization methods

Kernel-based regularization

$$\min_{f \in \mathcal{H}_{\mathbf{L}}} \left(\sum_{j=1}^m \sum_{i=1}^{\ell_j} V(y_{ij}, f_j(x_{ij})) + \|f\|_{\mathcal{H}_{\mathbf{L}}}^2 \right)$$

Representer theorem

$$f_j(x) = \sum_{k=1}^m \mathbf{L}_{jk} \left(\sum_{i=1}^{\ell_k} c_{ij} K_X(x_{ij}, x) \right)$$

How to choose the output kernel?

- Independent single-task learning: $\mathbf{L} = \mathbf{I}$.
- Pooled single-task learning: $\mathbf{L} = \mathbf{1}$.
- Design it using prior knowledge.
- Learn it from the data.

Multiple Kernel Learning

Multiple Kernel Learning (MKL)

$$K = \sum_{k=1}^N d_k K_k, \quad d_k \geq 0.$$

Multiple Kernel Learning

Multiple Kernel Learning (MKL)

$$K = \sum_{k=1}^N d_k K_k, \quad d_k \geq 0.$$

MKL with decomposable basis kernels

$$K((x_1, i), (x_2, j)) = \sum_{k=1}^N d_k K_X^k(x_1, x_2) K_Y^k(i, j)$$

Multiple Kernel Learning

Multiple Kernel Learning (MKL)

$$K = \sum_{k=1}^N d_k K_k, \quad d_k \geq 0.$$

MKL with decomposable basis kernels

$$K((x_1, i), (x_2, j)) = \sum_{k=1}^N d_k K_X^k(x_1, x_2) K_Y^k(i, j)$$

MKL with decomposable basis kernels (common input kernel)

$$K((x_1, i), (x_2, j)) = K_X(x_1, x_2) \left(\sum_{k=1}^N d_k K_Y^k(i, j) \right)$$

Multiple Kernel Learning

Multiple Kernel Learning (MKL)

$$K = \sum_{k=1}^N d_k K_k, \quad d_k \geq 0.$$

MKL with decomposable basis kernels

$$K((x_1, i), (x_2, j)) = \sum_{k=1}^N d_k K_X^k(x_1, x_2) K_Y^k(i, j)$$

MKL with decomposable basis kernels (common input kernel)

$$K((x_1, i), (x_2, j)) = K_X(x_1, x_2) \left(\sum_{k=1}^N d_k K_Y^k(i, j) \right)$$

Two issues:

- 1 The maximum number of kernels is limited by memory constraints.
- 2 Specifying the dictionary of basis kernels requires domain knowledge.

Optimization problem

$$\min_{\mathbf{L} \in \mathbb{S}_+} \left[\min_{f \in \mathcal{H}_{\mathbf{L}}} \left(\sum_{j=1}^m \sum_{i=1}^{\ell_j} V(y_{ij}, f_j(x_{ij})) + \|f\|_{\mathcal{H}_{\mathbf{L}}}^2 + \Omega(\mathbf{L}) \right) \right],$$

Optimization problem

$$\min_{\mathbf{L} \in \mathcal{S}_+} \left[\min_{f \in \mathcal{H}_{\mathbf{L}}} \left(\sum_{j=1}^m \sum_{i=1}^{\ell_j} V(y_{ij}, f_j(x_{ij})) + \|f\|_{\mathcal{H}_{\mathbf{L}}}^2 + \Omega(\mathbf{L}) \right) \right],$$

Examples:

- Squared Frobenius norm

$$\Omega(\mathbf{L}) = \|\mathbf{L}\|_F^2.$$

- Sparsity-inducing regularizer

$$\Omega(\mathbf{L}) = \|\mathbf{L}\|_1.$$

- Low-rank inducing regularizer

$$\Omega(\mathbf{L}) = \text{tr}(\mathbf{L}) + I(\text{rank}(\mathbf{L}) \leq p).$$

Low-Rank OKL

$$\min_{\mathbf{L} \in \mathcal{S}_+^{m,p}} \left[\min_{\mathbf{C} \in \mathbb{R}^{\ell \times m}} \left(\frac{\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{KCL})\|_F^2}{2\lambda} + \frac{\text{tr}(\mathbf{C}^T \mathbf{KCL})}{2} + \frac{\text{tr}(\mathbf{L})}{2} \right) \right].$$

Low-Rank OKL

$$\min_{\mathbf{L} \in \mathbb{S}_+^{m,p}} \left[\min_{\mathbf{C} \in \mathbb{R}^{\ell \times m}} \left(\frac{\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{KCL})\|_F^2}{2\lambda} + \frac{\text{tr}(\mathbf{C}^T \mathbf{KCL})}{2} + \frac{\text{tr}(\mathbf{L})}{2} \right) \right].$$

- A non-linear generalization of reduced-rank regression.

Low-Rank OKL

$$\min_{\mathbf{L} \in \mathbb{S}_+^{m,p}} \left[\min_{\mathbf{C} \in \mathbb{R}^{\ell \times m}} \left(\frac{\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{KCL})\|_F^2}{2\lambda} + \frac{\text{tr}(\mathbf{C}^T \mathbf{KCL})}{2} + \frac{\text{tr}(\mathbf{L})}{2} \right) \right].$$

- A non-linear generalization of reduced-rank regression.
- One of the reformulations only requires storing low-rank matrices.

Low-Rank OKL

$$\min_{\mathbf{L} \in \mathbb{S}_+^{m,p}} \left[\min_{\mathbf{C} \in \mathbb{R}^{\ell \times m}} \left(\frac{\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{KCL})\|_F^2}{2\lambda} + \frac{\text{tr}(\mathbf{C}^T \mathbf{KCL})}{2} + \frac{\text{tr}(\mathbf{L})}{2} \right) \right].$$

- A non-linear generalization of reduced-rank regression.
- One of the reformulations only requires storing low-rank matrices.

Low-Rank OKL

$$\min_{\mathbf{L} \in \mathbb{S}_+^{m,p}} \left[\min_{\mathbf{C} \in \mathbb{R}^{\ell \times m}} \left(\frac{\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{KCL})\|_F^2}{2\lambda} + \frac{\text{tr}(\mathbf{C}^T \mathbf{KCL})}{2} + \frac{\text{tr}(\mathbf{L})}{2} \right) \right].$$

- A non-linear generalization of reduced-rank regression.
- One of the reformulations only requires storing low-rank matrices.

Unconstrained reformulation

$$\min_{\substack{\mathbf{A} \in \mathbb{R}^{\ell \times p} \\ \mathbf{B} \in \mathbb{R}^{m \times p}} \left(\frac{\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{KAB}^T)\|_F^2}{2\lambda} + \frac{\text{tr}(\mathbf{A}^T \mathbf{KA})}{2} + \frac{\|\mathbf{B}\|_F^2}{2} \right),$$

- Current optimization strategy: block-coordinate descent + approximate Preconditioned Conjugate Gradient (PCG)

Low-Rank OKL

$$\min_{\mathbf{L} \in \mathcal{S}_+^{m,p}} \left[\min_{\mathbf{C} \in \mathbb{R}^{\ell \times m}} \left(\frac{\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{KCL})\|_F^2}{2\lambda} + \frac{\text{tr}(\mathbf{C}^T \mathbf{KCL})}{2} + \frac{\text{tr}(\mathbf{L})}{2} \right) \right].$$

- A non-linear generalization of reduced-rank regression.
- One of the reformulations only requires storing low-rank matrices.

Unconstrained reformulation

$$\min_{\substack{\mathbf{A} \in \mathbb{R}^{\ell \times p} \\ \mathbf{B} \in \mathbb{R}^{m \times p}}} \left(\frac{\|\mathbf{W} \odot (\mathbf{Y} - \mathbf{KAB}^T)\|_F^2}{2\lambda} + \frac{\text{tr}(\mathbf{A}^T \mathbf{KA})}{2} + \frac{\|\mathbf{B}\|_F^2}{2} \right),$$

- Current optimization strategy: block-coordinate descent + approximate Preconditioned Conjugate Gradient (PCG)

F. Dinuzzo, K. Fukumizu. *Learning low-rank output kernels*, ACML, 2011

F. Dinuzzo, *Learning output kernels for multi-task problems*, Neurocomputing, 2013

Part III

Experiments with OKL

Table: MovieLens datasets: total number of users, movies, and ratings.

Dataset	Users	Movies	Ratings
MovieLens100K	943	1682	10^5
MovieLens1M	6040	3706	10^6
MovieLens10M	69878	10677	10^7

Table: MovieLens datasets: total number of users, movies, and ratings.

Dataset	Users	Movies	Ratings
MovieLens100K	943	1682	10^5
MovieLens1M	6040	3706	10^6
MovieLens10M	69878	10677	10^7

Input Kernel (similarity between movies)

$$K(x_1, x_2) = \delta_K(x_1^{id}, x_2^{id}) + \exp(-d_H(x_1^g, x_2^g)),$$

Table: MovieLens datasets: total number of users, movies, and ratings.

Dataset	Users	Movies	Ratings
MovieLens100K	943	1682	10^5
MovieLens1M	6040	3706	10^6
MovieLens10M	69878	10677	10^7

Input Kernel (similarity between movies)

$$K(x_1, x_2) = \delta_K(x_1^{id}, x_2^{id}) + \exp(-d_H(x_1^g, x_2^g)),$$

Methods:

- Independent single-task learning.
- Pooled single-task learning.
- Regularized matrix factorization.
- Low-rank output kernel learning.

Setup: for each user, 50% of the ratings are used for training and the remaining ones for test.

Table: MovieLens datasets: test RMSE

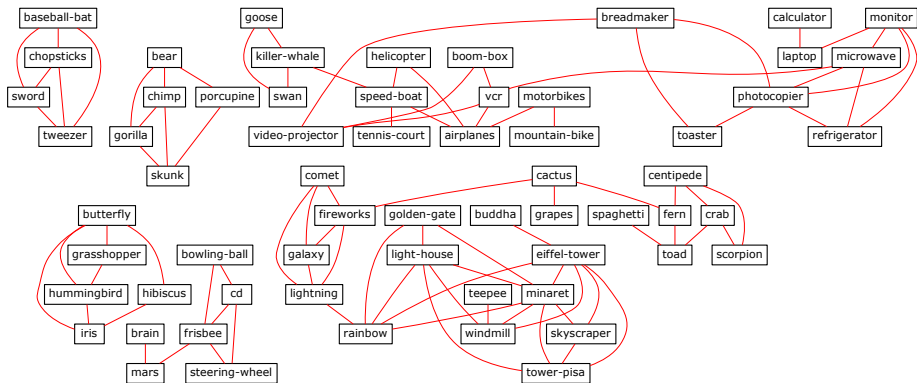
Dataset	Pooled	Independent	RMF	OKL
MovieLens100K	1.0209	1.0445	1.0300	0.9557
MovieLens1M	0.9811	1.0297	0.9023	0.8945
MovieLens10M	0.9441	0.9721	0.8627	0.8501

Caltech 256



- 257 classes, 30 training examples per class
- Input Kernel: designed as in (Gehler and Nowozin, ICCV, 2009)
- Output Kernel: learned using OKL

Structure discovery: how was this graph obtained?



Edges correspond to highest entries of the output kernel matrix \mathbf{L} .

Conclusions

- Many machine learning problems are structured and multi-task.
- Solving multiple problems simultaneously can improve performances.
- Output Kernel Learning methods can solve multi-task problems and automatically reveal inter-task relationships.
- Non-convex optimization problems, but with a special structure.
- Code available online.



F. Dinuzzo.

Learning output kernels for multi-task problems.

Neurocomputing, In Press, 2013.



F. Dinuzzo and K. Fukumizu.

Learning low-rank output kernels.

In *Proceedings of the Asian Conference on Machine Learning*, 2011.



F. Dinuzzo, C. S. Ong, P. Gehler, and G. Pilonetto.

Learning output kernels with block coordinate descent.

In *Proceedings of the International Conference on Machine Learning*, 2011.