Feature Learning for Multitask Learning Using $l_{2,1}$ Norm Minimization

Priya Venkateshan

Abstract

We look at solving the task of Multitask Feature Learning by way of feature selection. We find that formulating it as an $l_{2,1}$ norm minimization problem helps with both sparsity and uniformity, which are required for Multitask Feature Learning. Two approaches are explored, one which formulates an equivalent convex optimization problem and iteratively solves it, and another which formulates two equivalent smooth convex optimization problems, one using additional variables, and another constraining the problem with the $l_{2,1}$ ball, and solves these problems using first-order methods like Nestorov's method for increased efficiency. We look at the accuracy and speed of these algorithms.

1 Introduction

Multi-task learning is an approach to machine learning, that learns a problem together with other related problems at the same time, using a shared representation. This often leads to a better model for the main task, because it allows the learner to use the commonality among the tasks.
Tasks can be related in various ways. One way task relatedness can be modelled is by assuming that functions for all tasks are close to each other in some norm. Task relatedness can also be modelled by assuming that all the given tasks share a common underlying representation. For example, in modelling user preferences in collaborative filtering applications, predicting user preferences can be considered a separate task for each user in the system, and we can intuitively say that all these tasks are related as people make product choices using a common set of features describing these products. Additionally, multi-task learning comes of use when there are very few training examples for a given task, and training data for related tasks with the same feature representation can be pooled together in order to improve performance on all the tasks. We can learn a sparse representation underlying all the related tasks using the $l_{2,1}$ norm. As both sparsity as well as uniformity among the representations learned is desired in this task, the $l_{2,1}$ norm is ideal, because a small l_1 norm favours sparsity and small l_2 norm favours uniformity. Thus the $l_{2,1}$ norm encourages multiple predictors across tasks to share similar parameter sparsity patterns.
Here we look at two approaches for formulating the multitask feature learning problem as convex optimization problems that involve minimizing the $l_{2,1}$ norm. The first approach introduces the idea of reformulating the error function for multitask learning as a convex optimization problem in two variables and suggests an iterative algorithm for solving it. The second approach suggests two smooth reformulations of the nonsmooth $l_{2,1}$ norm which can then be solved by Nesterov's method, which being a first-order method, converges more quickly than iterative methods.

2 Feature Learning through $l_{2,1}$ Norm Minimization

Let $\Re$ be the set of real numbers, and $\Re_{++}$ the subset of non-negative ones. Let T be the number of tasks, and $N_T := 1...T$. For each task $t \in N_T$, we are given m input-output examples $(x_{t1}, y_{t1})...(x_{tm}, y_{tm}) \in \Re^d X \Re$. Given this data, we try to estimate T functions $f_t : \Re^d \rightarrow \Re, t \in N_t$.

The underlying assumption of this paper is that all the functions f_t are related, and they share a small set of features.

$$f_t(x) = \sum_{i=1}^{d} a_{it} h_i(x), t \in N_T ......(1.1)$$

where $h_i : \Re^d \to \Re$ are the features and $a_{it} \in \Re$ are the regression parameters. Our main assumption is that all the features but a few have zero coefficients across all the tasks.

For simplicity, we focus on linear features, that is, $h_i(x) =< u_i, x >$, where $u_i \in \Re$. In addition, we assume that the vectors u_i are orthonormal. Thus, if U denotes the dXd matrix with columns the vectors u_i, then U is orthogonal. The functions f_t are linear as well, that is $f_t(x) =< w_t, x >$, where $w_t = \sum_i a_{it} u_i$. Let us denote by W the dXT matrix whose columns are the vectors w_t and by A the dXT matrix with entries a_{it}. We then have that $W = UA$. Our assumption that the tasks share a small set of features means that the matrix A has many rows which are identically equal to zero and, so,the corresponding features (columns of matrix U) will not be used to represent the task parameters(columns of matrix W). In other words, matrix W is a low rank matrix. In the following, we describe our approach to computing the feature vectors u_i and the parameters a_{it}. We first consider the case that there is only one task (say task t) and the features ui are fixed. To learn the parameter vector $a_t \in \Re^d$ from data $\{(x_{ti}, y_{ti})\}_{i=1}^m$ we would like to minimize the empirical error $\sum_{i=1}^m L(y_{ti}, < a_t, U^T x_{ti} >)$ subject to an upper bound on the number of nonzero components of at, where $L : \Re X \Re \to \Re_+$ is a prescribed loss function which we assume to be convex in the second argument. This problem is intractable and is often relaxed by requiring an upper bound on the 1-norm of a_t. That is, we consider the problem $min\{\sigma_{i=1}^m L(y_{ti}, < a_t, U^T x_{ti} >) : ||a_t||_1^2 \leq \alpha^2\}$, or equivalently the unconstrained problem

$$min\{\sum_{i=1}^{m} L(y_{ti}, < a_t, U^T x_{ti} >) + \gamma \, ||\alpha_t||_1^2 : a_t \in \Re^d\} ....(1.2)$$

where $\gamma > 0$is the regularization parameter. It is well known that using the 1-norm leads to sparse solutions, that is, many components of the learned vector at are zero. Moreover, the number of nonzero components of a solution to problem (1.2) is typically a nonincreasing function of γ. We now generalize problem (1.2) to the multi-task case. For this purpose, we introduce the regularization error function

$$\varepsilon(A, U) = \sum_{t=1}^{T} \sum_{i=1}^{m} L(y_{ti}, < a_t, U^T x_{ti} >) + \gamma \, ||A||_{2,1}^2$$

(1.3) The first term in (1.3) is the average of the empirical error across the tasks while the second one is a regularization term which penalizes the (2, 1)-norm of the matrix A. It is obtained by first computing the 2-norm of the (across the tasks) rows a_i (corresponding to feature i) of matrix $\hat{A}$ and then the 1-norm of the vector $b(\hat{A}) = (||a^1||_2 \cdots ||a^d||_2)$. This norm combines the tasks and ensures that common features will be selected across them. Indeed, if the features U are prescribed and $\hat{A}$ minimizes the function ε over A, the number of nonzero components of the vector $b(\hat{A})$ will typically be non-increasing with like in the case of 1-norm single-task regularization. Moreover, the components of the vector $b(\hat{A})$ indicate how important each feature is and favor uniformity across the tasks for each feature. Since we do not simply want to select the features but also learn them, we further minimize the function ε over U, that is, we consider the optimization problem

$$min\{\varepsilon(A, U) : U orthogonal, A \in \Re^{dXt}\} ....(1.4)$$

This method learns a low-dimensional representation which is shared across the tasks. As in the single-task case, the number of features will be typically non-increasing with the regularization parameter.

2.1 Equivalent convex formulation

Solving problem (1.4) is a challenging task for two main reasons. First, it is a non-convex problem, although it is separately convex in each of the variables A and U. Second, the norm $||A||_{2,1}$ is nonsmooth which makes it more difficult to optimize. A main result in this paper is that problem

(1.4) can be transformed into an equivalent convex problem. To this end, for every $W \in \Re^{dXt}$ and $D \in S^d_+$, we define the function

$$R(W;D) = \sum_{t=1}^{T} \sum i = 1^m L(y_{ti}, < w_t, x_{ti} >) + \gamma \sum_{t=1}^{T}$$

(1.5) Theorem 1. Problem (1.4) is equivalent to the problem

$$min\{R(W,D) : W \in \Re^{dXT}, D \in S^d_+, trace(D) \leq 1, range(W) \subset range(D)\}$$

(1.6) That is, $(\hat{A};\hat{U})$ is an optimal solution for (1.4) if and only if $(\hat{W};\hat{D}) = (\hat{U}\hat{A};\hat{U}Diag(\hat{\lambda})\hat{U} >)$ is an optimal solution for (1.6), where

$$\lambda_i := \frac{\left|\left|a^i\right|\right|_2}{||A||_2} : ....(1.7)$$

Proof. Let $W = UA$ and $D = UDiag\frac{||a^i||_2}{||A||_2}U_T$. Then $\left|\left|a^i\right|\right|_2 = \left|\left|W^T u_i\right|\right|_2$ and hence $\sum_{t=1}^T < w_t, D^+ w_t >= trace(W^T D^+ W) = ||A||_{2,1} trace(W^T U Diag(\left|\left|W^T u_i\right|\right|_2)^+ U^T) = ||A||_{2,1} trace(\sum_{i=1}^d (\left|\left|W^T u_i\right|\right|_2)^+ W^T u_i u_i^T W) = ||A||_{2,1} \sum_{i=1}^d \left|\left|W^T u_i\right|\right|_2 = ||l||^2_{2,1}$.

Therefore, $min_{W,D} R(W,D) \leq min_{(A,U)} \varepsilon(A,U)$. Conversely, let $D = UDiag(\lambda)U^T$. Then

$$\sum_{t=1}^{T} < w_t, D^+ w_t >= trace(W^T U Diag(\lambda_i^+) U^T W) = trace(Diag(\lambda_i^+) A A^T) \geq ||A||^2_{2,1}$$

by Lemma 1. Therefore, $min_{A,U} \varepsilon(A,U) \leq min_(W,D) R_(W,D)$ We note that the rank of matrix D indicates how many common relevant features the tasks share. Indeed, it is clear from equation (1.7) that the rank of matrix D equals the number of nonzero rows of matrix A. We now show that the function R in equation (1.5) is jointly convex in W and D. For this purpose, we define the function $f(w;D) = w^T D^+ w, if D \in S^d_+$ otherwise. Clearly, R is convex provided f is convex. The latter is true since a direct computation expresses f as the supremum of a family of convex functions,

2.2 Learning Algorithm

We solve problem (1.6) by alternately minimizing the function R with respect to D and the w_t (recall that w_t is the t-th column of matrix W). When we keep D fixed, the minimization over wt simply consists of learning the parameters w_t independently by a regularization method, for example by an SVM or ridge regression type method. For a fixed value of the vectors w_t, we learn D by simply solving the minimization problem

$$min \sum_{t=1}^{T} < w_t, D^+ w_t >: D \in S^d_+, range(W) \subseteq range(D)....(1.8)$$

The following theorem characterizes the optimal solution of problem (1.8). Theorem 2. Let $C = WW^T$. The optimal solution of problem (1.8) is

$$D = \frac{C^{\frac{1}{2}}}{trace(c^{\frac{1}{2}})}(1.9)$$

and the optimal value equals $trace(c^{\frac{1}{2}})^2$. We first introduce the following lemma which is useful in our analysis. Lemma 1. For any $b = (b_1...b_d) \in \Re^d$, we have that

$$inf(\sum_{i=1}^{d})\frac{b_i^2}{\lambda_i^2} : \lambda_i > 0, \sum_{i=1}^{d} \lambda_i \leq 1) = ||b||^2_1 \,(1.10)$$

and any minimizing sequence converges to $\lambda_i = \frac{|b_i|}{||n||_1}, i \in N_d$.

Proof of Theorem 1.8. We write $D = UDiag(\lambda)U^T$, with orthogonal U and $\lambda \in \Re^d$ We first minimize over λ. For this purpose, we use Lemma 1 to obtain that

$$inf(trace(W^T UDiag(\lambda)^{-1}U^T W) : \sum_{i=1}^{d} \lambda_i \leq 1) = \left|\left|U^T W\right|\right|_{2,1}^{2} = (\sum_{i=1}^{d} \left|\left|W^T\right|\right|$$

. Next we show that

$$min \left|\left|U^T W\right|\right|_{2,1}^{2} = (traceC^{\frac{1}{2}})^2$$

and a minimizing U is a system of eigenvectors of C. To see this, note that

$$trace(WW^T u_i u_i^T) = trace(C^{\frac{1}{2}} u_i u_i^T C^{\frac{1}{2}} trace(u_i u_i^T u_i u_i^T) \geq trace(C^{\frac{1}{2}} u_i u_i^T) = u_i^T C^{\frac{1}{2}} u_i$$

The equality is verified if and only if $C^{\frac{1}{2}} u_i u_i^T = a u_i u_i^T$ which implies $C^{\frac{1}{2}} u_i = a u_i$, that is, if u_i is an eigenvector of C. The optimal a is $trace(C^{\frac{1}{2}})$. The expression $trace(WW^T)^{\frac{1}{2}}$ in (1.9) is simply the sum of the singular values of W and is sometimes called the trace norm. The trace norm is the convex envelope of rank(W) in the unit ball, which gives another interpretation of the relationship between the rank and in our experiments. Using the trace norm, problem (1.6) becomes a regularization problem which depends only on W. However, since the trace norm is nonsmooth, we have opted for the above alternating minimization strategy which is simple to implement and has a natural interpretation. Indeed, Algorithm 1 alternately performs a supervised and an unsupervised step, where in the latter step we learn common representations across the tasks and in the former step we learn task-specific functions using these representations.

Algorithm 1 Multitask Feature Learning

```
Set $D = \frac{I_{dXd}}{d}$
while Convergence condition is not true do
  for t = 1... T do
    $w_t = argmin\{\sum_{i=1}^{m} L(y_{ti}, < w, x_{ti} >) + \gamma < w, D^{+}w >: w \in \Re^d, w \in range(D)$
  end forset $D = \frac{(WW^T)^{\frac{1}{2}}}{trace(WW^T)^{\frac{1}{2}}}$
end while
```

3 Efficient Minimization using Nestorov's Method

In multi-task learning, we are given a training set of k tasks $\{(a_i^j, y_i^j)\}_{i=1}^{m_j}, j = 1, 2, ..., k$, where $a_i^j \in \Re^n$ denotes the i-th training sample for the j-th task, y_i^j denotes the corresponding output, m_j is the number of training samples for the j-th task, and $m = \sum_{j=1}^{k} mj$ is the total number of training samples (including possible duplicates). Let $A_j = [a_1^j, ..., a_{mj}^j]^T \in \Re^{m_j X n}$ denote the data matrix for the j-th task, $A = [A_1^T, ..., A_k^T]^T \in \Re^{m_j X n}, y_j = [y_1^j....y_{m_j}^j]^T \in \Re^{m_j}$ and $y = [y_1^T, ..., y_k^T]^T \in \Re^m$. In this approach, we consider linear models:

$$f_j(a) = w_j^T a, j = 1, ..., k, (2.1)$$

where $w_j \in \Re^n$ is the weight vector for the j-th task. The weight vectors for all k tasks form the weight matrix $W = [w_1, ..., w_k] \in \Re^{nXk}$, which needs to be estimated from the data. Assume that, given the value of a_j, the corresponding target y_j for the j-th task has a Gaussian distribution with mean $f_j(a_j)$ and precision σ_j ¿ 0 as

$$p(y_j|w_j, a_j, \sigma_j) = \sqrt{\frac{\sigma_j}{2\pi}} exp(\frac{-\sigma^j (y^j - w_j^T a^j)^2}{2})(2.2)$$

Denote $\sigma = [\sigma_1, ..., \sigma_k]^T \in \Re^k$ and assume that the data A, y is drawn independently from the distribution in Eq. (2), then the likelihood function can be written as

$$p(y|W, A, \sigma) = \prod_{j=1}^{k} \prod_{i=1}^{m_j} p(y_i^j|w_j, a_i^j, \sigma_j).(2.3)$$

To capture the task relatedness, a prior on W is defined as follows. The i-th row of W, denoted as $w_i \in \Re^{1Xk}$, corresponds to the i-th feature in all tasks. Assume that w_i is generated according to the exponential prior:

$$p(w_i|\delta^i) \propto exp(||(|| w^i)\delta^i), i = 1, 2, ..., n, (2.4)$$

where $\delta^i > 0$ is the hyperparameter. Note that when k = 1, the prior in Eq. (2.4) reduces to the Laplace prior. Denote $\delta = [\delta^1, ..., \delta^n]^T \in \Re^n$, and assume that $w^1, ..., w^n$ are drawn independently from the prior in Eq. (2.4). Then the prior for W can be expressed as

$$p(W|\delta) = \prod_{i=1}^{n} p(w^i|\delta^j)(2.5)$$

It follows that the posterior distribution for W, which is proportional to the product of the prior and the likelihood function, is given by:

$$p(W|A, y, \sigma, \delta) \propto p(y|W, A, \sigma)p(W|\delta).(2.6)$$

Taking the negative logarithm of Eq. (2.6) and combining with Eqs. (2.1-5), we obtain the maximum posterior estimation of W by minimizing

$$\frac{1}{2}\sum_{j=1}^{k} \sigma^j \left|\left|y_j - A_j w_j\right|\right|^2 + \sum_{i=1}^{n} \delta^t \left|\left|w^j\right|\right| (2.7)$$

For simplicity of discussion, we assume that $\sigma = \sigma^j, \forall j = 1, ..., k$ and $\delta = \delta_i, \forall i = 1, 2, ..., n$. We then obtain from Eq. (2.7) the following $l_{2,1}$-norm regularized least squares regression problem:

$$min_W \frac{1}{2}\sum_{j=1}^{k} \left|\left|y_j - A_j w_j\right|\right|^2 + \rho \left|\left|W\right|\right|_{2,1} (2.8)$$

where $\rho = \delta/\sigma$ and $||W||_{2,1} = \sum_{i=1}^{n} ||w_i||$ is the $l_{2,1}$-norm of the matrix W. The problem in Eq. (2.8) can be generalized to the following $l_{2,1}$-norm regularization problem:

$$min_{W \in \Re^{nXk}} loss(W) + \rho \left|\left|W\right|\right|_{2,1} (2.9)$$

where $\rho > 0$ is the regularization parameter, and loss(W) is a smooth convex loss function such as the least square loss in Eq. (2.8) and the logistic loss. When the number of tasks equals to one, the prior in Eq. (2.4) reduces to the Laplace prior distribution. It is easy to show that in this case the problem in Eq. (2.9) reduces to the l_1-norm regularized optimization problem. When there are multiple tasks, the weights corresponding to the i-th feature are grouped together via thel_2-norm of w_i. Thus, the $l_{2,1}$-norm regularization tends to select features based on the strength of the input variables of the k tasks jointly rather than on the strength of individual input variables as in the case of single task learning.

We propose to employ the Nesterov's method to solve the nonsmooth optimization problem in Eq. (2.9) by optimizing its equivalent smooth convex reformulations. Definition 1. The optimization problem

$$min_{x \in G} g(x)......(2.10)$$

is called a constrained smooth convex optimization problem if G is a closed convex set and g(x) is a convex and smooth function defined in G.

The objective function in Eq. (2.9) is nonsmooth, since $||W||_{2,1}$ is non-differentiable. It is known that the subgradient method for solving this problem requires $O(\frac{1}{\epsilon})$ iterations for achieving an accuracy of ϵ, which is significantly larger than $O(\frac{1}{\epsilon^2})$ required by the Nesterov's method for solving Eq. (2.10). We thus propose to solve the equivalent smooth convex reformulations of Eq. (2.9).

3.1 First Reformulation

We introduce an additional variable $t = [t1, t2, ..., tn]^T$, where t_i acts as the upper-bound of $\left|\left|w^T\right|\right|$. With t, we give the reformulation in the following theorem (proof given in Appendix A): *Theorem*

1 Let loss(W) be a smooth convex loss function. Then the problem in Eq. (2.9) is equivalent to the following constrained smooth convex optimization problem:

$$min_{(t,W)\in D)} loss(W) + \rho \sum_{i=1}^{n} t_i (2.11)$$

where $t = [t1, t2, ..., tn]^T$ and

$$D = \{(t, W)| \left|\left|w^i\right|\right| \leq ti, \forall i = 1, 2, ..., n\}(2.12)$$

is close and convex.

3.2 Second Reformulation

We can also derive an equivalent smooth reformulation by moving the nonsmooth $l_{2,1}$-norm term to the constraint. This results in the following equivalent reformulation (proof given in Appendix B): *Theorem 2*: Let loss(W) be a smooth convex loss function. Then the problem in Eq. (2.9) is equivalent to the following $l_{2,1}$-ball constrained smooth convex optimization problem:

$$min_{W\in Z} loss(W)(2.13)$$

where

$$Z = W \in \Re^{nXk} | \, ||W||_{2,1} \leq z, (2.14)$$

$z \geq 0$ is the radius of the $l_{2,1}$-ball, and there is a one-to-one correspondence between ρ and z. For simplicity of presentation, we focus on solving the general problem (2.10), which is a constrained smooth convex optimization problem.

3.3 Main Algorithm

We employ the Nesterov's method for solving (2.10). Recall that the Nesterov's method has a much faster convergence rate than the traditional methods such as subgradient descent and gradient descent. Specifically, the Nesterov's method has a convergence rate of $O(\frac{1}{d^2})$, while gradient descent and subgradient descent have convergence rates of $O(\frac{1}{d})$ and $O(\frac{1}{\sqrt{d}})$, respectively, where d denotes the number of iterations. The Nesterov's method is based on two sequences $\{x_i\}$ and $\{s_i\}$ in which $\{x_i\}$ is the sequence of approximate solutions, and $\{s_i\}$ is the sequence of search points. The search point s_i is the affine combination of x_{i-1} and x_i as

$$s_i = x_i + \alpha_i(x_i - x_{i-1}), (2.15)$$

where α_i is the combination coefficient. The approximate solution x_{i+1} is computed as a "gradient" step of s_i as

$$x_{i+1} = \pi_G(s_i = \frac{1}{\gamma_i} g'(s_i))(2.16)$$

where $\pi_G(v)$ is the Euclidean projection of v onto the convex set G:

$$\pi_G(v) = min_{x\in G} \frac{1}{2} ||x - v||^2 \, (2.17)$$

$\frac{1}{\gamma_i}$ is the stepsize, and γ_i is determined by the line search according to the Armijo-Goldstein rule so that γ_i should be appropriate for s_i.

3.4 Time Complexity

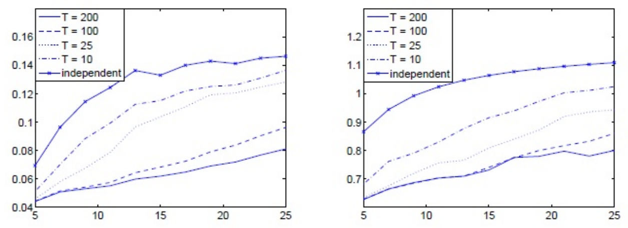
We first analyze the time complexity of the problem in Eq. (2.11). When loss(W) is either the least squares loss or the logistic loss, it costs O(mn) floating point operations (flops) for evaluating the function value and gradient of the objective function of (2.11) at each iteration. The Euclidean projections onto D can be analytically computed in a time complexity of O(nk). Therefore, through the equivalent reformulation (2.11), we can solve the $l_{2,1}$-norm regularized problem in Eq. (2.9) with a time complexity of $O(\frac{1}{\sqrt{\epsilon}}(mn + nk))$, where m, n, k and ϵ denote the total number of training samples, the sample dimensionality, the number of tasks, and the desired accuracy, respectively. Following a similar analysis, the time complexity for solving Eq. (2.9) via the reformulation in Eq. (2.13) is also $O(\frac{1}{\sqrt{\epsilon}}(mn + nk))$.

Algorithm 2 Algorithm 1 Nesterov's Method for Constrained Smooth Convex Optimization

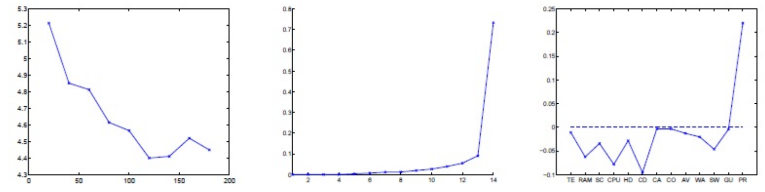
```
Require: g(.), g'(.), G, $L_0 > 0, x_0$
Ensure: : $x$
  Initialize $x_1 = x_0, t_1 = 0, t_0 = 1, \gamma_0 = L_0$
  for i = 1 to . . . do
    Set $\alpha_i = \frac{t_{i-2}-1}{t_{i.1}}, s_i = x_i + \alpha_i(x_i - x_{i.1})$
    for j = 0 to . . . do
      Set $\gamma = 2^j \gamma_{i-1}$
      Compute $x_{i+1} = \pi_G(s_i - \frac{1}{\gamma} g'(si))$
      if $g(x_{i+1}) \leq g_{\gamma,s_i}(x_{i+1})$ then
        $\gamma_i = \gamma$, break
      end if
    end for
    Set $t_i = (1 + \sqrt{1 + 4t_{t-1}^2})/2$
    if convergence then
      x = xi, terminate
    end if
  end for
```

4 Experiments and Results

The authors of these approaches have conducted a series of experiments to determine the efficiency and accuracy of their methods.



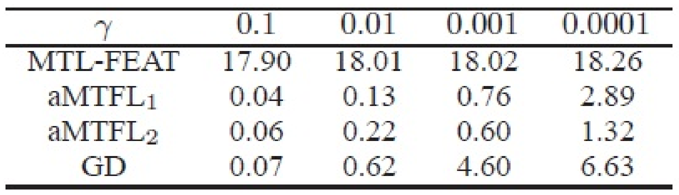
Test error (left) and residual of learned features (right) vs. dimensionality of the input.



Test error vs. number of tasks (left) for the computer survey data set. Significance of features (middle) and attributes learned by the most important feature (right).

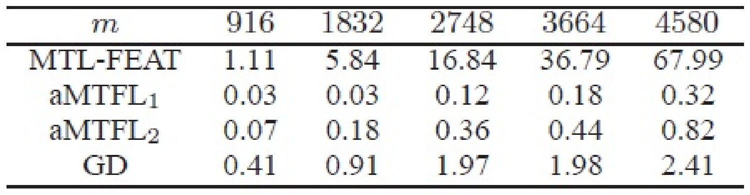
The first approach has been tested on two different datasets - One which used people's ratings of products. The data was taken from a survey of 180 people who rated the likelihood of purchasing one of 20 different PCs. The persons correspond to the tasks, and the PC models to examples. There are 13 binary attributes representing the input. The output is an integer rating on a scale of 0-10. 4 examples per task was used as the test data, and 8 for training. The first basic approach shows improved performance as the number of tasks increases. It also performs much better than independent ridge regressions. Additionally, attention was paid to the most important features shared by all persons. It turned out that the technical characcteristics of a computer like its RAM, CPU and CD-ROM were weighted against its price, going by the features selected by the algorithm.

The experiments conducted by the authors of the second approach were more focussed on establishing the efficiency of their approach compared to the first approach. For this, they used the School



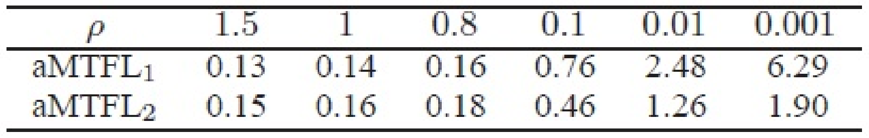
γ	0.1	0.01	0.001	0.0001
MTL-FEAT	17.90	18.01	18.02	18.26
aMTFL$_1$	0.04	0.13	0.76	2.89
aMTFL$_2$	0.06	0.22	0.60	1.32
GD	0.07	0.62	4.60	6.63

Comparison of the computational time (in seconds) among MTL-FEAT, the proposed two reformulations, and GD for the $\ell_{2,1}$-norm regularized least squares regression on the School data set.



m	916	1832	2748	3664	4580
MTL-FEAT	1.11	5.84	16.84	36.79	67.99
aMTFL$_1$	0.03	0.03	0.12	0.18	0.32
aMTFL$_2$	0.07	0.18	0.36	0.44	0.82
GD	0.41	0.91	1.97	1.98	2.41

Comparison of the computational time (in seconds) among MTL-FEAT, the proposed two reformulations, and GD for the $\ell_{2,1}$-norm regularized least squares regression on the Letter data set.



ρ	1.5	1	0.8	0.1	0.01	0.001
aMTFL$_1$	0.13	0.14	0.16	0.76	2.48	6.29
aMTFL$_2$	0.15	0.16	0.18	0.46	1.26	1.90

Comparison of the computational time (in seconds) between aMTFL$_1$ and aMTFL$_2$ for the $\ell_{2,1}$-norm regularized least squares regression on the School data set.

dataset and the Letter dataset. The School dataset consists of scores of 15,362 students from 139 secondary schools in London, with each sample having 28 attributes. The School dataset has 139 tasks of predicting student performance in each school. The Letter dataset consists of 8 defaut tasks of 2-class classification problems for the handwritten letters c/e, g/y, m/n, a/g, i/j, a/o, f/t, h/n. The writings are from 180 different writers, and has a total of 4,679 samples. As we can see in the table given, the both reformulations of the second approach (aMTFL1 and aMTFL2) are more efficient than the first approach (MTL-FEAT). Additionally, the two reformulations were compared with each other. When ρ is large, the solution is sparse, and the optimal value for aMTFL1 is comparable to that for aMTFL2, due to which the two perform similarly. However, when ρ is relatively small, the solution is less sparse and the optimal value for aMTFL1 is much larger than for aMTFL2, due to which aMTFL1 is less efficient.

Though the performances have been compared, the accuracies of these two different approaches have not been compared previously. We attempted to do so on the school dataset. We obtained the code for the first approach from the authors' website, as well as the School dataset. The code for the second reformulation of the second approach was implemented by us in MATLAB, and for Nesterov's Method, we used the package NESTA available online. We tried both approaches on a training set of 50 tasks and test set of 25. We found that though slower, the first approach outperforms the second. MTL-FEAT manages to achieve an accuracy of 84.38% across three trials, whereas aMTFL2 manages to achieve only an accuracy of 62.21% on the same.

5 Discussion and Conclusions

This project gave us an opportunity to explore Feature Learning for Multitask Learning, as well as a glimpse of how real-world problems are cast as convex optimization problems. Additionally, it was interesting to be introduced to first-order methods in convex optimization.
From the perusal of this literature, we have learned that there exist efficient methods for Multitask learning using Feature Learning, by minimizing a $l_{2,1}$ representation of the task. Though Nesterov's Method is supposed to trade-off accuracy for efficiency, we find that its usage still manages to give acceptable levels of accuracy.
From a Multitask Learning perspective, it might be interesting to extend these methods for hierarchical sets of tasks. It would also be interesting from an Optimization perspective to see if performance improves for the second set of reformulations if adaptive line search methods are used.

References

[1] Jun Liu, Shuiwang Ji, Jieping Ye , Multi-Task Feature Learning Via Efficient $\ell_{2,1}$-Norm Minimization, *UAI*

[2] Andreas Argyriou, Theodoros Evgeniou, Massimiliano Pontil (2006) Multi-Task Feature Learning *NIPS*

[3] Code for MTL-FEAT and School dataset http://www.cs.ucl.ac.uk/staff/A.Argyriou/code/

[4] Code for Nestorov's Method: http://www.acm.caltech.edu/ nesta/

5.0.1 Appendix A. Proof of Theorem 1

We begin with a lemma, which states that D defined in (12) is a closed convex set. Lemma 1. The set D defined in (12) is closed and convex. Proof:

$$Denote D_i = (t_i, w^i) | \left|\left|w^i\right|\right|_{2,1} \leq t_i, i = 1, 2, ..., n,$$

which is known as the ice-cream cone. Since the icecream cone is closed and convex,D_i is a closed convex cone. As the Cartesian product of closed convex cones is a closed convex cone, we conclude that $D = D1XD2X...XDn$ is a closed convex cone, and a closed convex set as well. We are now ready to prove Theorem 1. The objective function of (11) is smooth and convex, as both loss(W) and $\rho \sum_{i=1}^{n} t_i$ are smooth and convex. Moreover, from Lemma 1, D is a closed convex set. Therefore, (11) is a constrained smooth convex optimization problem. The problem in (9) is equivalent to (11), since at the optimal point of (11), we have $t_i = \left|\left|w^i\right|\right|, \forall i = 1, 2, ..., n.$

5.0.2 Appendix B. Proof of Theorem 2

We first prove that (13) is a constrained smooth convex optimization problem. It is easy to verify that $||W||_{2,1}$ is a norm. Since any norm is a closed convex function, we conclude that $||W||_{2,1}$ is a closed convex function. As the sublevel set of a closed convex function is either empty or a closed convex set, we conclude that

$$Z = \{W| \, ||W||_{2,1} < z\}$$

is closed and convex (note that, Z is not empty, since z ¿ 0 and the zero matrix belongs to Z). We conclude that (13) is a constrained smooth convex optimization problem. The equivalence relationship between (9) and (13) follows from the Lagrangian duality.