Efficient Output Kernel Learning for Multiple Tasks

Authored by:

Matthias Hein
Bernt Schiele
Maksim Lapin
Pratik Jawanpuria

Abstract

The paradigm of multi-task learning is that one can achieve better generalization by learning tasks jointly and thus exploiting the similarity between the tasks rather than learning them independently of each other. While previously the relationship between tasks had to be user-defined in the form of an output kernel, recent approaches jointly learn the tasks and the output kernel. As the output kernel is a positive semidefinite matrix, the resulting optimization problems are not scalable in the number of tasks as an eigendecomposition is required in each step. Using the theory of positive semidefinite kernels we show in this paper that for a certain class of regularizers on the output kernel, the constraint of being positive semidefinite can be dropped as it is automatically satisfied for the relaxed problem. This leads to an unconstrained dual problem which can be solved efficiently. Experiments on several multi-task and multi-class data sets illustrate the efficacy of our approach in terms of computational efficiency as well as generalization performance.

1 Paper Body

Multi-task learning (MTL) advocates sharing relevant information among several related tasks during the training stage. The advantage of MTL over learning tasks independently has been shown theoretically as well as empirically [1, 2, 3, 4, 5, 6, 7]. The focus of this paper is the question how the task relationships can be inferred from the data. It has been noted that naively grouping all the tasks together may be detrimental [8, 9, 10, 11]. In particular, outlier tasks may lead to worse performance. Hence, clustered multi-task learning algorithms [10, 12] aim to learn groups of closely related tasks. The information is then shared only within these clusters of tasks. This corresponds to learning the task covariance matrix, which we denote as the output kernel in this paper.

Most of these approaches lead to non-convex problems. In this work, we focus on the problem of directly learning the output kernel in the multi-task learning framework. The multi-task kernel on input and output is assumed to be decoupled as the product of a scalar kernel and the output kernel, which is a positive semidefinite matrix [1, 13, 14, 15]. In classical multi-task learning algorithms [1, 16], the degree of relatedness between distinct tasks is set to a constant and is optimized as a hyperparameter. However, constant similarity between tasks is a strong assumption and is unlikely to hold in practice. Thus recent approaches have tackled the problem of directly learning the output kernel. [17] solves a multi-task formulation in the framework of vector-valued reproducing kernel Hilbert spaces involving squared loss where they penalize the Frobenius norm of the output kernel as a regularizer. They formulate an invex optimization problem that they solve optimally. In comparison, [18] recently proposed an efficient barrier method to optimize a generic convex output kernel learning formulation. On the other hand, [9] proposes a convex formulation to learn low rank output kernel matrix by enforcing a trace constraint. The above approaches [9, 17, 18] solve the resulting optimization problem via alternate minimization between task parameters and the output kernel. Each step of the alternate minimization requires an eigen1

value decomposition of a matrix having as size the number of tasks and a problem corresponding to learning all tasks independently. In this paper we study a similar formulation as [17]. However, we allow arbitrary convex loss functions and employ general p-norms for p ? (1, 2] (including the Frobenius norm) as regularizer for the output kernel. Our problem is jointly convex over the task parameters and the output kernel. Small p leads to sparse output kernels which allows for an easier interpretation of the learned task relationships in the output kernel. Under certain conditions on p we show that one can drop the constraint that the output kernel should be positive definite as it is automatically satisfied for the unconstrained problem. This significantly simplifies the optimization and our result could also be of interest in other areas where one optimizes over the cone of positive definite matrices. The resulting unconstrained dual problem is amenable to efficient optimization methods such as stochastic dual coordinate ascent [19], which scale well to large data sets. Overall we do not require any eigenvalue decomposition operation at any stage of our algorithm and no alternate minimization is necessary, leading to a highly efficient methodology. Furthermore, we show that this trick not only applies to p-norms but also applies to a large class of regularizers for which we provide a characterization. Our contributions are as follows: (a) we propose a generic p-norm regularized output kernel matrix learning formulation, which can be extended to a large class of regularizers; (b) we show that the constraint on the output kernel to be positive definite can be dropped as it is automatically satisfied, leading to an unconstrained dual problem; (c) we propose an efficient stochastic dual coordinate ascent based method for solving the dual formulation; (d) we empirically demonstrate the superiority of our approach in terms of generalization performance as well as significant reduction in training time compared to other methods learning the output kernel. The paper is organized

as follows. We introduce our formulation in Section 2. Our main technical result is discussed in Section 3. The proposed optimization algorithm is described in Section 4. In Section 5, we report the empirical results. All the proofs can be found in the supplementary material.

2

The Output Kernel Learning Formulation

We first introduce the setting considered in this paper. We denote the number of tasks by T . We assume that all tasks have a common input space X and a common positive definite kernel function k : X ? X ? R. We denote by ?(?) the feature map and by Hk the reproducing kernel Hilbert space (RKHS) [20] associated with k. The training data is (xi , yi , ti)ni=1 , where xi ? X , ti is the task the i-th instance belongs to and yi is the corresponding label. Moreover, we have a positive T T definite matrix ? ? S+ on the set of tasks {1, . . . , T }, where S+ is the set of T ? T symmetric and positive semidefinite (p.s.d.) matrices. If one arranges the predictions of all tasks in a vector one can see multi-task learning as learning a vector-valued function in a RKHS [see 1, 13, 14, 15, 18, and references therein]. However, in this paper we use the one-to-one correspondence between real-valued and matrix-valued kernels, see [21], in order to limit the technical overhead. In this framework we define the joint kernel of input space and the set of tasks M : (X ? {1, . . . , T }) ? (X ? {1, . . . , T }) ? R as

M (x, s), (z, t) = k(x, z)?(s, t), (1) We denote the corresponding RKHS of functions on X ? {1, . . . , T } as HM and by k?kHM the corresponding norm. We formulate the output kernel learning problem for multiple tasks as n X 1 2 min C L yi , F (xi , ti) + kF kHM + ? V (?) (2) T ,F ?H 2 ??S+ M i=1 where L : R ? R ? R is the convex loss function (convex in the second argument), V (?) is a convex regularizer penalizing the complexity of the output kernel ? and ? ? R+ is the regularization 2 parameter. Note that kF kHM implicitly depends also on ?. In the following we show that (2) can be reformulated into a jointly convex problem in the parameters of the prediction function and the output kernel ?. Using the standard representer theorem [20] (see the supplementary material) for fixed output kernel ?, one can show that the optimal solution F ? ? HM of (2) can be written as F ? (x, t) =

T X n X

T X n X ?is M (xi , s), (x, t) = ?is k(xi , x)?(s, t).

s=1 i=1

s=1 i=1

2

(3)

With the explicit form of the prediction function one can rewrite the main problem (2) as min

T ,??Rn?T ??S+

C

n X

L yi ,

i=1

T n X 1 X ?ir ?js kij ?rs + ? V (?), ?js kji ?s ti + 2 r,s=1 i,j=1 s=1 j=1
T X n X
(4)

where ?rs = ?(r, s) and kij = k(xi , xj). Unfortunately, problem (4) is not jointly convex in ? and ? due to the product in the second term. A similar problem has been analyzed in [17]. They 2 could show that for the squared loss and V (?) = k?kF the corresponding optimization problem is invex and directly optimize it. For an invex function every stationary point is globally optimal [22]. We follow a different path which leads to a formulation similar to the one of [2] used for learning an input mapping (see also [9]). Our formulation for the output kernel learning problem is jointly convex in the task kernel ? and the task parameters. We present a derivation for the general RKHS Hk , analogous to the linear case presented in [2, 9]. We use the following variable transformation, ?it =

T X
?ts ?is , i = 1, . . . , n, s = 1, . . . , T,
?is =
resp.
s=1
T X
??1
? . st it
t=1

In the last expression ??1 has to be understood as the pseudo-inverse if ? is not invertible. Note that this causes no problems as in case ? is not invertible, we can without loss of generality restrict ? in (4) to the range of ?. The transformation leads to our final problem formulation, where the 2 prediction function F and its squared norm kF kHM can be written as F (x, t) =

n X
2
kF kHM =
?it k(xi , x),
n T X X
??1
? ? k(xi , xj). sr is jr
(5)
r,s=1 i,j=1
i=1

We get our final primal optimization problem min
T ,??Rn?T ??S+
C
n X
L yi ,
i=1
T n X
1 X ??1 sr ?is ?jr kij + ? V (?) ?jti kji + 2 r,s=1 i,j=1 j=1

n X
(6)

Before we analyze the convexity of this problem, we want Pn to illustrate the connection to the formulations in [9, 17]. With the task weight vectors wt = j=1 ?jt ?(xj) ? Hk we get predictions as F (x, t) = hwt , ?(x)i and one can rewrite 2

kF kHM =
T n X X
??1
? ? k(xi , xj) = sr is jr
r,s=1 i,j=1
T X
??1
sr
hws , wt i .
r,s=1

This identity is known for vector-valued RKHS, see [15] and references therein. When ? is ? times 2 PT 2 the identity matrix, then kF kHM = t=1 kw?t k and thus (2) is learning the tasks independently. As 2 mentioned before the convexity of the expression of kF kHM is crucial for the convexity of the full problem (6). The following result has been shown in [2] (see also [9]). T Lemma 1 Let R(?) denote the range of ? ? S+ and let ?? be the pseudoinverse. The extended T n?T function f : S+ ? R ? R ? {?} defined as (P

Pn T ? if ?i? ? R(?), ? i = 1, . . . , n, r,s=1 i,j=1 ? sr ?is ?jr k(xi , xj), f (?, ?) = , ? else .

is jointly convex. The formulation in (6) is similar to [9, 17, 18]. [9] uses the constraint Trace(?) ? 1 instead of a regularizer V (?) enforcing low rank of the output kernel. On the other hand, [17] employs squared Frobenius norm for V (?) with squared loss function. [18] proposed an efficient algorithm for convex V (?). Instead we think that sparsity of ? is better to avoid the emergence of spurious relations between tasks and also leads to output kernels which are easier to interpret. Thus we propose to use the following regularization functional for the output kernel ?: V (?) =

T X
p
—?tt0 —p = k?kp ,
t,t0 =1
3

for p ? [1, 2]. Several approaches [9, 17, 18] employ alternate minimization scheme, involving T costly eigendecompositions of T ? T matrix per iteration (as ? ? S+). In the next section we show that for a certain set of values of p one can derive an unconstrained dual optimization problem which T thus avoids the explicit minimization over the S+ cone. The resulting unconstrained dual problem can then be easily optimized by stochastic coordinate ascent. Having explicit expressions of the primal variables ? and ? in terms of the dual variables allows us to get back to the original problem.

3

Unconstrained Dual Problem Avoiding Optimization over S+T

The primal formulation (6) is a convex multi-task output kernel learning problem. The next lemma derives the Fenchel dual function of (6). This still involves the optimization over the primal variable T ? ? S+ . A main contribution of this paper is to show that this optimization problem over the T S+ cone can be solved with an analytical solution for a certain class of regularizers V (?). In the following we denote by ?r := {?i — ti = r} the dual variables corresponding to task r and by Krs the kernel matrix (k(xi , xj) — ti = r, tj = s) corresponding to the dual variables of tasks r and s. Lemma 2 Let L?i be the conjugate function of the loss Li : R ? R, u 7? L(yi , u), then q : Rn ? R, q(?) = ?C

n X

T ?

1 X i L?i ? ?rs h?r , Krs ?s i ? V (?) ? ? max T C 2? r,s=1 ??S+ i=1

(7)

is the dual function of (6), where ? ? Rn are the dual variables. The primal variable ? ? Rn?T in (6) and P the prediction function F can be expressed in terms of ? and ? as ?is = ?i ?sti and n F (x, s) = j=1 ?j ?stj k(xj , x) respectively, where tj is the task of the j-th training example. We now focus on the remaining maximization problem in the dual function in (7) max

T ??S+

T 1 X ?rs h?r , Krs ?s i ? V (?). 2? r,s=1

(8)

This is a semidefinite program which is computationally expensive to solve and thus prohibits to scale the output kernel learning problem to a large number of tasks. However, we show in the following that this problem has an analytical solution for a subset of the regularizers V (?) = PT 1 p r,s=1 —?rs — for p ? 1. For better readability we defer a more general result towards the end 2 of the section. The basic idea is to relax the constraint on ? ? RT ?T in (8) so that it is equivalent to the computation of the conjugate V ? of V . If the maximizer of the relaxed problem is positive semi-definite, one has found the solution of the original problem. Theorem 3 Let k ? N and p = max

T ??S+

T X

?rs ?rs ?

r,s=1

2k 2k?1 ,

then with ?rs =

1 2?

h?r , Krs ?s i we have

T T 1 X 1 2k ? 1 2k X r 2k —?rs —p = h? , Krs ?s i , 2 r,s=1 4k ? 2 2k? r,s=1

and the maximizer is given by the positive semi-definite matrix 2k ? 1 2k?1 2k?1 ??rs = h?r , Krs ?s i , r, s = 1, . . . , T. 2k?

(9)

(10)

Plugging the result of the previous theorem into the dual function of Lemma 2 we get for k ? N and p 2k p = 2k?1 with V (?) = k?kp the following unconstrained dual of our main problem (6): maxn ?C

??R

n X

T ? ? 2k ? 1 2k X r i 2k L?i ? ? h? , Krs ?s i . C 4k ? 2 2k? r,s=1 i=1

(11)

Note that by doing the variable transformation ?i := ?Ci we effectively have only one hyperparameter in (11). This allows us to cross-validate more efficiently. The range of admissible values for p in Theorem 3 lies in the interval (1, 2], where we get for k = 1 the value p = 2 and as k ? ? 4

Table 1: Examples of regularizers V (?) together with their generating function ? and the explicit 1 form of ?? in terms of the dual variables, ?rs = 2? h?r , Krs ?s i. The optimal value of (8) is given PT in terms of ? as max h?, ?i ? V (?) = r,s=1 ?(?rs). ??RT ?T

?(z)

??rs

V (?)

z 2k 2k ,

k?N

ez =

P?

2k?1 2k

k

z k=0 k!

cosh(z) ? 1 =

z 2k k=1 (2k)!

P?

T P

2k

?2k?1 rs

—?rs — 2k?1

r,s=1

? T ? P ? log(?) ? ? if ?rs ¿ 0?r, s rs rs rs r,s=1 ? ? else .

T p P ?rs arcsinh(?rs) ? 1 + ?2rs + T 2

e?rs arcsinh(?rs)

r,s=1

we have p ? 1. The regularizer for p = 2 together with the squared loss has been considered in the primal in [17, 18]. Our analytical expression of the dual is novel and allows us to employ stochastic dual coordinate ascent to solve the involved primal optimization problem. Please also note that by optimizing the dual, we have access to the duality gap and thus a well-defined stopping criterion. This is in contrast to the alternating scheme of [17, 18] for the primal problem which involves costly matrix operations. Our runtime experiments show that our solver for (11) outperforms the solvers of [17, 18].

Finally, note that even for suboptimal dual variables ?, the corresponding ? matrix in (10) is positive semidefinite. Thus we always get a feasible set of primal variables. Characterizing the set of convex regularizers V which allow an analytic expression for the dual function The previous theorem raises the question for which class of convex, separable regularizers we can get an analytical expression of the dual function by explicitly solving the optimization problem (8) over the positive semidefinite cone. A key element in the proof of the previous theorem is the characterization of functions f : R ? R which when applied elementwise T f (A) = (f (aij))Ti,j=1 to a positive semidefinite matrix A ? S+ result in a p.s.d. matrix, that is T f (A) ? S+ . This set of functions has been characterized by Hiai [23]. T . We denote by f (A) = (f (aij))Ti,j=1 the elementTheorem 4 ([23]) Let f : R ? R and A ? S+ T T if and only if f is analytic =? f (A) ? S+ wise application of f to A. It holds ? T ? 2, A ? S+ P? k and f (x) = k=0 ak x with ak ? 0 for all k ? 0.

Note that in the previous theorem the condition on f is only necessary when we require the implication to hold for all T . If T is fixed, the set of functions is larger and includes even (large) fractional powers, see [24]. We use the stronger formulation as we want that the result holds without any restriction on the number of tasks T . Theorem 4 is the key element used in our following characterization of separable regularizers of ? which allow an analytical expression of the dual function. P? ak k+1 Theorem 5 Let ? : R ? R be analytic on R and given as ?(z) = k=0 k+1 z where ak ? PT ? T ?T 0 ?k ? 0. If ? is convex, then, V (?) := r,s=1 ? (?rs), is a convex function V : R ? R and max h?, ?i ? V (?) = V ? (?) =

??RT ?T

T X

? ?rs ,

(12)

r,s=1

T T where the global maximizer fulfills ?? ? S+ if ? ? S+ and ??rs =

P?

k=0

ak ?krs .

Table 1 summarizes e.g. of functions ?, the corresponding V (?) and the maximizer ?? in (12).

4

Optimization Algorithm

The dual problem (11) can be efficiently solved via decomposition based methods like stochastic dual coordinate ascent algorithm (SDCA) [19]. SDCA enjoys low computational complexity per iteration and has been shown to scale effortlessly to large scale optimization problems. 5

Algorithm 1 Fast MTL-SDCA Input: Gram matrix K, label vector y, regularization parameter and relative duality gap parameter Output: ? (? is computed from ? using our result in 10) Initialize ? = ?(0) repeat Randomly choose a dual variable ?i Solve for ? in (13) corresponding to ?i ?i ? ?i + ? until Relative duality gap is below

Our algorithm for learning the output kernel matrix and task parameters is summarized in Algorithm 1 (refer to the supplementary material for more details). At each step of the iteration we optimize the dual objective over a randomly chosen ?i variable. Let ti = r be the task corresponding to ?i . We apply the update ?i ? ?i + ?. The optimization problem of solving (11) with respect to ? is as follows: X X

min L?i (??i ? ?)/C + ? (a?2 + 2brr ? + crr)2k + 2 (brs ? + crs)2k + c2k sz , (13) ??R

s6=r

s,z6=r

2k P ? 2k?1 where a = kii , brs = j:tj =s kij ?j ?s, csz = h?s , Ksz ?z i ?s, z and ? = C(4k?2) . 2k? This one-dimensional convex optimization problem is solved efficiently via Newton method. The complexity of the proposed algorithm is O(T) per iteration . The proposed algorithm can also be employed for learning output kernels regularized by generic V (?), discussed in the previous section. Special case p = 2(k = 1): For certain loss functions such as the hinge loss, the squared loss, etc., L?ti ? ?tiC+? yields a linear or a quadratic expression in ?. In such cases problem (13) reduces to finding the roots of a cubic equation, which has a closed form expression. Hence, our algorithm is highly efficient with the above loss functions when ? is regularized by the squared Frobenius norm.

5

Empirical Results

In this section, we present our results on benchmark data sets comparing our algorithm with existing approaches in terms of generalization accuracy as well as computational efficiency. Please refer to the supplementary material for additional results and details. 5.1

Multi-Task Data Sets

We begin with the generalization results in multi-task setups. The data sets are as follows: a) Sarcos: a regression data set, aim is to predict 7 degrees of freedom of a robotic arm, b) Parkinson: a regression data set, aim is to predict the Parkinson?s disease symptom score for 42 patients, c) Yale: a face recognition data with 28 binary classification tasks, d) Landmine: a data set containing binary classifications from 19 different landmines, e) MHC-I: a bioinformatics data set having 10 binary classification tasks, f) Letter: a handwritten letters data set with 9 binary classification tasks. We compare the following algorithms: Single task learning (STL), multi-task methods learning the output kernel matrix (MTL [16], CMTL [12], MTRL [9]) and approaches that learn both input and output kernel matrices (MTFL [11], GMTL [10]). Our proposed formulation (11) is denoted by FMTLp . We consider three different values for the p-norm: p = 2 (k = 1), p = 4/3 (k = 2) and p = 8/7 (k = 4). Hinge and -SVR loss functions were employed for classification and regression problems respectively. We follow the experimental protocol1 described in [11]. Table 2 reports the performance of the algorithms averaged over ten random train-test splits. The proposed FMTLp attains the best generalization accuracy in general. It outperforms the baseline MTL as well as MTRL and CMTL, which solely learns the output kernel matrix. Moreover, it achieves an overall better

performance than GMTL and MTFL. The FMTLp=4/3,8/7 give comparable generalization to p = 2 case, with the additional benefit of learning sparser and more interpretable output kernel matrix (see Figure 1). 1

The performance of STL, MTL, CMTL and MTFL are reported from [11].

6

Table 2: Mean generalization performance and the standard deviation over ten train-test splits. Data set

STL

MTL

CMTL

MTFL

GMTL

MTRL

Regression data sets: Explained Variance (%) Sarcos 40.5?7.6 34.5?10.2 33.0?13.4 49.9?6.3 45.8?10.6 41.6?7.1 2.7?3.6 16.8?10.8 33.6?9.4 12.0?6.8 Parkinson 2.8?7.5 4.9?20.0 Classification data sets: AUC (%) Yale 93.4?2.3 96.4?1.6 Landmine 74.6?1.6 76.4?0.8 MHC-I 69.3?2.1 72.3?1.9 Letter 61.2?0.8 61.0?1.6

95.2?2.1 75.9?0.7 72.6?1.4 60.5?1.1

97.0?1.6 76.4?1.0 71.7?2.2 60.5?1.8

91.9?3.2 76.7?1.2 72.5?2.7 61.2?0.9

46.7?6.9 27.0?4.4

2

2

4

4

4

6

6

6

8

8

8

10

10

10

12

12

12

14

14

14

16

16

16

18

18 4

6
8
10
12
14
16
18
p = 8/7
50.3?5.8 48.4?5.8 27.0?4.4 27.0?4.4
96.1?2.1 97.0?1.2 97.0?1.4 76.1?1.0 76.8?0.8 76.7?1.0 71.5?1.7 71.7?1.9 70.8?2.1 60.3?1.4 61.4?0.7 61.5?1.0
2
2
FMTLp p = 4/3
p=2
96.8?1.4 76.4?0.9 70.7?1.9 61.4?1.0
18 2
4
(p = 2)
6
8
10
12
14
16
18
2
4
6
(p = 4/3)
8
10
12
14
16
18
(p = 8/7)

Figure 1: Plots of —?— matrices (rescaled to [0,1] and averaged over ten splits) computed by our solver FMTLp for the Landmine data set for different p-norms, with cross-validated hyper-parameter values. The darker regions indicate higher value. Tasks (landmines) numbered 1-10 correspond to highly foliated regions and those numbered 11-19 correspond to bare earth or desert regions. Hence, we expect two groups of tasks (indicated by the red squares). We can observe that the learned ? matrix at p = 2 depicts much more spurious task relationships than the ones at p = 4/3 and p = 8/7. Thus, our sparsifying regularizer improves interpretability.

Table 3: Mean accuracy and the standard deviation over five train-test splits.
Data set
STL
MTL-SDCA
MNIST USPS
84.1?0.3 90.5?0.3
86.0?0.2 90.6?0.2
5.2
GMTL
MTRL
p=2
FMTLp -H p = 4/3
p = 8/7
p=2
FMTLp -S p = 4/3
p = 8/7
84.8?0.3 85.6?0.4 86.1?0.4 85.8?0.4 86.2?0.4 82.2?0.6 82.5?0.4 82.4?0.3 91.6?0.3
92.4?0.2 92.4?0.2 92.6?0.2 92.6?0.1 87.2?0.4 87.7?0.3 87.5?0.3
Multi-Class Data Sets

The multi-class setup is cast as T one-vs-all binary classification tasks, corresponding to T classes. In this section we experimented with two loss functions: a) FMTLp -H ? the hinge loss employed in SVMs, and b) FMTLp -S ? the squared loss employed in OKL [17]. In these experiments, we also compare our results with MTL-SDCA, a state-of-the-art multi-task feature learning method [25]. USPS & MNIST Experiments: We followed the experimental protocol detailed in [10]. Results are tabulated in Table 3. Our approach FMTLp -H obtains better accuracy than GMTL, MTRL and MTL-SDCA [25] on both data sets. MIT Indoor67 Experiments: We report results on the MIT Indoor67 benchmark [26] which covers 67 indoor scene categories. We use the train/test split (80/20 images per class) provided by the authors. FMTLp -S achieved the accuracy of 73.3% with p = 8/7. Note that this is better than the ones reported in [27] (70.1%) and [26] (68.24%). SUN397 Experiments: SUN397 [28] is a challenging scene classification benchmark [26] with 397 classes. We use m = 5, 50 images per class for training, 50 images per class for testing and report the average accuracy over the 10 standard splits. We employed the CNN features extracted with the 7

Table 4: Mean accuracy and the standard deviation over ten train-test splits on SUN397. m
STL
MTL
MTL-SDCA
5 50
40.5?0.9 55.0?0.4
42.0?1.4 57.0?0.2
41.2?1.3 54.8?0.3
p=2

FMTLp -H p = 4/3
p = 8/7
p=2
FMTLp -S p = 4/3
p = 8/7
41.5?1.1 55.1?0.2
41.6?1.3 55.6?0.3
41.6?1.2 55.1?0.3
44.1?1.3 58.6?0.1
44.1?1.1 58.5?0.1
44.0?1.2 58.6?0.2
(Time by baseline) / (Time by FMTL2?S)
3
10
FMTL ?S 2
2
ConvexOKL
Time (log10 scale), s
10
OKL 1
10
0
10
?1
10
?2
10
50
100
150
200 250 300 Number of Tasks
350
400
(a)
20 18 16
MIT Indoor67, OKL SUN397, OKL MIT Indoor67, ConvexOKL . SUN397, ConvexOKL
14 12 10 8 6 4 2 0 3
3.5
4
4.5
5 5.5 Log10(?)
6
6.5
7
(b)

Figure 2: (a) Plot compares the runtime of various algorithms with varying number of tasks on SUN397. Our approach FMTL2 -S is 7 times faster that OKL [17] and 4.3 times faster than ConvexOKL [18] when the number of tasks is maximum. (b) Plot showing the factor by which FMTL2 S outperforms OKL and ConvexOKL over the hyper-parameter range on various data sets. On SUN397, we outperform OKL and ConvexOKL by factors of 5.2 and 7 respectively. On MIT Indoor67, we are better than OKL and ConvexOKL by factors of 8.4 and 2.4 respectively. convolutional neural network (CNN) [26] using Places 205 database. The results are tabulated in Table 4. The ? matrices computed by FMTLp -S are discussed in the supplementary material. 5.3

Scaling Experiment

We compare the runtime of our solver for FMTL2 -S with the OKL solver of [17] and the ConvexOKL solver of [18] on several data sets. All the three methods solve the same optimization problem. Figure 2a shows the result of the scaling experiment where we vary the number of tasks (classes). The parameters employed are the ones obtained via cross-validation. Note that both OKL and ConvexOKL algorithms do not have a well defined stopping criterion whereas our approach can easily compute the relative duality gap (set as 10?3). We terminate them when they reach the primal objective value achieved by FMTL2 -S . Our optimization approach is 7 times and 4.3 times faster than the alternate minimization based OKL and ConvexOKL, respectively, when the number of tasks is maximal. The generic FMTLp=4/3,8/7 are also considerably faster than OKL and ConvexOKL. Figure 2b compares the average runtime of our FMTLp -S with OKL and ConvexOKL on the crossvalidated range of hyper-parameter values. FMTLp -S outperform them on both MIT Indoor67 and SUN397 data sets. On MNIST and USPS data sets, FMTLp -S is more than 25 times faster than OKL, and more than 6 times faster than ConvexOKL. Additional details of the above experiments are discussed in the supplementary material.

6

Conclusion

We proposed a novel formulation for learning the positive semi-definite output kernel matrix for multiple tasks. Our main technical contribution is our analysis of a certain class of regularizers on the output kernel matrix where one may drop the positive semi-definite constraint from the optimization problem, but still solve the problem optimally. This leads to a dual formulation that can be efficiently solved using stochastic dual coordinate ascent algorithm. Results on benchmark multi-task and multi-class data sets demonstrates the effectiveness of the proposed multi-task algorithm in terms of runtime as well as generalization accuracy. Acknowledgments. P.J. and M.H. acknowledge the support by the Cluster of Excellence (MMCI). 8

2 References

[1] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. JMLR, 6:615?637, 2005. [2] A. Argyriou, T. Evgeniou, and

M. Pontil. Convex multi-task feature learning. ML, 73:243?272, 2008. [3] K. Lounici, M. Pontil, A. B. Tsybakov, and S. van de Geer. Taking advantage of sparsity in multi-task learning. In COLT, 2009. [4] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A dirty model for multi-task learning. In NIPS, 2010. [5] P. Jawanpuria and J. S. Nath. Multi-task multiple kernel learning. In SDM, 2011. [6] A. Maurer, M. Pontil, and B. Romera-paredes. Sparse coding for multitask and transfer learning. In ICML, 2013. [7] P. Jawanpuria, J. S. Nath, and G. Ramakrishnan. Generalized hierarchical kernel learning. JMLR, 16:617? 652, 2015. [8] R. Caruana. Multitask learning. ML, 28:41?75, 1997. [9] Y. Zhang and D. Y. Yeung. A convex formulation for learning task relationships in multi-task learning. In UAI, 2010. [10] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In ICML, 2011. [11] P. Jawanpuria and J. S. Nath. A convex feature learning formulation for latent task structure discovery. In ICML, 2012. [12] L. Jacob, F. Bach, and J. P. Vert. Clustered multi-task learning: A convex formulation. In NIPS, 2008. [13] C. A. Micchelli and M. Pontil. Kernels for multitask learning. In NIPS, 2005. [14] A. Caponnetto, C. A. Micchelli, M. Pontil, and Y. Ying. Universal multi-task kernels. JMLR, 9:1615? 1646, 2008. ? [15] M. A. Alvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. Foundations and Trends in Machine Learning, 4:195?266, 2012. [16] T. Evgeniou and M. Pontil. Regularized multi?task learning. In KDD, 2004. [17] F. Dinuzzo, C. S. Ong, P. Gehler, and G. Pillonetto. Learning output kernels with block coordinate descent. In ICML, 2011. [18] C. Ciliberto, Y. Mroueh, T. Poggio, and L. Rosasco. Convex learning of multiple tasks and their structure. In ICML, 2015. [19] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss. JMLR, 14(1):567?599, 2013. [20] B. Sch?olkopf and A. Smola. Learning with Kernels. MIT Press, 2002. [21] M. Hein and O. Bousquet. Kernels, associated structures and generalizations. Technical Report TR-127, Max Planck Institute for Biological Cybernetics, 2004. [22] A. Ben-Israel and B. Mond. What is invexity ? J. Austral. Math. Soc. Ser. B, 28:1?9, 1986. [23] F. Hiai. Monotonicity for entrywise functions of matrices. 431(8):1125 ? 1146, 2009.

Linear Algebra and its Applications,

[24] R. A. Horn. The theory of infinitely divisible matrices and kernels. Trans. Amer. Math. Soc., 136:269?286, 1969. [25] M. Lapin, B. Schiele, and M. Hein. Scalable multitask representation learning for scene classification. In CVPR, 2014. [26] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In NIPS, 2014. [27] M. Koskela and J. Laaksonen. Convolutional network features for scene recognition. In Proceedings of the ACM International Conference on Multimedia, 2014. [28] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In CVPR, 2010.

9