Correction Note on the Results of Multi-task Gaussian Process Prediction

Edwin V. Bonilla, Kian Ming A. Chai, Christopher K. I. Williams
School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, UK
edwin.bonilla@ed.ac.uk, K.M.A.Chai@sms.ed.ac.uk, c.k.i.williams@ed.ac.uk

January 13, 2009

Abstract

This note rectifies the results presented in section 6 of the publication "Multi-task Gaussian Process Prediction" [1] regarding the compiler and school applications.

1 Introduction

In [1], the method "Multi-task Gaussian Process Prediction" is proposed as a technique for exploiting transference across different machine learning tasks. This technique is evaluated on two different applications: a) the compiler data and b) the school data (see section 6 of the original paper). This note provides a correction of the results on these applications.

2 New Results

Here we present the updated results for the compiler and school data. Although on average across all tasks the benefits of multi-task Gaussian process for the compiler application seem to have diminished, this method either improves on or provides the same performance as the "no transfer" method on most tasks. The superior performance of this method over the "no transfer" case still holds on the school data. Furthermore, multi-task GP has been proved successful for applications where the assumed covariance decomposition occurs naturally. For example, recently Chai et al. [2] have applied Multi-task Gaussian processes to the problem of learning robot inverse dynamics and have shown clear advantages of this method over the "no transfer" scenario.

2.1 Compiler Data

We have come to realize that the results presented for the compiler application in [1, Figure 1] were not all based on the same partitioning of the training and test data. The MAEs (mean absolute errors) of the "no transfer" and "transfer parametric" (dotted and dashed lines of [1, Figure 1]) methods were taken from the previously published results of [3, Figure 3]. In contrast, the MAEs for "transfer free-form" were obtained with a different partition. Consequently, it is difficult to determine if the difference in performance of the methods is due to the different training partitions used or due to the methods themselves.

Figure 1 shows the new results when the "no transfer" and "transfer free-form" methods use the same data partitioning, which is a block-design, i.e. where the input locations are the same across all tasks, with the input locations (for each replication) drawn at random. This is the same partitioning used in [1, Figure 1] for the "transfer free-form" technique, so the dotted line corresponding to the "no transfer" scenario is the only one that has changed. Note that, by definition, the "transfer parametric" method uses a different partitioning, as the samples selected are based on the *canonical responses* (see [3] for details). However, a paired evaluation of this method with respect to the "no

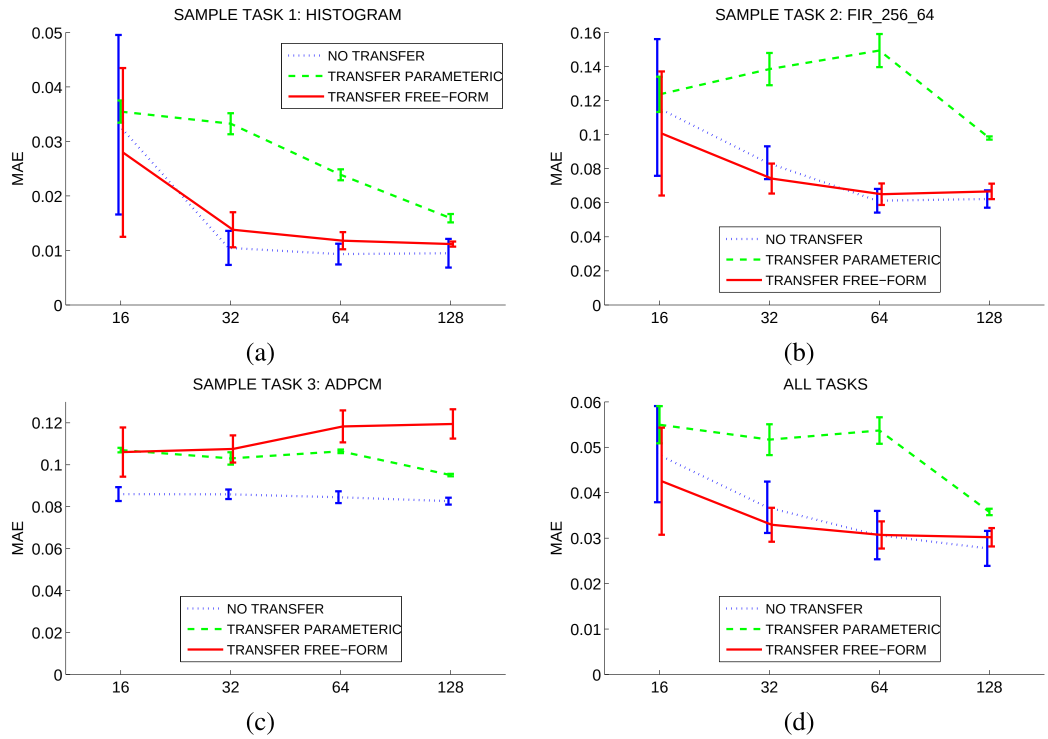

Figure 1: Panels (a), (b) and (c) show the average mean absolute error on the compiler data as a function of the number of training points for the sample tasks presented in [1]. *no transfer* stands for the use of a single GP for each task separately; *transfer parametric* is the use of a GP with a joint parametric (SE) covariance function as in [3]; and *transfer free-form* is multi-task GP with a "free form" covariance matrix over tasks. The error bars show $\pm$ two standard errors taken over the 10 replications. Panel (d) shows the average MAE over all 11 tasks, and the error bars show the average of the two standard errors over all 11 tasks. Note that the plots have been slightly offset horizontally so that the error bars can be easily distinguished.

transfer" scenario is presented in [3] by also showing the performance of the "no transfer" method when using only the canonical responses. For consistency with the results presented in [1, Figure 1], Figure 1 also shows the performance of the "transfer parametric" method.

We see in Figure 1 that (as in [1, Figure 1]) for *sample task 3* the "transfer free-form" method degrades performance, although all the methods perform similarly. Additionally, (unlike [1, Figure 1]) for *sample task 1* and *sample task 2* "transfer free-from" does not perform significantly better than "no transfer". On average over all 11 tasks, there is only a very small difference between these two methods. However, by examining the results on the remaining benchmarks in Figure 2, we see that, in general, the "transfer free-form" method either improves or provides the same performance than the "no transfer" method.

Results on a Non-block Design

As shown in [1, sec 2.3], a cancellation of inter-task transfer occurs when having noiseless observations and a full-block design. Hence, we may expect better benefits from multi-task GP by using a non-block design. Figure 3 shows the standardised mean absolute error (SMAE) for a non-block training sample design of the compiler data across 30 replications, where we have standardised the mean absolute error by the MAE obtained with the the simple median predictor of the test data. We see that "transfer free-form" clearly outperforms "no transfer" on four benchmarks (*iir*, *latnrm*, *lpc* and *spectral estimation*) and, on average across all tasks multi-task Gaussian process prediction outperforms the single task (no transfer) scenario (see bottom right plot in Figure 3).

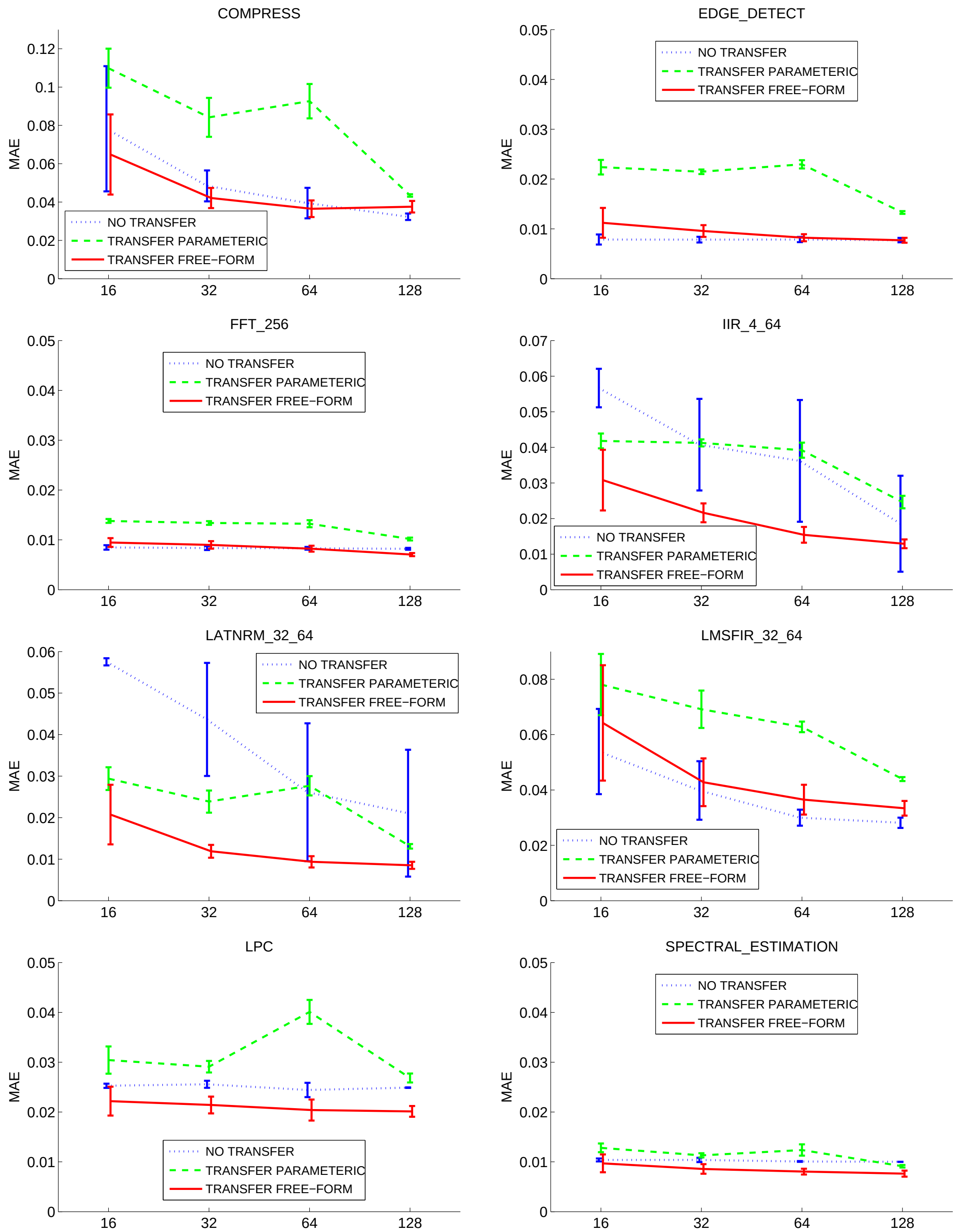

Figure 2: The average mean absolute error on the compiler data as a function of the number of training points for the remaining tasks. *no transfer* stands for the use of a single GP for each task separately; *transfer parametric* is the use of a GP with a joint parametric (SE) covariance function as in [3]; and *transfer free-form* is multi-task GP with a "free form" covariance matrix over tasks. The error bars show $\pm$ two standard errors taken over the 10 replications. Note that the plots have been slightly offset horizontally so that the error bars can be easily distinguished.

2.2 School Data

We have recently found that in our setup for the school data we used a single mean across all tasks for pre-processing of the target values (exam scores). This has a significant impact on the final

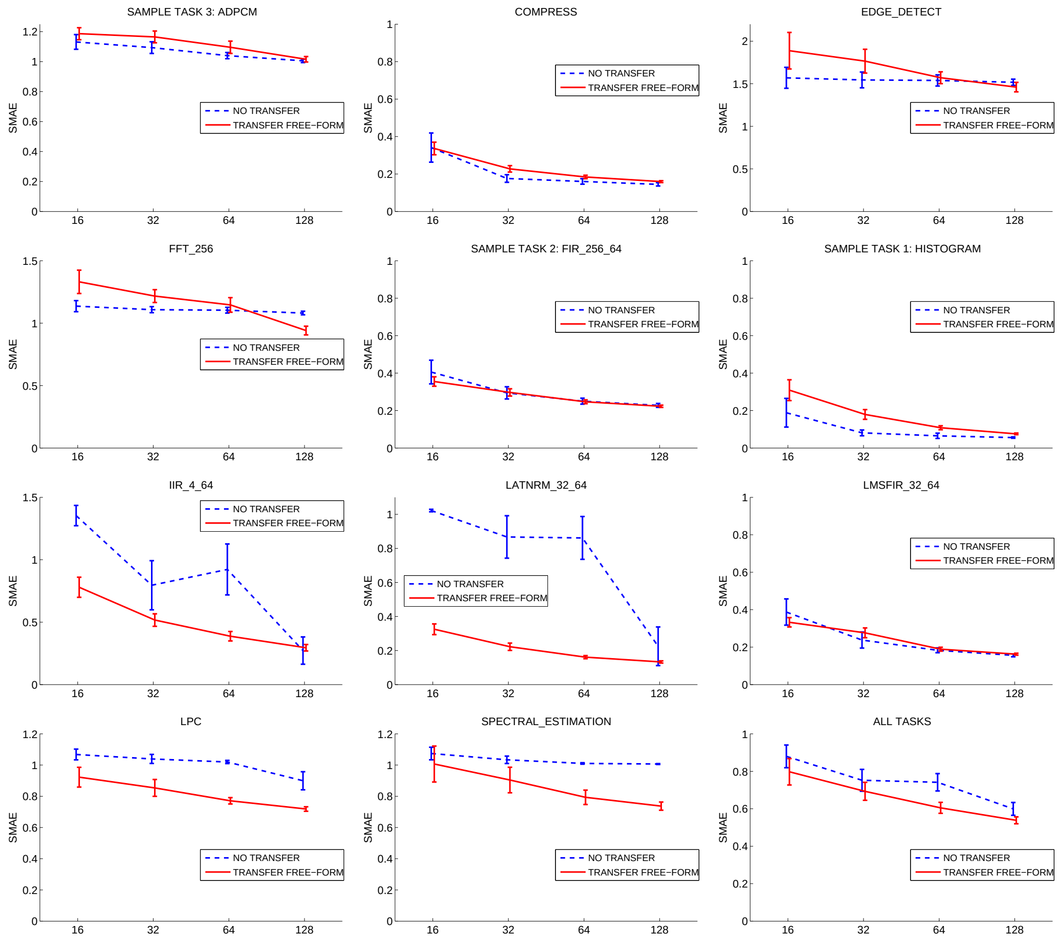

Figure 3: The average standardised mean absolute error (SMAE) when using a non-block design on the compiler data as a function of the number of training points. *no transfer* stands for the use of a single GP for each task separately and *transfer free-form* is multi-task GP with a "free form" covariance matrix over tasks. The error bars show $\pm$ two standard errors taken over the 30 replications. Note that the plots have been slightly offset horizontally so that the error bars can be easily distinguished.

results, and using task-specific means leads to an increase in performance. The updated results are given in Table 1. Here we see that multi-task GP consistently outperforms the "no transfer" method

no transfer	parametric	rank 1	rank 2	rank 3	rank 5
31.12 (1.33)	35.83 (1.09)	34.95 (0.95)	36.16 (0.99)	35.54 (1.08)	33.44 (1.32)

Table 1: Percentage variance explained on the school dataset for various situations. The figures in brackets are standard deviations obtained from the ten replications.

and therefore, for this particular application, there is benefit from transfer across the different tasks.

3 Conclusion

Although it is unfortunate that the results presented in [1] have changed significantly, we have shown that multi-task GP provides equal or better performance than the "no transfer" method for most tasks on the compiler application and that it consistently outperforms the single-task learning scenario on the school data. Finally, multi-task GP has also been shown to have a significant and positive impact in applications where the assumed covariance decomposition occurs naturally [2].

References

[1] Edwin V. Bonilla, Kian Ming A. Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *In Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.

[2] Kian Ming A. Chai, Christopher K. I. Williams, Stefan Klanke, and Sethu Vijayakumar. Multi-task Gaussian process learning of robot inverse dynamics. In *Advances in Neural Information Processing Systems 21*. 2009. To appear.

[3] Edwin V. Bonilla, Felix V. Agakov, and Christopher K. I. Williams. Kernel Multi-task Learning using Task-specific Features. In *Proceedings of the 11th AISTATS*, March 2007.