Exclusivity Regularized Machine: A New Ensemble SVM Classifier

Xiaojie Guo[†*] **Xiaobo Wang**[‡*] **Haibin Ling**[§]
[†]State Key Laboratory of Information Security, IIE, Chinese Academy of Sciences
[*]University of Chinese Academy of Sciences
[‡]National Laboratory of Pattern Recognition, IA, Chinese Academy of Sciences
[§]Department of Computer and Information Sciences, Temple University
xj.max.guo@gmail.com xiaobo.wang@ia.ac.cn hbling@temple.edu

Abstract

The diversity of base learners is of utmost importance to a good ensemble. This paper defines a novel measurement of diversity, termed as *exclusivity*. With the designed exclusivity, we further propose an ensemble SVM classifier, namely *Exclusivity Regularized Machine* (ExRM), to jointly suppress the training error of ensemble and enhance the diversity between bases. Moreover, an Augmented Lagrange Multiplier based algorithm is customized to effectively and efficiently seek the optimal solution of ExRM. Theoretical analysis on convergence, global optimality and linear complexity of the proposed algorithm, as well as experiments are provided to reveal the efficacy of our method and show its superiority over state-of-the-arts in terms of accuracy and efficiency.

1 Introduction

Classification is a major task in machine learning and pattern recognition. In binary classification, a hypothesis is constructed from a feasible hypothesis space based on the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\{\mathbf{x}_i\}_{i=1}^N$ is a set of data points with $\mathbf{x}_i \in \mathbb{R}^d$ sampled i.i.d. under a distribution from an input subspace, and $\{y_i\}_{i=1}^N$ with $y_i \in \{-1, +1\}$ is the corresponding label set. The obtained hypothesis, also known as classifier, is "good" when it is capable to generalize well the "knowledge" learned from the training data to unseen instances. Multi-class cases can be analogously accomplished by a group of binary classifiers.

Over the past decades, numerous classifiers have been devised. The k-nearest neighbor (k-NN) [Altman, 1992] is an intuitive and simple one. However, its performance heavily depends on the selection of k and is sensitive to noisy data, making k-NN vulnerable in practice. Decision tree (DT) learning is to construct a tree by partitioning the source set into subsets according to a feature test. CART [Breiman *et al.*, 1984] and C4.5 [Quinlan, 1993] are two examples of DT. The main drawbacks of decision trees are their sub-optimality and overfitting problems. Support vector machine (SVM) [Vapnik, 1995], as one of the most robust and accurate approaches, aims to find a hyperplane with the maximum margin between classes. Besides, linear discriminant analysis [Fisher, 1936] and naive Bayes [Domingos and Pazzani, 1997] are other classic classifiers.

As has been well recognized, a combination of various classifiers can improve predictions. Ensemble approaches, with Boosting [Freund and Schapire, 1997] and Bagging [Breiman, 1996] as representatives, make use of this recognition and achieve strong generalization performance [Zhou, 2012]. The generalization error of ensemble mainly depends on two factors, formally expressed as $E = \bar{E} - \bar{A}$, where E is the mean-square error of the ensemble, $\bar{E}$ represents the average mean-square error of component learners and $\bar{A}$ stands for the average squared difference (diversity) between the ensemble and the components. *Error-Ambiguity decomposition* [Krogh and Vedelsby, 1995], *Bias-Variance-Covariance decomposition* [Ueda and Nakano, 1996] and *Strength-Correlation decomposition* [Breiman, 2001] all confirm the above principle. Specifically, Boosting is a family of ensemble learning meta-algorithms, which improves the classification performance through letting the subsequent base learner pay more attention on data that have been wrongly grouped by the previous bases. AdaBoost is probably the most prominent Boosting scheme. While Bagging represents another category that tries to train a set of diverse weak classifiers by utilizing the bootstrap sampling technique. Several specific algorithms, such as random subspace [Ho, 1998], random forest [Breiman, 2001] and rotation forest [Rodriguez *et al.*, 2006], have been proposed to further enforce the diversity of ensemble members. Certainly, both Boosting and Bagging can be applied to (most of) the conventional classifiers, *e.g.* DT and SVM.

Considering the popularity of SVM and the potential of ensemble, it would be beneficial to equip SVM with ensemble thoughts. Employing SVM as the base learner of Bagging or Boosting is a natural manner. Alternatively, this work provides a model to train a set of SVMs and integrate them as a homogeneous ensemble. The model jointly minimizes the training error and maximizes the diversity of base learners. Different from Bagging, our method does not rely on sampling schemes to achieve the diversity, although it is flexible to embrace such schemes. Additionally, the proposed method simultaneously trains all the base learners unlike Boosting methods doing the job sequentially. Concretely, the contribution of this paper can be summarized as follows: 1) we define a new measurement, namely (relaxed) exclusivity, to manage

the diversity between base learners, 2) we propose a novel ensemble, called Exclusivity Regularized Machine (ExRM), which concerns the training error and the diversity of components simultaneously, and 3) we design an Augmented Lagrange Multiplier based algorithm to efficiently seek the solution of ExRM. Theoretical analysis on convergence, global optimality and linear complexity of the proposed algorithm is also provided.

2 Our Method

2.1 Preliminary

Arguably, SVM [Vapnik, 1995][Keerthi and DeCoste, 2005] is one of the most popular classifiers due to its promising performance. In general, the primal SVM (PSVM) can be modeled as follows:

$$\underset{\{\mathbf{w},b\}}{\operatorname{argmin}} \Psi(\mathbf{w}) + \lambda \sum_{i=1}^{N} f(y_i, \phi(\mathbf{x}_i)^T \mathbf{w} + b), \tag{1}$$

where $f(\cdot)$ is a penalty function, $\Psi(\mathbf{w})$ performs as a regularizer on the learner $\mathbf{w}$ and b is the bias. The function $\phi(\cdot)$ is to map $\mathbf{x}_i$ from the original D-dimensional feature space to a new M-dimensional one. Moreover, λ is a non-negative coefficient that provides a trade-off between the loss term and the regularizer. If PSVM adopts the hinge loss as penalty, the above (1) turns out to be:

$$\underset{\{\mathbf{w},b\}}{\operatorname{argmin}} \Psi(\mathbf{w}) + \lambda \sum_{i=1}^{N} \left(1 - (\phi(\mathbf{x}_i)^T \mathbf{w} + b) y_i\right)_+^p, \tag{2}$$

where the operator $(u)_+ := \max(u, 0)$ keeps the input scalar u unchanged if u is non-negative, returns zero otherwise, the extension of which to vectors and matrices is simply applied element-wise. Furthermore, p is a constant typically in the range $[1, 2]$ for being meaningful. In practice, p is often selected to be either 1 or 2 for ease of computation, which correspond to ℓ^1-norm and ℓ^2-norm loss PSVMs, respectively. As for the regularization term, $\Psi(\mathbf{w}) := \frac{1}{2}\|\mathbf{w}\|_2^2$ (ℓ^2 regularizer) and $\Psi(\mathbf{w}) := \|\mathbf{w}\|_1$ (ℓ^1 regularizer) are two classical options.

2.2 Exclusivity Regularized Machine

It is natural to extend the traditional PSVM (2) to the following ensemble version with C components as:

$$\underset{\{\mathbf{W},\mathbf{b}\}}{\operatorname{argmin}} \Psi(\mathbf{W}) + \lambda \sum_{c=1}^{C} \sum_{i=1}^{N} \left(1 - (\phi(\mathbf{x}_i)^T \mathbf{w}_c + b_c) y_i\right)_+^p, \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{M \times C} := [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_C]$ and $\mathbf{b} \in \mathbb{R}^C := [b_1, b_2, .., b_C]^T$. Suppose we simply impose $\frac{1}{2}\|\mathbf{W}\|_F^2$ or $\|\mathbf{W}\|_1$ on $\mathbf{W}$, all the components (and the ensemble) will have no difference with the hypothesis directly calculated from (2) using the same type of regularizer.[1] From this view, $\Psi(\mathbf{W})$ is critical to achieving the diversity.

[1] We note that splitting the training data into C sub-sets, like the bootstrap sampling, and training C classifiers separately on the subsets would lead to some difference between the components. However, this strategy does not essentially advocate the diversity of base learners. Although how to sample training subsets is not the focus of this work, our model has no difficulties to adopt such sampling strategies.

Prior to introducing our designed regularizer, we first focus on the concept of diversity. Although the diversity has no formal definition so far, the thing in common among studied measurements is that the diversity enforced in a pairwise form between members strikes a good balance between complexity and effectiveness. The evidence includes Q-statistics measure [Kuncheva *et al.*, 2003], correlation coefficient measure [Kuncheva *et al.*, 2003], disagreement measure [Ho, 1998], double-fault measure [Giacinto and Roli, 2001], k-statistic measure [Dietterich, 2000], mutual angular measure [Yu *et al.*, 2011; Xie *et al.*, 2015b; 2015a] and competition measure [Du and Ling, 2014]. These measures somehow enhance the diversity, however, most of them are heuristic. One exception is Diversity Regularized Machine [Yu *et al.*, 2011], which attempts to seek the globally-optimal solution. Unfortunately, it often fails because the condition required for the global optimality, say $\|\mathbf{w}_c\|_2 = 1$ for all c, is not always satisfied. Further, Li *et al.* proposed a pruning scheme to improve the performance of DRM [Li *et al.*, 2012]. But, DRM requires too much time to converge, which limits its applicability. Below, we define a new measure of diversity, *i.e.* (relaxed) exclusivity.

Definition 1. (**Exclusivity**) *Exclusivity between two vectors $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^m$ is defined as $\mathcal{X}(\mathbf{u}, \mathbf{v}) := \|\mathbf{u} \odot \mathbf{v}\|_0 = \sum_{i=1}^m \mathbf{u}(i) \cdot \mathbf{v}(i) \neq 0$, where $\odot$ designates the Hadamard product, and $\|\cdot\|_0$ is the ℓ^0 norm.*

From the definition, we can observe that the exclusivity encourages two vectors to be as orthogonal as possible. Due to the non-convexity and discontinuity of ℓ^0 norm, we have the following relaxed exclusivity.

Definition 2. (**Relaxed Exclusivity**) *The definition of relaxed exclusivity between $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^m$ is given as $\mathcal{X}_r(\mathbf{u}, \mathbf{v}) := \|\mathbf{u} \odot \mathbf{v}\|_1 = \sum_{i=1}^m |\mathbf{u}(i)| \cdot |\mathbf{v}(i)|$, where $|u|$ is the absolute value of u. The relaxation is similar with that of the ℓ^1 norm to the ℓ^0 norm.*

It can be easily verified that $\|\mathbf{u}\|_0 = \mathcal{X}(\mathbf{u}, \mathbf{1})$, $\|\mathbf{u}\|_1 = \mathcal{X}_r(\mathbf{u}, \mathbf{1})$ and $\|\mathbf{u}\|_2^2 = \mathcal{X}_r(\mathbf{u}, \mathbf{u})$, where $\mathbf{1} \in \mathbb{R}^m$ is the vector with all of its m entries being 1.

Instead of directly employing $\sum_{1 \leq \tilde{c} \neq c \leq C} \mathcal{X}_r(\mathbf{w}_c, \mathbf{w}_{\tilde{c}})$ as the final $\Psi(\mathbf{W})$, we adopt the following:

$$\begin{aligned} \Psi(\mathbf{W}) &:= \frac{1}{2}\|\mathbf{W}\|_F^2 + \sum_{1 \leq \tilde{c} \neq c \leq C} \mathcal{X}_r(\mathbf{w}_c, \mathbf{w}_{\tilde{c}}) \\ &= \frac{1}{2} \sum_{i=1}^{M} \left(\sum_{c=1}^{C} |\mathbf{w}_c(i)| \right)^2 = \frac{1}{2}\|\mathbf{W}^T\|_{1,2}^2. \end{aligned} \tag{4}$$

The main reasons of bringing $\frac{1}{2}\|\mathbf{W}\|_F^2$ into the regularizer are: 1) it essentially enhances the stability of solution, 2) it tends to mitigate the scale issue by penalizing large columns, and 3) as the relaxed exclusivity itself is non-convex, the introduction guarantees the convexity of the regularizer. Finally, the proposed Exclusivity Regularized Machine (ExRM) can be written in the following shape:

$$\min_{\{\mathbf{w}_c, b_c\}} \frac{1}{2}\|\mathbf{W}^T\|_{1,2}^2 + \lambda \sum_{c=1}^{C} \sum_{i=1}^{N} \left(1 - (\phi(\mathbf{x}_i)^T \mathbf{w}_c + b_c) y_i\right)_+^p. \tag{5}$$

Remarks As expressed in Eq. (4), we have motivated the $\ell_{1,2}$ regularizer from a novel perspective. It has been verified that, as one of mixed norms, the $\ell_{1,2}$ is in nature able to capture some structured sparsity [Kowalski, 2009]. In general, the regression models using such mixed norms can be solved by a modified FOCUSS algorithm [Kowalski, 2009]. Zhou *et al.* [Zhang *et al.*, 2010] introduced the $\ell_{1,2}$ regularizer into a specific task, *i.e.* multi-task feature selection, and used the subgradient method to seek the solution of the associated optimization problem. The responsibility of the $\ell_{1,2}$ regularizer is to enforce the negative correlation among categories [Zhang *et al.*, 2010]. Recently, Kong *et al.* [Kong *et al.*, 2014] utilized $\ell_{1,2}$ norm to bring out sparsity at intra-group level in feature selection, and proposed an effective iteratively re-weighted algorithm to solve the corresponding optimization problem. In this work, besides the view of motivating the $\ell_{1,2}$ regularizer, its role in our target problem, say constructing an ensemble of SVMs, is also different with the previous work [Kowalski, 2009; Zhang *et al.*, 2010; Kong *et al.*, 2014]. The functionalities of [Zhang *et al.*, 2010] and [Kong *et al.*, 2014] are the intra-exclusivity of multiple hypotheses (tasks) and the inter-exclusivity of a single hypothesis respectively, while our principle is the diversity of multiple components of a single ensemble hypothesis.

2.3 Optimization

With the trick that $1-(\phi(\mathbf{x}_i)^T\mathbf{w}_c+b_c)y_i = y_iy_i-(\phi(\mathbf{x}_i)^T\mathbf{w}_c+b_c)y_i = y_i(y_i-(\phi(\mathbf{x}_i)^T\mathbf{w}_c+b_c))$, we introduce auxiliary variables $e_i^c := y_i-(\phi(\mathbf{x}_i)^T\mathbf{w}_c+b_c)$. In the sequel, the minimization of (5) can be converted into:

$$\begin{aligned}&\underset{\{\mathbf{W},\mathbf{b}\}}{\operatorname{argmin}}\frac{1}{2}\|\mathbf{W}^T\|_{1,2}^2+\lambda\big(\mathbf{Y}\odot\mathbf{E}\big)_+^p\\&\text{s.t.}\quad \mathbf{P}=\mathbf{W};\ \ \mathbf{E}=\mathbf{Y}-(\mathbf{X}^T\mathbf{P}+\mathbf{1}\mathbf{b}^T),\end{aligned}\tag{6}$$

where $\mathbf{X}\in\mathbb{R}^{M\times N} := [\phi(\mathbf{x}_1),\phi(\mathbf{x}_2),...,\phi(\mathbf{x}_N)]$, $\mathbf{e}^c\in\mathbb{R}^N := [e_1^c,e_2^c,...,e_N^c]^T$, $\mathbf{E}\in\mathbb{R}^{N\times C} := [\mathbf{e}^1,\mathbf{e}^2,...,\mathbf{e}^C]$ and $\mathbf{y}\in\mathbb{R}^N := [y_1,y_2,...,y_N]^T$. And each column of $\mathbf{Y}\in\mathbb{R}^{N\times C}$ is $\mathbf{y}$. Please note that, the constraint $\mathbf{P}=\mathbf{W}$ is added to make the objective separable and thus solvable by the ALM framework. It is worth mentioning that, thanks to the convexity of each term in the objective and the linearity of the constraints, the target problem is convex. The Lagrangian function of (6) can be written in the following form:

$$\begin{aligned}&\mathcal{L}(\mathbf{W},\mathbf{b},\mathbf{E},\mathbf{P}) := \frac{1}{2}\|\mathbf{W}^T\|_{1,2}^2+\lambda\big(\mathbf{Y}\odot\mathbf{E}\big)_+^p+\\&\Phi(\mathbf{Q},\mathbf{P}-\mathbf{W})+\Phi(\mathbf{Z},\mathbf{E}-\mathbf{Y}+\mathbf{X}^T\mathbf{P}+\mathbf{1}\mathbf{b}^T),\end{aligned}\tag{7}$$

with the definition $\Phi(\mathbf{U},\mathbf{V}) := \frac{\mu}{2}\|\mathbf{V}\|_F^2+\langle\mathbf{U},\mathbf{V}\rangle$, where $\langle\cdot,\cdot\rangle$ represents matrix inner product and μ is a positive penalty scalar. In addition, $\mathbf{Q}\in\mathbb{R}^{M\times C}$ and $\mathbf{Z}\in\mathbb{R}^{N\times C}$ are Lagrangian multipliers. The proposed solver iteratively updates one variable at a time by fixing the others. Below are the solutions to sub-problems.

$\mathbf{W}$ sub-problem With the variables unrelated to $\mathbf{W}$ fixed, we have the sub-problem of $\mathbf{W}$:

$$\mathbf{W}^{(t+1)}=\underset{\mathbf{W}}{\operatorname{argmin}}\frac{1}{2}\|\mathbf{W}^T\|_{1,2}^2+\Phi(\mathbf{Q}^{(t)},\mathbf{P}^{(t)}-\mathbf{W}).\tag{8}$$

As observed from the problem (8), it can be split into a set of smaller problems. For each row $\mathbf{W}_{\cdot j}$, instead of directly optimizing (8), we resolve the following equivalent objective:

$$\mathbf{W}_{\cdot j}^{(t+1)}=\underset{\mathbf{W}_{\cdot j}}{\operatorname{argmin}}\frac{1}{2}\mathbf{W}_{\cdot j}\mathbf{G}\mathbf{W}_{\cdot j}^T+\Phi(\mathbf{Q}_{\cdot j}^{(t)},\mathbf{P}_{\cdot j}^{(t)}-\mathbf{W}_{\cdot j}),\tag{9}$$

where $\mathbf{G}$ is formed by:

$$\mathbf{G} := \operatorname{Diag}\left(\left[\frac{\|\mathbf{W}_{\cdot j}\|_1}{|\mathbf{W}_{\cdot j}(1)|+\epsilon},...,\frac{\|\mathbf{W}_{\cdot j}\|_1}{|\mathbf{W}_{\cdot j}(C)|+\epsilon}\right]\right),\tag{10}$$

where $\epsilon\to 0^+$ (a small constant) is introduced to avoid zero denominators.[2] Since both $\mathbf{G}$ and $\mathbf{W}_{\cdot j}$ depend on $\mathbf{W}_{\cdot j}$, to find out the solution to (9), we employ an efficient reweighted algorithm to iteratively update $\mathbf{G}$ and $\mathbf{W}_{\cdot j}$. As for $\mathbf{W}_{\cdot j}$, with $\mathbf{G}$ fixed, equating the partial derivative of (9) with respect to $\mathbf{W}_{\cdot j}$ to zero yields:

$$\mathbf{W}_{\cdot j}^{(k+1)}=(\mu^{(t)}\mathbf{P}_{\cdot j}^{(t)}+\mathbf{Q}_{\cdot j}^{(t)})(\mathbf{G}^{(k)}+\mu^{(t)}\mathbf{I})^{-1}.\tag{11}$$

Then $\mathbf{G}^{(k+1)}$ is updated using $\mathbf{W}_{\cdot j}^{(k+1)}$ as in (10). The procedure summarized in Alg. 1 terminates when converged.

$\mathbf{b}$ sub-problem Dropping the terms independent on $\mathbf{b}$ leads to a least squares regression problem:

$$\begin{aligned}\mathbf{b}^{(t+1)}&=\underset{\mathbf{b}}{\operatorname{argmin}}\,\Phi(\mathbf{Z}^{(t)},\mathbf{E}^{(t)}-\mathbf{Y}+\mathbf{X}^T\mathbf{P}^{(t)}+\mathbf{1}\mathbf{b}^T)\\&=\big(\mathbf{Y}-\mathbf{E}^{(t)}-\mathbf{X}^T\mathbf{P}^{(t)}-\frac{\mathbf{Z}^{(t)}}{\mu^{(t)}}\big)^T\big(\frac{1}{N}\mathbf{1}\big).\end{aligned}\tag{12}$$

$\mathbf{E}$ sub-problem Similarly, picking out the terms related to $\mathbf{E}$ gives the following problem:

$$\mathbf{E}^{(t+1)}=\underset{\mathbf{E}}{\operatorname{argmin}}\frac{\lambda}{\mu^{(t)}}\big(\mathbf{Y}\circ\mathbf{E}\big)_+^p+\frac{1}{2}\|\mathbf{E}-\mathbf{S}^{(t)}\|_F^2,\tag{13}$$

where $\mathbf{S}^{(t)} := \mathbf{Y}-\mathbf{X}^T\mathbf{P}^{(t)}-\mathbf{1}\mathbf{b}^{(t+1)T}-\frac{\mathbf{Z}^{(t)}}{\mu^{(t)}}$. It can be seen that the above is a single-variable 2-piece piecewise function. Thus, to seek the minimum of each element in $\mathbf{E}$, we just need to pick the smaller between the minima when $y_ie_i^c\geq 0$ and $y_ie_i^c<0$. Moreover, we can provide the explicit solution when $p := 1$ or 2 (for arbitrary p we will discuss it latter). When $p := 1$:

$$\mathbf{E}^{(t+1)}=\mathbf{\Omega}\circ\mathcal{S}_{\frac{\lambda}{\mu^{(t)}}}[\mathbf{S}^{(t)}]+\bar{\mathbf{\Omega}}\circ\mathbf{S}^{(t)}.\tag{14}$$

For $p := 2$:

$$\mathbf{E}^{(t+1)}=\mathbf{\Omega}\circ\mathbf{S}^{(t)}/(1+\frac{2\lambda}{\mu^{(t)}})+\bar{\mathbf{\Omega}}\circ\mathbf{S}^{(t)},\tag{15}$$

where $\mathbf{\Omega}\in\mathbb{R}^{N\times C} := (\mathbf{Y}\circ\mathbf{S}^{(t)}>0)$ is an indicator matrix, and $\bar{\mathbf{\Omega}}$ is the complementary support of $\mathbf{\Omega}$. The definition of shrinkage operator on scalars is $\mathcal{S}_{\epsilon>0}[u] := \operatorname{sgn}(u)\max(|u|-\epsilon,0)$. The extension of the shrinkage operator to vectors and matrices is simply applied element-wise.

[2] The derived algorithm can be proved to minimize $\|\mathbf{W}^T+\epsilon\|_{1,2}^2$. Certainly, when $\epsilon\to 0^+$, $\|\mathbf{W}^T+\epsilon\|_{1,2}^2$ infinitely approaches to $\|\mathbf{W}^T\|_{1,2}^2$.

Algorithm 1: $\mathbf{W}$ Solver

Input: $\mathbf{W}^{(t)}$, $\mathbf{P}^{(t)}$, $\mathbf{Q}^{(t)}$, $\mu^{(t)}$.
Initial.: $k \leftarrow 0$; $\mathbf{H}^{(k)} \leftarrow \mathbf{W}^{(t)}$;
for $j = 0 : M$ **do**
  while *not converged* **do**
    Update $\mathbf{G}^{(k+1)}$ via Eq. (10);
    Update $\mathbf{H}_{\cdot j}^{(k+1)}$ via Eq. (11); $k \leftarrow k + 1$;
  end
end
Output: $\mathbf{W}^{(t+1)} \leftarrow \mathbf{H}^{(k)}$

Algorithm 2: Exclusivity Regularized Machine

Input: Training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, positive integer C and positive real value λ.
Initial.: $t \leftarrow 0$; $\mathbf{W}^{(t)} \in \mathbb{R}^{M\times C} \leftarrow \mathbf{1}$; $\mathbf{b}^{(t)} \in \mathbb{R}^{C} \leftarrow \mathbf{0}$;
$\mathbf{P}^{(t)} \in \mathbb{R}^{M\times C} \leftarrow \mathbf{0}$;$\mathbf{Q}^{(t)} \in \mathbb{R}^{M\times C} \leftarrow \mathbf{1}$;
$\mathbf{Z}^{(t)} \in \mathbb{R}^{N\times C} \leftarrow \mathbf{0}$; $\mu^{(t)} \leftarrow 1$; $\rho \leftarrow 1.1$;
while *not converged* **do**
  Update $\mathbf{W}^{(t+1)}$ via Alg. 1;
  Update $\mathbf{b}^{(t+1)}$ via Eq. (12);
  Update $\mathbf{E}^{(t+1)}$ via Eq. (14) or (15);
  Update $\mathbf{P}^{(t+1)}$ via Eq. (17);
  Update Multipliers and $\mu^{(t+1)}$ via Eq. (18);
  $t \leftarrow t + 1$;
end
Output: Final Ensemble $\frac{1}{C}(\sum_{i=1}^{C} \mathbf{w}_c^{(t)}, \sum_{i=1}^{C} b_c^{(t)})$

$\mathbf{P}$ sub-problem There are two terms involving $\mathbf{P}$. The associated optimization problem reads:

$$\begin{aligned}\mathbf{P}^{(t+1)} = \underset{\mathbf{P}}{\operatorname{argmin}}\, \Phi(\mathbf{Q}^{(t)}, \mathbf{P} - \mathbf{W}^{(t+1)}) + \\ \Phi(\mathbf{Z}^{(t)}, \mathbf{E}^{(t+1)} - \mathbf{Y} + \mathbf{X}^T\mathbf{P} + \mathbf{1}\mathbf{b}^{(t+1)T}).\end{aligned} \tag{16}$$

Its closed-form solution can be obtained by:

$$\mathbf{P}^{(t+1)} = \mathbf{K}\big(\mathbf{W}^{(t+1)} - \frac{\mathbf{Q}^{(t)}}{\mu^{(t)}} + \mathbf{X}(\mathbf{M} - \mathbf{E}^{(t+1)})\big), \tag{17}$$

where we denote $\mathbf{K} := (\mathbf{I} + \mathbf{X}\mathbf{X}^T)^{-1}$ and $\mathbf{M} := \mathbf{Y} - \mathbf{1}\mathbf{b}^{(t+1)T} - \frac{\mathbf{Z}^{(t)}}{\mu^{(t)}}$.

Multipliers and μ Two multipliers and μ are updated by:

$$\begin{aligned}&\mathbf{Z}^{(t+1)} = \mathbf{Z}^{(t)} + \mu^{(t)}(\mathbf{E}^{(t+1)} - \mathbf{Y} + \mathbf{X}^T\mathbf{P}^{(t+1)} + \mathbf{1}\mathbf{b}^{(t+1)T});\\ &\mathbf{Q}^{(t+1)} = \mathbf{Q}^{(t)} + \mu^{(t)}(\mathbf{P}^{(t+1)} - \mathbf{W}^{(t+1)});\\ &\mu^{(t+1)} = \rho\mu^{(t)}, \rho > 1.\end{aligned} \tag{18}$$

For clarity, the procedure of solving (2) is outlined in Algorithm 2. The algorithm should not be terminated until the change of objective value is smaller than a pre-defined threshold (in the experiments, we use 0.05). Please see Algorithm 2 for other details that we can not cover in the text.

3 Theoretical Analysis

First, we come to the loss term of ExRM (5), which assesses the total penalty of base learners as $\sum_{c=1}^{C}\sum_{i=1}^{N}\big(1 - (\phi(\mathbf{x}_i)^T\mathbf{w}_c + b_c)y_i\big)_+^p$, where $p \geq 1$. Alternatively, the loss of the ensemble $\{\mathbf{w}_e, b_e\} := \{\frac{1}{C}\sum_{c=1}^{C}\mathbf{w}_c, \frac{1}{C}\sum_{c=1}^{C} b_c\}$ is $\sum_{i=1}^{N}\big(1 - (\phi(\mathbf{x}_i)^T\mathbf{w}_e + b_e)y_i\big)_+^p$. We have the relationship between the two losses as described in Proposition 1.

Proposition 1. *Let $\{\mathbf{w}_1, b_1\}$,..., $\{\mathbf{w}_C, b_C\}$ be the component learners obtained by ExRM (Alg. 2), and $\{\mathbf{w}_e, b_e\} := \{\frac{1}{C}\sum_{c=1}^{C}\mathbf{w}_c, \frac{1}{C}\sum_{c=1}^{C} b_c\}$ the ensemble, the loss of $\{\mathbf{w}_e, b_e\}$ is bounded by the average loss of the base learners.*

Proof. This can be established by Jensen's inequality. □

The proposition indicates that as we optimize ExRM (5), an upper bound of the loss of the ensemble is also minimized. Thus, incorporating with our proposed regularizer, ExRM is able to achieve the goal of simultaneously optimizing the training error of ensemble and the diversity of components.

One may wonder why not minimizing the loss of the ensemble, which seems reasonable, like:

$$\min_{\{\mathbf{W},b\}} \frac{1}{2}\|\mathbf{W}^T\|_{1,2}^2 + \tilde{\lambda}\sum_{i=1}^{N}\big(1 - (\phi(\mathbf{x}_i)^T\mathbf{w}_e + b_e)y_i\big)_+^p. \tag{19}$$

The reason is that the optimal solution of (19) is exactly that of PSVM (2). The solution of (19) satisfies that one entry in the i-th row of $\mathbf{W}$ is $\mathbf{w}_e(i)$ and the rest ones are all zero, having the columns of $\mathbf{W}$ absolutely independent. But, the exclusivity achieved in this way is meaningless. According to Proposition 1, their equivalence is reached when each $\mathbf{w}_c$ is identical with $\mathbf{w}_e$. Based on the analysis above, we can say that, compared with (19), ExRM is more proper, which desires to find a balance between the diversity and consistency (training error) of base learners.

Next, we shall consider the convergence and optimality of the designed algorithms. We first confirm the property of Alg. 1, which is established by Theorem 1.

Theorem 1. *The updating rules* (10) *and* (11) *for solving the problem* (9)*, i.e. Algorithm 1, converges and the obtained optimal solution is exactly the global optimal solution of the problem* (8)*.*

Proof. Algorithm 1 is actually a special case of the algorithm proposed in [Kong *et al.*, 2014]. We refer readers to [Kong *et al.*, 2014] for the detailed proof. □

With Theorem 1, we have come to the convergence and optimality of our proposed Algorithm 2.

Theorem 2. *The solution consisting of the limit of the sequences $\{\mathbf{W}^{(t)}\}$, $\{\mathbf{b}^{(t)}\}$ and $\{\mathbf{E}^{(t)}\}$ generated by Algorithm 2, i.e. $(\mathbf{W}^{(\infty)}, \mathbf{b}^{(\infty)}, \mathbf{E}^{(\infty)})$, is global optimal to ExRM* (5)*, and the convergence rate is at least $o(\frac{1}{\mu^{(t)}})$.*

Proof. Due to space limit, instead of the complete proof, we here only provide the proof sketch including three main steps: 1) $(\mathbf{W}^{(t)}, \mathbf{b}^{(t)}, \mathbf{E}^{(t)})$ approaches to a feasible solution, *i.e.* $\lim_{t\to\infty}\|\mathbf{P}^{(t)} - \mathbf{W}^{(t)}\|_F = 0$ and $\lim_{t\to\infty}\|\mathbf{E}^{(t)} - \mathbf{Y} + \mathbf{X}^t\mathbf{P}^{(t)} + \mathbf{1}\mathbf{b}^{(t)T}\|_F = 0$; 2) All of

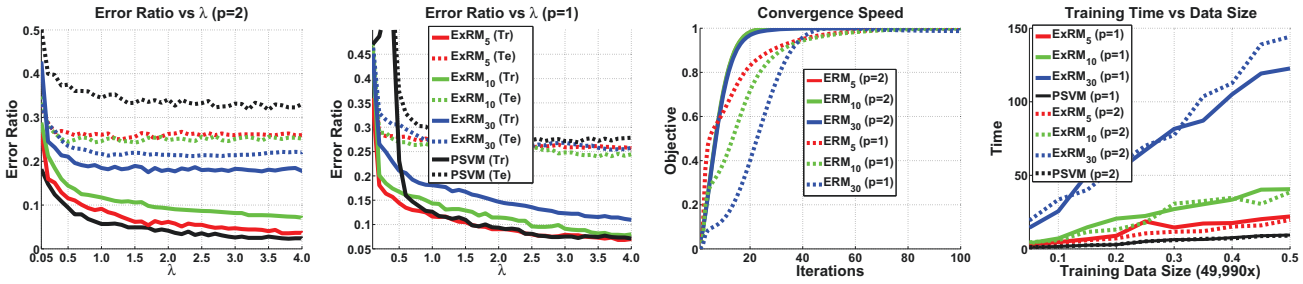

Figure 1: Parameter effect of λ, convergence speed and training time

$\{\mathbf{P}^{(t)}\}$, $\{\mathbf{Z}^{(t)}\}$, $\{\mathbf{Q}^{(t)}\}$, $\{\mathbf{W}^{(t)}\}$, $\{\mathbf{b}^{(t)}\}$ and $\{\mathbf{E}^{(t)}\}$ are bounded; 3) $\lim_{t\to\infty} \frac{1}{2}\|\mathbf{W}^{(t)}\|_{1,2}^2 + \lambda\|(\mathbf{Y} \circ \mathbf{E}^{(t)})_+\|_p^p = \lim_{t\to\infty} \mathcal{L}_{\mu^{(t-1)}}(\mathbf{W}^{(t)}, \mathbf{b}^{(t)}, \mathbf{E}^{(t)}, \mathbf{P}^{(t)}, \mathbf{Q}^{(t-1)}, \mathbf{Z}^{(t-1)})$ due to $\mathcal{L}_{\mu^{(t-1)}}(\mathbf{W}^{(t)}, \mathbf{b}^{(t)}, \mathbf{E}^{(t)}, \mathbf{P}^{(t)}, \mathbf{Q}^{(t-1)}, \mathbf{Z}^{(t-1)}) = \frac{1}{2}\|\mathbf{W}^{(t)}\|_{1,2}^2 + \lambda\|(\mathbf{Y} \circ \mathbf{E}^{(t)})_+\|_p^p + \frac{\|\mathbf{Q}^{(t)}\|_F^2 - \|\mathbf{Q}^{(t-1)}\|_F^2}{2\mu^{(t-1)}} + \frac{\|\mathbf{Z}^{(t)}\|_F^2 - \|\mathbf{Z}^{(t-1)}\|_F^2}{2\mu^{(t-1)}}$. Then the conclusion can be drawn with the feasibility of solution by Alg. 2, the convexity of problem (5), and the (descending) property of an ALM algorithm. □

In addition, we show the complexity of Alg. 2. Updating each row of $\mathbf{W}$ takes $\mathcal{O}(qC^2)$ and $\mathcal{O}(qC)$ for (11) and (10) respectively, where q is the (inner) iteration number of Alg. 1. Please note that, due to the diagonality of $\mathbf{G}$, the inverse of $\mathbf{G} + \mu\mathbf{I}$ only needs $\mathcal{O}(C)$. Therefore, the cost of Alg. 1 is $\mathcal{O}(qC^2M)$. The $\mathbf{b}$ sub-problem requires $\mathcal{O}(CMN)$. The complexity of the $\mathbf{E}$ sub-problem is $\mathcal{O}(CMN)$, for both $p := 1$ and $p := 2$. Solving $\mathbf{P}$ spends $\mathcal{O}(CMN + CM^2)$. Besides, the update of the multipliers is at $\mathcal{O}(CMN)$ expense. In summary, Alg. 2 has the complexity of $\mathcal{O}(tCM(qC + N + M))$, where t is the number of (outer) iterations required to converge.

4 Experimental Verification

We adopt several popular UCI benchmark datasets for performance evaluation.[3] All experiments are conducted on a machine with 2.5 GHz CPU and 64G RAM.

Parameter Effect Here, we evaluate the training and testing errors of ExRM$_C$ ($C \in \{5, 10, 30\}$ means the component number) against varying values of λ in the range $[0.05, 4]$. All the results shown in this experiment are averaged over 10 independent trials, each of which randomly samples half data from the *sonar* dataset for training and the other half for testing. The first picture in Fig. 1 displays the training error and testing error plots of L_2 loss ExRM with L_2 loss PSVM [Nie *et al.*, 2014] (denoted as L_2-PSVM) as reference. From the curves, we can observe that, as λ grows, the training errors drop, as well as composing less base learners leads to a smaller training error. This is because more and more effort is put on fitting data. As regards the testing error, the order is reversed, which corroborates the recognition that the predication gains from the diversity of classifiers, and reveals the effectiveness of our design in comparison with L_2-PSVM. Besides, the testing errors change very slightly in a relatively large range of λ, which implies the insensitivity of ExRM to λ. The second picture corresponding to $p := 1$ shows an additional evidence to $p := 2$. Although the performance gaps between the different cases shrink, the improvement of ExRM is still noticeable. Based on this evaluation, we set λ to 2 for both L_1-ExRM and L_2-ExRM in the rest experiments.

Convergence Speed & Training Time Although the convergence rate and complexity of the proposed algorithm have been theoretically provided, it would be more intuitive to see its empirical behavior. Thus, we here show how quick the algorithm converges, without loss of generality, on the *ijcnn1* dataset. From the third picture in Fig. 1, we can observe that, when $p := 2$, all the three cases converge with about 30 iterations. The cases correspond to $p := 1$ take more iterations than $p := 2$ (about 70 iterations), but they are still very efficient. Please note that, for a better view of different settings, the objective plots are normalized into the range $[0, 1]$. The rightmost picture in Fig. 1 gives curves of how the CPU-time used for training increases with respect to the number of training samples. Since the training time is too short to be accurately recorded, we carry out each test for 10 independent trials, and report the total training time (in seconds). As can be seen, the training time for both $p := 1$ and 2 is quasi linear with respect to the size of training data. For all the three cases that correspond to ExRM$_5$, ExRM$_{10}$ and ExRM$_{30}$, the choice of p barely brings differences in time. The gaps between the three cases dominantly come from the number of base learners. The PSVM only needs to learn one classifier while ExRM requires to train multiple bases.[4]

Performance Comparison This part first compares our proposed ExRM with the classic ensemble models including AdaBoost and Bagging with Tree as the base learner (denoted as AdaTree and BagTree, respectively), and the recently designed DRM. The codes of DRM are downloaded from the authors' website, while those of AdaTree and BagTree are integrated in the Matlab statistics toolbox (*fitensemble*). The base of DRM, ν-SVM, is from LibSVM.

Table 1 provides the quantitative comparison among the competitors. We report the mean testing errors over 10 independent trials. Since the sizes and distributions of the datasets vary, to avoid the effect brought by the amount of training da-

[3] Available at www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets

[4] In [Nie *et al.*, 2014], the authors have revealed via extensive experiments, that PSVM (SVM-ALM) is much more efficient than SVM^{perf} [Joachims, 2006], Pegasos [Shalev-Shwartz *et al.*, 2007], BMRM [Teo *et al.*, 2010], and TRON [Lin *et al.*, 2008], PCD [Chang *et al.*, 2008] and DCD [Hsieh *et al.*, 2008].

Table 1: Testing errors (mean ± standard deviation, %) on benchmark datasets

Method (R)	***german***	***diabetes***	***australian***	***sonar***	***splice***	***liver***	***heart***	***ionosphere***	**A.R.**
L_1-ExRM$_{10}$	26.08±1.21 (5)	24.73±1.44 (4)	14.59±0.84 (6)	23.62±3.64 (4)	26.53±1.66 (7)	42.82±2.41 (6)	17.17±1.25 (2)	13.68±2.60 (4)	4.8
L_2-ExRM$_{10}$	26.00±1.16 (3)	**24.34±0.92 (1)**	14.13±0.40 (3)	23.79±4.36 (5)	26.75±2.26 (8)	**36.00±4.05 (1)**	17.83±1.81 (4)	13.03±2.20 (2)	3.4
AdaTree$_{10}$	30.51±1.38 (8)	29.24±2.73 (7)	48.39±1.91 (9)	27.59±6.08 (10)	14.86±3.03 (3)	46.92±3.59 (10)	47.83±3.25 (9)	30.05±3.13 (8)	8.0
BagTree$_{10}$	31.59±2.59 (9)	30.63±3.38 (9)	46.43±2.58 (7)	25.00±8.26 (6)	19.26±1.78 (4)	46.00±3.58 (8)	47.58±1.94 (7)	38.71±3.02 (10)	7.5
DRM$_{10}$	25.98±1.20 (2)	24.47±0.89 (2)	14.09±1.55 (2)	27.07±3.05 (9)	35.96±2.85 (9)	36.77±3.79 (3)	19.00±2.22 (5)	19.50±3.67 (5)	4.6
L_1-ExRM$_{30}$	26.27±1.02 (6)	33.50±1.92 (10)	14.24±1.02 (4)	23.62±2.82 (3)	25.64±1.63 (5)	42.77±2.48 (5)	17.17±2.30 (3)	13.30±2.17 (3)	4.9
L_2-ExRM$_{30}$	**25.75±1.10 (1)**	25.42±1.41 (5)	**14.02±0.81 (1)**	**21.55±3.66 (1)**	26.07±1.80 (6)	40.05±3.48 (4)	**17.08±1.26 (1)**	**12.99±1.95 (1)**	2.5
AdaTree$_{30}$	32.22±2.32 (10)	29.85±2.77 (8)	48.94±2.13 (10)	25.17±7.63 (7)	14.45±2.53 (2)	46.26±3.97 (9)	47.58±3.27 (8)	32.34±3.25 (9)	7.9
BagTree$_{30}$	29.16±1.02 (7)	28.58±2.81(6)	46.59±1.39 (8)	21.72±4.24 (2)	**14.00±1.96 (1)**	45.85±2.23 (7)	48.08±4.21 (10)	27.86±2.27 (7)	6.0
DRM$_{30}$	26.05±1.14 (4)	24.47±0.89 (2)	14.43±1.65 (5)	26.55±3.91 (8)	35.98±2.84 (10)	36.67±4.03 (2)	19.00±2.51 (6)	19.70±3.48 (6)	5.4

Table 2: Average training time in seconds

L_1-ExRM$_{10}$	**L_2-ExRM$_{10}$**	**AdaTree$_{10}$**	**BagTree$_{10}$**	**DRM$_{10}$**	**L_1-ExRM$_{30}$**	**L_2-ExRM$_{30}$**	**AdaTree$_{30}$**	**BagTree$_{30}$**	**DRM$_{30}$**
0.0568	0.0575	0.1758	0.3019	16.86	0.1358	0.1091	0.2841	0.2512	59.38

ta and test the generalization ability of the ensembles learned from different types of data, we randomly sample 150 data points from a dataset as its training set and the rest as the testing. AdaTree and BagTree are inferior to ExRM and DRM in most cases. The exception is on the *splice* dataset. As for our ExRM, we can see that it significantly outperforms the others on *australian*, *sonar*, *heart* and *ionosphere*, and competes very favorably on the *german* dataset. On each dataset, we assign ranks to methods. The average ranks (A.R.) of the competitors over the involved datasets are given in the last column of Tab. 1. It can be observed that the top five average ranks are all lower than 5.0, and four of which are from ExRM methods. The best and the second best belong to L_2-ExRM$_{30}$ (A.R.=2.5) and L_2-ExRM$_{10}$ (A.R.=3.4) respectively, while the fourth and fifth places are taken by L_1-ExRM$_{10}$ (A.R.=4.8) and L_1-ExRM$_{30}$ (A.R.=4.9) respectively. The third goes to DRM$_{10}$, the average rank of which is 4.6. The results on the *ijcnn1* are not included in the table, as all the methods perform very closely to each other, which may lead to an unreliable rank.

Another issue should be concerned is the efficiency. Table 2 lists the mean training time over all the datasets and each dataset executes 10 runs. From the numbers, we can see the clear advantage of our ExRM. L_1-ExRM$_{10}$ and L_2-ExRM$_{10}$ only spend about $0.05s$ on training, while the ExRMs with 30 components, *i.e.* L_1-ExRM$_{30}$ and L_2-ExRM$_{30}$, cost less than $0.14s$. Both AdaTree and BagTree are sufficiently efficient, which take less than $0.3s$ to accomplish the task. But the training uses $16.86s$ and $59.38s$ by DRM for the 10-base and 30-base cases respectively. We would like to mention that the core of DRM is implemented in C++, while our ExRM is in pure Matlab. Moreover, as theoretically analyzed and empirically verified, our algorithm is linear with respect to the size of training set.

As aforementioned, employing SVM as the base of AdaBoost or Bagging is a way to construct an ensemble SVM. Here, we repeat the previous experiment to see the difference among AdaBoost *plus* L_2-PSVM (AdaSVM), Bagging *plus* L_2-PSVM (BagSVM), DRM and our L_2-ExRM. Please note that L_2-ExRM and DRM reduce to PSVM and ν-SVM respectively, when the base number is 1. Table 3 provides the average testing errors over all the datasets (each method runs 10 times). From the numbers, we can observe that both AdaSVM and BagSVM outperform DRM for all the cases, while ExRM shows the best results among the competitors.

Table 3: Average testing error comparison

# Base	**1**	**5**	**10**	**30**
L_1-ExRM	26.30 (L_1-PSVM)	25.65	25.24	25.09
L_2-ExRM	**25.68** (L_2-PSVM)	**24.80**	**24.41**	**24.85**
AdaSVM	–	26.51	26.59	26.64
BagSVM	–	26.34	25.93	25.69
DRM	28.31(ν-SVM)	27.85	27.70	27.89

5 Conclusion and Discussion

The diversity of component learners is critical to the ensemble performance. This paper has defined a new measurement of diversity, *i.e.* exclusivity. Incorporating the designed regularizer with the hinge loss function gives a birth to a novel model, namely ExRM. The convergence of the proposed ALM-based algorithm to a global optimal solution is theoretically guaranteed. ExRM can take into account more elaborate treatments for further improvement. For instance, thanks to the relationship $\|\mathbf{u}\|_1 = \mathcal{X}_r(\mathbf{u}, \mathbf{1})$, the sparsity on $\mathbf{W}$ can be promoted by extending $\tilde{\mathbf{W}}$ to $[\mathbf{W}, \beta\mathbf{1}]$, where β is a weight coefficient of the sparsity. In addition, it is difficult to directly solve the $\mathbf{E}$ sub-problem (13) with arbitrary given p. Fortunately, in this work, it is always that $p \geq 1$. Thus the partial derivative of (13) with respect to $\mathbf{E}$ is monotonically increasing. The binary search method can be employed to narrow the possible range of $\mathbf{E}$ by half via each operation. In this work, we did not take any advantages of Boosting and Bagging. It is positive that ExRM is able to act as the base learner for both Boosting and Bagging. By doing so, Boosting ExRM and Bagging ExRM can be viewed as ensembles of ensemble and expected to further boost the performance.

Acknowledgments

Xiaojie Guo is supported by National Natural Science Foundation of China (grant no. 61402467). Haibin Ling is supported by National Natural Science Foundation of China (grant no. 61528204) and National Science Foundation (grant no. 1350521).

References

[Altman, 1992] N. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[Breiman *et al.*, 1984] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.

[Breiman, 1996] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Breiman, 2001] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Chang *et al.*, 2008] K. Chang, C. Hsieh, and C. Lin. Coordinate descent method for large-scale L_2-loss linear svm. *Journal of Machine Learning Research*, 9:1369–1398, 2008.

[Dietterich, 2000] T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

[Domingos and Pazzani, 1997] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[Du and Ling, 2014] L. Du and H. Ling. Exploiting competition relationship for robust visual recognition. In *AAAI*, pages 2746–2752, 2014.

[Fisher, 1936] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[Freund and Schapire, 1997] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[Giacinto and Roli, 2001] G. Giacinto and F. Roli. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9-10):699–707, 2001.

[Ho, 1998] T. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[Hsieh *et al.*, 2008] C. Hsieh, K. Chang, S. Keerthi, S. Sundararajan, and C. Lin. A dual coordinate descent method for large-scale linear svm. In *ICML*, pages 408–415, 2008.

[Joachims, 2006] T. Joachims. Training linear svms in linear time. In *ACM SIGKDD*, pages 217–226, 2006.

[Keerthi and DeCoste, 2005] S. Keerthi and D. DeCoste. A modified finite newton method for fast solution of large scale linear svms. *Journal of Machine Learning Research*, 6:627–650, 2005.

[Kong *et al.*, 2014] D. Kong, R. Fujimaki, J. Liu, F. Nie, and C. Ding. Exclusive feature learning on arbitrary structures via $\ell_{1,2}$-norm. In *NIPS*, pages 1655–1663, 2014.

[Kowalski, 2009] M. Kowalski. Sparse regression using mixed norms. *Applied and Computational Harmonic Analysis*, 27(3):303–324, 2009.

[Krogh and Vedelsby, 1995] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In *NIPS*, pages 231–238, 1995.

[Kuncheva *et al.*, 2003] L. Kuncheva, C. Whitaker, C. Shipp, and R. Duin. Limits on the majority vote accuracy in classification fusion. *Pattern Analysis and Applications*, 6(1):22–31, 2003.

[Li *et al.*, 2012] N. Li, Y. Yu, and Z. Zhou. Diversity regularized ensemble pruning. In *ECML PKDD*, pages 330–345, 2012.

[Lin *et al.*, 2008] C. Lin, R. Weng, and S. Keerthi. Trust region newton method for large-scale logistic regression. *Journal of Machine Learning Research*, 9:627–650, 2008.

[Nie *et al.*, 2014] F. Nie, Y. Huang, X. Wang, and H. Huang. New primal svm solver with linear computational cost for big data classifications. In *ICML*, Beijing, China, 2014.

[Quinlan, 1993] J. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann Publisher, San Mateo, 1993.

[Rodriguez *et al.*, 2006] J. Rodriguez, L. Kubcheva, and C. Alonso. Rotation forest: A new classifier ensemble method. *TPAMI*, 28(10):1619–1630, 2006.

[Shalev-Shwartz *et al.*, 2007] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated subgradient solver for svm. In *ICML*, pages 807–814, 2007.

[Teo *et al.*, 2010] C. Teo, S. Vishwanathan, A. Smola, and Q. Le. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365, 2010.

[Ueda and Nakano, 1996] P. Ueda and R. Nakano. Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Network (ICNN)*, pages 90–95, 1996.

[Vapnik, 1995] V. Vapnik. *The Nature of Statistical Learning Theory*. Springe-Verlag, New York, NY, 1995.

[Xie *et al.*, 2015a] P. Xie, Y. Deng, and E. Xing. Latent variable modeling with diversity-inducing mutual angular regularization. *arXiv:1512.07336v1*, 2015.

[Xie *et al.*, 2015b] P. Xie, Y. Deng, and E. Xing. On the generalization error bounds of neural networks under diversity-inducing mutual angular regularization. *arXiv:1511.07110v1*, 2015.

[Yu *et al.*, 2011] Y. Yu, Y. Li, and Z. Zhou. Diversity regularized machine. In *IJCAI*, pages 1603–1608, 2011.

[Zhang *et al.*, 2010] Y. Zhang, R. Jin, and S. Hoi. Exclusive lasso for multi-task feature selection. In *AISTATS*, pages 988–995, 2010.

[Zhou, 2012] Z. Zhou. *Ensemble Methods: Foundations and Algorithms*. Taylor & Francis Group, LLC, Boca Raton, FL, 2012.