An Efficient Projection for $l_{1,\infty}$ Regularization

Ariadna Quattoni ARIADNA@CSAIL.MIT.EDU
Computer Science and Artificial Intelligence Laboratory, MIT, 32 Vassar St., Cambridge MA 02139 USA
UC Berkeley EECS and ICSI, 1947 Center St., Berkeley CA 94704 USA

Xavier Carreras CARRERAS@CSAIL.MIT.EDU
Michael Collins MCOLLINS@CSAIL.MIT.EDU
Computer Science and Artificial Intelligence Laboratory, MIT, 32 Vassar St., Cambridge MA 02139 USA

Trevor Darrell TREVOR@EECS.BERKELEY.EDU
UC Berkeley EECS and ICSI, 1947 Center St., Berkeley CA 94704 USA

Abstract

In recent years the $l_{1,\infty}$ norm has been proposed for joint regularization. In essence, this type of regularization aims at extending the l_1 framework for learning sparse models to a setting where the goal is to learn a set of jointly sparse models. In this paper we derive a simple and effective projected gradient method for optimization of $l_{1,\infty}$ regularized problems. The main challenge in developing such a method resides on being able to compute efficient projections to the $l_{1,\infty}$ ball. We present an algorithm that works in $O(n \log n)$ time and $O(n)$ memory where n is the number of parameters. We test our algorithm in a multi-task image annotation problem. Our results show that $l_{1,\infty}$ leads to better performance than both l_2 and l_1 regularization and that it is is effective in discovering jointly sparse solutions.

1. Introduction

Learning algorithms based on l_1 regularized loss functions have had a relatively long history in machine learning, covering a wide range of applications such as sparse sensing (Donoho, 2004), l1-logistic regression (Ng, 2004), and structure learning of Markov networks (Lee et al., 2007). A well known property of l_1 regularized models is their ability to recover sparse solutions. Because of this they are suitable for applications where discovering significant features is of value and where computing features is expensive. In addition, it has been shown that in some cases l_1 regularization can lead to sample complexity bounds that are logarithmic in the number of input dimensions, making it suitable for learning in high dimensional spaces (Ng, 2004).

In recent years the $l_{1,\infty}$ norm has been proposed for joint regularization (Turlach et al., 2005; Tropp, 2006; Quattoni et al., 2008; Schmidt et al., 2008). The $l_{1,\infty}$ norm is a matrix norm that penalizes the sum of maximum absolute values of each row. This regularizer encourages row sparsity: i.e., it encourages entire rows of the matrix to have zero elements.

As one example application, consider a multi-task setting with m problems where the $l_{1,\infty}$ regularizer is applied to a parameter matrix $W = [\mathbf{w}_1, \ldots, \mathbf{w}_m]$, where $\mathbf{w}_i$ is a parameter vector for the i-th task. In this case the $l_{1,\infty}$ regularizer is used to promote feature sharing across tasks and discover solutions where only a few features are non-zero in any of the m tasks (i.e. jointly sparse solutions) (Quattoni et al., 2008). Other applications of $l_{1,\infty}$ regularization are simultaneous sparse signal approximation (Tropp, 2006; Turlach et al., 2005) and structure learning (Schmidt et al., 2008).

In this paper we present an efficient projected subgradient method for optimization of $l_{1,\infty}$ regularized convex objectives, which we formulate as a constrained convex optimization problem. Projected gradient methods iterate between performing unconstrained gradient-based updates followed by projections to the feasible set, which in our case is an $l_{1,\infty}$ ball. These methods have been shown to scale well and have been proposed for solving constrained optimization problems involving large numbers of variables. For

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009.

example, (Shalev-Shwartz et al., 2007) developed a projected gradient method for l_2 regularization and (Duchi et al., 2008) proposed an analogous algorithm for l_1. However, the problem of finding an efficient projected method for $l_{1,\infty}$ constraints remains open. The main challenge in developing a projected gradient algorithm for $l_{1,\infty}$ constraints resides on being able to efficiently compute Euclidean projections onto the $l_{1,\infty}$ ball. We show that this can be done in $O(n \log n)$ time and $O(n)$ memory, where n is the number of parameters of our model.

We apply our algorithm to a multi-task image annotation problem where the goal is to predict keywords for a given image. We show that $l_{1,\infty}$ regularization performs significantly better than both independent l_2 and independent l_1 regularization. Furthermore, we show that $l_{1,\infty}$ is able to find jointly sparse solutions (i.e. parameter matrices with few non-zero rows).

2. Previous Work

Turlach et al. (2005) developed an interior point algorithm for optimizing a twice differentiable objective regularized with an $l_{1,\infty}$ norm. One of the limitations of this approach is that it requires the exact computation of the Hessian of the objective function. This might be computationally expensive for some applications both in terms of memory and time. An alternative approach was proposed by Schmidt et al. (2008), who combined a gradient-descent method with independent l_∞ projections.

For the special case of a linear objective the regularization problem can be expressed as a linear program (Quattoni et al., 2008). While this is feasible for small problems it does not scale to problems with large number of variables.

Our $l_{1,\infty}$ projection algorithm is related to the l_1 projection of Duchi et al. (2008) in that theirs is a special case of our algorithm for $m = 1$. The derivation of the general case for $l_{1,\infty}$ regularization is significantly more involved as it requires reducing a set of l_∞ regularization problems tied together through a common l_1 norm to a problem that can be solved efficiently.

Similarly to the $l_{1,\infty}$ norm, the $l_{1,2}$ has also been proposed for joint sparse approximation. This norm penalizes the sum of the l_2 norms of each row of W (Yuan & Lin, 2006; Meier et al., 2006; Similä & Tikka, 2007; Park & Hastie, 2006; Obozinski et al., 2006; Argyriou et al., 2007; Schmidt et al., 2009).

3. A projected gradient method for $l_{1,\infty}$ regularization

In this section we start by describing a convex constrained optimization formulation of $l_{1,\infty}$ regularization, followed by a concrete application to multi-task learning. We finish by introducing the projected gradient approach that we will use to solve the problem.

3.1. Constrained Convex Optimization Formulation

Assume we have a dataset $D = (z_1, z_2, \ldots, z_n)$ with points z belonging to some set Z, and a $d \times m$ parameter matrix $W = [\mathbf{w}_1, \ldots, \mathbf{w}_m]$ where $\mathbf{w}_k \in R^d$ for $k = \{1, \ldots, m\}$ is the k-th column of W. For example in a multi-task setting $\mathbf{w}_k$ would be the parameters for the k-th task. We also have a convex loss function $L(z, W)$ that measures the loss incurred by W on a training sample z. Let us now define the $l_{1,\infty}$ norm:

$$||W||_{1,\infty} = \sum_{j=1}^{d} \max_k |W_{j,k}| \tag{1}$$

When used as a regularization norm $l_{1,\infty}$ induces solutions where only a few rows will contain non-zero values. In fact Tropp (2006) showed that under certain conditions the $l_{1,\infty}$ regularization norm is a convex relaxation of a pseudo-norm that counts the number of non-zero rows of W.

One possibility for defining the $l_{1,\infty}$ regularization problem is to set it as a soft constraint:

$$\min_W \sum_{i=1}^{n} L(z_i, W) + \lambda ||W||_{1,\infty} \tag{2}$$

Here λ is a parameter that captures the trade-off between error and sparsity. Another natural way of formulating the regularization problem is to set it as a constrained convex optimization:

$$\min_W \sum_{i=1}^{n} L(z_i, W) \quad \text{s.t.} \quad ||W||_{1,\infty} \leq C \tag{3}$$

In this case C is a bound on $||W||_{1,\infty}$ and serves a role analogous to that of λ in the previous formulation. In this paper we concentrate on this latter formulation.

3.2. An Application: Multi-task Learning

To give a concrete application let us describe the multi-task joint regularization setting. The goal here is to train m jointly-sparse linear classifiers, one for each task. By jointly sparse we mean that we wish only a few features to be non-zero in any of the m problems.

We can formulate this problem as an $l_{1,\infty}$ regularized objective.

Following the notation from the previous section, Z is the set of tuples: $(\mathbf{x}_i, y_i, l_i)$ for $i = 1 \dots n$ where each $\mathbf{x}_i \in R^d$ is a feature vector, $l_i \in \{1, \dots, m\}$ is a label specifying to which of the m tasks the example corresponds to, and $y_i \in \{+1, -1\}$ is the label for the example. Assume that we wish to learn m linear classifiers of the form $f_k(\mathbf{x}) = \mathbf{w}_k \cdot \mathbf{x}$, and let $W = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m]$ be a $d \times m$ matrix where $W_{j,k}$ corresponds to the j-th parameter of the k-th problem. So in the ideal case we would like a solution matrix W with a few non-zero rows (i.e. a few active features). In order to assess the classification performance multiple classification losses could be used, in this paper we used the hinge loss:

$$L_{\mathrm{H}}(z, W) = \max(0, 1 - f_k(\mathbf{x})y) \qquad (4)$$

In this formulation, the hinge loss encourages correct classification of the points and the $l_{1,\infty}$ norm is similar to l_1 regularization, but it encourages joint sparsity across the different tasks.

3.3. A Projected Gradient Method

Our algorithm for optimizing equation (3) is based on the projected subgradient method for minimizing a convex function $\mathrm{F}(W)$ subject to convex constraints of the form $W \in \Omega$, where Ω is a convex set (Bertsekas, 1999). In our case $\mathrm{F}(W)$ is some convex loss function, W is a parameter matrix and Ω is the set of all matrices with $||W||_{1,\infty} \leq C$.

A projected subgradient algorithm works by generating a sequence of solutions W^t via $W^{t+1} = \mathrm{P}_\Omega(W^t - \eta \nabla^t)$. Here ∇^t is a subgradient of F at W^t and $\mathrm{P}_\Omega(W)$ is is the Euclidean projection of W onto Ω, given by:

$$\min_{W' \in \Omega} ||W' - W||^2 = \min_{W' \in \Omega} \sum_{j,k} (W'_{j,k} - W_{j,k})^2 \qquad (5)$$

Finally, η is the learning rate that controls the amount by which the solution changes at each iteration.

Standard results in optimization literature (Bertsekas, 1999) show that when $\eta = \frac{\eta_0}{\sqrt{t}}$ and $\mathrm{F}(W)$ is a convex Lipschitz function the projected algorithm will converge to an ϵ-accurate solution in $O(1/\epsilon^2)$ iterations.

For the hinge loss case described in the previous section, computing the subgradient of the objective of equation (3) is straightforward. The subgradient for the parameters of each task can be computed independently of the other tasks, and for the k-th task it is given by summing examples of the task whose margin is less than one:

$$\nabla_k^t = \sum_{i \,:\, l_i = k,\ f_k(\mathbf{x}_i) y_i < 1} y_i \mathbf{x}_i \qquad (6)$$

In the next section we show how to compute the projection onto the $l_{1,\infty}$ ball efficiently.

4. Efficient Projection onto the $l_{1,\infty}$ Ball

We start this section by using the Lagrangian of the projection to characterize the optimal solution. This will allow us to map the projection to a simpler problem for which we can develop an efficient algorithm, that we present in the second part of the section.

4.1. Characterization of the solution

We now describe the projection of a matrix A to the $l_{1,\infty}$ ball. For now, we assume that all entries in A are non-negative; later we will show that this assumption imposes no limitation. The projection can be formulated as finding a matrix B that solves the following convex optimization problem:

$$\begin{aligned}
\mathbf{P}_{\mathbf{1},\infty}: \quad \min_{B,\boldsymbol{\mu}} \quad & \frac{1}{2}\sum_{i,j}(B_{i,j} - A_{i,j})^2 && (7)\\
\text{s.t.} \quad & \forall i,j \ \ B_{i,j} \leq \mu_i && (8)\\
& \sum_i \mu_i = C && (9)\\
& \forall i,j \ \ B_{i,j} \geq 0 && (10)\\
& \forall i \ \ \mu_i \geq 0 && (11)
\end{aligned}$$

In the above problem, the objective (7) corresponds to the Euclidean distance between A and B, whereas the constraints specify that B is in the boundary of the $l_{1,\infty}$ ball of radius C.[1] To do so, there are variables $\boldsymbol{\mu}$ that stand for the the maximum coefficients of B for each row i, as imposed by constraints (8), and that sum to the radius of the ball, as imposed by constraint (9). Constraints (10) and (11) stand for non-negativity of the new coefficients and maximum values.

We now present the Lagrangian of problem $\mathrm{P}_{1,\infty}$ and three lemmas that will be used to derive an efficient algorithm. The Lagrangian is:

$$\begin{aligned}
\mathcal{L}(B, \boldsymbol{\mu}, \boldsymbol{\alpha}, \theta, \boldsymbol{\beta}, \boldsymbol{\gamma}) \quad = \quad & \frac{1}{2}\sum_{i,j}(B_{i,j} - A_{i,j})^2 \\
& + \sum_{i,j} \alpha_{i,j}(B_{i,j} - \mu_i) + \theta\Big(\sum_i \mu_i - C\Big) \\
& - \sum_{i,j} \beta_{i,j} B_{i,j} \ - \ \sum_i \gamma_i \mu_i
\end{aligned}$$

[1] It is simple to show that the optimal B is on the boundary of the ball, providing that $||A||_{1,\infty} >= C$.

Lemma 1 *At the optimal solution of* $\mathrm{P}_{1,\infty}$ *there exists a constant* $\theta \geq 0$ *such that for every* i*: either (a)* $\mu_i > 0$ *and* $\sum_j (A_{i,j} - B_{i,j}) = \theta$*; or (b)* $\mu_i = 0$ *and* $\sum_j A_{i,j} \leq \theta$.

Proof: Differentiating $\mathcal{L}$ with respect to $B_{i,j}$ gives the optimality condition $\frac{\partial \mathcal{L}}{\partial B_{i,j}} = B_{i,j} - A_{i,j} + \alpha_{i,j} - \beta_{i,j} = 0$. Differentiating $\mathcal{L}$ with respect to μ_i gives the optimality condition $\frac{\partial \mathcal{L}}{\partial \mu_i} = \theta - \sum_j \alpha_{i,j} - \gamma_i = 0$.

We now assume $\mu_i > 0$ to prove (a). The complementary slackness conditions imply that whenever $\mu_i > 0$ then $\gamma_i = 0$ and therefore $\theta = \sum_j \alpha_{i,j}$. If we assume that $B_{i,j} > 0$, by complementary slackness then $\beta_{i,j} = 0$, and therefore $\alpha_{i,j} = A_{i,j} - B_{i,j}$. If $B_{i,j} = 0$ then $B_{i,j} - \mu_i \neq 0$ and so $\alpha_{i,j} = 0$ due to complementary slackness; we then observe that $A_{i,j} = -\beta_{i,j}$, but since $A_{i,j} \geq 0$ and $\beta_{i,j} \geq 0$ it must be that $A_{i,j} = 0$; so we can express also $\alpha_{i,j} = A_{i,j} - B_{i,j}$. Thus, $\theta = \sum_j (A_{i,j} - B_{i,j})$ which proves (a).

When $\mu_i = 0$, $B_{i,j} = 0$ because of (8) and (10). Then, using the optimality conditions $\frac{\partial \mathcal{L}}{\partial B_{i,j}}$ we get that $\alpha_{i,j} = A_{i,j} + \beta_{i,j}$. Plugging this into $\frac{\partial \mathcal{L}}{\partial \mu_i}$ we get $\theta = \sum_j (A_{i,j} + \beta_{i,j}) + \gamma_i$. By definition $\beta_{i,j} \geq 0$ and $\gamma_i \geq 0$, which proves (b). □

Lemma 1 means that when projecting A, for every row whose sum is greater than θ, the sum of the new values in the row will be reduced by a constant θ. The rows whose sum is less than θ will become zero.

The next lemma reveals how to obtain the coefficients of B given the optimal maximums $\boldsymbol{\mu}$.

Lemma 2 *Let* $\boldsymbol{\mu}$ *be the optimal maximums of problem* $\mathrm{P}_{1,\infty}$*. The optimal matrix* B *of* $\mathrm{P}_{1,\infty}$ *satisfies that:*

$$A_{i,j} \geq \mu_i \implies B_{i,j} = \mu_i \qquad (12)$$

$$A_{i,j} \leq \mu_i \implies B_{i,j} = A_{i,j} \qquad (13)$$

$$\mu_i = 0 \implies B_{i,j} = 0 \qquad (14)$$

Proof: If $\mu_i = 0$ the lemma follows directly from (8) and (10). The rest of the proof assumes that $\mu_i > 0$.

To prove (12), assume that $A_{i,j} \geq \mu_i$ but $B_{i,j} \neq \mu_i$. We consider two cases. When $B_{i,j} > 0$, by (8), if $B_{i,j} \neq \mu_i$ then $B_{i,j} < \mu_i$, which means $\alpha_{i,j} = 0$ due to complementary slackness. This together with $\beta_{i,j} = 0$ imply that $B_{i,j} = A_{i,j}$, and therefore $A_{i,j} < \mu_i$, which contradicts the assumption. When $B_{i,j} = 0$ then $\alpha_{i,j} = 0$ and $A_{i,j} = 0$ (see proof of Lemma 1), which contradicts the assumption.

To prove (13), assume that $A_{i,j} \leq \mu_i$ but $B_{i,j} \neq A_{i,j}$. We again consider two cases. If $B_{i,j} > 0$, $\beta_{i,j} = 0$; given that $B_{i,j} \neq A_{i,j}$, then $\alpha_{i,j} > 0$, and so $B_{i,j} = \mu_i$ due to complementary slackness. But since $\alpha_{i,j} > 0$, $A_{i,j} > B_{i,j} = \mu_i$, which contradicts the assumption. If $B_{i,j} = 0$ then $A_{i,j} = 0$ (see proof of Lemma 1), which contradicts the assumption. □

With these results, the problem of projecting into the $l_{1,\infty}$ ball can be reduced to the following problem, which finds the optimal maximums $\boldsymbol{\mu}$:

$$\mathbf{M}_{1,\infty}: \quad \text{find} \quad \boldsymbol{\mu}\ ,\ \theta \qquad (15)$$

$$\text{s.t.} \quad \sum_i \mu_i = C \qquad (16)$$

$$\sum_{j:A_{i,j} \geq \mu_i} (A_{i,j} - \mu_i) = \theta\ ,\ \forall i \text{ s.t. } \mu_i > 0 \qquad (17)$$

$$\sum_j A_{i,j} \leq \theta\ ,\ \forall i \ \text{ s.t. } \ \mu_i = 0 \qquad (18)$$

$$\forall i \ \ \mu_i \geq 0 \ \ ; \ \ \theta \geq 0 \qquad (19)$$

With $\boldsymbol{\mu}$ we can create a matrix B using Lemma 2. Intuitively, the new formulation reduces finding the projection to the $l_{1,\infty}$ ball to finding a new vector of maximum absolute values that will be used to truncate the original matrix. The constraints express that the cumulative mass removed from a row is kept constant across all rows, except for those rows whose coefficients become zero. A final lemma establishes that there is a unique solution to $\mathrm{M}_{1,\infty}$. Therefore, the original projection $\mathrm{P}_{1,\infty}$ reduces to finding the solution of $\mathrm{M}_{1,\infty}$.

Lemma 3 *For any matrix* A *and a constant* C *such that* $C < ||A||_{1,\infty}$*, there is a unique solution* $\boldsymbol{\mu}^*, \theta^*$ *to the problem* $\mathrm{M}_{1,\infty}$.

Proof: For any $\theta \geq 0$ there is a unique $\boldsymbol{\mu}$ that satisfies (17), (18) and (19). To see this, consider $\theta \geq \sum_j A_{i,j}$. In this case we must have $\mu_i = 0$, by equation (18). For $\theta < \sum_j A_{i,j}$ we have $\mu_i = \mathrm{f}_i(\theta)$ where f_i is the inverse of the function

$$\mathrm{g}_i(\mu) = \sum_{j:A_{i,j} \geq \mu} (A_{i,j} - \mu)$$

$\mathrm{g}_i(\mu)$ is a strictly decreasing function in the interval $[0, \max_j A_{i,j}]$ with $\mathrm{g}_i(0) = \sum_j A_{i,j}$ and $\mathrm{g}_i(\max_j A_{i,j}) = 0$. Therefore it is clear that $\mathrm{f}_i(\theta)$ is also well defined on the interval $[0, \sum_j A_{i,j}]$.

Next, define

$$\mathrm{N}(\theta) = \sum_i \mathrm{h}_i(\theta)$$

where $\mathrm{h}_i(\theta) = 0$ if $\theta > \sum_j A_{i,j}$ and $\mathrm{h}_i(\theta) = \mathrm{f}_i(\theta)$ otherwise. $\mathrm{N}(\theta)$ is strictly increasing in the interval $[0, \max_i \sum_j A_{i,j}]$. Hence there is a unique θ^* that satisfies $\mathrm{N}(\theta^*) = C$; and there is a unique $\boldsymbol{\mu}^*$ such that $\mu_i^* = \mathrm{h}_i(\theta^*)$ for each i. □

So far we have assumed that the input matrix A is non-negative. For the general case, it is easy to prove that the optimal projection never changes the sign of a coefficient (Duchi et al., 2008). Thus, given the coefficient matrix W used by our learning algorithm, we can run the $l_{1,\infty}$ projection on A, where A is a matrix made of the absolute values of W, and then recover the original signs after truncating each coefficient.

4.2. An efficient projection algorithm

In this section we describe an efficient algorithm to solve problem $\mathrm{M}_{1,\infty}$. Given a $d \times m$ matrix A and a ball of radius C, the goal is to find a constant θ and a vector $\boldsymbol{\mu}$ of maximums for the new projected matrix, such that $C = \sum_i \mu_i$.

As we have shown in the proof of Lemma 3, $\boldsymbol{\mu}$ and θ can be recovered using functions $\mathrm{N}(\theta)$ and $\mathrm{h}_i(\theta)$. Each function $\mathrm{h}_i(\theta)$ is piecewise linear with $m+1$ intervals. Furthermore $\mathrm{N}(\theta)$, the sum of functions $\mathrm{h}_i(\theta)$, is also piecewise linear with $dm + 1$ intervals. Appendix A describes the intervals and slopes of h_i.

Our algorithm builds these functions piece by piece, until it finds a constant θ that satisfies the conditions of problem $\mathrm{M}_{1,\infty}$; it then recovers $\boldsymbol{\mu}$. The cost of the algorithm is dominated by sorting and merging the x-coordinates of the h_i functions, which form the intervals of N. Therefore the complexity is $O(dm \log dm)$ time and $O(dm)$ in memory, where dm is the number of parameters in A. As a final note, the algorithm only needs to consider non-zero parameters of A. Thus, in this complexity cost, dm can be interpreted as the number of non-zero parameters. This property is particularly attractive for learning methods that maintain sparse coefficients.

5. Synthetic Experiments

For the synthetic experiments we considered a multi-task setting where we compared the $l_{1,\infty}$ projection with both independent l_2 projections for each task and independent l_1 projections. In all cases we used a projected subgradient method, thus the only difference is in the projection step. For all the experiments we used the sum of average hinge losses per task as our objective function. For these experiments as well as the experiments in the following section the learning rate was set to $\eta_0/\sqrt{t}$, where η_0 was chosen to minimize the objective on the training data (we tried values 0.1, 1, 10 and 100). All models were run for 200 iterations.

To create data for these experiments we first generated parameters $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m]$ for all tasks, where each entry was sampled from a normal distribution with 0 mean and unit variance. To generate jointly sparse vectors we randomly selected 10% of the features to be the global set of relevant features V. Then for each task we randomly selected a subset $v \subseteq V$ of relevant features for the task. The size of v was sampled uniformly at random from $\{|V|/2, \ldots, |V|\}$. All parameters outside v were zeroed.

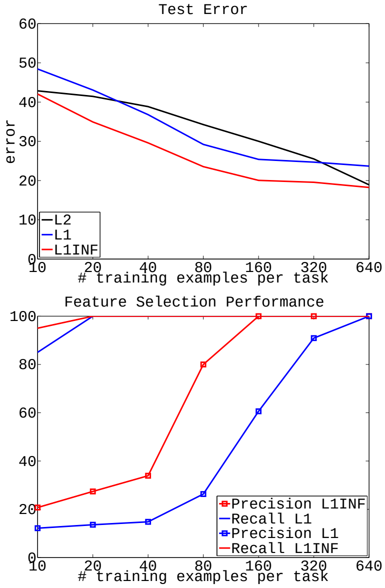

Figure 1. Synthetic experiments, for 60 problems and 200 features (10% relevant). Top: test error. Bottom: feature selection performance.

Each of the dimensions of the training points $\mathbf{x}_i^k$ for each task was also generated from a normal distribution with 0 mean and unit variance. All vectors were then normalized to have unit norm. The corresponding labels y_i^k were set to $\text{sign}(\mathbf{w}_k \cdot \mathbf{x}_k^i)$. The test data was generated in the same fashion. The number of dimensions for these experiments was set to 200 and the number of problems to 60.

We evaluated three types of projections: $l_{1,\infty}$, independent l_2 and independent l_1. For each projection the ball constraints C were set to be the true norm of the corresponding parameters.[2] That is for the $l_{1,\infty}$ norm we set $C = ||W||_{1,\infty}$. For the independent l_1 and l_2 norms we set $C_k = ||\mathbf{w}_k||_1$ and $C_k = ||\mathbf{w}_k||_2$, resp., where C_k is the regularization parameter for task k.

We trained models with different number of training examples ranging from 10 to 640 examples per task and evaluated the classification error of the resulting classifier on the test data.

Figure 1 shows the results of these experiments. As we would expect, given the amount of feature sharing between tasks, the $l_{1,\infty}$ projection results in better generalization than both independent l_1 and independent l_2 projections. Since we know the relevant feature set V, we can evaluate how well the l_1 and $l_{1,\infty}$ projections recovered these features. For each model we take the coefficient matrix W learned and select all the features corresponding to non-zero coefficients for

[2] We also conducted experiments where C was chosen using a validation set and obtained very similar results.

at least one task. The bottom plot shows precision and recall of feature selection for each model, as we increase the number of training examples per task. As we can see both the l_1 model and the $l_{1,\infty}$ can easily recognize that a feature is in the relevant set: the recall for both models is high even with very few training examples. The main difference between the two models is in the precision at recovering relevant features: the $l_{1,\infty}$ model returns significantly sparser solutions, and thus has higher precision.

6. Image Annotation Experiments

In these experiments we use our algorithm in a multi-task learning application. We compare the performance of independent l_2, independent l_1, and joint $l_{1,\infty}$ regularization. In all cases we used the sum of hinge losses as our objective function. To train the l_1 regularized models we used the projected method of (Duchi et al., 2008) (which is a special case of our projection when $m = 1$). To train the l_2 regularized models we used the standard SVM-Light software [3].

For these experiments we collected a dataset from the Reuters news-website. Images on the Reuters website have associated captions. We selected the 40 most frequent content words as our target prediction tasks (a content word is defined as not being in a list of *stop* words). That is, each task involved the binary prediction of whether a content word was an appropriate annotation for an image. Examples of words include: awards, president, actress, actor, match, team, people.

We partitioned the data into three sets: 10,589 images for training, 5,000 images as validation data, and 5,000 images for testing. For each of the 40 most frequent content words we created multiple training sets of different sizes, $n = \{40, 80, 160, 320, 640\}$: each training set contained n positive examples and $2n$ negative examples. All examples for each task were randomly sampled from the pool of supervised training data.

Our image representation is a multi-resolution vocabulary histogram (Nister & Stewenius, 2006; Grauman & Darrell, 2008); each pyramid level is concatenated to form our feature space. As a preprocessing step we performed SVD to obtain a new basis of the image space where features are uncorrelated.

6.1. Evaluation and Significance Testing

To compare the performance of different classifiers we use the AUC criterion, which is commonly used in evaluation on retrieval tasks. For a single task, assuming a labeled test set (x_i, y_i) for $i = 1 \dots n$, the AUC measure for a function f can be expressed (e.g., see Agarwal et al. (2005)) as

$$\frac{1}{n^+} \sum_{i:y_i=+1} \sum_{j:y_j=-1} \frac{I[f(x_i) > f(x_j)]}{n^-} \tag{20}$$

where n^+ is the number of positive examples in the test set, n^- is the number of negative examples, and $I[\pi]$ is the indicator function which is 1 if π is true, 0 otherwise. The AUC measure can be interpreted (Agarwal et al., 2005) as an estimate of the following expectation, which is the probability that the function f correctly ranks a randomly drawn positive example over a randomly drawn negative item:

$$AUC(f) = \mathbf{E}_{X^+ \sim D^+, X^- \sim D^-} \left[I[f(X^+) > f(X^-)] \right]$$

Here D^+ is the distribution over positively labeled examples, and D^- is the distribution over negative examples.

This interpretation allows to develop a simple significance test, based on the sign test, to compare the performance of two classifiers f and g (more specifically, to develop a significance test for the hypothesis that $AUC(f) > AUC(g)$). Assuming that $n^+ < n^-$ (which is the case in all of our experiments), and given a test set, we create pairs of examples (x_i^+, x_i^-) for $i = 1 \dots n^+$. Here each x_i^+ is a positive test example, and x_i^- is an arbitrary negative test example; each positive and negative example is a distinct item from the test set. Given the n^+ pairs, we can calculate the following counts:

$$\begin{aligned} s^+ &= \sum_i I[(f(x_i^+) > f(x_i^-)) \wedge (g(x_i^+) < g(x_i^-))] \\ s^- &= \sum_i I[(f(x_i^+) < f(x_i^-)) \wedge (g(x_i^+) > g(x_i^-))] \end{aligned}$$

These counts are used to calculate significance under the sign test.

In our experiments, the set-up is slightly more complicated, in that we have a multi-task setting where we are simultaneously measuring the performance on several tasks rather than a single task. The test set in our case consists of examples (x_i, y_i, l_i) for $i = 1 \dots n$ where l_i specifies the task for the i-th example. We replace the AUC measure in Eq. 20 with the following measure:

$$\frac{1}{n^+} \sum_l \sum_{i:l_i=l, y_i=+1} \sum_{j:l_i=l, y_j=-1} \frac{I[f_l(x_i) > f_l(x_j)]}{n_l^-}$$

where n^+ is the total number of positive examples in the test set, n_l^- is the number of negative examples for

[3] http://svmlight.joachims.org/

Table 1. Significance tests for the Image Annotation task, comparing $l_{1,\infty}$ with l_2 and l_1.

# samples	l_2 p-value	l_1 p-value
4,800	0.0042	0.00001
9,600	0.0002	0.009
19,200	0.00001	0.00001
38,400	0.35	0.36
63,408	0.001	0.46

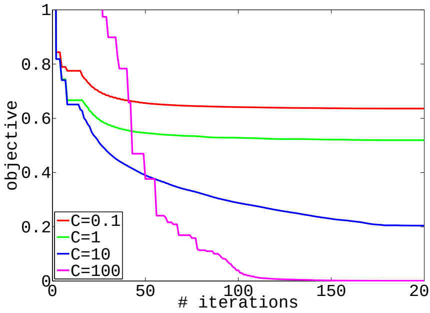

Figure 2. Convergence of $l_{1,\infty}$ models for different values of C and best η_0 in the Image Annotation task using 63,408 samples.

the l-th task, and f_l is the classifier for the l-th task. It is straightforward to develop a variant of the sign test for this setting; for brevity the details are omitted.

6.2. Results

We trained models for $C = \{0.1, 1, 10, 100\}$. Figure 2 shows the convergence of the $l_{1,\infty}$ models for the largest training set. To validate the constant C for each model we assume that we have 10 words for which we have validation data. We chose the C that maximized the AUC.

Figure 3 shows results on the Reuters dataset for l_2, l_1 and $l_{1,\infty}$ regularization as a function of the total number training samples. Table 1 shows the corresponding significance tests for the difference between $l_{1,\infty}$ and the other two models. As we can see the $l_{1,\infty}$ regularization performs significantly better than both l_1 and l_2 for training sizes of less than $20,000$ samples, and for larger sizes all models seem to perform similarly.

Figure 4 shows the accuracy of the learned model as a function of the total number of non-zero features (i.e. the number of features that were used by any of the tasks), where we obtained solutions of different sparsity by controlling the C parameter. Notice that the $l_{1,\infty}$ model is able to find a solution of 66% average AUC using only 30% of the features while the l_1 model needs to use 95% of the features to achieve a performance of 65%.

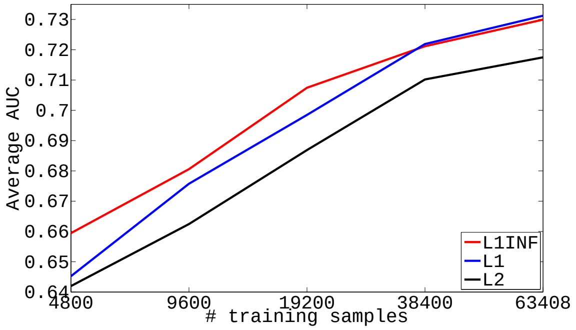

Figure 3. Model Comparison in the Image Annotation task: $l_{1,\infty}$, l_1 and l_2.

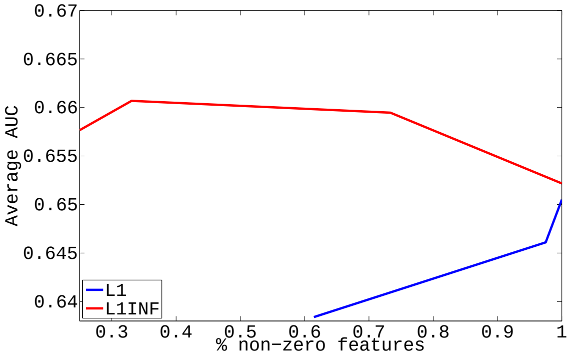

Figure 4. AUC measure with respect joint sparsity in the Image Annotation task, for l_1 and $l_{1,\infty}$.

7. Conclusion

In this paper we have presented a simple and effective projected gradient method for training joint models with $l_{1,\infty}$ constraints. We have shown that projections onto the $l_{1,\infty}$ ball can be done efficiently by presenting an algorithm that can compute them in $O(n \log n)$ time and $O(n)$ memory. This matches the computational cost of the most efficient algorithm (Duchi et al., 2008) for computing l_1 projections.

We have applied our algorithm to a multi-task image annotation problem. Our results show that $l_{1,\infty}$ leads to better performance than both l_2 and l_1 regularization. Furthermore $l_{1,\infty}$ is effective in discovering jointly sparse solutions.

One advantage of our approach is that it can be easily extended to work on an online convex optimization setting (Zinkevich, 2003). Future work should explore this possibility.

References

Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., & Roth, D. (2005). Generalization bounds for the area under the roc curve. *Journal of Machine Learning Research*, *6*, 393–425.

Argyriou, A., Evgeniou, T., & Pontil, M. (2007). Multi-task feature learning. *Advances in Neural Information Processing Systems 19* (pp. 41–48).

Bertsekas, D. (1999). *Nonlinear programming.* Athena Scientific.

Donoho, D. (2004). *For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution.* (Technical Report). Statistics Dept., Stanford University.

Duchi, J., Shalev-Shwartz, S., Singer, Y., & Chandra, T. (2008). Efficient projections onto the l1-ball for learning in high dimensions. *Proc. of Intl. Conf. on Machine Learning* (pp. 272–279).

Grauman, K., & Darrell, T. (2008). The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, *8*, 725–760.

Lee, S. I., Ganapathi, V., & Koller, D. (2007). Effcient structure learning of markov networks using l1-regularization. *Advances in Neural Information Processing Systems 19* (pp. 817–824).

Meier, L., van de Geer, S., & Buhlmann, P. (2006). *The group lasso for logistic regression* (Technical Report). ETH Seminar fur Statistik.

Ng, A. Y. (2004). Feature selection, l1 vs. l2 regularization, and rotational invariance. *Proc. of Intl. Conf. on Machine Learning.*

Nister, D., & Stewenius, H. (2006). Scalable recognition with a vocabulary tree. *Proc. of Conf. on Computer Vision and Pattern Recognition.*

Obozinski, G., Taskar, B., & Jordan, M. (2006). *Multitask feature selection* (Technical Report). Statistics Dept., University of California, Berkeley.

Park, M. Y., & Hastie, T. (2006). *Regularization path algorithms for detecting gene interactions* (Technical Report). Stanford University.

Quattoni, A., Collins, M., & Darrell, T. (2008). Transfer learning for image classification with sparse prototype representations. *Proc. of Conf. on Computer Vision and Pattern Recognition.*

Schmidt, M., Murphy, K., Fung, G., & Rosale, R. (2008). Structure learning in random fields for heart motion abnormality detection. *Proc. of Conf. on Computer Vision and Pattern Recognition.*

Schmidt, M., van den Berg, E., Friedlander, M., & Murphy, K. (2009). Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. *Proc. of Conf. on Artificial Intelligence and Statistics* (pp. 456–463).

Shalev-Shwartz, S., Singer, Y., & Srebro, N. (2007). Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. *Proc. of Intl. Conf. on Machine Learning* (pp. 807–814).

Similä, T., & Tikka, J. (2007). Input selection and shrinkage in multiresponse linear regression. *Computational Statistics and Data Analysis*, *52*, 406–422.

Tropp, J. (2006). Algorithms for simultaneous sparse approximation, part ii: convex relaxation. *Signal Processing* (pp. 589–602).

Turlach, B., Venables, W., & Wright, S. (2005). Simultaneous variable selection. *Technometrics*, *47*, 349–363.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. Royal Statistical Society Series B*, *68*, 49–67.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. *Proc. of Intl. Conf. on Machine Learning* (pp. 928–936).

Acknowledgements

We would like to thank John Duchi and Yoram Singer for useful discussions about this work. A. Quattoni and M. Collins were supported by NSF grant 0347631.

A. Appendix: Computation of h_i

In this section we describe the intervals and slope of piecewise linear functions h_i. Let $\mathbf{s}_i$ be a vector of the coefficients of row i in A sorted in decreasing order, with an added 0 at position $m+1$, $s_{i,1} \geq s_{i,2} \geq \ldots \geq s_{i,m} \geq s_{i,m+1} = 0$. Then, let us define points $r_{i,k} = \sum_{j:A_{i,j} \geq s_{i,k}} (A_{i,j} - s_{i,k}) = \sum_{j=1}^{k} (s_{i,j} - s_{i,k}) = \sum_{j=1}^{k-1} s_{i,j} - (k-1)s_{i,k}$, for $1 \leq k \leq m+1$. Each point $r_{i,k}$ corresponds to the reduction in magnitude for row i that is obtained if we set the new maximum to $s_{i,k}$. Clearly $\mathrm{h}_i(r_{i,k}) = s_{i,k}$. Furthermore it is easy to see that h_i is piecewise linear with intervals $[r_{i,k}, r_{i,k+1}]$ for $1 \leq k \leq m$ and slope:

$$\frac{s_{i,k+1} - s_{i,k}}{\sum_{j=1}^{k} s_{i,j} - k s_{i,k+1} - \sum_{j=1}^{k-1} s_{i,j} + (k-1) s_{i,k}} = -\frac{1}{k}$$

After point $r_{i,m+1}$ the function is constant and its value is zero. Note that this comes from equation (18) that establishes that $\mu_i = 0$ for $\theta > r_{i,m+1} = \sum_{j=1}^{m} A_{i,j}$.