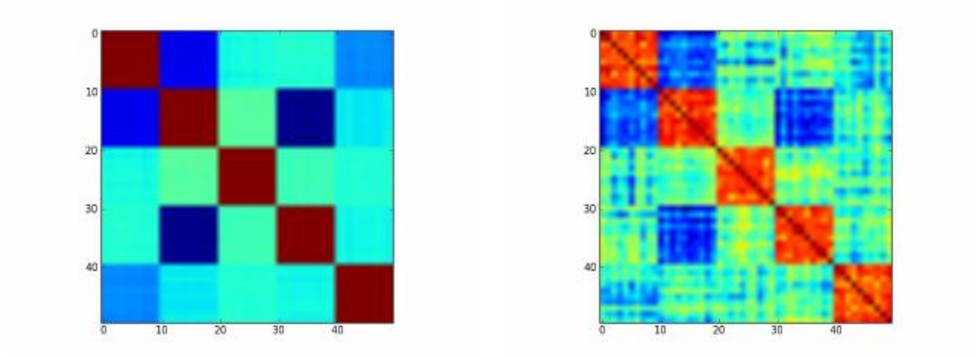

Figure 4. Left: Plot of the correlation matrix of the ground-truth weight vectors of the 50 tasks. Right: Inferred correlation matrix

- Independently learned tasks - **STL**: assumes the tasks are independent (no information sharing).
- Multitask Feature Learning - **MTFL**: assumes the tasks share a common set of features (Argyriou et al., 2007).
- Shared Gaussian prior over the weight vectors - **PRIOR** (Chelba & Acero, 2006): assumes the tasks are drawn from a shared Gaussian prior with a unknown but fixed mean and covariance.
- Single shared subspace - **RANK** (Zhang et al., 2006; Rai & Daumé III, 2010): assumes the tasks live close to a linear subspace (also equivalent to the matrix of the weight vector being low-rank).
- DP mixture model based task clustering - **DP-MTL** (Xue et al., 2007): assumes the weight vectors are generated from a mixture model, each component being a full-rank Gaussian.
- **L**earning with **W**hom to **S**hare - **LWS** (Kang et al., 2011). It is an integer programming based method that learn the task grouping structure (with pre-specified number of groups) and encourages the tasks within each group to share features.

Of these baselines, MTFL and LWS were used for regression problems only since the publicly available implementations are for regression. In the experiments, we would refer to our model as **MFA-MTL** (**M**ixture of **F**actor **A**nalyzers for **M**ulti**T**ask **L**earning). In all our experiments, we set the hyperparameters $\alpha_1 = 1$ and $\alpha_2 = 5$, as these values performed reasonably in preliminary experiments. The truncation level for the DP can be chosen to be equal to the number of tasks T, and for the IBP, to be the minimum of T and the number of features D in the data. This is often more than necessary and in most of our experiments, much smaller truncation levels were found to be sufficient.

For our multitask regression experiments, we compared MFA-MTL with STL, MTFL, and LWS (we skip the other baselines as they performed comparably or worse than MTFL/LWS). For this experiment, we used three datasets - one synthetic dataset used in (Kang et al., 2011), and two real-world datasets used commonly in the multitask learning literature: (1) **School**: This dataset consists of the examination scores of 15362 students from 139 schools in London. Each school is a task so there are a total of 139 tasks for this dataset. (2) **Computer**: This dataset consists of a survey of 190 students about the chances of purchasing 20 different personal computers. There are a total of 190 tasks, 20 examples per task, and 13 features per example. For the synthetic data, we followed the similar procedure for train/test split as used by (Kang et al., 2011). For School and Computer datasets, we split the data equally into training and test set and further only used 20% of the training data (training set deliberately kept small as is often the case with multitask learning problems in practice). The average mean squared errors (i.e., across tasks) in predicting the responses by each method are shown in Table 1. As shown in Table 1, MFA-MTL outperforms the other baselines on all the datasets. Moreover, for the synthetic data, we found that it also inferred the number of task groups (3) correctly (the LWS baseline needs this number to be specified - we ran it with the ground truth). On the school and computer datasets, MFA-MTL outperforms STL and LWS and does slightly better than MTFL. For LWS on these two datasets, we report the best results as obtained by varying the number of groups from 1 to 20.

	Synthetic	**School**	**Computer**
STL	1.35	468.7	153.3
MTFL	0.36	376.1	30.4
LWS	0.37	430.9	30.2
MFA-MTL	**0.18**	**374.5**	**29.8**

Table 1. Mean squared error (MSE) of various methods on multitask regression problems

	Landmine	**20ng**
STL	52.9%	69.3%
PRIOR	52.9%	75.8%
RANK	53.8%	75.8%
DP-MTL	53.8%	75.7%
MFA-MTL	**62.4%**	**76.9%**

Table 2. Multitask classification accuracies of various methods on the **Landmine** and **20ng** datasets

We next experiment with the classification setting. For this, we chose two datasets: (1) **Landmine**: The landmine detection dataset is a subset of the dataset used in the symmetric multitask learning experiment by (Xue et al., 2007). It contains 19 classification tasks and the tasks are known to be clustered for this data. (2) **20ng:** We did the standard training/test split of 20 Newsgroups for multitask learning, following Raina et al. (2006), and used a 50/50 split for the landmine data. The classification accuracies reported by our