Multi-task Averaging via Task Clustering

David Martínez-Rego[1] and Massimiliano Pontil[2]

[1] LIDIA Group, Department of Computer Science
University of A Coruña
Campus de Elviña, s/n 15071 A Coruña, Spain
dmartinez@udc.es

[2] Centre for Computational Statistics and Machine Learning
Department of Computer Science, University College London
Malet Place, Gower Street, London, WC1E 6BT
m.pontil@cs.ucl.ac.uk

Abstract. Multi-task averaging deals with the problem of estimating the means of a set of distributions jointly. It has its roots in the fifties when it was observed that leveraging data from related distributions can yield superior performance over learning from each distribution independently. Stein's paradox showed that, in an average square error sense, it is better to estimate the means of T Gaussian random variables using data sampled from all of them. This phenomenon has been largely disregarded and has recently emerged again in the field of multi-task learning. In this paper, we extend recent results for multi-task averaging to the n-dimensional case and propose a method to detect from data which tasks/distributions should be considered as related. Our experimental results indicate that the proposed method compares favorably to the state of the art.

Keywords: multi-task averaging, information theory, spectral clustering.

1 Introduction

Multi-task averaging (MTA) problem can be posed as follows: we have T datasets $\{\mathbf{x}_{t1}, \mathbf{x}_{t2}, \ldots, \mathbf{x}_{tN_t}\}$, $t = 1, \ldots, T$ each of which is sampled from a fixed but unknown probability distribution (N_t denotes the size of dataset t). Our goal is to estimate the means of each distribution. The first direct approach would be to estimate the means one at a time. However, it turns out that leveraging data from related distributions/tasks[1] can yield superior performance over learning each mean independently. Early evidence of this phenomenon dates back in the fifties from Stein's work, who showed that it is better (in an average square error sense) to estimate each of the means of T Gaussian random variables using data sampled from all of them, even if the random variables are independent

[1] Throughout the paper we use the words "distribution", "task" and "mean" interchangeably.

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 148–159, 2013.

and have different means. This surprising result is often referred to as Stein's paradox [3]. A recent work [4], studies MTA problem in one dimension (that is, taking $\mathbb{R}$ as input space) and presents different optimal results both for MTA mean estimator formula and its hyper-parameters. The proposed estimators are proved to be more accurate than those previously studied in the literature [6,7], but the study of their performance in an n-dimensional space is not treated and the proposed optimal hyper-parameter expression is only valid for the case when all the tasks are related to each other.

In this paper, we study MTA problem in $\mathbb{R}^n$ and also explore the impact of task grouping on the estimation accuracy of the estimation method. We propose optimal formulas for the n-dimensional case and a practical algorithm for task grouping based on information theoretic divergence measures and spectral clustering. When combining these two results, a practical algorithm for MTA in $\mathbb{R}^n$ is obtained. It will be showed that in certain circumstances, when not all the tasks at hand should be considered as related, then the optimal estimators presented in [4] have a null improvement when compared with independent mean estimation for each of the T tasks. On the other hand, we will demonstrate that the proposed method can improve estimation accuracy in an average mean square error sense. These findings may pave the way for more accurate algorithms in a multi-task scenario.

The paper is organized in the following manner. In Section 2, a summary of the key results in [4] are reviewed as the base for the present work. Section 3 presents the extension of the estimators in [4] for the n-dimensional case. Section 4 presents the proposed k-MTA method. In Section 5, we report on our numerical experiments with this method and with previous approaches. Finally, Section 6 contains concluding remarks and suggestions for future work.

2 Background

In the recent paper [4], MTA estimation in $\mathbb{R}$ is presented as the optimal solution to the following convex problem:

$$\mathbf{c}^* = \arg\min_{\mathbf{c}\in\mathbb{R}^T} \left\{ \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N_t} \frac{(x_{ti}-c_t)^2}{\sigma_t^2} + \frac{\gamma}{T^2}\sum_{s,t=1}^{T} A_{st}(c_s-c_t)^2 \right\}$$

where $x_{t1}, x_{t2}, \ldots, x_{tN_t}$ are independent and identically distributed (iid) random samples for each task $t = 1, \ldots, T$, σ_t^2 is the variance of t-th distribution and $\mathbf{c} = (c_1, \ldots, c_T)$ is the vector of means we wish to estimate. Matrix $\mathbf{A} = (A_{st})_{s,t=1}^{T}$ describes the relatedness or similarity of any pair of the T tasks (with $A_{tt} = 0$ for all t without loss of generality because the diagonal self-similarity terms are canceled in the objective). It can be noted that the proposed MTA objective regularizes the estimates of each of the means, that is, it ties them together. The regularization parameter γ balances the empirical risk (error) and the multi-task regularizer. Note that if $\gamma = 0$, the MTA objective decomposes into T separate minimization problems, producing the simple separate sample averages

$\hat{x}_t = 1/N_t \sum_{i=1}^{N_t} x_{ti}$. Tasks' similarity matrix $\mathbf{A}$ for a specific problem at hand can be specified from the knowledge of a domain expert, but often this side information is not available or it may not be clear how to transform semantic notions of tasks' similarity into an appropriate choice for the values in $\mathbf{A}$. In addition to this difficulty, parameter γ has a great impact on the final result and an optimal choice from a mean square error perspective is desirable. However, the problem of finding an optimal formula for this parameter for a general form of matrix $\mathbf{A}$ is often analytically intractable. In [4], the optimal solution in cases when $\mathbf{A} = a\mathbf{1}\mathbf{1}'$ (called "constant MTA") was found. We restate this result for completeness:

Lemma 1 (constant MTA). *Assume that $\mathbf{A} = a\mathbf{1}\mathbf{1}'$ and $0 < \frac{\sigma_t^2}{N_t} < \infty$ for all t. The optimal $\mathbf{c}^*$ (in terms of mean square error) is given by the formula*

$$\mathbf{c}^* = (I_T + \frac{a}{T}\mathbf{\Sigma} L(\mathbf{1}\mathbf{1}'))^{-1}\hat{\mathbf{x}}$$

where

$$a = \frac{2}{\frac{1}{T(T-1)} \sum\limits_{s,t=1}^{T} (\mu_s - \mu_t)^2}. \tag{1}$$

In the above formula $\mathbf{\Sigma} = \text{diag}(\frac{\sigma_1^2}{N_1}, \ldots, \frac{\sigma_T^2}{N_T})$, $L(\mathbf{A})$ is the Laplacian of matrix $\mathbf{A}$ and μ_t is the true mean of task t. Note that in this result γ is considered equal to 1 without loss of generality.

There are two main issues when applying this lemma in a practical situation. First, the result involves σ_t^2 and μ_t, both quantities which are not known in practice (the second quantity is indeed the one that we are trying to estimate). This issue is solved in [4] using empirical estimates for both quantities and proved to be accurate in practice. Therefore such approach is also used in this paper. The second issue has to do with the form of matrix $\mathbf{A}$ considered in Lemma 1. With $\mathbf{A} = a\mathbf{1}\mathbf{1}'$ we are assuming that all the T tasks are mutually related, which is very unlikely to happen in practice. An analytical result for the case when $T = 2$ proves that the proposed MTA estimation is better than single task estimation only if the true means are close with respect to the variances of their distributions. This observation will be experimentally observed in Section 5. In addition, a closer look at formula (1) shows us that, if far apart tasks are considered as related, the optimal value of parameter a will approximate 0, so that the MTA estimator will bring no benefit.

In order to use the above results in a general case, in addition to extend them to $\mathbb{R}^n$, it is necessary to devise a strategy that, directly from data, estimates which tasks should be considered as related. In the remaining part of the paper we will tackle these problems and demonstrate experimentally that our strategy yields improved results in an average mean square error sense when compared to previous strategies.

3 MTA in High Dimensional Spaces

In this section, we extend the problem presented in [4] to $\mathbb{R}^n$ in a straightforward manner. This will be the first step towards the general MTA algorithm presented in Section 4. MTA in $\mathbb{R}^n$ consist in finding the optimal solution to the problem

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \left\{ \sum_{t=1}^{T} \sum_{i=1}^{N_t} \frac{\|\mathbf{x}_{ti} - \mathbf{c}_t\|^2}{\sigma_t^2} + \frac{\gamma}{2T} \sum_{s,t=1}^{T} A_{st} \|\mathbf{c}_s - \mathbf{c}_t\|^2 \right\} \tag{2}$$

where $\mathbf{c} \in \mathbb{R}^{Tn}$ denotes the vector with all the means $\mathbf{c}_t$, $t = 1, \ldots, T$ concatenated and γ is a hyper-parameter that balances the weighting of the two terms. Problem (2) is similar to equation proposed in [4] but including the 2-norm in $\mathbb{R}^n$ instead of in $\mathbb{R}$. The next two lemmas will be proved in the appendix.

Lemma 2 (MTA in $\mathbb{R}^n$). *The optimal solution of problem (2) is given by*

$$\mathbf{c}^* = ((I_T + \frac{\gamma}{T} \mathbf{\Sigma} L(\mathbf{A}))^{-1} \otimes I_n) \hat{\mathbf{x}} \tag{3}$$

where I_T (resp. I_n) is the $T \times T$ (resp. $n \times n$) identity matrix, $\mathbf{\Sigma} = \mathrm{diag}(\frac{\sigma_1^2}{N_1}, \ldots, \frac{\sigma_T^2}{N_T})$, $L(\mathbf{A})$ is the Laplacian of matrix $\mathbf{A}$ and $\hat{\mathbf{x}} \in \mathbb{R}^{Tn}$ is the vector of independent means $\hat{\mathbf{x}}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{x}_{ti}$ concatenated in the same order as in $\mathbf{c}^*$.

Lemma 3 (constant MTA in $\mathbb{R}^n$). *Assume that $\mathbf{A} = a\mathbf{1}\mathbf{1}'$ and $0 < \frac{\sigma_t^2}{N_t} < \infty$ for all t. The optimal (in a mean square error sense) mean estimator is given by*

$$\mathbf{c}^* = ((I_T + \frac{a}{T} \mathbf{\Sigma} L(\mathbf{1}\mathbf{1}'))^{-1} \otimes I_n) \hat{\mathbf{x}} \tag{4}$$

for

$$a = \frac{2n}{\frac{1}{T(T-1)} \sum_{s,t=1}^{T} \|\mu_s - \mu_t\|^2} \tag{5}$$

where n is the dimension of the input space and μ_t are the true mean vectors of the distributions of each task.

Note that the obtained formulas for the estimator involve the inverse of a matrix which depends neither on the dimension of the space nor on the sample sizes. Hence, its calculation can be done in a very efficient way. Estimators from data of the actual values of μ_t and σ_t^2 in equations (4) and (5) will be used in the practical implementation of these formulas.

4 *k*-MTA: Multi-task Averaging via Information Theoretic Clustering

In this section, k-MTA algorithm is proposed. It is divided in two phases: (a) first, the sets of tasks which should be considered as related are detected via spectral clustering; (b) for each cluster of tasks, equations (5) and (4) are applied separately in order to find the means of each task in the cluster. This approach aims at tackling the limitations of the direct application of the results in [4] when a clustered set of tasks is presented and their respective means are required. Following the results sketched in [4], MTA is only effective when the distance between the true means of the tasks is small when compared to the variance of their distributions. So, for this first phase, we need a measure of divergence between tasks which is able to detect (from data samples) whether the supports of probability distributions largely overlap or not. Based on those similarities, tasks are subsequently clustered and their means estimated. In the next section, we present the divergence measure which will be used in this paper. Subsequently, the spectral clustering algorithm used to construct the groups is described.

4.1 Information Theoretic Tasks' Similarity Measure

The work in [10] considers the quadratic Renyi's entropy as the basic expression for building cost functions for clustering, linear models, and other machine learning problems. The cost and divergence measures developed under the Renyi's entropy framework have been proved effective when dealing with these different learning problems. In particular, the divergence measure between probability density functions (pdfs) called *euclidean pdf distance* is given by

$$D_{ED} = \int\limits_{\mathbb{R}^n} (f(\mathbf{x}) - g(\mathbf{x}))^2 d\mathbf{x}. \tag{6}$$

In the absence of an expression for both f and g, in [10], a parzen estimation using a Gaussian Kernel [11] of both is considered. Using these approximations for f and g, this quantity can be rewritten as:

$$\begin{aligned} D_{ED} &= \int\limits_{\mathbb{R}^n} (f(\mathbf{x}) - g(\mathbf{x}))^2 d\mathbf{x} = \int\limits_{\mathbb{R}^n} f(\mathbf{x})^2 d\mathbf{x} + \int\limits_{\mathbb{R}^n} g(\mathbf{x})^2 d\mathbf{x} - 2\int\limits_{\mathbb{R}^n} f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x} \\ &= \int\limits_{\mathbb{R}^n} \left(\frac{1}{N}\sum_{i=1}^{N} G_\sigma(\mathbf{x} - \mathbf{x}_i^f)\right)^2 d\mathbf{x} + \int\limits_{\mathbb{R}^n} \left(\frac{1}{M}\sum_{i=1}^{M} G_\sigma(\mathbf{x} - \mathbf{x}_i^g)\right)^2 d\mathbf{x} \\ &- 2\int\limits_{\mathbb{R}^n} \Big(\frac{1}{N}\sum_{i=1}^{N} G_\sigma(\mathbf{x} - \mathbf{x}_i^f)\Big)\Big(\frac{1}{M}\sum_{i=1}^{M} G_\sigma(\mathbf{x} - \mathbf{x}_i^g)\Big) d\mathbf{x} \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sqrt{2}\sigma}(\mathbf{x}_j^f - \mathbf{x}_i^f)^2 + \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} G_{\sqrt{2}\sigma}(\mathbf{x}_j^g - \mathbf{x}_i^g)^2 \\ &- \frac{2}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M} G_{\sqrt{2}\sigma}(\mathbf{x}_j^g - \mathbf{x}_i^f)^2 = \hat{V}_f + \hat{V}_g - 2\hat{V}_c \end{aligned} \tag{7}$$

where σ is the width of the gaussian kernel and has to be selected. This measure has proven to be an effective way of computing the divergence between two pdfs represented by a sample in many learning scenarios and in this work will be used as the similarity measure between tasks. Specifically, a normalized version of this measure is used

$$D_{ED}^{N}(f,g) = 2 - 2\hat{V}_c/\hat{V}_f\hat{V}_g. \tag{8}$$

This expression still maintains the properties of a divergence and has the advantage of being normalized in the interval $[0,2]$ which will be useful for graph construction in the k-MTA algorithm. Since the clustering technique presented below requires a similarity measure, we transform the aforementioned divergence D_{ED}^{N} into the following similarity measure in the interval $[0,1]$:

$$S_{ij} = \frac{2 - D_{ED}^{N}(f_i,f_j)}{2} \tag{9}$$

4.2 Spectral Clustering

Spectral clustering [14] aims at clustering similar objects $o_i,\ i = 1,\ldots,T$ into k groups given a similarity graph G between all these objects. It can be used virtually with a sample of any kind of items as long as we are given a similarity measure between them. These similarities are used to build a similarity graph G which subsequently is fed into the clustering subroutine. When constructing similarity graphs the goal is to model the local neighborhood relationships between the data points. There are several popular constructions to transform a given set $o_1,\ldots,o_n$ of objects with pairwise similarities S_{ij} into a graph: (a) ϵ-neighborhood, where all points whose pairwise similarities are greater than ϵ are connected; (b) k-nearest neighbor graphs, where if a vertex v_i is among the k-nearest neighbors of v_j those two vertex are connected, and (c) fully connected graph, in which all points are connected with positive similarity given by S_{ij}. In this work we will use ϵ-neighborhood strategy to build the similarity graph.

Once we have the similarity graph G, the graph Laplacian of matrix G is constructed. At this point three main algorithms are proposed in the literature depending on the kind of Laplacian used: unnormalized spectral clustering [14] and the works in [9,13] which use a normalized Laplacian. In this work we will use the version of [13] since it has proved more accurate and stable in practice. Algorithm 1 summarizes the steps of this algorithm (more details can be found in [14]).

4.3 Proposed Algorithm

In this section, we combine the results and components described in previous sections in the proposed algorithm k-MTA. Algorithm 2 summarizes its main steps. First, the task clusters are detected combining the similarity measure presented in Section 4.1 with the spectral clustering algorithm of Section 4.2. Thanks to this step, we will apply the MTA formula derived in Section 3 to the task groups which are similar to each other and we will not blend in tasks which are completely dissimilar, thus avoiding negative transfer.

Algorithm 1. Spectral clustering main steps

```
Input: Similarity matrix $\mathbf{S} \in \mathbb{R}^{T \times T}$, number of clusters $k$, barrier $\epsilon$.
Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid o_j \in C_i\}$

1. Construct a similarity graph $G$ by $\epsilon$-neighborhood based on $S$.
2. Compute the unnormalized Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{G}$.
3. Compute the first $k$ generalized eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ of the genera lized eigenproblem $\mathbf{Lu} = \lambda \mathbf{Du}$.
4. Let $\mathbf{U} \in \mathbb{R}^{T \times k}$ be the matrix containing the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ as columns.
5. Let $\mathbf{y}_i \in \mathbb{R}^k$ be the vector corresponding to the $i$-th row of $\mathbf{U}$ (each $\mathbf{y}_i$ corresponds to each object $o_i$).
6. Cluster the points $\mathbf{y}_i \in \mathbb{R}^k, \ i = 1, \ldots, T$ with the $k$-means algorithm into clusters $A_1, \ldots, A_k$.
```

Algorithm 2. k-MTA algorithm

```
Input: Similarity matrix $\mathbf{S} \in \mathbb{R}^{T \times T}$, number of clusters $k$, barrier $\epsilon$.
Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid o_j \in C_i\}$

1. Construct a similarity graph $G$ by $\epsilon$-neighborhood based on $S$.
2. Compute the unnormalized Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{G}$.
3. Compute the first $k$ generalized eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ of the genera lized eigenproblem $\mathbf{Lu} = \lambda \mathbf{Du}$.
4. Let $\mathbf{U} \in \mathbb{R}^{T \times k}$ be the matrix containing the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ as columns.
5. Let $\mathbf{y}_i \in \mathbb{R}^k$ be the vector corresponding to the $i$-th row of $\mathbf{U}$ (each $\mathbf{y}_i$ corresponds to each object $o_i$).
6. Cluster the points $\mathbf{y}_i \in \mathbb{R}^k, \ i = 1, \ldots, T$ with the $k$-means algorithm into clusters $A_1, \ldots, A_k$.
```

5 Experimental Results

In this section, we explore the performance of k-MTA when compared to its predecessor MTA in [4] and with the single task mean calculation method. To this end, we test the methods on both an artificial dataset which exhibits the behavior of all the methods when clusters of tasks are present, as well as a real dataset where final marks of groups of students are to be predicted.

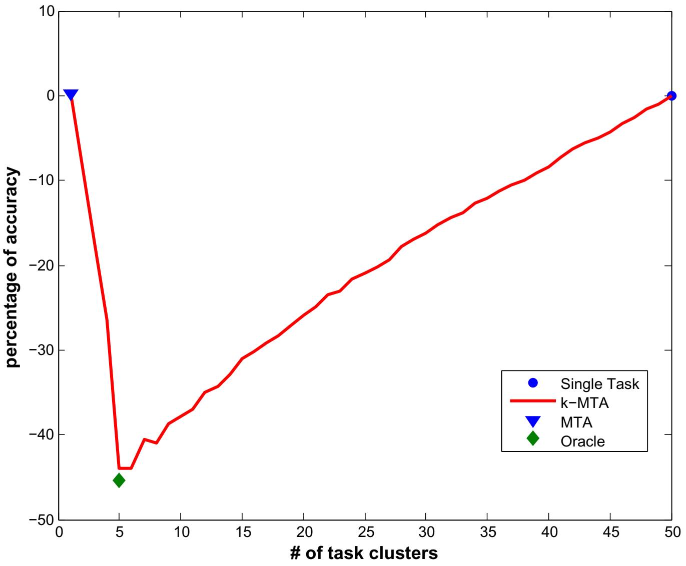

Fig. 1. Mean square distance to the actual means compared to single task result

5.1 Artificial Dataset

The artificial generated dataset has the following properties:

- Number of tasks: 50;
- Number of clusters of tasks: 5;
- Number of task per cluster: 10;
- Input Space: $\mathbb{R}^{10}$;
- Distribution of data: first the means of the Gaussians are selected from 5 Gaussians $\mu_t \sim N(\mu_c \mathbf{1}, \sigma I_d))$ for $\mu_c = \{-10, -5, 0, 5, 10\}$ and $\sigma = 0.1$, where I_d denotes the $d \times d$ identity matrix. Ten centers are selected for each value of μ_c. Then, for each task, a set of iid random data points are generated as $\mathbf{x}_{ti} \sim N(\mu_t, I_d)$.

In this selection, we obtain a convenient distribution of the data for k-MTA since the task are clustered in 5 distant clusters and the expected distance between their centers is small compared to the variance of each task. Figure 1 depicts the average mean square distance from the estimated means to the actual ones compared to the average mean distances obtained with single task means. The results in the figure are the averages of 30 random runs, having 5 data points per task (a scarce sample when compared to the number of parameters to be estimated). The value k of the x axis is the number of clusters k that were configured for k-MTA (optimal ϵ was selected from the interval $[0, 0.5]$). It can be observed how MTA directly applied to the data does not bring any benefit when compared with the single task means while k-MTA obtains an increment of up to -45% when the exact number of clusters is given. Also, the mean result for the oracle, when the correct clustering is always provided to the k-MTA, is shown. It can be observed how the risk of the k-MTA is similar to the oracle one when the correct number of clusters is given as value for k. In addition, it should be noted that even when the number of asked clusters is not exactly the number of actual clusters in the dataset, k-MTA is indeed able to obtain very good accuracy increments.

5.2 School Dataset

The goal of this application is to predict the final class grades $\mu_1, \ldots, \mu_T$ of T students, given only each student's N homework grades y_{ti}, $i = 1, \ldots, N$. The final class grades include all tests and final exams made by the students but only homework grades are used to predict the final grade. The 16 anonymized datasets were provided by instructors at the University of Washington Department of Electrical Engineering. We consider each class as an experiment and the students in that class the tasks. All grades are normalized in the interval $[0,\ 100]$ and never handed homework was assigned 0 points. For each class, a single pooled variance estimate was used for all tasks. In other words $\sigma_t^2 = \sigma^2$, for every $t = 1, \ldots, T$. Table 1 shows the results obtained when compared with MTA. The reported results are the gains in percentage in final marks prediction when compared with single task means, thus lower value is better.

Table 1. School dataset results

# of stud.	68	69	72	44	50	50	47	16
k-MTA	-37.29	**-38.73 (*)**	-26.92	-36.91	**-18.14 (*)**	-26.58	-8.62	**-1.80 (*)**
MTA	-37.29	-38.42	-26.94	-36.91	3.33	-26.58	-8.62	1.0
# of stud.	29	36	57	48	58	39	149	110
k-MTA	-10.26	-13.99	**-3.82 (*)**	**-12.80 (*)**	-12.35	-5.38	-9.15	-11.52
MTA	-10.26	-13.99	-3.47	-11.53	-12.35	-5.38	-9.15	-11.52

In the table it can be observed that, since k-MTA includes MTA as an special case (when $k = 0$) it has always an equal or better performance than MTA. It is important to note that k-MTA performs better in 5 out of 16 classes and that it always presents a gain with respect to single task means. It is able to obtain a gain even when MTA can not improve single task means. This may be due the presence of clusters in those classes, which are not treated by MTA. In this case, optimal values were selected from the intervals $k = [1, 30]$ and $\epsilon = [0, 0.5]$.

6 Conclusions and Future Work

We have proposed a new algorithm for multi-task averaging. It extends the work in [4] to a n-dimensional space and tackles a key issue when dealing with real data, namely the presence of clusters of related tasks. The algorithm is based on two steps. First, tasks are clustered based on their samples and subsequently MTA is applied for each cluster of tasks. Experimental results show that direct application of MTA in a case where tasks are clustered is useless compared with the results obtained by the single task means. On the other hand, k-MTA is able to detect the underlying clusters of tasks and obtains a significant increment of accuracy. The experiments also suggest that, when dealing with more than two tasks, their relatedness should reflect the similarity between their distributions and this issue should be taken into account when building algorithms like for example multitask one-class classifiers [5,15]. In the future it would be interesting to study extension of the ideas presented here to learn multiple mean embeddings in reproducing kernel Hilbert spaces (see e.g. [2]). Another interesting direction of research is to consider different models of task relatedness and groupping such as in [1,8].

Acknowledgements. This work was supported by Secretaría de Estado de Investigación of the Spanish Government under project TIN 2009-02402, and by the Xunta de Galicia through projects CN2011/007 and CN2012/211, all partially supported by the European Union ERDF. David Martinez-Rego is supported by the FPU Grant Program of the Ministry of Education of the Spanish Government.

A Appendix: Proof of Lemmas

Proof of Lemma 2

We first rewrite the objective function in equation (2) as

$$\sum_{t=1}^{T}(a_t + \frac{N_t}{\sigma_t^2}\|\mathbf{c}_t\|^2 - 2\frac{N_t}{\sigma_t^2}\mathbf{c}_t'\mathbf{c}_t) + \frac{\gamma}{2T}\sum_{s,t=1}^{T} A_{st}(\|\mathbf{c}_s\|^2 + \|\mathbf{c}_t\|^2 - 2\mathbf{c}_s'\mathbf{c}_t)$$

where $a_t := \sum_{i=1}^{N_t} \frac{\|\mathbf{x}_{ti}\|^2}{\sigma_t^2}$

Next we rewrite this equation as in terms of $\mathbf{c} \in \mathbb{R}^{Tn}$ and $\hat{\mathbf{x}} \in \mathbb{R}^{Tn}$ as

$$\sum_{t=1}^{T} \mathbf{a}_t + \mathbf{c}'(\mathbf{\Sigma}^{-1} \otimes I_n)\mathbf{c} - 2\mathbf{c}'(\mathbf{\Sigma}^{-1} \otimes I_n)\hat{\mathbf{x}} + \frac{\gamma}{T}\mathbf{c}'(L(\mathbf{A}) \otimes I_d)\mathbf{c}.$$

Taking the derivative with respect to $\mathbf{c}$ and setting it equal it to $\mathbf{0}$ yields that

$$\mathbf{c}^* = (I_{Tn} + \frac{\gamma}{T}(\mathbf{\Sigma} \otimes I_n)(L(\mathbf{A}) \otimes I_n))^{-1}\hat{\mathbf{x}}.$$

Applying the mixed-product property of the kronecker product to the second term of the inverse, then the associativity of the kronecker product and the inverse property we find that

$$(I_T \otimes I_n + \frac{\gamma}{T}(\mathbf{\Sigma}L(\mathbf{A})) \otimes I_n)^{-1} = ((I_T + \frac{\gamma}{T}\mathbf{\Sigma}L(\mathbf{A}))^{-1} \otimes I_n).$$

The result follows.

A Proof of Lemma 3

Without loss of generality we assume that $\gamma = 1$. Let $\sigma = \frac{\mathrm{tr}(\boldsymbol{\Sigma})}{T}$ and observe that

$$\begin{aligned}
\mathbf{c}^* &= ((I_T + \frac{a}{T}\boldsymbol{\Sigma} L(\mathbf{1}\mathbf{1}'))^{-1} \otimes I_n)\hat{\mathbf{x}} \\
&= ((I_T + \frac{a}{T}\boldsymbol{\Sigma}(T I_T - \mathbf{1}\mathbf{1}')^{-1} \otimes I_n)\hat{\mathbf{x}} \\
&= ((I_T + a\boldsymbol{\Sigma} I_T - \frac{a}{T}\boldsymbol{\Sigma}\mathbf{1}\mathbf{1}')^{-1} \otimes I_n)\hat{\mathbf{x}} \\
&= ((I_T + a\boldsymbol{\Sigma} I_T)^{-1} + \frac{(I_T + a\boldsymbol{\Sigma} I_T)^{-1}\frac{a}{T}\boldsymbol{\Sigma}\mathbf{1}\mathbf{1}'(I_T + a\boldsymbol{\Sigma} I_T)^{-1}}{1 - \frac{a}{T}\mathbf{1}'(I_T + a\boldsymbol{\Sigma} I_T)^{-1}\boldsymbol{\Sigma}\mathbf{1}}) \otimes I_n)\hat{\mathbf{x}} \\
&= (\frac{1}{a\sigma + 1}\left(I_T + a\frac{\sigma}{T}\mathbf{1}\mathbf{1}'\right) \otimes I_n)\hat{\mathbf{x}}
\end{aligned}$$

where we have made use of the Sherman-Morrison formula for the inverse and omitted some tedious algebra. We will call the matrix on the right-hand side $\mathbf{Z}$ when substituting.

Next, we define the expression for the expected mean square error of an estimator of the form $\mathbf{W}\hat{\mathbf{x}}$ of a mean vector μ, where $\hat{\mathbf{x}}$ is the simple average of each task. We have that:

$$\begin{aligned}
R(\mathbf{W}\hat{\mathbf{x}}, \mu) &= E(\|\mathbf{W}\hat{\mathbf{x}} - \mu\|^2) \\
&= E((\mathbf{W}\hat{\mathbf{x}} - \mu)'(\mathbf{W}\hat{\mathbf{x}} - \mu)) \\
&= \mathrm{tr}(\mathbf{W}\boldsymbol{\Sigma}\mathbf{W}') + \mu'(\mathbf{W} - I)'(\mathbf{W} - I)\mu
\end{aligned}$$

where the expected value is taken with respect to the random sample and μ and $\boldsymbol{\Sigma}$ are the actual mean and covariance of the distribution. In this work we will suppose that all the distributions have an isotropic diagonal covariance matrix so we can use this expression with $\mu \in R^{Tn}$ and covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_T \otimes I_n$ with $\boldsymbol{\Sigma}_{\mathbf{T}} = \mathrm{diag}(\frac{\sigma_1^2}{N_1}, \ldots, \frac{\sigma_T^2}{N_T})$. If we substitute the optimal expression for $\mathbf{W}$ in this expression we have that:

$$\begin{aligned}
R(\mathbf{W}\hat{\mathbf{x}}, \mu) &= \mathrm{tr}((\mathbf{Z} \otimes I_n)\boldsymbol{\Sigma}(\mathbf{Z} \otimes I_n)') + \mu'((\mathbf{Z} \otimes I_n) - I_{Tn})'((\mathbf{Z} \otimes I_n) - I_{Tn})\mu \\
&= \mathrm{tr}((\mathbf{Z}\boldsymbol{\Sigma}\mathbf{Z}') \otimes I_n) + \mu'((\mathbf{Z} - I_T)'(\mathbf{Z} - I_T) \otimes I_n)\mu \\
&= \mathrm{tr}(\mathbf{Z}\boldsymbol{\Sigma}\mathbf{Z}')\mathrm{tr}(I_n) + \mu'((\mathbf{Z} - I_T)'(\mathbf{Z} - I_T) \otimes I_n)\mu \\
&= n\left[\frac{\sigma}{(a\sigma + 1)^2}(T + 2a\sigma + (a\sigma)^2)\right] + \frac{(a\sigma)^2}{(a\sigma + 1)^2}\mu'\left[L(\frac{1}{T}\mathbf{1}\mathbf{1}') \otimes I_n\right]\mu
\end{aligned}$$

where $\sigma = \frac{\mathrm{tr}(\boldsymbol{\Sigma})}{T}$, we have used the idempotency of matrix $L(\frac{1}{T}\mathbf{1}\mathbf{1}')$ and omitted some tedious algebra in the last step. The derivative of this expression with respect to a is given by

$$\frac{\delta R((\mathbf{Z} \otimes I_n)\hat{\mathbf{x}}, \mu)}{\delta a} = \frac{2\sigma^2[(1 - T)n + a\mu'\left[L(\frac{1}{T}\mathbf{1}\mathbf{1}') \otimes I_n\right]\mu]}{(a\sigma + 1)^3}. \tag{10}$$

In order for this expression to be equal to zero, the numerator must be zero. The result follows.

References

1. Argyriou, A., Maurer, A., Pontil, M.: An algorithm for transfer learning in a heterogeneous environment. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 71–85. Springer, Heidelberg (2008)
2. Gretton, A., Sriperumbudur, B., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K.: Optimal kernel choice for large-scale two-sample tests. In: Advances in Neural Information Processing Systems 25, pp. 1214–1222 (2012)
3. Efron, B., Morris, C.N.: Stein's paradox in statistics. Scientific American 236(5), 119–127 (1977)
4. Feldman, S., Gupta, M.R., Frigyik, B.A.: Multi-task averaging. In: Advances in Neural Information Processing Systems 25, pp. 1178–1186 (2012); MMM This paper seems obscure. can you add volume, page number etc? I would otherwise remove this citation unless strictly needed PPP
5. He, X., Mourot, G., Maquin, D., Ragot, J.: One-class SVM in multi-task learning. In: Advances in Safety, Reliability and Risk Management (2012)
6. James, W., Stein, C.: Estimation with quadratic loss. In: Proc. Fourth Berkeley Symposium on Mathematical Statistics and Probability, pp. 361–379 (1961)
7. Lehmann, E.L., Casella, G.: Theory of Point Estimation. Springer, New York (1998)
8. Maurer, A., Pontil, M., Romera-Paredes, B.: Sparse coding for multitask and transfer learning. In: Proceedings of the 30th International Conference on Machine Learning (2013)
9. Ng, A.Y., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems (2002)
10. Principe, J.C.: Information Theoretic Learning. Renyi's Entropy and Kernel Perspectives. Springer (2000)
11. Parzen, E.: On Estimation of a Probability Density Function and Mode. Annals of Mathematics and Statistics 33(3), 1065–1076 (1962)
12. Stein, C.: Inadmissibility of the usual estimator for the mean of a multivariate distribution. In: Proc. Third Berkeley Symposium on Mathematical Statistics and Probability, pp. 197–206 (1956)
13. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
14. von Luxburg, U.: A Tutorial on Spectral Clustering. Statistics and Computing 17(4), 395–416 (2007)
15. Yang, H., King, I., Lyu, M.R.: Multi-task Learning for One-class Classification. In: International Joint Conference on Neural Networks (2010)