Mach Learn
DOI 10.1007/s10994-014-5465-9

Multilabel classification through random graph ensembles

Hongyu Su · Juho Rousu

Received: 16 January 2014 / Accepted: 14 July 2014

Abstract We present new methods for multilabel classification, relying on ensemble learning on a collection of random output graphs imposed on the multilabel, and a kernel-based structured output learner as the base classifier. For ensemble learning, differences among the output graphs provide the required base classifier diversity and lead to improved performance in the increasing size of the ensemble. We study different methods of forming the ensemble prediction, including majority voting and two methods that perform inferences over the graph structures before or after combining the base models into the ensemble. We put forward a theoretical explanation of the behaviour of multilabel ensembles in terms of the diversity and coherence of microlabel predictions, generalizing previous work on single target ensembles. We compare our methods on a set of heterogeneous multilabel benchmark problems against the state-of-the-art machine learning approaches, including multilabel AdaBoost, convex multitask feature learning, as well as single target learning approaches represented by Bagging and SVM. In our experiments, the random graph ensembles are very competitive and robust, ranking first or second on most of the datasets. Overall, our results show that our proposed random graph ensembles are viable alternatives to flat multilabel and multitask learners.

Keywords Multilabel classification · Structured output · Ensemble methods · Kernel methods · Graphical models

1 Introduction

Multilabel and multitask classification rely on representations and learning methods that allow us to leverage the dependencies between the different labels. When such dependencies

Editors: Cheng Soon Ong, Tu Bao Ho, Wray Buntine, Bob Williamson, and Masashi Sugiyama.

H. Su (✉) · J. Rousu
Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, Aalto University, Konemiehentie 2, 02150 Espoo, Finland
e-mail: hongyu.su@aalto.fi

J. Rousu
e-mail: juho.rousu@aalto.fi

Published online: 15 August 2014

are given in form of a graph structure such as a sequence, a hierarchy or a network, structured output prediction (Taskar et al. 2004; Tsochantaridis et al. 2004; Rousu et al. 2006) becomes a viable option, and has achieved a remarkable success. For multilabel classification, limiting the applicability of the structured output prediction methods is the very fact they require the predefined output structure to be at hand, or alternatively auxiliary data where the structure can be learned from. When these are not available, flat multilabel learners or collections of single target classifiers are thus often resorted to.

In this paper, we study a different approach, namely using ensembles of graph labeling classifiers, trained on randomly generated output graph structures. The methods are based on the idea that variation in the graph structures shifts the inductive bias of the base learners and causes diversity in the predicted multilabels. Each base learner, on the other hand, is trained to predict as good as possible multilabels, which makes them satisfy the weak learning assumption, necessary for successful ensemble learning.

Ensembles of multitask or multilabel classifiers have been proposed, but with important differences. The first group of methods, boosting type, rely on changing the weights of the training instances so that difficult to classify instances gradually receive more and more weights. The AdaBoost boosting framework has spawned multilabel variants (Schapire and Singer 2000; Esuli et al. 2008). In these methods the multilabel is considered essentially as a flat vector. The second group of methods, Bagging, are based on bootstrap sampling the training set several times and building the base classifiers from the bootstrap samples. Thirdly, randomization has been used as the means of achieving diversity by Yan et al. (2007) who select different random subsets of input features and examples to induce the base classifiers, and by Su and Rousu (2011) who use majority voting over random graphs in drug bioactivity prediction context. Here we extend the last approach to two other types of ensembles and a wider set of applications, with gain in prediction performances. A preliminary version of this article appeared as Su and Rousu (2013).

The remainder of the article is structured as follows. In Sect. 2 we present the structured output model used as the graph labeling base classifier. In Sect. 3 we present three aggregation strategies based on random graph labeling. In Sect. 4 we present empirical evaluation of the methods. In Sect. 5 we present concluding remarks.

2 Multilabel classification through graph labeling

We start by detailing the graph labeling classification methods that are subsequently used as the base classifier. We examine the following multilabel classification setting. We assume data from a domain $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is a set and $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_k$ is the set of multilabels, represented by a Cartesian product of the sets $\mathcal{Y}_j = \{1, \ldots, l_j\}$, $j = 1, \ldots, k$. In particular, setting $k = 1, l_1 = 2$ ($\mathcal{Y} = \{1, 2\}$) corresponds to binary classification problem. A vector $\mathbf{y} = (y_1, \ldots, y_k) \in \mathcal{Y}$ is called the *multilabel* and the components y_j are called the *microlabels*. We assume that a training set $\{(x_i, \mathbf{y}_i)\}_{i=1}^m \subset \mathcal{X} \times \mathcal{Y}$ has been given. A pair $(x_i, \mathbf{y})$ where x_i is a training pattern and $\mathbf{y} \in \mathcal{Y}$ is arbitrary, is called a *pseudo-example*, to denote the fact that the pair may or may not be generated by the distribution generating the training examples. The goal is to learn a model $F : \mathcal{X} \mapsto \mathcal{Y}$ so that the expected loss over predictions on future instances is minimized, where the loss function is chosen suitably for multilabel learning problems. By $\mathbf{1}_{\{\cdot\}}$ we denote the indicator function $\mathbf{1}_{\{A\}} = 1$, if A is true, $\mathbf{1}_{\{A\}} = 0$ otherwise.

Here, we consider solving multilabel classification with graph labeling classifiers that, in addition to the training set, assumes a graph $G = (V, E)$ with nodes $V = \{1, \ldots, k\}$ corresponding to microlabels and edges $E \subseteq V \times V$ denoting potential dependencies between the

microlabels. For any edge $e = (j, j') \in E$, we denote by $\mathbf{y}_e = (y_j, y_{j'})$ the edge label of e in multilabel $\mathbf{y}$, induced by concatenating the microlabels corresponding to end points of e, with corresponding domain of edge labels $\mathcal{Y}_e = \mathcal{Y}_j \times \mathcal{Y}_{j'}$. By $\mathbf{y}_{ie}$ we denote the label of the edge e in the i'th training example. Hence, for a fixed multilabel $\mathbf{y}$, we can compute corresponding node label y_j of node $j \in V$ and edge label $\mathbf{y}_e$ of edge $e \in E$. We also use separate notation for node and edge labels that are free, that is, not bound to any particular multilabel. We denote by u_j a possible label of node j, and $\mathbf{u}_e$ a possible labels of edge e. Naturally, $u_j \in \mathcal{Y}_j$ and $\mathbf{u}_e \in \mathcal{Y}_e$. See supplementary material for a comprehensive list of notations.

2.1 Graph labeling classifier

As the graph labeling classifier in this work we use max-margin structured output prediction, with the aim to learn a compatibility score for pairs $(x, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$, indicating how well an input goes together with an output. Naturally, such a score for coupling an input x with the correct multilabel $\mathbf{y}$ should be higher than the score of the same input with an incorrect multilabel $\mathbf{y}'$. The compatibility score between an input x and a multilabel $\mathbf{y}$ takes the form

$$\psi(x, \mathbf{y}) = \langle w, \varphi(x, \mathbf{y}) \rangle = \sum_{e \in E} \langle w_e, \varphi_e(x, \mathbf{y}_e) \rangle = \sum_{e \in E} \psi_e(x, \mathbf{y}_e), \tag{1}$$

where by $\langle \cdot, \cdot \rangle$ we denote the inner product and parameter w contains the feature weights to be learned. $\psi_e(x, \mathbf{y}_e)$ is a shorthand for the compatibility score, or the potential, between the input x and an edge label $\mathbf{y}_e$, defined as $\psi_e(x, \mathbf{y}_e) = \langle w_e, \varphi_e(x, \mathbf{y}_e) \rangle$, where w_e is the element of w that associates to edge e.

The joint feature map

$$\varphi(x, \mathbf{y}) = \phi(x) \otimes \Upsilon(\mathbf{y}) = \phi(x) \otimes (\Upsilon_e(\mathbf{y}_e))_{e \in E} = (\varphi_e(x, \mathbf{y}_e))_{e \in E}$$

is given by a tensor product of an input feature $\phi(x)$ and the feature space embedding of the multilabel $\Upsilon(\mathbf{y}) = (\Upsilon_e(\mathbf{y}_e))_{e \in E}$, consisting of edge label indicators $\Upsilon_e(\mathbf{y}_e) = \left(\mathbf{1}_{\{\mathbf{y}_e = \mathbf{u}_e\}}\right)_{\mathbf{u}_e \in \mathcal{Y}_e}$. The benefit of the tensor product representation is that context (edge label) sensitive weights can be learned for input features and no prior alignment of input and output features needs to be assumed.

The parameters w of the model are learned through max-margin optimization, where the primal optimization problem takes the form (e.g. Taskar et al. 2004; Tsochantaridis et al. 2004; Rousu et al. 2006)

$$\begin{aligned} &\min_{w, \xi} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{m} \xi_i \\ &\text{s.t. } \langle w, \varphi(x_i, \mathbf{y}_i) \rangle \geq \max_{\mathbf{y} \in \mathcal{Y}} \left(\langle w, \varphi(x_i, \mathbf{y}) \rangle + \ell(\mathbf{y}_i, \mathbf{y}) \right) - \xi_i, \\ &\quad \xi_i \geq 0, \ \forall i \in \{1, \ldots, m\}, \end{aligned} \tag{2}$$

where ξ_i denotes the slack allotted to each example, $\ell(\mathbf{y}_i, \mathbf{y})$ is the loss function between pseudo-label and correct label, and C is the slack parameter that controls the amount of regularization in the model. The primal form can be interpreted as maximizing the minimum margin between the correct training example and incorrect pseudo-examples, scaled by the loss function. The intuition behind loss-scaled margin is that example with nearly correct multilabel would require smaller margin than with multilabel that is quite different from the correct one. Denoting $\Delta\varphi(x_i, \mathbf{y}) = \varphi(x_i, \mathbf{y}_i) - \varphi(x_i, \mathbf{y})$, the Lagrangian of the problem is given by

$$L(w, \xi, \alpha, \rho) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i,\mathbf{y}}\alpha(i,\mathbf{y})\left(\langle w, \Delta\varphi(x_i,\mathbf{y})\rangle + \ell(\mathbf{y}_i,\mathbf{y})\right) - \sum_i \xi_i\rho_i,$$

where by setting derivatives to zero with respect to w we obtain

$$w = \sum_{i,\mathbf{y}}\alpha(i,\mathbf{y})\Delta\varphi(x_i,\mathbf{y}), \tag{3}$$

and the zero derivatives for ξ give the box constraint $\sum_y \alpha(i, y) \leq C$ for all i, while the dual variables ρ_i are canceled out. Maximization with respect to α's gives the dual optimization problem as

$$\begin{aligned} \max_{\alpha\geq 0}\ & \alpha^T\ell - \frac{1}{2}\alpha^T K\alpha \\ \text{s.t.}\ & \sum_{\mathbf{y}}\alpha(i,\mathbf{y}) \leq C, \forall i \in \{1,\ldots,m\}, \end{aligned} \tag{4}$$

where $\alpha = (\alpha(i,\mathbf{y}))_{i,\mathbf{y}}$ denotes the vector of dual variables and $\ell = (\ell(\mathbf{y}_i,\mathbf{y}))_{i,\mathbf{y}}$ is the vector of losses for each pseudo-example $(x_i, \mathbf{y})$. The joint kernel

$$\begin{aligned} K(x_i,\mathbf{y};x_j,\mathbf{y}') &= \langle\varphi(x_i,\mathbf{y}_i) - \varphi(x_i,\mathbf{y}), \varphi(x_j,\mathbf{y}_j) - \varphi(x_j,\mathbf{y}')\rangle \\ &= \langle\phi(x_i),\phi(x_j)\rangle_\phi \cdot \langle(\Upsilon(\mathbf{y}_i) - \Upsilon(\mathbf{y}), \Upsilon(\mathbf{y}_j) - \Upsilon(\mathbf{y}')\rangle_\Upsilon \\ &= K_\phi(x_i,x_j)\cdot\left(K_\Upsilon(\mathbf{y}_i,\mathbf{y}_j) - K_\Upsilon(\mathbf{y}_i,\mathbf{y}') - K_\Upsilon(\mathbf{y},\mathbf{y}_j) + K_\Upsilon(\mathbf{y},\mathbf{y}')\right) \\ &= K_\phi(x_i,x_j)\cdot K_{\Delta\Upsilon}(\mathbf{y}_i,\mathbf{y};\mathbf{y}_j,\mathbf{y}') \end{aligned}$$

is composed by product of input kernel $K_\phi(x_i, x_j) = \langle x_i, x_j\rangle_\phi$ and output kernel

$$K_{\Delta\Upsilon}(\mathbf{y}_i,\mathbf{y};\mathbf{y}_j,\mathbf{y}') = \left(K_\Upsilon(\mathbf{y}_i,\mathbf{y}_j) - K_\Upsilon(\mathbf{y}_i,\mathbf{y}') - K_\Upsilon(\mathbf{y},\mathbf{y}_j) + K_\Upsilon(\mathbf{y},\mathbf{y}')\right),$$

where $K_\Upsilon(\mathbf{y},\mathbf{y}') = \langle\Upsilon(\mathbf{y}'),\Upsilon(\mathbf{y})\rangle$.

2.2 Factorized dual form

The dual optimization problem (4) is a challenging one to solve due to the exponential-sized dual variable space, thus efficient algorithms are required. A tractable form is obtained via factorizing the problem according to the graph structure. Following Rousu et al. (2007), we transform (4) into the factorized dual form, where the edge-marginals of dual variables are used in place of the original dual variables

$$\mu(i,e,\mathbf{u}_e) = \sum_{\mathbf{y}\in\mathcal{Y}}\mathbf{1}_{\{\Upsilon_e(\mathbf{y})=\mathbf{u}_e\}}\alpha(i,\mathbf{y}), \tag{5}$$

where $e = (j, j') \in E$ is an edge in the output graph and $\mathbf{u}_e \in \mathcal{Y}_j \times \mathcal{Y}_{j'}$ is a possible label for the edge (j, j').

The output kernel decomposes by the edges of the graph as

$$K_\Upsilon(\mathbf{y},\mathbf{y}') = \langle\Upsilon(\mathbf{y}'),\Upsilon(\mathbf{y})\rangle = \sum_e K_{\Upsilon,e}(\mathbf{y}_e,\mathbf{y}'_e),$$

where $K_{\Upsilon,e}(u,u') = \langle \Upsilon_e(u), \Upsilon_e(u') \rangle_\Upsilon$. Given the joint features defined by the tensor product, the joint kernel also decomposes as

$$\begin{aligned} K(x_i, \mathbf{y}; x_j, \mathbf{y}') &= K_\phi(x, x') K_{\Delta\Upsilon}(\mathbf{y}_i, \mathbf{y}; \mathbf{y}_j, \mathbf{y}') = \\ &= \sum_e K_\phi(x, x') K_{\Delta\Upsilon,e}(\mathbf{y}_e, \mathbf{y}'_e) = \sum_e K_e(x, \mathbf{y}_e; x', \mathbf{y}'_e), \end{aligned}$$

where we have denoted

$$\begin{aligned} &K_{\Delta\Upsilon,e}(\mathbf{y}_{ie}, \mathbf{y}_e; \mathbf{y}_{je}, \mathbf{y}'_e) = \\ &\qquad \left(K_{\Upsilon,e}(\mathbf{y}_{ie}, \mathbf{y}_{je}) - K_{\Upsilon,e}(\mathbf{y}_{ie}, \mathbf{y}'_e) - K_{\Upsilon,e}(\mathbf{y}_e, \mathbf{y}_{je}) + K_{\Upsilon,e}(\mathbf{y}_e, \mathbf{y}'_e) \right). \end{aligned}$$

Using the above, the quadratic part of the objective factorizes as follows

$$\begin{aligned} \alpha^T K \alpha &= \sum_e \sum_{i,j} \sum_{\mathbf{y},\mathbf{y}'} \alpha(i,\mathbf{y}) K_e(x_i, \mathbf{y}_e; x_j, \mathbf{y}'_e) \alpha(j,\mathbf{y}') \\ &= \sum_e \sum_{i,j} \sum_{\mathbf{u},\mathbf{u}'} K_e(x_i, \mathbf{u}; x_j, \mathbf{u}') \sum_{\mathbf{y}:\mathbf{y}_e=\mathbf{u}} \sum_{\mathbf{y}':\mathbf{y}'_e=\mathbf{u}'} \alpha(i,\mathbf{y}) \alpha(j,\mathbf{y}') \\ &= \sum_e \sum_{i,j} \sum_{\mathbf{u},\mathbf{u}'} \mu(i,e,\mathbf{u}) K_e(x_i, \mathbf{u}; x_j, \mathbf{u}') \mu(j,e,\mathbf{u}') \\ &= \mu^T K_E \mu, \end{aligned} \tag{6}$$

where $K_E = \operatorname{diag}(K_e, e \in E)$ is a block diagonal matrix with edge-specific kernel blocks K_e and $\mu = (\mu(i,e,u))_{i,e,u}$ is the vector of marginal dual variables. We assume a loss function that can be expressed in a decomposed form as

$$\ell(\mathbf{y}, \mathbf{y}') = \sum_e \ell_e(\mathbf{y}_e, \mathbf{y}'_e),$$

a property that is satisfied by the Hamming loss family, counting incorrectly predicted nodes (i.e. microlabel loss) or edges, and are thus suitable for our purpose. With a decomposable loss function, the linear part of the objective satisfies

$$\begin{aligned} \sum_{i=1}^m \sum_{\mathbf{y}\in\mathcal{Y}} \alpha(i,\mathbf{y}) \ell(\mathbf{y}_i, \mathbf{y}) &= \sum_{i=1}^m \sum_{\mathbf{y}} \alpha(i,\mathbf{y}) \sum_e \ell_e(\mathbf{y}_{ie}, \mathbf{y}_e) = \\ &= \sum_{i=1}^m \sum_{e\in E} \sum_{u\in\mathcal{Y}_e} \sum_{\mathbf{y}:\mathbf{y}_e=u} \alpha(i,\mathbf{y}) \ell_e(\mathbf{y}_{ie}, u) \\ &= \sum_{i=1}^m \sum_{e\in E} \sum_{u\in\mathcal{Y}_e} \mu(e,u) \ell_e(\mathbf{y}_{ie}, u) = \sum_{i=1}^m \mu_i^T \ell_i = \mu^T \ell_E, \end{aligned} \tag{7}$$

where $\ell_E = (\ell_i)_{i=1}^m = (\ell_e(i,u))_{i=1,e\in E,u\in\mathcal{Y}_e}^m$ is the vector of losses. Given the above, we can state the dual problem (4) in equivalent form (c.f. Taskar et al. 2004; Rousu et al. 2007) as

$$\max_{\mu\in\mathcal{M}} \boldsymbol{\mu}^T \ell - \frac{1}{2} \boldsymbol{\mu}^T K_E \boldsymbol{\mu}, \tag{8}$$

where the constraint set is the marginal polytope (c.f. Rousu et al. 2007; Wainwright et al. 2005)

$$\mathcal{M} = \{\mu | \exists \alpha \in \mathcal{A} \text{ s.t. } \mu(i, e, \mathbf{u}_e) = \sum_{\mathbf{y} \in \mathcal{Y}} \mathbf{1}_{\{\mathbf{y}_{ie} = \mathbf{u}_e\}} \alpha(i, \mathbf{y}), \forall i, \mathbf{u}_e, e\}$$

of the dual variables, the set of all combinations of marginal dual variables (5) of training examples that correspond to some α in the original dual feasible set $\mathcal{A} = \{\alpha | \alpha(i, \mathbf{y}) \geq 0, \sum_{\mathbf{y}} \alpha(i, \mathbf{y}) \leq C, \forall i\}$ in (4). Note that the above definition of $\mathcal{M}$ automatically takes care of the consistency constraints between marginal dual variables (c.f. Rousu et al. 2007).

The factorized dual problem (8) is a quadratic program with a number of variables *linear* in both the size of the output network and the number of training examples. There is an exponential reduction in the number of dual variables from the original dual (4), however, with the penalty of more complex feasible polytope $\mathcal{M}$. For solving (8) we use MMCRF (Rousu et al. 2007) that relies on a conditional gradient method. Update directions are found in linear time via probabilistic inference (explained in the next section), making use of the the exact correspondence of maximum margin violating multilabel in the primal (2) and steepest feasible gradient of the dual objective (4).

2.3 Inference

In both training and prediction, efficient inference is required over the multilabel spaces. In training any of the models (2, 4, 8), one needs to iteratively solve the *loss-augmented inference* problem

$$\begin{aligned} \bar{\mathbf{y}}(x_i) &= \operatorname*{\mathbf{argmax}}_{\mathbf{y} \in \mathcal{Y}} \left(\langle w, \varphi(x_i, \mathbf{y}) \rangle + \ell(\mathbf{y}_i, \mathbf{y}) \right) \\ &= \operatorname*{\mathbf{argmax}}_{\mathbf{y} \in \mathcal{Y}} \sum_e \langle w_e, \varphi_e(x_i, \mathbf{y}_e) \rangle + \ell_e(\mathbf{y}_e, \mathbf{y}_{ie}) \end{aligned} \tag{9}$$

that finds for each example the multilabel that violates its margins the most (i.e. the worst margin violator) given the current w. Depending on the optimization algorithm, the worst-margin violator may be used to grow a constraint set (column-generation methods), or to define an update direction (structured perceptron, conditional gradient).

In the prediction phase, the inference problem to be solved is simply to find the highest scoring multilabel for each example:

$$\hat{\mathbf{y}}(x) = \operatorname*{\mathbf{argmax}}_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}, \varphi(x, \mathbf{y}) \rangle = \operatorname*{\mathbf{argmax}}_{\mathbf{y} \in \mathcal{Y}} \sum_e \langle w_e, \varphi_e(x, \mathbf{y}) \rangle \tag{10}$$

Both of the above inference problems can be solved in the factorized dual, thus allowing us to take advantage of kernels for complex and high-dimensional inputs, as well as the linear-size dual variable space.

Next, we put forward a lemma that shows explicitly how the compatibility score $\psi_e(x, \mathbf{y}_e) = \langle w_e, \varphi_e(x, \mathbf{y}_e) \rangle$ of labeling an edge e as $\mathbf{y}_e$ given input x can be expressed in terms of kernels and marginal dual variables. We note that the property is already used in marginal dual based structured output methods such as MMCRF, however, below we make the property explicit, to facilitate the description of the ensemble learning methods.

Lemma 1 *Let w be the solution to (2), $\varphi(x, \mathbf{y})$ be the joint feature map, and let $G = (V, E)$ be the graph defining the output graph structure, and let us denote*

$$H_e(x_i, \mathbf{u}_e; x, \mathbf{y}_e) = K_\phi(x, x_i) \cdot \left(K_{\Upsilon,e}(\mathbf{y}_{ie}, \mathbf{y}_e) - K_{\Upsilon,e}(\mathbf{u}_e, \mathbf{y}_e) \right).$$

Then, we have

$$\psi_e(x, \mathbf{y}_e) = \langle w_e, \varphi_e(x, \mathbf{y}_e)\rangle = \sum_{i,\mathbf{u}_e} \mu(i, e, \mathbf{u}_e) \cdot H_e(x_i, \mathbf{u}_e; x, \mathbf{y}_e),$$

where μ is the marginal dual variable learned by solving optimization problem (8).

Proof Using (3) and (5), and elementary tensor algebra, the compatibility score of a example $(x, \mathbf{y}')$ is given by

$$\begin{aligned}
\langle w, \varphi(x, \mathbf{y}')\rangle &= \sum_i \sum_{\mathbf{y}} \alpha(i, \mathbf{y}) \langle \Delta\varphi(x_i, \mathbf{y}), \varphi(x, \mathbf{y}')\rangle \\
&= \sum_i \sum_e \sum_{\mathbf{u}_e} \sum_{\mathbf{y}:\mathbf{y}_e=\mathbf{u}_e} \alpha(i, \mathbf{y}) \langle \Delta\varphi_e(x_i, \mathbf{u}_e), \varphi_e(x, \mathbf{y}'_e)\rangle \\
&= \sum_e \sum_i \sum_{\mathbf{u}_e} \mu(i, e, \mathbf{u}_e) \langle \phi(x_i) \otimes (\Upsilon_e(\mathbf{y}_{ie}) - \Upsilon_e(\mathbf{u}_e)), \phi(x) \otimes \Upsilon_e(\mathbf{y}'_e)\rangle \\
&= \sum_e \sum_i \sum_{\mathbf{u}_e} \mu(i, e, \mathbf{u}_e) K_\phi(x_i, x) \langle \Upsilon_e(\mathbf{y}_{ie}) - \Upsilon_e(\mathbf{u}_e), \Upsilon_e(\mathbf{y}'_e)\rangle \\
&= \sum_e \sum_i \sum_{\mathbf{u}_e} \mu(i, e, \mathbf{u}_e) K_\phi(x_i, x) \cdot \left(K_{\Upsilon,e}(\mathbf{y}_{ie}, \mathbf{y}'_e) - K_{\Upsilon,e}(\mathbf{u}_e, \mathbf{y}'_e)\right) \\
&= \sum_e \sum_{i,\mathbf{u}_e} \mu(i, e, \mathbf{u}_e) \cdot H_e(x_i, \mathbf{u}_e; x, \mathbf{y}'_e).
\end{aligned}$$

The loss-augmented inference problem can thus be equivalently expressed in the factorized dual by

$$\begin{aligned}
\bar{\mathbf{y}}(x) &= \mathbf{argmax}_{\mathbf{y}\in\mathcal{Y}} \sum_e \psi_e(x, \mathbf{y}_e) + \ell_e(\mathbf{y}_e, \mathbf{y}_{ie}) \\
&= \mathbf{argmax}_{\mathbf{y}\in\mathcal{Y}} \sum_{e,i,\mathbf{u}_e} \mu(i, e, \mathbf{u}_e) H_e(i, \mathbf{u}_e; x, \mathbf{y}_e) + \ell_e(\mathbf{y}_e, \mathbf{y}_{ie}).
\end{aligned} \tag{11}$$

Similarly, the inference problem (10) solved in prediction phase can be solved in the factorized dual by

$$\begin{aligned}
\hat{\mathbf{y}}(x) &= \mathbf{argmax}_{\mathbf{y}\in\mathcal{Y}} \sum_e \psi_e(x, \mathbf{y}_e) = \mathbf{argmax}_{\mathbf{y}\in\mathcal{Y}} \sum_e \langle \mathbf{w}_e, \varphi_e(x, \mathbf{y}_e)\rangle \\
&= \mathbf{argmax}_{\mathbf{y}\in\mathcal{Y}} \sum_{e,i,\mathbf{u}_e} \mu(i, e, \mathbf{u}_e) H_e(i, \mathbf{u}_e; x, \mathbf{y}_e).
\end{aligned} \tag{12}$$

To solve (11) or (12) any commonly used inference technique for graphical models can be applied. In this paper we use MMCRF that relies on the message-passing method, also referred as loopy belief propagation (LBP). We use early stopping in inference of LBP, so that the number of iterations is limited by the diameter of the output graph G.

3 Learning graph labeling ensembles

In this section we consider generating ensembles of multilabel classifiers, where each base model is a graph labeling classifier. Algorithm 1 depicts the general form of the learning approach. We assume a function to output a random graph $G^{(t)}$ for each stage of the ensemble,

a base learner $F^{(t)}(\cdot)$ to learn the graph labeling model, and an aggregation function $A(\cdot)$ to compose the ensemble model. The prediction of the model is then obtained by aggregating the base model predictions

$$F(x) = A(F^{(1)}(x), \ldots, F^{(T)}(x)).$$

Given a set of base models trained on different graph structures we expect the predicted labels of the ensemble have diversity which is known to be necessary for ensemble learning. At the same time, since the graph labeling classifiers aim to learn accurate multilabels, we expect the individual base classifiers to be reasonably accurate, irrespective of the slight changes in the underlying graphs. Indeed, in this work we use randomly generated graphs to emphasize this point. We consider the following three aggregation methods:

– In *majority-voting-ensemble*, each base learner gives a prediction of the multilabel. The ensemble prediction is obtained by taking the most frequent value for each microlabel. Majority voting aggregation is admissible for any multilabel classifier.

Second, we consider two aggregation strategies that assume the base classifier has a conditional random field structure:

– In *average-of-maximum-marginals aggregation*, each base learner infers local maximum marginal scores for each microlabel. The ensemble prediction is taken as the value with highest average local score.
– In *maximum-of-average-marginals aggregation*, the local edge potentials of each base model are first averaged over the ensemble and maximum global marginal scores are inferred from the averages.

In the following, we detail the above aggregation strategies.

3.1 Majority voting ensemble (MVE)

The first ensemble model we consider is the majority voting ensemble (MVE), which was introduced in drug bioactivity prediction context by Su and Rousu (2011). In MVE, the ensemble prediction on each microlabel is the most frequently appearing prediction among the base classifiers

$$F_j^{\text{MVE}}(x) = \mathop{\mathbf{argmax}}_{y_j \in \mathcal{Y}_j} \left(\frac{1}{T} \sum_{i=1}^{T} \mathbf{1}_{\{F_j^{(t)}(x) = y_j\}} \right),$$

where $F^{(t)}(x) = (F_j^{(t)}(x))_{j=1}^{k}$ is the predicted multilabel in the t'th base classifier. When using (8) as the base classifier, predictions $F^{(t)}(x)$ are obtained via solving the inference problem (12). We note, however, in principle, any multilabel learner will fit into the MVE

Input: Training sample $S = \{(x_i, \mathbf{y}_i)\}_{i=1}^{m}$, ensemble size T, graph generating oracle function $outputGraph : t \in \{1, \ldots, T\} \mapsto \mathcal{G}_k$, aggregation function $A(\cdot) : \mathcal{F} \times \cdots \times \mathcal{F} \mapsto \mathcal{Y}$
Output: Multilabel classification ensemble $F(\cdot) : \mathcal{X} \mapsto \mathcal{Y}$
1: **for** $t \in \{1, \ldots, T\}$ **do**
2: $G^{(t)} = \text{outputGraph}(t)$
3: $F^t(\cdot) = \text{learnGraphLabelingClassifier}((x_i)_{i=1}^{m}, (\mathbf{y}_i)_{i=1}^{m}, G^{(t)})$
4: **end for**
5: $F(\cdot) = A(F^{(1)}(\cdot), \ldots, F^{(T)}(\cdot))$

Algorithm 1: Graph Labeling Ensemble Learning

framework as long as it adapts to a collection of output graphs $\mathcal{G} = \{G^{(1)}, \cdots, G^{(T)}\}$ and generates multilabel predictions accordingly from each graph.

3.2 Average-of-max-marginal aggregation (AMM)

Next, we consider an ensemble model where we perform inference over the graph to extract information on the learned compatibility scores in each base model. Thus, we assume that we have access to the compatibility scores between the inputs and edge labels

$$\Psi_E^{(t)}(x) = (\psi_e^{(t)}(x, \mathbf{u}_e))_{e \in E^{(t)}, \mathbf{u}_e \in \mathcal{Y}_e}.$$

In the AMM model, our goal is to infer for each microlabel u_j of each node j its *max-marginal* (Wainwright et al. 2005), that is, the maximum score of a multilabel that is consistent with $y_j = u_j$

$$\tilde{\psi}_j(x, u_j) = \max_{\{\mathbf{y} \in \mathcal{Y} : y_j = u_j\}} \sum_e \psi_e(x, \mathbf{y}_e). \tag{13}$$

One readily sees (13) as a variant of the inference problem (12), with similar solution techniques. The maximization operation fixes the label of the node $y_j = u_j$ and queries the optimal configuration for the remaining part of output graph. In message-passing algorithms, only slight modification is needed to make sure that only the messages consistent with the microlabel restriction are considered. To obtain the vector $\tilde{\Psi}(x) = (\tilde{\psi}_j(x, u_j))_{j,u_j}$ the same inference is repeated for each target-microlabel pair (j, u_j), hence it has quadratic time complexity in the number of edges in the output graph.

Given the max-marginals of the base models, the AMM ensemble is constructed as follows. Let $\mathcal{G} = \{G^{(1)}, \cdots, G^{(T)}\}$ be a set of output graphs, and let $\{\tilde{\Psi}^{(1)}(x), \cdots, \tilde{\Psi}^{(T)}(x)\}$ be the max-marginal vectors of the base classifiers trained on the output graphs. The ensemble prediction for each target is obtained by averaging the max-marginals of the base models and choosing the maximizing microlabel for the node:

$$F_j^{\text{AMM}}(x) = \operatorname*{argmax}_{u_j \in \mathcal{Y}_j} \frac{1}{|T|} \sum_{t=1}^{T} \tilde{\psi}_{j,u_j}^{(t)}(x),$$

and the predicted multilabel is composed from the predicted microlabels

$$F^{\text{AMM}}(x) = \left(F_j^{\text{AMM}}(x)\right)_{j \in V}.$$

An illustration of AMM ensemble scheme is shown in Fig. 1. Edge information on individual base learner are not preserved during AMM ensemble, which is shown as dash line in Fig. 1. In principle, AMM ensemble can give different predictions compared to MVE, since the most frequent label may not be the ensemble prediction if it has lower average max-marginal score.

3.3 Maximum-of-average-marginals aggregation (MAM)

The next model, the maximum-of-average-marginals (MAM) ensemble, first collects the local compatibility scores $\Psi_E^{(t)}(x)$ from individual base learners, averages them and finally performs inference on the global consensus graph with averaged edge potentials. The model is defined as

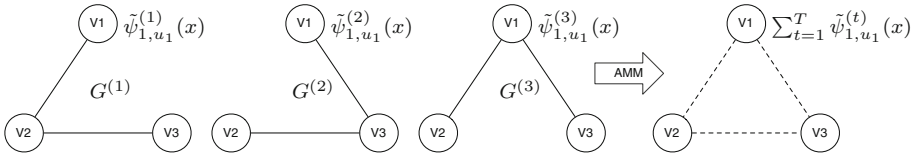

Fig. 1 An example of AMM scheme, where three base models are learned on the output graph $G^{(1)}, G^{(2)}, G^{(3)}$. Given an example x, each base model computes for node v_1 local max-marginals $\tilde{\psi}^{(t)}_{1,u_1}$ for all $u_1 \in \{+,-\}$. The AMM collects local max-marginals $\sum_{t=1}^{T}\tilde{\psi}^{(t)}_{1,u_1}(x)$, and outputs $F_1(x) = +$ if $\sum_{t=1}^{T}\tilde{\psi}^{(t)}_{1,+}(x) \geq \sum_{t=1}^{T}\tilde{\psi}^{(t)}_{1,-}(x)$, otherwise outputs $F_1(x) = -$

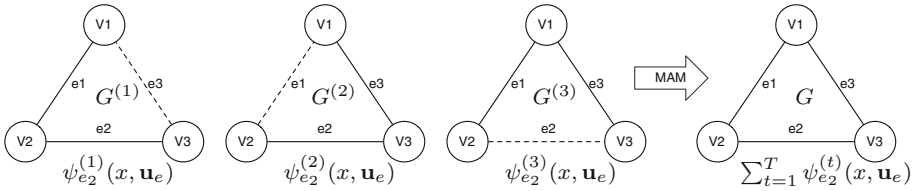

Fig. 2 An example of MAM scheme, where three base models are learned on the output graph $G^{(1)}, G^{(2)}, G^{(3)}$. Given an example x, each base model computes for edge e_2 local edge potentials $\psi^{(t)}_{e_2}(x,\mathbf{u}_e)$ for all $\mathbf{u}_e = \{--,-+,+-,++\}$. For graph $G^{(3)}$ where $e_2 \notin E^{(3)}$, we first impute corresponding marginal dual variable of e_2 on $G^{(3)}$according to local consistency constraints. Similar computations are required for edge e_1 and e_3. The finial prediction is through inference over averaged edge potentials on consensus graph G

$$F^{\text{MAM}}(x) = \operatorname*{\mathbf{argmax}}_{\mathbf{y}\in\mathcal{Y}} \sum_{e\in E}\frac{1}{T}\sum_{t=1}^{T}\psi^{(t)}_e(x,\mathbf{y}_e) = \operatorname*{\mathbf{argmax}}_{\mathbf{y}\in\mathcal{Y}} \frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E}\langle \mathbf{w}^{(t)}_e, \varphi_e(x,\mathbf{y}_e)\rangle.$$

With the factorized dual representation, this ensemble scheme can be implemented simply and efficiently in terms of marginal dual variables and the associated kernels, which saves us from explicitly computing the local compatibility scores from each base learner. Using Lemma (1), the above can be equivalently expressed as

$$\begin{aligned} F^{\text{MAM}}(x) &= \operatorname*{\mathbf{argmax}}_{\mathbf{y}\in\mathcal{Y}} \frac{1}{T}\sum_{t=1}^{T}\sum_{i,e,\mathbf{u}_e}\mu^{(t)}(i,e,\mathbf{u}_e)\cdot H_e(i,\mathbf{u}_e;x,\mathbf{y}_e) \\ &= \operatorname*{\mathbf{argmax}}_{\mathbf{y}\in\mathcal{Y}} \sum_{i,e,\mathbf{u}_e}\bar{\mu}(i,e,\mathbf{u}_e)H_e(i,\mathbf{u}_e;x,\mathbf{y}_e), \end{aligned}$$

where we denote by $\bar{\mu}(i,e,\mathbf{u}_e) = \frac{1}{T}\sum_{t=1}^{T}\mu^{(t)}(i,e,\mathbf{u}_e)$ the marginal dual variable averaged over the ensemble.

We note that $\mu^{(t)}$ is originally defined on edge set $E^{(t)}$, $\mu^{(t)}$ from different random output graph are not mutually consistent. In practice, we first construct a consensus graph $G = (E,V)$ by pooling edge sets $E^{(t)}$, then complete $\mu^{(t)}$ on E where missing components are imputed via exploring local consistency conditions and solving constrained least square problem. Thus, the ensemble prediction can be computed in marginal dual form without explicit access to input features, and the only input needed from the different base models are the values of the marginal dual variables. An example that illustrates the MAM ensemble scheme is shown in Fig. 2.

3.4 The MAM ensemble analysis

Here, we present theoretical analysis of the improvement of the MAM ensemble over the mean of the base classifiers. The analysis follows the spirit of the single-label ensemble analysis by Brown and Kuncheva (2010), generalizing it to the multilabel MAM ensemble.

Assume there is a collection of T individual base learners, indexed by $t \in \{1, \cdots, T\}$, that output compatibility scores $\psi_e^{(t)}(x, \mathbf{u}_e)$ for all $t \in \{1, \ldots, T\}$, $e \in E^{(t)}$, and $\mathbf{u}_e \in \mathcal{Y}_e$. For the purposes of this analysis, we express the compatibility scores in terms of the nodes (microlabels) instead of the edges and their labels. We denote by

$$\psi_j(x, y_j) = \sum_{\substack{e=(j,j'),\\ e \in N(j)}} \mathbf{1}_{\{y_j = u_j\}} \frac{1}{2} \psi_e(x, \mathbf{u}_e)$$

the sum of compatibility scores of the set of edges $N(j)$ incident to node j with consistent label $\mathbf{y}_e = (y_j, y_{j'})$, $y_j = u_j$. Then, the compatibility score for the input and the multilabel in (1) can be alternatively expressed as

$$\psi(x, \mathbf{y}) = \sum_{e \in E} \psi_e(x, \mathbf{y}_e) = \sum_{j \in V} \psi_j(x, y_j).$$

The compatibility score from MAM ensemble can be similarly represented in terms of the nodes by

$$\psi^{\text{MAM}}(x, \mathbf{y}) = \frac{1}{T} \sum_t \psi^{(t)}(x, \mathbf{y}) = \sum_{e \in E} \bar{\psi}_e(x, \mathbf{y}_e) = \sum_{j \in V} \bar{\psi}_j(x, y_j),$$

where we have denoted $\bar{\psi}_j(x, y_j) = \frac{1}{T} \sum_t \psi_j^{(t)}(x, y_j)$ and $\bar{\psi}_e(x, \mathbf{y}_e) = \frac{1}{T} \sum_t \psi_e^{(t)}(x, \mathbf{y}_e)$.

Assume now the ground truth, the optimal compatibility score of an example and multilabel pair $(x, \mathbf{y})$, is given by $\psi^*(x, \mathbf{y}) = \sum_{j \in V} \psi_j^*(x, y_j)$. We study the reconstruction error of the compatibility score distribution, given by the squared distance of the estimated score distributions from the ensemble and the ground truth. The reconstruction error of the MAM ensemble can be expressed as

$$\Delta^R_{\text{MAM}}(x, \mathbf{y}) = \left(\psi^*(x, \mathbf{y}) - \psi^{\text{MAM}}(x, \mathbf{y})\right)^2,$$

and the average reconstruction error of the base learners can be expressed as

$$\Delta^R_I(x, \mathbf{y}) = \frac{1}{T} \sum_t \left(\psi^*(x, \mathbf{y}) - \psi^{(t)}(x, \mathbf{y})\right)^2.$$

We denote by $\Psi_j(x, y_j)$ a random variable of the compatibility scores obtained by the base learners and $\{\psi_j^{(1)}(x, y_j), \cdots, \psi_j^{(T)}(x, y_j)\}$ as a sample from its distribution. We have the following result:

Theorem 1 *The reconstruction error of compatibility score distribution given by MAM ensemble $\Delta^R_{MAM}(x, \mathbf{y})$ is guaranteed to be no greater than the average reconstruction error given by individual base learners $\Delta^R_I(x, \mathbf{y})$.*

In addition, the gap can be estimated as

$$\Delta^R_I(x, \mathbf{y}) - \Delta^R_{MAM}(x, \mathbf{y}) = \boldsymbol{Var}\left(\sum_{j \in V} \Psi_j(x, y_j)\right) \geq 0.$$

The variance can be further expanded as

$$\boldsymbol{Var}\left(\sum_{j\in V}\Psi_j(x,y_j)\right)=\underbrace{\sum_{j\in V}\boldsymbol{Var}(\Psi_j(x,y_j))}_{diversity}+\underbrace{\sum_{\substack{p,q\in V,\\ p\neq q}}\boldsymbol{Cov}(\Psi_p(x,y_p),\Psi_q(x,y_q))}_{coherence}.$$

Proof By expanding and simplifying the squares we get

$$\begin{aligned}\Delta_I^R(x,\mathbf{y})-\Delta_{\text{MAM}}^R(x,\mathbf{y}) &= \frac{1}{T}\sum_t\left(\psi^*(x,\mathbf{y})-\psi^{(t)}(x,\mathbf{y})\right)^2-\left(\psi^*(x,\mathbf{y})-\psi^{\text{MAM}}(x,\mathbf{y})\right)^2\\ &= \frac{1}{T}\sum_t\left(\sum_{j\in V}\psi_j^*(x,y_j)-\sum_{j\in V}\psi_j^{(t)}(x,y_j)\right)^2\\ &\quad-\left(\sum_{j\in V}\psi_j^*(x,y_j)-\sum_{j\in V}\frac{1}{T}\sum_t\psi_j^{(t)}(x,y_j)\right)^2\\ &= \frac{1}{T}\sum_t\left(\sum_{j\in V}\psi_j^{(t)}(x,y_j)\right)^2-\left(\frac{1}{T}\sum_t\sum_{j\in V}\psi_j^{(t)}(x,y_j)\right)^2\\ &= \boldsymbol{Var}(\sum_{j\in V}\Psi_j(x,y_j))\\ &\geq 0.\end{aligned}$$

The expression of variance can be further expanded as

$$\begin{aligned}\boldsymbol{Var}\left(\sum_{j\in V}\Psi_j(x,y_j)\right) &= \sum_{p,q\in V}\boldsymbol{Cov}(\Psi_p(x,y_p),\Psi_q(x,y_q))\\ &= \sum_{j\in V}\boldsymbol{Var}(\Psi_j(x,y_j))+\sum_{\substack{p,q\in V,\\ p\neq q}}\boldsymbol{Cov}(\Psi_p(x,y_p),\Psi_q(x,y_q)).\end{aligned}$$

The Theorem 1 states that the reconstruction error from MAM ensemble is guaranteed to be less than or equal to the average reconstruction error from the individuals. In particular, the improvement can be further addressed by two terms, namely *diversity* and *coherence*. The classifier diversity measures the variance of predictions from base learners independently on each single labels. It has been previously studied in single-label classifier ensemble context by Krogh and Vedelsby (1995). The diversity term prefers the variabilities of individuals that are learned from different perspectives. It is a well known factor to improve the ensemble performance. The coherence term, that is specific to the multilabel classifiers, indicates that the more the microlabel predictions vary together, the greater advantage multilabel ensemble gets over the base learners. This supports our intuitive understanding that microlabel correlations are keys to successful multilabel learning.

Table 1 Statistics of multilabel datasets used in our experiments. For *NCI60* and *Fingerprint* dataset where there is no explicit feature representation, the rows of kernel matrix is assumed as feature vector

Dataset	Statistics				
	Instances	Labels	Features	Cardinality	Density
Emotions	593	6	72	1.87	0.31
Yeast	2417	14	103	4.24	0.30
Scene	2407	6	294	1.07	0.18
Enron	1702	53	1001	3.36	0.06
Cal500	502	174	68	26.04	0.15
Fingerprint	490	286	490	49.10	0.17
NCI60	4547	60	4547	11.05	0.18
Medical	978	45	1449	1.14	0.03
Circle10	1000	10	3	8.54	0.85
Circle50	1000	50	3	35.63	0.71

4 Experiments

4.1 Datasets

We experiment on a collection of ten multilabel datasets from different domains, including chemical, biological, and text classification problems. The *NCI60* dataset contains 4547 drug candidates with their cancer inhibition potentials in 60 cell line targets. The *Fingerprint* dataset links 490 molecular mass spectra together to 286 molecular substructures used as prediction targets. Four text classification datasets[1] are also used in our experiment. In addition, two artificial *Circle* dataset are generated according to Bian et al. (2012) with different amount of labels. An overview of the datasets is shown in Table 1, where *cardinality* is defined as the average number of positive microlabels for each example

$$cardinality = \frac{1}{m}\sum_{i=1}^{m} |\{j|\mathbf{y}_{ij} = 1\}|,$$

and *density* is the average number of labels for each example divided by the size of label space defined as

$$density = \frac{cardinality}{k}.$$

4.2 Kernels

We use kernel methods to describe the similarity between complex data objects in some experiment datasets. We calculate linear kernel on datasets where instants are described by feature vectors. For text classification datasets, we first compute weighted features with term frequency inverse document frequency model (TF-IDF) (c.f. Rajaraman and Ullman 2011). TF-IDF weights reflect how important a word is to a document in a collection of corpus defined as the ratio between the word frequency in a document and the word frequency in the a collection of corpus. We compute linear kernel of the weighted features.

[1] Available at http://mulan.sourceforge.net/datasets.html.

As the input kernel of the *Fingerprint* dataset where we have for each instant a mass spectrometry (MS) data, we calculated quadratic kernel over the 'bag' of mass/charge peak intensities. As the input kernel of the *cancer* dataset where each object is described as a molecular graph, we used the hash fingerprint Tanimoto kernel (Ralaivola et al. 2005) that enumerates all linear fragments up to length n in a molecule x. A hash function assigns each fragment a hash value that determines its position in descriptor space $\phi(x)$. Given two binary bit vectors $\phi(x)$ and $\phi(y)$ as descriptors, Tanimoto kernel is defined as

$$K(x, y) = \frac{|I(\phi(x)) \cap I(\phi(y))|}{|I(\phi(x)) \cup I(\phi(y))|},$$

where $I(\phi(x))$ denotes the set of indices of 1-bits in $\phi(x)$.

In practice, some learning algorithms required kernelized input while others need feature representation of input data. Due to the intractability of using explicit features for complex data and in order to achieve a fair comparison, we take precomputed kernel matrix as rows of feature vectors for the learning algorithms that required input of feature vectors.

4.3 Obtaining random output graphs

The structure of the output graph is significant both in term of efficiency of learning and inference, and the predictive performance. We consider the following two approaches to generate random output graphs.

- In the random pair approach, one takes each vertex in turn, randomly draw another vertex and couples the two with an edge.
- In the random spanning tree approach, one first draws a random $k \times k$ weight matrix W with non-negative edge weights and then extracts a maximum weight spanning tree out of the matrix, using w_{ij} as the weight for edge connecting labels i and j.

The random pair approach generally produces a set of disconnected graphs, which may not let the base learner to fully benefit from complex multilabel dependencies. On the other hand, the learning of the base classifier is potentially made more efficient due to the graph simplicity. The random spanning tree approach connects all targets so that complex multilabel dependencies can be learned. Also, the tree structure facilitates efficient inference.

4.4 Compared classification methods

For comparison, we choose the following established classification methods form different perspectives towards multilabel classification, accounting for single-label and multilabel, as well as ensemble and standalone methods:

- **MMCRF** (Rousu et al. 2007) is used both as a standalone multilabel classifier and the base classifier in the ensembles. Individual MMCRF models are trained with two kinds of output graphs, random tree and random pair graph.
- **SVM** is a discriminative learning method that has become very popular over recent years, described in several textbooks (Cristianini and Shawe-Taylor 2000; Schölkopf and Smola 2001). For multilabel classification task, we split the multilabel task into a collection of single-label classification problems. Then we apply SVM on each single problem and compute the predictions. The drawback of SVM on multilabel classification task is the computation becomes infeasible as the number of the labels increases. Beside, this approach assumes independency between labels, it does not get any benefit from dependencies defined by complex structures of the label space. SVM serves as the single-label non-ensemble baseline learner.

- **MTL** is a multi-task feature learning methods developed in Argyriou et al. (2008), which is used as multilabel baseline learner. The underlying assumption of MTL is that the task specific functions are related such that they share a small subset of features.
- **Adaboost** is an ensemble method that has been extensively studied both empirically and theoretically since it was developed in Freund and Schapire (1997). The idea behind the model is that a distribution is assigned over data points. In each iteration, a weak hypothesis is calculated based on current distribution, and the distribution is updated according to the performance of the weak hypothesis. As a results, the difficult examples will receive more weight (probability mass) after the update, and will be emphasized by the base learner in the next round.
  In addition, Adaboost for multilabel classification using Hamming loss (AdaboostMH), is designed for incorporating multilabel learner into Adaboost framework (Schapire and Singer 1998). The only difference is the distribution is assigned to each example and microlabel pair and updated accordingly. In our study, we use real-valued decision tree with at most 100 leaves as base learner of AdaboostMH, and generate an ensemble with 180 weak hypothesises.
- **Bagging** (bootstrapping aggregation) was introduced in Breiman (1996) as an ensemble method of combining multiple weak leaners. It creates individual weak hypothesises for its ensemble by calling base learner repeatedly on the random subsets of the training set. The training set for the weak learner in each round is generated by randomly sampling with replacement. As a result, many original training examples may be repeated many times while others may be left out. In our experiment, we randomly select 40% of the data as input to SVM to compute a weak hypothesis, and repeat the process until we collect an ensemble of 180 weak hypotheses.

4.5 Parameter selection and evaluation measures

We first sample 10 % data uniform at random from each experimental dataset for the purpose of parameter selection. SVM, MMCRF and MAM ensemble each have a margin softness parameter C, which potentially needs to be tuned. We tested the value of parameter C from a set $\{0.01, 0.1, 0.5, 1, 5, 10\}$ based on tuning data, then keep the best ones for the following validation step. We also perform extensive selection on γ parameters in MTL model in the same range as margin softness parameters.

We observe that most of the multilabel datasets are highly biased with regards to multilabel density. Therefore, we use the following *stratified* 5*-fold cross validation* scheme in the experiments reported, such that we group examples in equivalence classes based on the number of positive labels they have. Each equivalence class is then randomly split into five local folds, after that the local folds are merged to create five global folds. The proposed procedure ensures that also the smaller classes have representations in all folds.

To quantitatively evaluate the performance of different classifiers, we adopt several performance measures. We report *multilabel accuracy* which counts the proportion of multilabel predictions that have all of the microlabels being correct, *microlabel accuracy* as the proportion of microlabel being correct, and *microlabel* F_1 *score* that is the harmonic mean of microlabel precision and recall $F_1 = 2 \cdot \frac{Pre \times Rec}{Pre + Rce}$.

4.6 Comparison of different ensemble approaches

We evaluate our proposed ensemble approaches by learning ensemble with 180 base learners. The learning curves as the size of ensemble on varying datasets are shown in Figs. 3, 4, and

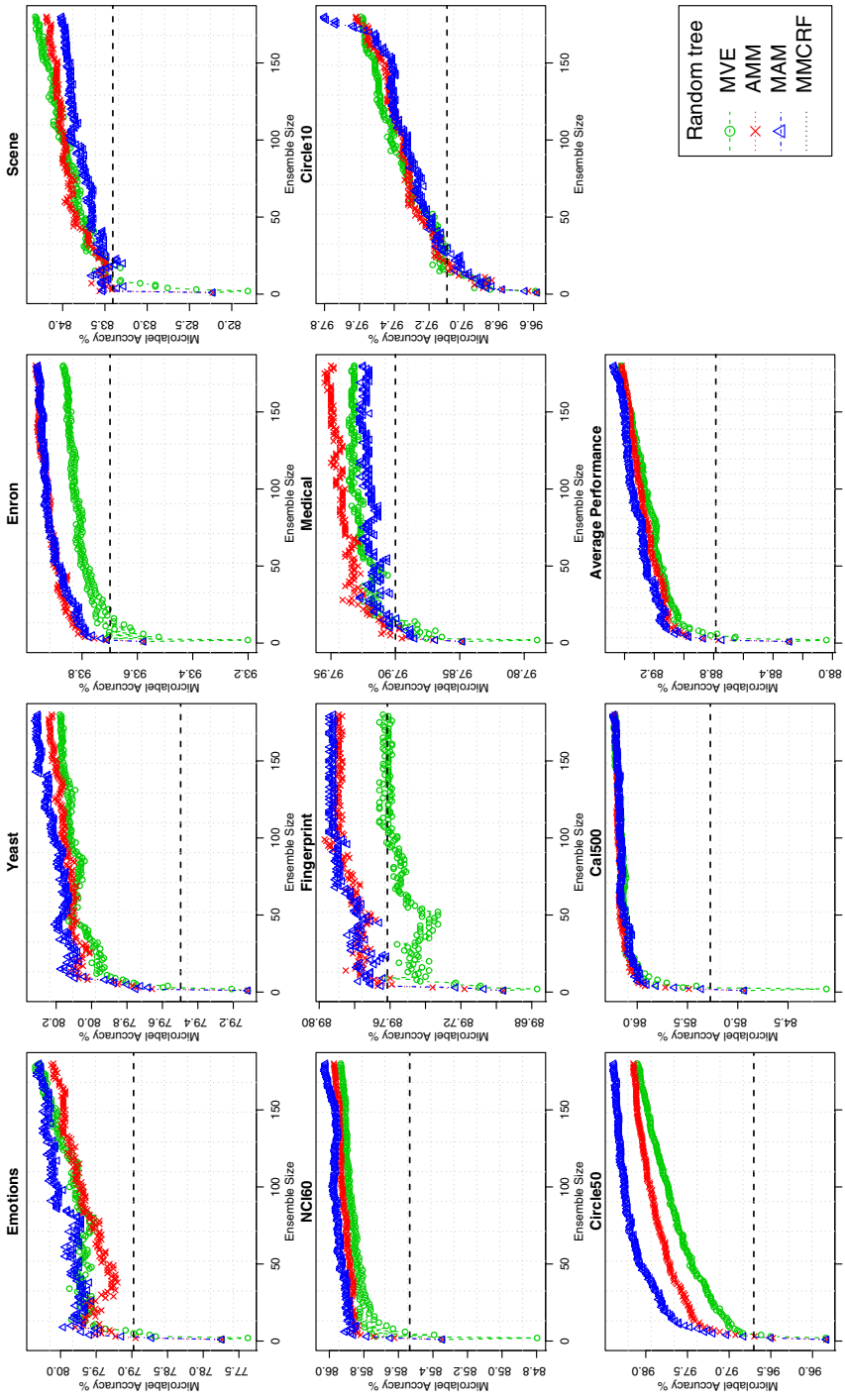

Fig. 3 Ensemble learning curve (microlabel accuracy) plotted as the size of ensemble. Average performance over datasets is shown as the last plot

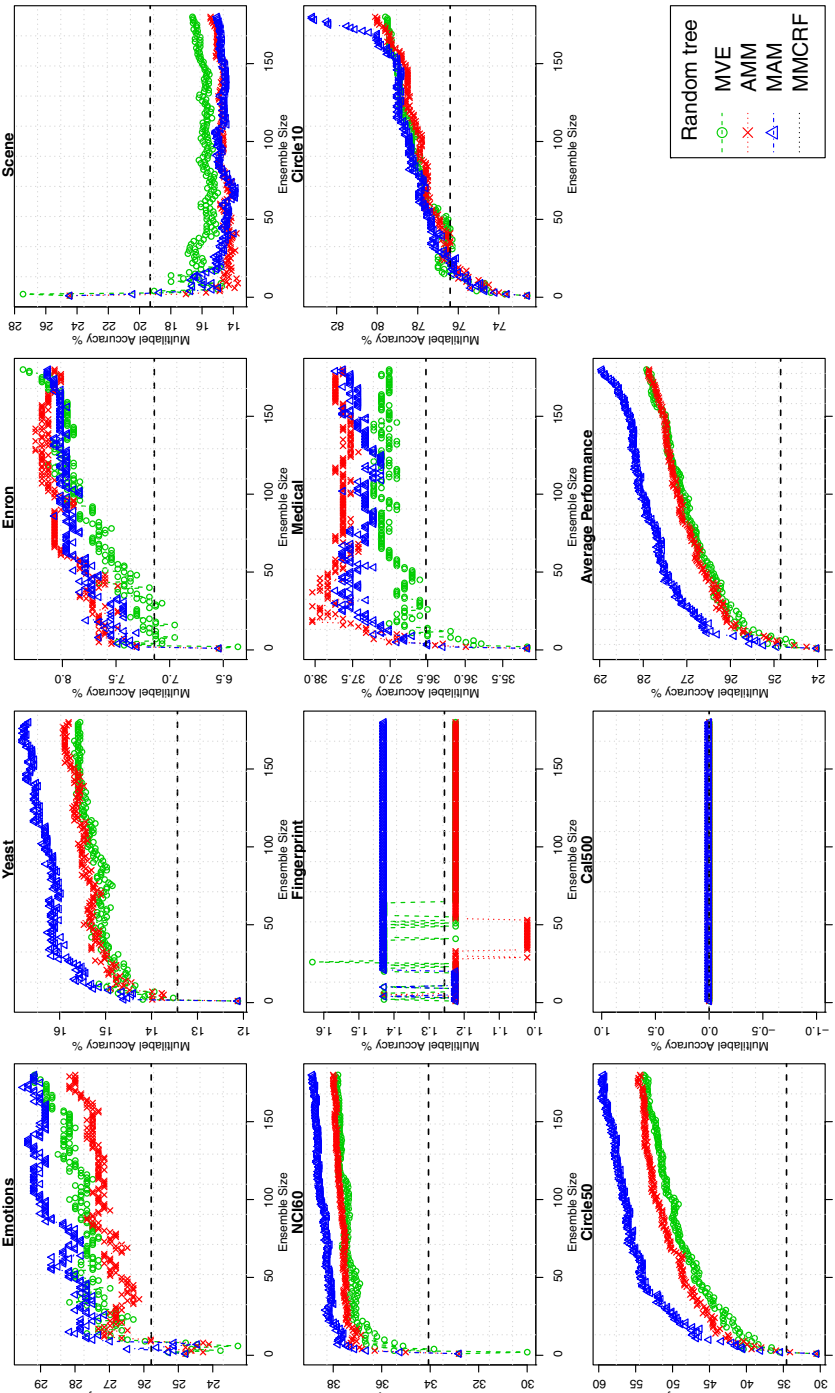

Fig. 4 Ensemble learning curve (multilabel accuracy) plotted as the size of ensemble. Average performance over datasets is shown as the last plot

5 for microlabel accuracy, multilabel accuracy, and microlabel F1 score, respectively. The base learners are trained with random tree as output graph structure.

There is a clear trend of improving microlabel accuracy for proposed ensemble approaches as more individual base models are combined. On most datasets and algorithms the ensemble accuracy increases fast and levels off rather quickly, the most obvious exception being the Circle10 dataset where improvement can be still seen beyond ensemble size 180. In addition, all three proposed ensemble learners (MVE, AMM, MAM) outperform their base learner MMCRF (horizontal dash lines) with consistent and noticeable margins, which is best seen from the learning curves of the average performance.

Similar patterns of learning curves are also observed in microlabel F_1 (Fig. 4) and multilabel accuracy (Fig. 5), with a few exceptions. The Fingerprint and Cal500 datasets prove to be difficult to learn in that very few multilabels are perfectly predicted, this is not surprising as these datasets have a large number of microlabels. The datasets also have the largest proportion of positive microlabels, which is reflected in the low F1 score. *Scene* dataset is the only exception where increasing the number of base learners seems to hurt the ensemble performance in microlabel F_1 and multilabel accuracy. In fact *Scene* is practically a single-label multiclass dataset, having very few examples with more than one positive microlabel. This contradicts the implicit assumption of graph based learners that there are rich dependency structures between different labels that could be revealed by the different random graphs. Among the extreme label sparsity, the ensemble learners appear to predict more negative labels for each example which leads to decreased performances in F_1 and multilabel accuracy space. We also observe large fluctuations in the initial part of MVE learning curves of *Fingerprint* and *Cal500* datasets in F_1 score space, implying MVE is not as stable as AMM and MAM approaches.

In particular, the performance of MAM ensemble surpasses MVE and AMM in eight out of ten datasets, the exceptions being *Scene* and *Medical*, making it the best among all proposed ensemble approaches. Consequently, we choose MAM for the further studies described in the following sections.

4.7 Effect of the structure of output graph

To find out which is the more beneficial output graph structure, we carry out empirical studies on MAM ensemble with random tree and random pair graph as output graph structure. Table 2 illustrates the performance of two output structures in terms of microlabel accuracy, multilabel accuracy and microlabel F_1 score. The results show that random tree and random pair graph are competitive output graph structures in terms of microlabel accuracy and F_1 score, with random tree achieves slightly better results. In addition, we observe noticeable difference in multilabel accuracy, where random tree behaves better than random pair graph. One way to understand this is to realize that random tree is able to connect all output labels so that learning and inference can work over the the whole label space. On the other hand, random pair approach divides the label space into isolated pairs where there is no cross-talk between pairs.

We continue by studying learning curves of average performance of MAM ensemble on two different output structures. Fig. 6 illustrates that MAM ensemble with random tree as output structure consistently outperforms random pair in accuracy space. The performance differences in F_1 space are not clear where we see the random pair approach fluctuating around random tree curve. Base on the experiments, we deem random tree the better of the two output graph structures.

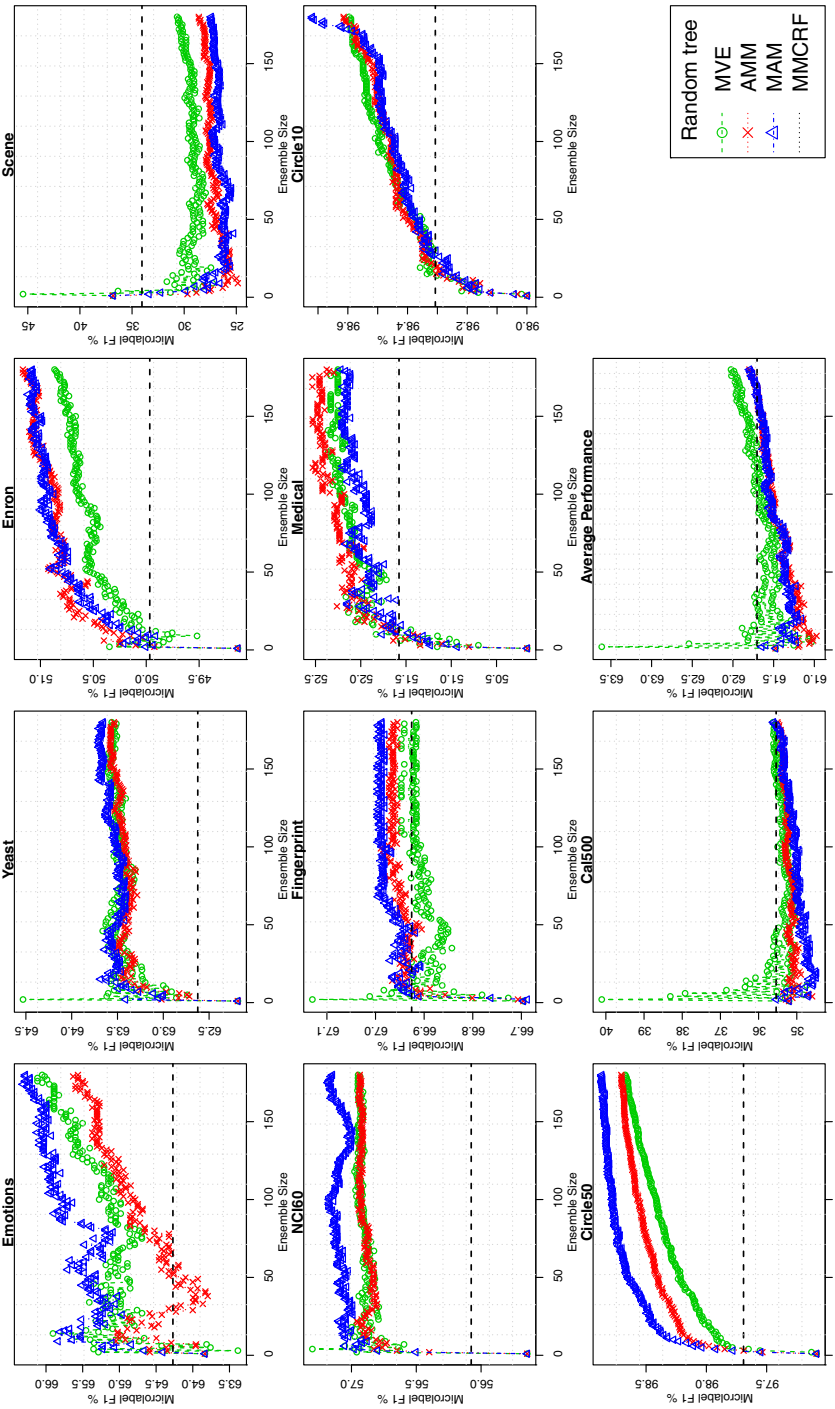

Fig. 5 Ensemble learning curve (microlabel F_1 score) plotted as the size of ensemble. Average performance over datasets is shown as the last plot

Table 2 Prediction performance of MAM ensemble with random tree and random pair graph in terms of microlabel accuracy, multilabel accuracy, and microlabel F_1 score

Dataset	Microlabel Acc %		Multilabel Acc %		Microlabel F_1 %	
	Pair	Tree	Pair	Tree	Pair	Tree
Emotions	**80.4** ± 2.4	80.3 ± 1.4	27.8 ± 3.4	**29.2** ± 4.2	65.7 ± 4.3	**66.3** ± 2.3
Yeast	80.2 ± 0.7	**80.3** ± 0.5	15.9 ± 1.1	**16.7** ± 0.4	63.5 ± 1.4	**63.7** ± 1.1
Scene	84.0 ± 0.5	**84.0** ± 0.1	**16.4** ± 1.9	15.0 ± 0.9	**28.9** ± 2.5	27.4 ± 2.4
Enron	**94.1** ± 0.1	94.0 ± 0.2	7.7 ± 1.0	**8.1** ± 2.3	51.1 ± 1.9	**51.1** ± 1.3
Cal500	**86.2** ± 0.1	86.2 ± 0.2	0.0 ± 0.0	0.0 ± 0.0	35.2 ± 0.8	**35.6** ± 0.4
Fingerprint	89.8 ± 0.5	**89.8** ± 0.3	1.2 ± 0.6	**1.4** ± 0.6	66.9 ± 2.5	**67.0** ± 1.9
NCI60	85.9 ± 0.8	**86.0** ± 0.9	37.9 ± 1.2	**38.9** ± 1.2	57.1 ± 3.8	**57.1** ± 3.2
Medical	97.9 ± 0.2	**97.9** ± 0.1	37.6 ± 4.3	**37.6** ± 2.5	52.2 ± 4.6	**52.2** ± 3.2
Circle10	97.5 ± 0.4	**97.8** ± 0.4	79.0 ± 2.0	**83.2** ± 3.5	98.5 ± 0.2	**98.7** ± 0.3
Circle50	97.6 ± 0.3	**98.4** ± 0.3	47.6 ± 5.9	**59.4** ± 5.6	98.3 ± 0.2	**98.9** ± 0.2

4.8 Multilabel prediction performance

In the following experiments we examine whether our proposed ensemble model (MAM) can boost the prediction performance in multilabel classification problems. Therefore, we compare our model with other advanced methods including both single-label and multilabel classifiers, both standalone and ensemble frameworks. Table 3 shows the performance of difference methods in terms of microlabel accuracy, multilabel accuracy and microlabel F_1 score, where the best performance in each dataset is emphasised in boldface and the second best is shown in italics.

We observe from Table 3 that MAM receives in general higher evaluation scores than the competitors. In particular, it achieves nine times as top two performing methods in microlabel accuracy, eight times in multilabel accuracy, and eight times in microlabel F_1 score. The only datasets where MAM is consistently outside the top two is the *Scene* dataset. As discussed above, the dataset is practically a single-label multiclass dataset. On this dataset the single target classifiers SVM and Bagging outperform all compared multilabel classifiers.

In these experiments, MMCRF also performs robustly, being in top two on half of the datasets with respect to microlabel and multilabel accuracy. However, it quite consistently trails to MAM in all three evaluation scores, the *Scene* dataset again being the exception. We also notice that the standalone single target classifier SVM is competitive against most multilabel methods, performs better than Bagging, AdaBoost and MTL with respect to microlabel and microlabel accuracy.

4.9 Statistical evaluation

To statistically evaluate the performance of different methods over multiple datasets, we first apply paired t-test on the values shown in Table 3. In particular, we compute a test statistic (with a p-value) for each ordered pair of methods to assess whether the average performance of the first is better than the second in a statistically significant manner. The result, shown in Table 4, indicates that, in terms of microlabel accuracy, MAM significantly outperforms MMCRF, AdaBoost and Bagging and almost significantly outperforms MTL, while the performance is not significantly different from SVM. In multilabel accuracy, MAM

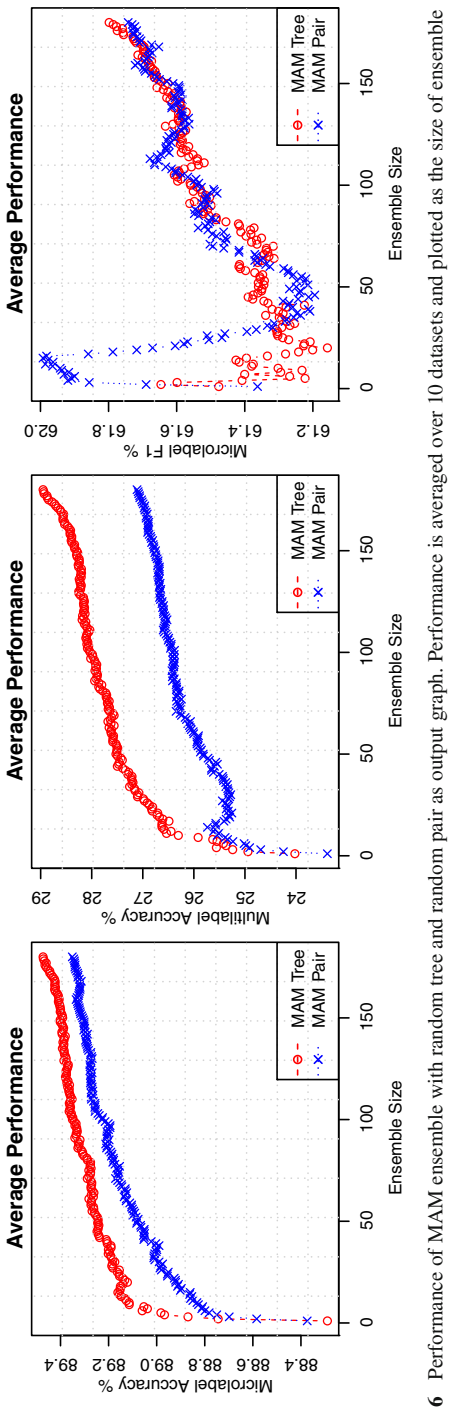

Fig. 6 Performance of MAM ensemble with random tree and random pair as output graph. Performance is averaged over 10 datasets and plotted as the size of ensemble

Table 3 Prediction performance of methods in terms of microlabel accuracy (top), microlabel F_1 score (middle), and multilabel accuracy (bottom). '–' represents no positive predictions. 'Avg. Rank' is the average rank of the performance over datasets

Dataset	SVM	Bagging	AdaBoost	MTL	MMCRF	MAM
Microlabel accuracy %						
Emotions	77.3±1.9	74.1±1.8	76.8±1.6	79.8±1.8	*79.0*±0.9	**80.3**±1.4
Yeast	*80.0±0.7*	78.4±0.9	74.8±0.7	79.3±0.5	79.5±0.6	**80.3**±0.5
Scene	**90.2**±0.3	87.8±0.5	84.3±0.9	*88.4±0.5*	83.4±0.3	84.0±0.1
Enron	93.6±0.2	*93.7±0.1*	86.2±0.3	93.5±0.2	93.7±0.2	**94.0**±0.2
Cal500	**86.3**±0.3	86.0±0.2	74.9±0.7	*86.2±0.3*	85.3±0.3	*86.2±0.2*
Fingerprint	89.7±0.3	85.0±0.4	84.1±0.7	82.7±0.6	**89.8**±0.6	**89.8**±0.3
NCI60	84.7±0.7	79.5±0.4	79.3±0.8	84.0±0.6	*85.5±1.3*	**86.0**±0.9
Medical	97.4±0.0	97.4±0.1	91.4±0.3	97.4±0.1	**97.9**±0.1	**97.9**±0.1
Circle10	94.8±0.9	92.9±0.7	**98.0**±0.3	93.7±0.7	97.1±0.3	*97.8±0.4*
Circle50	94.1±0.5	91.7±0.5	96.6±0.3	93.8±0.5	*96.7±0.3*	**98.4**±0.3
Avg. Rank	3.0	4.5[a]	4.8[a]	4.0[a]	3.0	**1.8**
Microlabel F_1 score %						
Emotions	57.1±4.4	61.5±3.1	*66.2±2.9*	64.6±3.0	64.3±1.2	**66.3**±2.3
Yeast	62.6±1.1	**65.5**±1.4	63.5±1.2	60.2±1.2	62.6±1.2	*63.7±1.1*
Scene	*68.3±1.4*	**69.9**±1.4	64.8±2.1	61.5±2.1	34.0±2.7	27.4±2.4
Enron	29.4±1.5	38.8±1.0	42.3±1.1	–	*50.0±1.0*	**51.1**±1.3
Cal500	31.4±0.6	*40.1±0.8*	**44.3**±1.5	28.6±1.3	35.5±0.4	35.6±0.4
Fingerprint	66.3±0.7	64.4±0.5	62.8±1.2	0.4±0.3	*66.9±0.8*	**67.0**±1.9
NCI60	45.9±3.6	53.9±1.2	32.9±2.7	32.9±3.4	*56.1±3.7*	**57.1**±3.2
Medical	–	–	33.7±1.2	–	*51.6±2.7*	**52.2**±3.2
Circle10	97.0±0.6	96.0±0.4	**98.8**±0.2	96.4±0.4	*98.3±0.2*	*98.7±0.3*
Circle50	96.0±0.3	94.5±0.3	97.6±0.2	95.7±0.3	*97.7±0.3*	**98.9**±0.2
Avg. Rank	4.2[a]	3.8[b]	3.0	5.2[a]	3.0	**1.9**
Multilabel accuracy %						
Emotions	21.2±3.4	20.9±2.6	23.8±2.3	25.5±3.5	*25.8±3.1*	**29.2**±4.2
Yeast	*14.0±2.8*	13.1±1.9	7.5±1.3	11.3±1.0	13.4±1.5	**16.7**±0.4
Scene	**52.8**±1.4	*46.5±1.9*	34.7±2.2	44.8±3.6	19.3±1.2	15.0±0.9
Enron	0.4±0.3	0.1±0.2	0.0±0.0	0.4±0.4	*7.1±2.8*	**8.1**±2.3
Cal500	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
Fingerprint	1.0±0.7	0.0±0.0	0.0±0.0	0.0±0.0	*1.2±0.5*	**1.4**±0.6
NCI60	*43.1±1.3*	21.1±0.9	2.5±0.6	**47.0**±2.0	34.1±1.4	38.9±1.2
Medical	8.2±2.1	8.2±2.7	5.1±2.0	8.2±2.3	*36.5±3.3*	**37.6**±2.5
Circle10	69.1±3.8	64.8±3.3	**86.0**±2.7	66.8±3.4	76.4±2.1	*83.2±3.5*
Circle50	29.7±2.0	21.7±3.9	28.9±3.4	27.7±3.3	*34.6±4.5*	**59.4**±5.5
Avg. Rank	3.1	4.7[a]	4.5[a]	3.9[b]	2.9	**2.0**

The average rank is marked with [a] (resp. [b]) if the algorithm performs significantly different at p-value = 0.05 (resp. at p-value = 0.1) from the top performing one according to two-tailed Bonferroni–Dunn test

Table 4 Paired t-test to assess whether the method from group A outperforms the one from group B in a significant manner. By '†, ∗, ‡' we denote the performance in microlabel accuracy, microlabel F_1 score, and multilabel accuracy, respectively

Group A	Group B					
	SVM	Bagging	AdaBoost	MTL	MMCRF	MAM
SVM	–	††, ‡	††	∗	–	–
Bagging	∗∗	–	–	∗∗	–	–
Adaboost	–	–	–	∗∗	–	–
MTL	–	†	††	–	–	–
MMCRF	–	††	††	∗∗	–	–
MAM	–	††, ‡	††, ‡	†, ∗∗	††, ‡	–

By double marks (e.g. '††') we denote p-value = 0.05, and by single mark (e.g. '†') we denote p-value = 0.1. By '–' we denote not significant given above p-values

outperforms Bagging, Adaboost and MMCRF in almost significant manner. SVM, MTL and MMCRF perform similarly to each other, and are better than Bagging and Adaboost with respect to microlabel accuracy. In addition, in microlabel F_1 score, we notice that all methods are competitive against MTL, and SVM performs better than Bagging.

As suggested by Demšar (2006), several critical assumptions might be violated when performing paired t-test to compare classifiers over multiple datasets. Similarly, other commonly used statistical tests might also be ill-posed in this scope (e.g. sign test, Wilcoxon signed-ranks test, ANOVA with post-hoc Tukey test). We therefore follow the test procedure proposed in Demšar (2006). First, we compute the rank of each model based on the performance on different datasets, where the best performing algorithm getting the rank of one. In case of ties, averaged ranks are then assigned. Then we use Friedman test (Friedman 1937) which compares the average rank of each algorithm, and under null hypothesis, states that all algorithms are equivalent with equal average ranks. P-values calculated from Friedman test for microlabel accuracy, microlabel F_1 score, and multilabel accuracy are 0.001, 0.002 and 0.005, respectively. As a result, we reject the null-hypothesis and proceed with post-hoc two-tailed Bonferroni-Dunn test (Dunn 1961), where all other methods are compared against the top performing control (MAM). We compute the *critical difference* $CD = 2.2$ at p-value = 0.05, and $CD = 1.9$ at p-value = 0.1 (see details in supplementary material). The performance of an algorithm is significantly different from the control if the corresponding average ranks differ by at least CD. The corresponding rank is marked with 'b' (at p-value = 0.1) or 'a' (at p-value = 0.05) in Table 3. We observe from the results that in microlabel accuracy and multilabel accuracy, the performance differences of SVM and MMCRF to MAM fail to be statistically significant. On the other hand, Bagging, Adaboost and MTL perform significantly worse than MAM in terms of microlabel accuracy and multilabel accuracy. In addition, with respect to microlabel F_1 score, the performances of MMCRF and Adaboost are not significantly different from MAM, while SVM, Bagging and MTL perform worse than MAM in a significant manner.

Overall, the results indicate that ensemble by MAM is a robust and competitive alternative for multilabel classification.

4.10 Effect of diversity and coherence

To explain the performance of MAM as well as to empirically validate the diversity and coherence arguments stated in Theorem 1, we conduct the following experiment.

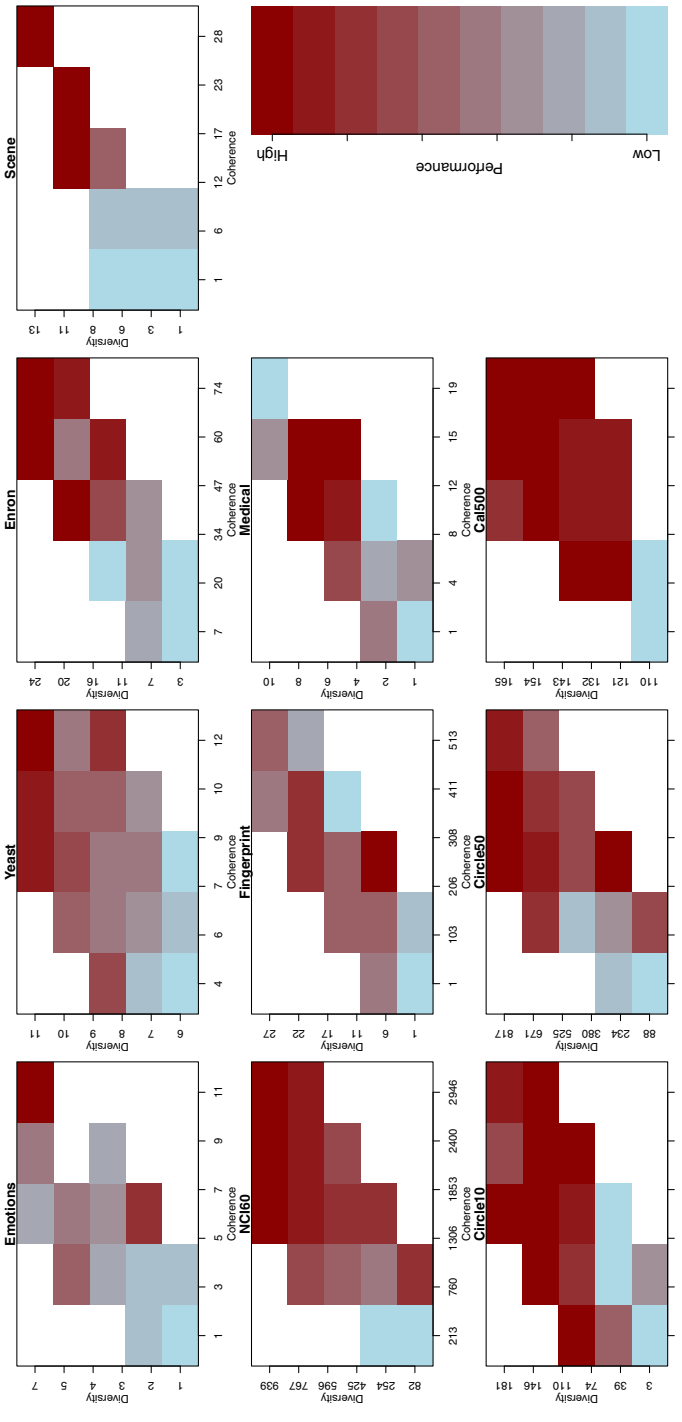

Fig. 7 Performance of MAM ensemble plotted in diversity and coherence space. Color of the blocks depicts average performance in term of microlabel accuracy computed from the data points in the block. White area means there is no examples with corresponding diversity and coherence. Colors are normalized for datasets so that worst and best performances are shown as light blue and red

We train a MAM ensemble model for each dataset consist of 30 base learners with a random spanning tree as output graph structure. For each example-label pair $(x_i, \mathbf{y}_i)$ and the corresponding set of microlabels $\mathbf{y}_i = \{y_{i,1}, \cdots, y_{i,l}\}$, we then calculate from each base learner t a set of node compatibility scores $\{\psi^t(x_i, y_{i,1}), \cdots, \psi^t(x_i, y_{i,l})\}$. Next, the node compatibility scores from different base learners are pooled together to get $\Psi_j(x_i, y_{i,j}) = \{\psi^1(x_i, y_{i,j}), \cdots, \psi^{30}(x_i, y_{i,j})\}$ for all $j \in \{1, \cdots, l\}$. Diversity and coherence of pair $(x_i, \mathbf{y}_i)$ can be calculated from $\{\Psi_j(x_i, y_{i,j})\}_{j=1}^{l}$ according to

$$Diversity = \sum_{j \in \{1 \cdots l\}} \boldsymbol{Var}(\Psi_j(x_i, y_{ij})),$$

$$Coherence = \sum_{\substack{p,q \in \{1 \cdots l\}, \\ p \neq q}} \boldsymbol{Cov}(\Psi_p(x_i, y_{ip}), \Psi_q(x_i, y_{iq})),$$

which locates pair $(x_i, \mathbf{y}_i)$ in the diversity-coherence space. We also compute the microlabel accuracy from the microlabels in $\mathbf{y}_i$ based on the prediction from MAM ensemble. The accuracy of different diversity-coherence region in the space is computed as the average microlabel accuracy of examples in that region. The results are shown in Fig. 7.

We observe from the results a pattern of increasing prediction performance from lower left corner to upper right corner. In particular, microlabel accuracy are lower for examples with both low diversity and coherence computed based on current set of base learners, shown as the light blue blocks in lower left corner. On the other hand, we achieve higher prediction accuracy on examples with high diversity and coherence, which are shown as red blocks in the upper right corner. In addition, fixing one factor while increasing the other usually leads to improved performance.

The observations demonstrates both diversity and coherence have positive effects on the performance of MAM ensemble. They reflect different aspects of the ensemble. To improve the quality of the prediction, one should aim to increase either the diversity of the base learner on a single microlabel or the coherence among microlabel pairs.

5 Conclusions

In this paper we have put forward new methods for multilabel classification, relying on ensemble learning on random output graphs. In our experiments, models thus created have favourable predictive performances on a heterogeneous collection of multilabel datasets, compared to several established methods. The theoretical analysis of the MAM ensemble highlights the covariance of the compatibility scores between the inputs and microlabels learned by the base learners as the quantity explaining the advantage of the ensemble prediction over the base learners.

We note in passing that it is straightforward to generalize the theoretical analysis to any multilabel classifiers that give scores to microlabels; there is no dependency on random graph classifiers in the analysis.

The empirical evaluation supports the theoretical analysis, explaining the performance of the proposed ensemble models. Our results indicate that structured output prediction methods can be successfully applied to problems where no prior known output structure exists, and thus widen the applicability of the structured output prediction.

Acknowledgments The work was financially supported by Helsinki Doctoral Programme in Computer Science (Hecse), Academy of Finland grant 118653 (ALGODAN), IST Programme of the European Community under the PASCAL2 Network of Excellence, ICT-2007-216886. This publication only reflects the authors' views.

References

Argyriou, A., Evgeniou, T., & Pontil, M. (2008). Convex multi-task feature learning. *Machine Learning*, *73*(3), 243–272.

Bian, W., Xie, B., & Tao, D. (2012). Corrlog: Correlated logistic models for joint prediction of multiple labels. In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12)*, vol. 22, pp. 109–117.

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

Brown, G., & Kuncheva, LI. (2010). Good and bad diversity in majority vote ensembles. In: *Multiple classifier systems* (pp. 124–133). Berlin: Springer.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods* (1st ed.). Cambridge: Cambridge University Press.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, *7*, 1–30.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, *56*(293), 52–64.

Esuli, A., Fagni, T., & Sebastiani, F. (2008). Boosting multi-label hierarchical text categorization. *Information Retrieval*, *11*(4), 287–313.

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

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, *32*(200), 675–701.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In: *Advances in neural information processing systems* (pp. 231–238). Cambridge, MA: MIT Press.

Rajaraman, A., & Ullman, J. (2011). *Mining of massive datasets*. Cambridge: Cambridge University Press.

Ralaivola, L., Swamidass, S., Saigo, H., & Baldi, P. (2005). Graph kernels for chemical informatics. *Neural Networks*, *18*, 1093–1110.

Rousu, J., Saunders, C., Szedmak, S., & Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *The Journal of Machine Learning Research*, *7*, 1601–1626.

Rousu, J., Saunders, C., Szedmak, S., & Shawe-Taylor, J. (2007). Efficient algorithms for max-margin structured classification. In *Predicting structured data*, pp. 105–129.

Schapire, R., & Singer, Y. (1998). Improved boosting algorithms using confidence-rated predictions. In: *Proceedings of the Annual Conference on Computational Learning Theory* (pp. 80–91). New York: ACM Press.

Schapire, R. E., & Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine Learning*, *39*(2/3), 135–168.

Schölkopf, B., & Smola, A. (2001). *Learning with Kernels*. Cambridge, MA: MIT Press.

Su, H., & Rousu, J. (2001). Multi-task drug bioactivity classification with graph labeling ensembles. In *Proceedings of the 6th International Conference on Pattern Recognition in Bioinformatics (PRIB2011), Lecture Note in Computer Science (LNCS)*, (Vol. 7035, pp.157–167).

Su, H., & Rousu, J. (2013). Multilabel classification through random graph ensembles. In: *Proceedings, 5th Asian conference on machine learning (ACML2013), Journal of Machine Learning Research W&CP*, vol. 29, pp. 404–418.

Taskar, B., Guestrin, G., & Koller, D. (2004). Max-Margin Markov networks. In S. Thrun, L. K. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems*, (Vol. 16, pp. 25–32). MIT Press.

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In: *Proceedings of the twenty-first international conference on machine learning* ICML'04, pp. 823–830.

Wainwright, M., Jaakkola, T., & Willsky, A. (2005). MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, *51*(11), 3697–3717.

Yan, R., Tesic, J., & Smith, J. (2007). Model-shared subspace boosting for multi-label classification. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 834–843). ACM