Journal of Machine Learning Research 1 (2003) 1-48 Submitted 4/00; Published 10/00

Coupled Transductive Ensemble Learning of Kernel Models

Bart Hamers BART.HAMERS@ESAT.KULEUVEN.AC.BE
Johan A.K. Suykens JOHAN.SUYKENS@ESAT.KULEUVEN.AC.BE
Bart De Moor BART.DEMOOR@ESAT.KULEUVEN.AC.BE
K.U.Leuven, ESAT-SCD-SISTA
Kasteelpark Arenberg 10
B-3001 Leuven (Heverlee), Belgium

Editor: Leslie Pack Kaelbling

Abstract

In this paper we propose the concept of coupling for ensemble learning. In the existing literature, all submodels that are considered within an ensemble are trained independently from each other. Here we study the effect of coupling the individual training processes within an ensemble of regularization networks. The considered coupling set gives the opportunity to work with a transductive set for both regression and classification problems. We discuss links between this coupled learning and multitask learning and explain how it can be interpreted as a form of group regularization. The methods are illustrated with experiments on classification and regression data sets.

Keywords: Ensemble learning, kernel models, multitask learning, coupled minimizers, transductive learning

1. Introduction

It is a well-known fact in the literature that combining learning algorithms is a fruitful way for creating improved estimators. The idea behind this is that instead of using one single learning algorithm to model the data, to take a set or ensemble of models trained on the data. The ensemble model then combines the individual submodels in order to make predictions. In recent years many different strategies of ensemble learning have been developed. To situate our work we will give a brief overview of the current literature on this topic. One basically considers two groups of methods.

In a first group one trains a set of learning algorithms where each of the submodels is trained on the whole data set. The ensemble model is then constructed by averaging of the outputs of the submodels. It is proven in literature that the performance of the ensemble model is better than the mean performance of the submodels. One of the first studies on this topic was done by Breiman (1996) who proved that the error of a model created by averaging the results of a set of learning algorithms, each trained on a bootstrap sample of the original data set, is lower than the average error of the individuals. The requirement for an improved performance is that learning algorithms should be unstable. Notice that these bootstrap samples are made by random sampling N elements with replacement of the original data set of size N. The disadvantage of these methods is that there is no reduction in

©2003 B. Hamers, J.A.K. Suykens, B. De Moor; Corresponding author Johan Suykens.

computational complexity of the individual learning algorithms. These methods are known as *Bagging* (Breiman (1996)). Other developed models based on this idea are strategies like Boosting (Schapire (1999)), and Committee Networks (Perrone and Cooper (1993), Bishop (1995)). Those methods differ in terms of better sampling techniques or better weighting of the individual models.

In a second group of methods the individual learning algorithms operate on *sub-samples* of the original data set. This is often known as *mixture of experts* (Bishop (1995)). Hereby every submodel or 'expert' specializes on a certain task induced by its training set. Parallel to the submodels, a *gater* is trained. This gater network also sees the input and it is trained to judge which of the trained submodels is best suited to determine the output.

Notice that both of these strategies can be applied to several models such as neural networks or decision trees. In this work we take kernel models as individual experts. These models have proven to show good generalization performances on both classification and regressions tasks. However, one of the disadvantages of kernel models is that the training often needs heavy computational and memory resources that worst case scales quadratically with the number of data points, depending on the algorithms used. Therefore, subdividing the data set can be a promising method for tackling this problem. By creating ensembles of models trained on subsamples of the original data set, one aims at reducing the memory and computational complexity of the training algorithm without giving up too much in performance. In this way our proposed method will be related to averaging methods like bagging, but trained on subsets.

The idea of combining kernel models trained on subsets is not new. One of the first attempts of creating combinations of kernel methods for classification was done by Pavlov et al. (2000) and Evgeniou et al. (2000a). Pavlov experimentally showed that *Boosting* could be used to create ensembles of SVMs. Making ensembles of 10 to 15 SVMs, each trained on subsets between 2-4 % of the original data set, results into algorithms whose performance is comparable to a standard SVM trained on the whole data set. Moreover, computationally, speed-ups up to a factor 10 were measured. In Evgeniou et al. (2000a) it was theoretically prove and experimentally illustrated that bagging on subsets leads to better generalization performances. Other combinations of ensembles based on the boosting optimization methods where studied by Yamana et al. (2003). Another theoretical justification was recently given by Elisseeff and Pontil (2003). They showed that combining kernel models, which are often not stable, increases the stability.

Another approach for classification has been presented by Collobert et al. (2002) who proposed to take a mixture of SVMs each trained on a subset of the training data. The combination of the outputs of the individual outputs of the SVMs is then evaluated by an MLP Neural Network which acts as a *gater*. A crucial step in their approach is that the assignment of the data points to the individual SVM is done in several loops. Hereby the reassignment is based according to the believe that the gater addresses to the performances of one of the individual experts to a certain data point. An important advantage of this method is that it can be easily implemented in parallel. However, according to their results, much depends on the complexity of the gater. The number of neurons and hidden layers of the MLP has a big influence on the performance. In this way, the computational bottleneck is now partially transferred to the training of the gater.

Another related technique is the Bayesian Committee Support Vector Machine presented in Schwaighofer and Tresp (2001). This technique also partitions the training set into several randomly chosen subsets. On each of the these subsets an SVM is trained. Using a probabilistic output of the SVM, the combination scheme is based on the covariance of the test data. In this way this method can be seen as a *transductive* method (Vapnik (1998)). The disadvantage of this is that it cannot operate on a single data point and the whole test set should be known in advance.

In all previous mentioned methods the training of the individual submodels is done independently from each other. In Hamers et al. (2003) a technique with coupling of the individual learning processes was presented. In this paper we go deeper into this subject. The idea behind the method is that the different estimators of the ensemble are each trained on disjoint training sets, but have to agree on a predefined set of chosen data points. This *coupling set* can be chosen freely. It can be part of the training set, chosen at random or via more intelligent sampling criteria. Moreover, it can also be chosen as the test set. In this way the coupling can be done in a *transductive* way. The idea originates from the method of coupled local minimizers for solving optimization problems and coupled training processes that has been proposed in Suykens et al. (2001). There the coupling is done by means of coupling constraints. In this paper the outputs of individual models are coupled on a chosen set of data points, called here the coupling set. Further links and motivations based on multitask learning (Caruana (1997)) will be given. The method is then applied to the training of *parameterized kernel models* which take a similar functional form as kernel models arising in Gaussian Processes (MacKay (1998)), Kriging (Krige (1951)), RBF networks and Regularization Networks (Evgeniou et al. (2000b), Poggio and Girosi (1990)) and Least Squares SVMs (Suykens et al. (2002)) (which differ in the way they view the kernel model: e.g. parameterized, semi-parametric, Bayesian or with primal-dual optimization problem formulations). The method is illustrated on UCI data sets.

This paper is organized as follows. In Section 2 we discuss the individual models that form the ensemble. In Section 3 we explain how the submodels are combined. In Section 4 we introduce the concept of coupling set and explain how it can be used in ensemble learning. We explain the links between coupling and multitask learning and numerical aspects of the training process. In Section 5 we show experiments that illustrate the advantages of coupling for ensemble learning. The following notations are used. Vectors are denoted in bold case $\mathbf{x} \in \mathbb{R}^d$ with elements x_i; $\mathbf{1}_d = [1; ...; 1] \in \mathbb{R}^d$. The inner product between two vectors is defined as $\mathbf{x}^T\mathbf{z} = \sum_{i=1}^d x_i z_i$ where $\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$. The vector norm used in this paper is the Euclidean norm $\|\mathbf{x}\|_2 = (\sum_{i=1}^d x_i^2)^{1/2}$. Matrices are denoted by capital letters $A \in \mathbb{R}^{m \times n}$. A^T denotes the transpose of A and I_m the $m \times m$ identity matrix.

2. Parameterized Kernel Methods

In this paper we consider true models of the form $\mathbf{y} = f(\mathbf{x}) + \mathbf{e}$ where $E[\mathbf{e}] = \mathbf{0}$, $E\left[\mathbf{e}\mathbf{e}^T\right] = I\sigma^2$ and $\sigma^2 < \infty$. Based on a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times \mathbb{R}$ a model $\hat{f}(\mathbf{x}) = \hat{f}(\mathbf{x}; \mathcal{D})$ is estimated. Since we will focus on *ensemble learning*, this estimate $\hat{f}(\mathbf{x})$ results from combining the elements of a population of q estimators $\{\hat{f}^{(j)}\}_{j=1}^q$.

In order to construct our population of submodels, *parameterized kernel models* are considered. Let us take ensembles of estimators $\{\hat{f}^{(j)}\}_{j=1}^q$ where the individual models have

the form

$$\hat{f}^{(j)}(\mathbf{x}) = \sum_{p=1}^{g_j} \alpha_p^{(j)} k(\mathbf{x}_p^{(j)}, \mathbf{x})$$

where $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} : (\mathbf{x}, \mathbf{x}') \mapsto k(\mathbf{x}, \mathbf{x}')$. Unlike standard SVMs where k should be a Mercer kernel (positive definite kernel), this is not the case here as we consider parameterized kernel models. The kernel (or Gram) matrix constructed over the data set $\mathcal{D}^{(j)}$ of size g_j will be indicated by $K^{(j)}$. Each of the elements of these matrices are denoted by $K_{lm}^{(j)} = k(\mathbf{x}_l^{(j)}, \mathbf{x}_m^{(j)}), l, m = 1, ..., g_j$ where $(\mathbf{x}_l^{(j)}, y_l^{(j)}) \in \mathcal{D}^{(j)}$. The unknown parameters of each model $\hat{f}^{(j)}$ are $\alpha^{(j)} = [\alpha_1^{(j)} ... \alpha_{g_j}^{(j)}]^T$.

The estimation of these parameters is based on the least squares loss function with regularization (ridge regression). This leads to the following cost function $U^{(j)}(\alpha^{(j)})$ corresponding to the training of the submodels (Poggio and Girosi (1990))

$$\min_{\alpha^{(j)}} U^{(j)}\left(\alpha^{(j)}\right) = \frac{1}{2}\left\|K^{(j)}\alpha^{(j)} - y^{(j)}\right\|_2^2 + \frac{1}{2\gamma}\left\|\alpha^{(j)}\right\|_2^2. \tag{1}$$

The influence of the regularization term is controlled by the hyperparameter γ. The solution of this convex cost function can be found by setting $\nabla_{\alpha^{(j)}} U^{(j)}(\alpha^{(j)}) = 0$. This results into a linear system for each individual submodel:

$$(K^{(j)T}K^{(j)} + \frac{1}{\gamma} I_g)\alpha^{(j)} = K^{(j)T}\mathbf{y}^{(j)}, \tag{2}$$

where the output variables of the training data points of subset $\mathcal{D}^{(j)}$ are contained in the vector $\mathbf{y}^{(j)}$. Although the kernel used is not necessarily positive definite, the solution of this linear system is unique as long as the matrix is full rank, which can be ensured through the regularization term (note that $K^{(j)T}K^{(j)}$ is positive semidefinite).

3. Uncoupled Ensembles and Committee Networks

To combine the elements of the ensemble we will use *weighted averaging methods* (or *committee networks* as discussed e.g. in Bishop (1995))

$$\hat{f}(\mathbf{x},\mathcal{D}) = \sum_{j=1}^{q} \beta^{(j)} \hat{f}^{(j)}(\mathbf{x}). \tag{3}$$

We simplify the notation as $\hat{f}^{(j)}(\mathbf{x}) = \hat{f}^{(j)}(\mathbf{x}; \mathcal{D}^{(j)})$. To avoid that the whole data set $\mathcal{D}$ has to be learned by only one single model, we consider training the different submodels on separate subsets of the original data set. Therefor we divide $\mathcal{D}$ into q subsets, each of size g_j. Each of the submodels $\hat{f}^{(j)}$ is then trained on one of the mutually disjoint subsets $\mathcal{D}^{(j)} = \{(\mathbf{x}_p^{(j)}, y_p^{(j)})\}_{p=1}^{g_j} \subset \mathcal{D}$, for $j = 1, ..., q$. This training on the subsets introduces a variance on the performance of these different models which is reduced by taking a weighted average of the individual models. This variance reduction capability was already proven by Breiman (1996) on neural networks and decision tree models.

In the previous Section we explained a method for constructing the individual learning algorithms of our ensemble. But how should we interpret this for the whole ensemble? The

training of the ensemble corresponds to minimizing a cost function U_e which is the sum of the individual cost functions

$$U_e(\alpha^{(1)}, ..., \alpha^{(q)}) = \sum_{j=1}^{q} U^{(j)}(\alpha^{(j)}).$$

Setting $\nabla_{\alpha^{(j)}} U_e = 0, \forall j$ gives the solution

$$\begin{bmatrix} H^{(1)} & 0 & \cdots & 0 \\ 0 & H^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H^{(q)} \end{bmatrix} \begin{bmatrix} \alpha^{(1)} \\ \alpha^{(2)} \\ \vdots \\ \alpha^{(q)} \end{bmatrix} = \begin{bmatrix} K^{(1)T} y^{(1)} \\ K^{(2)T} y^{(2)} \\ \vdots \\ K^{(q)T} y^{(q)} \end{bmatrix}, \tag{4}$$

where $H^{(j)} = K^{(j)T} K^{(j)} + \frac{1}{\gamma} I_{g_j}$ for $j = 1, ..., q$. Note that the matrix of this linear system is a block-diagonal matrix where each individual subsystem is of the form (2). Hence there is no coupling between the learning processes for the parameter vectors $\alpha^{(j)}$. The training of an individual model on a data set $\mathcal{D}^{(j)}$ can be seen as the learning of one single task. Therefor, we call them *Single Task Learners*. This concept of Single Task Learners originates from Caruana (1997). Ensembles of Single Task Learners found via this training are called here *Uncoupled Ensembles* as opposed to coupled versions discussed in the next Section.

After finding the optimal parameter vectors of the individual submodels, the overall committee model, based on (3) becomes

$$\begin{aligned} \hat{f}(\mathbf{x}) &= \sum_{j=1}^{q} \beta^{(j)} \hat{f}^{(j)}(\mathbf{x}) \\ &= \sum_{j=1}^{q} \beta^{(j)} \sum_{p=1}^{g_j} \alpha_p^{(j)} k(\mathbf{x}_p^{(j)}, \mathbf{x}). \end{aligned}$$

From this expression one can clearly see that the overall models consist of two layers (Figure 1). The first layer is parameterized in terms of $\{\alpha^{(j)}\}_{j=1}^{q}$. The second layer is parameterized by $\{\beta^{(j)}\}_{j=1}^{q}$.

The optimal weights of the second layer can be determined e.g. according to committee networks (Perrone and Cooper (1993), Bishop (1995)) where the individual error functions are defined as $\epsilon^{(j)}(\mathbf{x}) = \hat{f}^{(j)}(\mathbf{x}) - f(\mathbf{x})$ and the committee network takes the form

$$\hat{f}(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^{q} \beta^{(j)} \epsilon^{(j)}(\mathbf{x}) \tag{5}$$

where one assumes that $\sum_{j=1}^{q} \beta^{(j)} = 1$. The cost function considered for the committee network is

$$\begin{aligned} J &= E\left[\left(\hat{f}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] \\ &= E\left[\left(\sum_{j=1}^{q} \beta^{(j)} \epsilon^{(j)}(\mathbf{x})\right)\left(\sum_{j=1}^{q} \beta^{(j)} \epsilon^{(j)}(\mathbf{x})\right)\right] \\ &\simeq \beta^T S \beta \end{aligned}$$

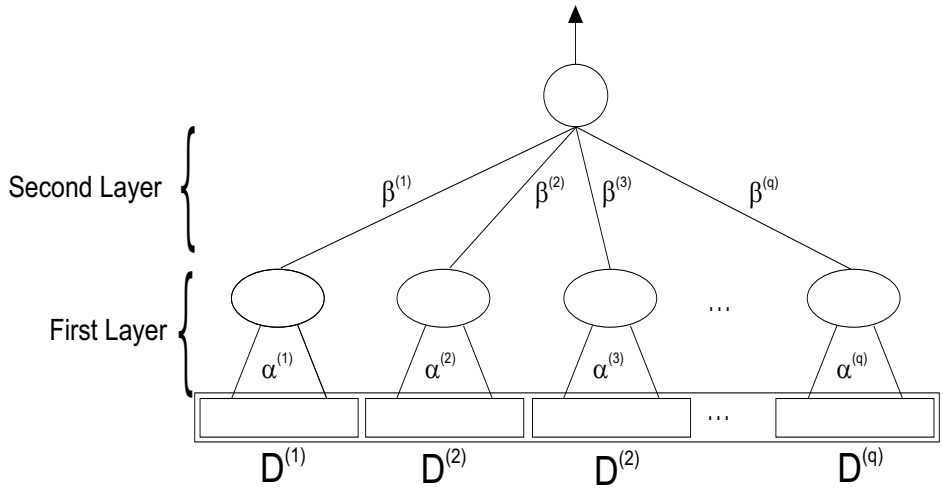

Figure 1: Architecture of the Uncoupled Ensemble Model.

where S denotes the error covariance matrix with $s_{ij} = E\left[\epsilon^{(i)}(\mathbf{x})\,\epsilon^{(j)}(\mathbf{x})\right]$, $\beta = [\beta^{(1)}; ...; \beta^{(q)}]$ and $E[\cdot]$ the expected value. This error covariance matrix can be computed by making use of a finite-sample approximation of $S = [s_{ij}] \in \mathbb{R}^{q\times q}$ based on a data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{p}$ of size p such that

$$s_{ij} \approx \frac{1}{p+1}\sum_{j=1}^{p}\left(\hat{f}(\mathbf{x}_i) - y_i\right)\left(\hat{f}(\mathbf{x}_j) - y_j\right).$$

This data set on which the error covariance matrix is built is typically a subset of the original training set.

An optimal choice of the parameters β can be found by minimizing

$$\min_{\beta}\ \beta^T S\beta \qquad \text{such that} \quad \sum_{j=1}^{q}\beta^{(j)} = 1. \tag{6}$$

The solution follows from the Lagrangian $\mathcal{L}(\beta,\lambda) = \frac{1}{2}\beta^T S\beta - \lambda\left(\sum_{j=1}^{q}\beta^{(j)} - 1\right)$ with Lagrange multiplier λ. The conditions for optimality are given by

$$\begin{aligned}\nabla_\beta\mathcal{L}(\beta,\lambda) &= S\beta - \lambda\mathbf{1}_q = 0\\ \nabla_\lambda\mathcal{L}(\beta,\lambda) &= \mathbf{1}_q^T\beta - 1 = 0\end{aligned}$$

with optimal solution

$$\beta = \frac{S^{-1}\mathbf{1}_q}{\mathbf{1}_q^T S^{-1}\mathbf{1}_q}$$

and corresponding error $J = \left(\mathbf{1}_q^T S^{-1}\mathbf{1}_q\right)^{-1}$.

4. Ensemble Learning using a Coupling Set

In relation to the literature on multitask learning (Caruana (1997)), one may interpret the learning of the different submodels on disjoint data sets as different tasks that have to be learned. The fact that each model $\hat{f}^{(j)}$ is trained on its individual data set $\mathcal{D}^{(j)}$ can be interpreted as if it is trained to perform a specific task induced by this data set. Therefor it is called *Single Task Learning* (Caruana (1997)).

Multitask learning was first introduced by Caruana (1997), who showed that when learning algorithms are trained on different but related tasks, they share knowledge and are able to learn from each other. The central idea of multitask learning is sharing what is learned by different tasks while these tasks are trained in *parallel*. This is explained via the concept of *inductive bias*.

An inductive bias is anything that causes an inductive learning algorithm to prefer some set of parameters, or hypotheses, over another hypothesis. When the training of a model on a certain task is influenced by tasks other than the main task, these other tasks may serve as bias. This information exchange between the different models results into an improved generalization performance. Caruana tested this idea on neural networks (MLPs) that share the hidden layer for different tasks. But in the end, the goal was to test the algorithm on its main task.

The idea to use this approach on ensemble models was investigated by Bakker and Heskes (2003) where the goal is to train different models on the same task and the knowledge sharing is achieved through model clustering. An ensemble model is created on top of these submodels found by the multitask learning scheme.

The central aspect in the multitask learning is the information exchange between the different models. This causes that the generalization performance increases. In our approach we induce this information exchange by coupling of the different submodels. This idea is inspired on the idea of Coupled Local Minimizers (CLM) introduced in (Suykens et al. (2001)), where the strategy has been proposed as a new way for solving non-convex optimization problems. By coupling of the different trajectories of a multi-start optimization method, very good local minima of the non-convex optimization problem are found in an efficient way. In certain ill-conditioned problems it has been illustrated that the coupling mechanism may serve as a regularization mechanism by itself (without having any regularization term in the cost function). During the optimization process there is a continuous information exchange between the different multi-start optimizers through the coupling (as illustrated for state vector synchronization of search and learning processes). This methodology was applied to the training of neural networks. The CLM approach improved the generalization performances of the neural networks even without regularization. The coupling between the models acts as information exchanging procedure. In this way the coupling acts as a collective intelligence among the different models. Notice that the training of the different models is done in parallel. It is this idea of coupled learning models that is further studied in the following subsections.

4.1 Ensemble Learning through Coupling

As explained in the previous Section, it is our goal to increase the generalization performance of the ensemble model through coupling of the individual learning processes of the

submodels. Since an ensemble model is constructed on top of the collection of submodels $\{\hat{f}^{(j)}\}_{j=1}^{q}$ the performance of the ensemble model may increase if the individual submodels have a better performance on their individual task. Instead of constructing an ensemble model out of Single Task Learners, we now construct an ensemble with coupling of the training processes of the submodels. Since each of the individual models $\hat{f}^{(j)}$ is trained on its individual data set $\mathcal{D}^{(j)}$, this can be understood as different tasks that have to be learned. The coupling between the individual models will result into a multitask learning scheme. But, how will these models be coupled?

During the parallel training of the individual models, a distance measure between the models will be monitored. Based on this distance measure, corrections on the minimization of the individual cost functions are made. Our experience about a good distance measure between the models is in correspondence with the results of Bakker and Heskes (2003). The 'natural' distance function based on the model outputs rather than a measure based on the model parameters should be used to couple the submodels. Each of the q submodels is trained by minimizing a cost function. Since training sets $\mathcal{D}^{(j)}$ are all sampled from the same distribution, we expect the submodels to give similar results on the same input $\mathbf{x}$. This can be used as an extra coupling between the submodels. We will use this constraint as an extra *output coupling* between the models on a predefined *coupling set* $\mathcal{D}_c = \{\mathbf{x}_i^c\}_{i=1}^{N_c}$ of size N_c. This means that during training we will demand that the neighboring models are found with as additional objective

$$\min_{\mathbf{x}_i^c \in \mathcal{D}_c} \sum_{i=1}^{Nc} \left\| \hat{f}^{(j)}\left(\mathbf{x}_i^c\right) - \hat{f}^{(j+1)}\left(\mathbf{x}_i^c\right) \right\|_2^2 \ \forall j.$$

Notice that we don't need the output values of the coupling points. This means that we have the freedom to chose the coupling set points. We consider three schemes of coupling sets $\mathcal{D}_c$:

- the coupling set $\mathcal{D}_c$ is a subset of the training set,
- the coupling set $\mathcal{D}_c$ is a randomly generated in the input space,
- the coupling set $\mathcal{D}_c$ is the test set. In some applications we have the test set in advance. We can use this information during the training without actually knowing the output value. In this way the coupling set will act as Transductive Learning set. By doing this we achieve a way of *Transductive Learning* (Vapnik (1998), Joachims (1999)).

4.2 Parameterized Kernel Models with Coupled Learning Processes

When we tested the idea of coupled learning on the parameterized kernel models, it turned out the models as defined in (2) are too stringent. In this way the coupling between the models only has minor effects. In order to construct models with more degrees of freedom, we consider an *overparameterization* of our individual submodels. Each submodel j is constructed based on a Gram matrix on the corresponding training set $\mathcal{D}^{(j)}$ and on the *adjoining subsets* $\mathcal{D}^{(j-1)}$ and $\mathcal{D}^{(j+1)}$. The submodels have the form

$$\hat{f}^{(j)}\left(\mathbf{x}; \tilde{\alpha}^{(j)}\right) = \tilde{\alpha}^{(j)T} \tilde{\mathbf{k}}^{(j)}\left(\mathbf{x}\right),$$

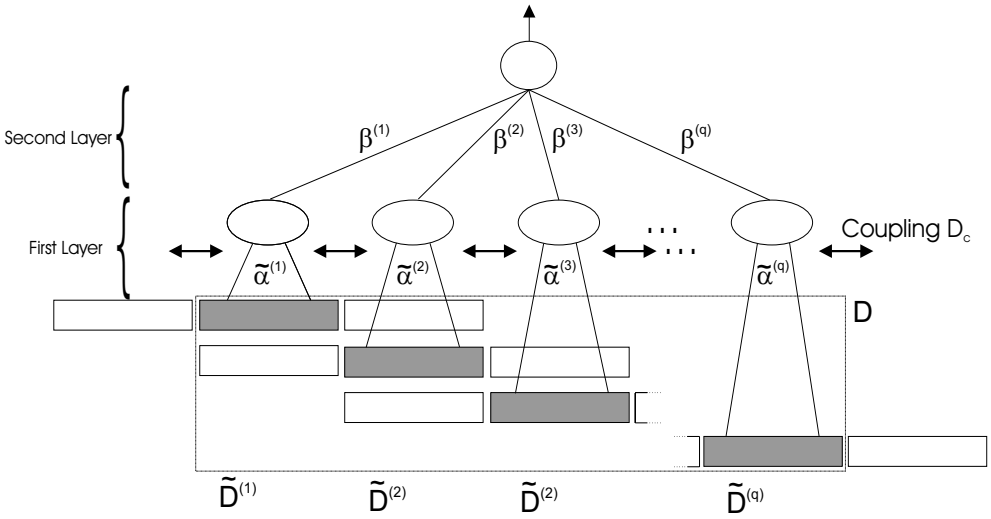

Figure 2: Architecture of the Coupled Ensemble Model.

where $\tilde{\mathbf{k}}^{(j)}(\mathbf{x}) = [k(\mathbf{x}_1^{(j-1)}, \mathbf{x})...k(\mathbf{x}_g^{(j-1)}, \mathbf{x})\ k(\mathbf{x}_1^{(j)}, \mathbf{x})...\ k(\mathbf{x}_g^{(j)}, \mathbf{x})\ k(\mathbf{x}_1^{(j+1)}, \mathbf{x})\ldots k(\mathbf{x}_g^{(j+1)}, \mathbf{x})]^T$ and $\tilde{\alpha}^{(j)} \in \mathbb{R}^{3g\times 1}$ (without loss of generality we consider here $g = g_1 = ... = g_q$). This is done e.g. in a circular way, such that $\mathbf{x}_i^{(0)} = \mathbf{x}_i^{(q)}$ and $\mathbf{x}_i^{(q+1)} = \mathbf{x}_i^{(1)}$.

For each subset $\mathcal{D}^{(j)}$ we consider now minimization of

$$\min_{\alpha^{(j)}} \frac{1}{2} \left\| \tilde{K}^{(j)} \tilde{\alpha}^{(j)} - y^{(j)} \right\|_2^2 + \frac{1}{2\gamma} \left\| \tilde{\alpha}^{(j)} \right\|_2^2,$$

where $\tilde{K}^{(j)} = [K^{(j,j-1)}\ K^{(j,j)}\ K^{(j,j+1)}]$ and $K^{(i,j)} \in \mathbb{R}^{g\times g}$ is a kernel matrix built on the subsets $\mathcal{D}^{(i)}$ and $\mathcal{D}^{(j)}$ defined by $K_{kl}^{(i,j)} = k(\mathbf{x}_k^{(i)}, \mathbf{x}_l^{(j)})$, $k, l = 1, ..., g$ and $(\mathbf{x}_k^{(i)}, y_k^{(i)}) \in \mathcal{D}^{(i)}$ and $(\mathbf{x}_l^{(j)}, y_l^{(j)}) \in \mathcal{D}^{(j)}$. We see that similar to (1), this cost function is also a combination of a squared loss error minimization and a regularization term which can be controlled by the parameter γ. The final architecture has a similar two layered structure and is shown in Figure 2.

4.3 The Training Process

As explained in the previous Section we will use the idea of Coupled Local Minimizers (CLM) to achieve multitask learning via coupling. The coupling of the different submodels is done by imposing an extra constraint in the optimization process. In Section 3 we explained how an ensemble of Single Task Learners can be trained by minimizing the sum of the individual cost functions. The CLM method suggests now to couple these individual models by imposing additional coupling constraints in the optimization process. An important point hereby is that we don't need the output values at the coupling points. In the same way as given in Section 3, we now have the ability to create an ensemble of these models with a

circular coupling via

$$\min_{\tilde{\alpha}^{(j)},e_i^{(j)}} \frac{1}{2}\sum_{j=1}^{q}\left\|\tilde{K}^{(j)}\tilde{\alpha}^{(j)}-y^{(j)}\right\|_2^2+\frac{1}{2\gamma}\left\|\tilde{\alpha}^{(j)}\right\|_2^2+\frac{\nu}{2}\sum_{j=1}^{q}\sum_{i=1}^{N_c}\left(e_i^{(j)}\right)^2$$
$$\text{s.t. } e_i^{(j)}=\tilde{\alpha}^{(j)T}\tilde{\mathbf{k}}^{(j)}\left(\mathbf{x}_i^c\right)-\tilde{\alpha}^{(j+1)T}\tilde{\mathbf{k}}^{(j+1)}\left(\mathbf{x}_i^c\right),$$
$$x_i^c\in\mathcal{D}_c; i=1,....,N_c; j=1,...,q.$$

The coupling strength can be adjusted by the coupling parameter ν. The constraints are added to the problem in a similar fashion as in least squares SVMs (Suykens et al. (2002)). If we put $\nu=0$ we end up with a model similar to (3). When $\nu>0$ a synchronization between the models on the predefined coupling set is achieved. Notice that if $\nu<0$ the models will be de-synchronized. This last option should be used with care since it can violate the convexity of the loss function. Depending on the value of ν, the matrix involved in the resulting linear system may become singular.

As mentioned, in the case that $\nu>0$ the optimization problem has a unique solution due to its convexity. This is an important property compared to the results on multitask learning with Neural Networks where one typically has a non-convex optimization problem. The solution of the convex problem can be found via the Karush-Kuhn-Tucker optimality conditions. Therefor we take the Lagrangian

$$\mathcal{L}\left(\tilde{\alpha}^{(j)},e_i^{(j)},\mu_i^{(j)}\right)=\frac{1}{2}\sum_{j=1}^{q}\left\|\tilde{K}^{(j)}\tilde{\alpha}^{(j)}-y^{(j)}\right\|_2^2+\frac{1}{2\gamma}\left\|\tilde{\alpha}^{(j)}\right\|_2^2+\frac{\nu}{2}\sum_{j=1}^{q}\sum_{i=1}^{N_c}\left(e_i^{(j)}\right)^2$$
$$-\sum_{j=1}^{q}\sum_{i=1}^{N_c}\mu_i^{(j)}\left\{\tilde{\alpha}^{(j)T}\tilde{\mathbf{k}}^{(j)}\left(\mathbf{x}_i^c\right)-\tilde{\alpha}^{(j+1)T}\tilde{\mathbf{k}}^{(j)}\left(\mathbf{x}_i^c\right)-e_i^{(j)}\right\}.$$

The Karush-Kuhn-Tucker conditions are given by

$$\nabla_{\alpha^{(j)}}\mathcal{L}=\left(\tilde{K}^{(j)T}\tilde{K}^{(j)}+\frac{1}{\gamma}I\right)\tilde{\alpha}^{(j)}-2y^{(j)}\tilde{K}^{(j)}$$
$$+\sum_{i=1}^{N_c}\mu_i^{(j-1)}\mathbf{k}^{(j)}\left(\mathbf{x}_i^c\right)-\sum_{i=1}^{N_c}\mu_i^{(j)}\mathbf{k}^{(j)}\left(\mathbf{x}_i^c\right)=0,\forall j=1,...,q,$$
$$\nabla_{e_i^{(j)}}\mathcal{L}=\nu e_i^{(j)}+\mu_i^{(j)}=0,\forall j=1,...,q,\forall i=1,...,N_c,$$
$$\nabla_{\mu_i^{(j)}}\mathcal{L}=\alpha^{(j)^T}\tilde{\mathbf{k}}^{(j)}\left(\mathbf{x}_i^c\right)-\alpha^{(j+1)^T}\tilde{\mathbf{k}}^{(j+1)}\left(\mathbf{x}_i^c\right)-e_i^{(j)}=0,\ \forall i=1,...,N_c.$$

After some simple matrix algebra this can be rewritten as

$$\left(\tilde{K}^{(j)T}\tilde{K}^{(j)}+\frac{1}{\gamma}I+\nu G^{(j,j)}\right)\tilde{\alpha}^{(j)}-\nu G^{(j,j-1)}\tilde{\alpha}^{(j-1)}-\nu G^{(j,j+1)}\tilde{\alpha}^{(j+1)}$$
$$=2\tilde{K}^{(j)T}y^{(j)},\ \forall j=1,...,q,$$

with $G^{(q,w)}=\sum_{i=1}^{N_c}\tilde{\mathbf{k}}^{(q)}(\mathbf{x}_i^c)\tilde{\mathbf{k}}^{(w)}(\mathbf{x}_i^c)^T$.

In order to show the links with (4) we rewrite this as

$$\begin{bmatrix} H^{(1)} & -\nu G^{(1,2)} & 0 & \cdots & 0 & -\nu G^{(1,q)} \\ -\nu G^{(2,1)} & H^{(2)} & -\nu G^{(2,3)} & & & 0 \\ 0 & -\nu G^{(3,2)} & H^{(3)} & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & & & \ddots & \ddots & -\nu G^{(q-1,q)} \\ -\nu G^{(q,1)} & 0 & \cdots & 0 & -\nu G^{(q,q-1)} & H^{(q)} \end{bmatrix} \begin{bmatrix} \tilde{\alpha}^{(1)} \\ \tilde{\alpha}^{(2)} \\ \vdots \\ \\ \tilde{\alpha}^{(q)} \end{bmatrix} = \begin{bmatrix} 2\tilde{K}^{(1)T} y^{(1)} \\ 2\tilde{K}^{(2)T} y^{(2)} \\ \vdots \\ \\ 2\tilde{K}^{(q)T} y^{(q)} \end{bmatrix},$$

where $H^{(j)} = \tilde{K}^{(j)^T} \tilde{K}^{(j)} + \frac{1}{\gamma} I + \nu G^{(j,j)}$ for $j = 1, ..., q$. The effect of the coupling manifests itself in two ways. The most noticeable are the off-diagonal blocks. These represent the coupling between the neighboring submodels. A second effect is the extra term added to the diagonal elements of the matrix. This term acts as an additional regularization term. In this way we can see that $H^{(j)}$ consists out of two regularization terms. The first term I/γ is an individual regularization for the submodel. This was also present in the uncoupled ensemble. The second term $\nu G^{(j,j)}$ is an effect of the coupling and can be seen as a *group regularization* caused by the coupling.

4.4 Towards Large Data Sets

In the introduction we already mentioned that one of the goals of ensemble methods is that one can reduce the computational and memory complexity of the algorithms towards large data sets. Since both models consist out of two separate layers that are also computed separately, will we handle them apart.

The memory complexity for the training of the first layer of the uncoupled ensemble model is $O(N^2/q)$ which is clear from (4). But since each of the parameters $\alpha^{(j)}$ can be computed separately, the memory complexity drops to $O(N^2/q^2)$.

For the first layer of the coupled ensemble models, the memory complexity increased due to the overparameterization and the coupling. The memory complexity of this model is $O(27N^2/q)$. This can be seen as follows. Each of the block matrices in the coupled ensemble formulation has a memory storage $O((3N/q)^2)$ where N is the number of data points and q the number of submodels. The factor 3 originates from the overparameterization of the models. Since we have a number of q block diagonal elements and $2q$ block off-diagonal elements, this adds up to a memory complexity of $O(3q(3N/q)^2) = O(27N^2/q)$. For large data sets the structure and sparseness of this large system (which is positive definite) can be exploited. One sees that if $q \geq 27$ the memory complexity of the coupled ensemble model is lower then other kernel model like RBF, SVM or LS-SVM which have a memory complexity [1] of $O(N^2)$.

A second solution is to transform the linear system into a block tridiagonal matrix. For this purpose, we eliminate here the coupling between the outer models. On tests this effect did not influence on the performance. The solution of the block tridiagonal system can be

1. This memory complexity holds for direct methods. Better memory efficiency can be achieved by using iterative methods like conjugate gradient algorithms.

found by a *block LU factorization*. Since the matrices $H^{(i)}$ are all non-singular[2], this LU factorization is proven to converge (Golub and Loan (1989)). As will be shown, this divides the original problem into smaller sub-problems.

In general the *block LU factorization* is defined as follows. Let a linear system $Ax = b$ to be solved have the following block tridiagonal form:

$$\begin{bmatrix} A_1 & C_1 & 0 & \cdots & 0 \\ B_1 & A_2 & C_2 & 0 & \vdots \\ 0 & B_2 & A_3 & \ddots & 0 \\ \vdots & 0 & \ddots & \ddots & C_{q-1} \\ 0 & \cdots & 0 & B_{q-1} & A_q \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_q \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_q \end{bmatrix}.$$

According to Golub and Loan (1989) the LU-factorization of this matrix A is given by

$$A = \begin{bmatrix} I & 0 & & \cdots & 0 \\ L_1 & I & 0 & & \vdots \\ 0 & L_2 & I & \ddots & \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & L_{q-1} & I \end{bmatrix} \begin{bmatrix} U_1 & C_1 & 0 & \cdots & 0 \\ 0 & U_2 & C_2 & 0 & \vdots \\ & 0 & U_3 & \ddots & 0 \\ \vdots & & \ddots & \ddots & C_{q-1} \\ 0 & \cdots & & 0 & U_q \end{bmatrix}.$$

The unknown submatrices L_i and U_i can be computed via the following iterative process:

$U_1 = A_1$
for $i = 2 : q$
    $L_{i-1} = B_{i-1}U_{i-1}^{-1}$
    $U_i = A_i - L_{i-1}C_{i-1}$
end

The procedure is defined as long as the matrices U_i are nonsingular. After this factorization, the solution of the linear system follows from forward and backward substitution:

$y_1 = b_1$
for $i = 2 : 1 : q$
    $y_i = b_i - L_{i-1}y_{i-1}$
end
$x_q = U_q^{-1}y_q$
for $i = q - 1 : -1 : 1$
    $x_i = U_i^{-1}\left(y_i - C_i x_{i+1}\right)$
end

Notice that all the matrix inversions can be avoided by rewriting it into linear systems that can be solved by appropriate methods.

2. If $\nu > 0$ then the matrix $H^{(j)}$ is a positive definite matrix ($\tilde{K}^{(j)^T}\tilde{K}^{(j)} + \frac{1}{\gamma}I$ is positive definite and $G^{(j,j)}$ which is a sum of rank one matrices, is also positive definite).

At the level of the second layer also improvements for large data sets are possible. As explained in (3), the optimal weights $\{\beta^{(j)}\}_{j=1}^{q}$ can be found by solving (6). The solution to this problem is in most cases non-sparse which means that all the values of $\beta^{(j)}$ are non-zero. However, by imposing extra constraints $\beta^{(j)} \geq 0, \forall j$, one obtains (Bishop (1995), Perrone and Cooper (1993)):

$$\min_{\beta} \beta^T S \beta \quad \text{such that} \quad \begin{cases} \sum_{j=1}^{q} \beta^{(j)} = 1 \\ \beta^{(j)} \geq 0,\ j = 1, ..., q. \end{cases} \tag{7}$$

This is a quadratic optimization problem of dimension q. Since the error covariance matrix is positive definite, the solution of this QP problem is global and unique. The advantage is that the solution now becomes sparse which simplifies the evaluation at new points

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{\mathrm{SM}} \beta^{(j)} \sum_{p=1}^{g} \alpha_p^{(j)} k(\mathbf{x}_p^{(j)}, \mathbf{x}),$$

where SM denotes the number of $\beta^{(j)}$ output weights that are different from zero. Similar to the support vector machine literature (Vapnik (1998), Cristianini and Shawe-Taylor (2000)) these models can be viewed as *support models*. This sparseness sometimes also leads to a better performance as is illustrated in the next Section.

5. Experiments

We illustrate coupled learning of regularization networks on examples of regression and classification.

5.1 Regression

5.1.1 Toy Example: Sine Function

In this example we give a comparison between models constructed via an uncoupled ensemble (as defined in (3)) versus a coupled overparameterized ensemble (as defined in Section (4.2)) on a sine function estimation problem. We did the training of the submodels on 4 disjoint subsets locally sampled in the input space. Although this local sampling of the input space is usually not possible in practice, we will use it here for demonstration purposes in order to visualize the effect of coupling. The coupling points are uniformly spread over the x-axis. Although subsets of different sizes can be taken, we assume here that the subsets are of equal size $g = N/q$. One can clearly see in Figure 3 that the individual models of the coupled overparameterized system have improved performances in the regions where they are not trained. Compared with the individual models of the uncoupled ensemble, the latter clearly shows a bad performance in regions where there are no training points.

Furthermore, we constructed a weighted combination based on these ensembles. We see that the performance of the coupled overparameterized models is better than for the uncoupled ensemble. This example illustrates that the coupling improves the generalization performance, which is caused by the information exchange between the different submodels during the learning process. The reason is that averaging of individual models that have

better generalization performance will probably lead to a better performance of the ensemble model.

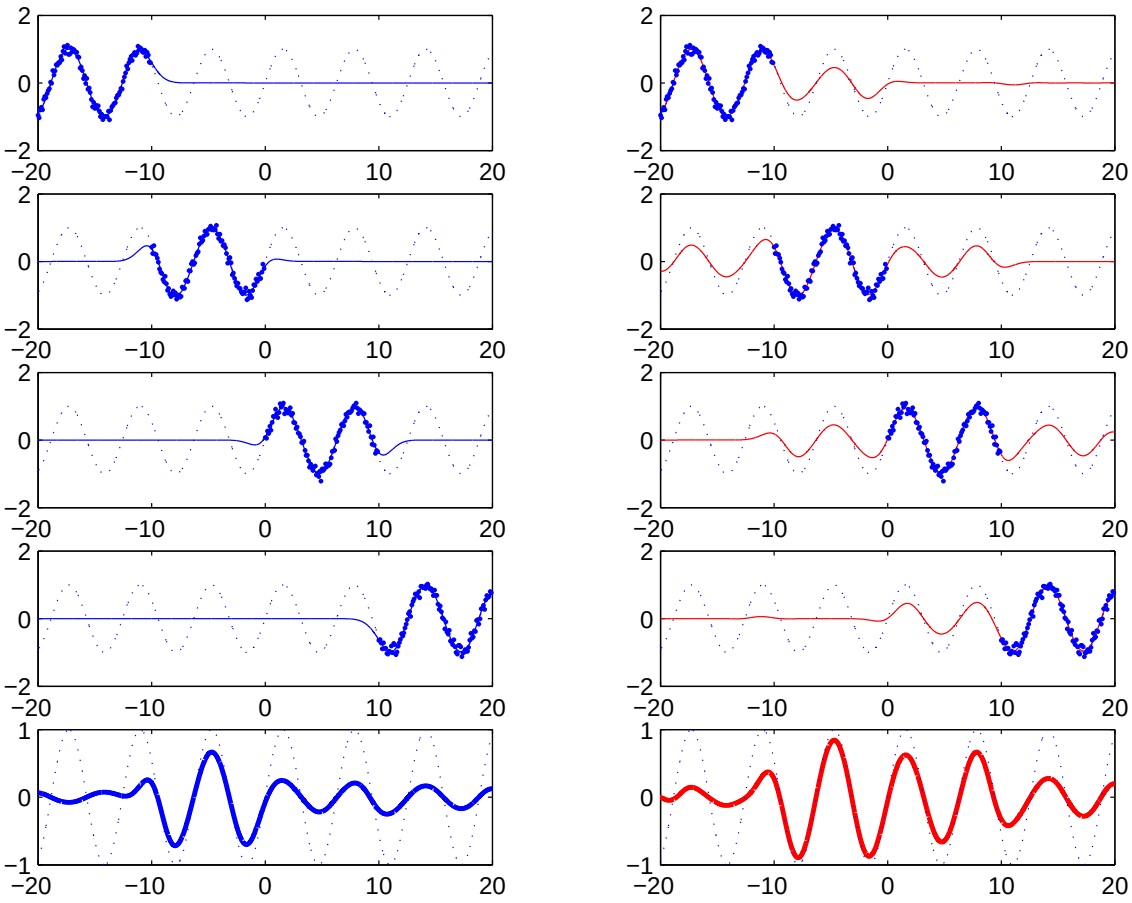

Figure 3: This figure shows 4 submodels of the uncoupled (left) and coupled (right) ensembles for a sine function (true function: dotted) with their corresponding training sets. The bottom figures illustrate the combined models of both ensembles (uncoupled (left); coupled (right)). This shows the effect and improvement obtained by coupling of the learning processes for the individual submodels.

5.1.2 Boston Housing Data Set

The Boston housing data set is a multivariate regression data set of 506 cases in 14 attributes. It has two prototasks: NOX, in which the nitrous oxide level is to be predicted, and price MEDV, in which the median value of a home is to be predicted. In these tests we do a regression on the data sets with the models constructed via a simple ensemble and via a coupled overparameterized ensemble on randomly sampled subsets. We trained on 400 training points and use the remaining 106 points as test set and this for different randomizations.

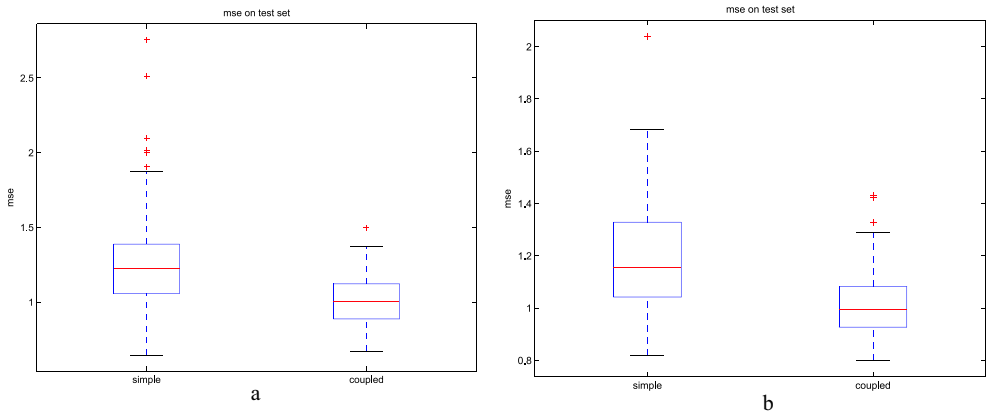

Figure 4: This figure shows the distribution of the mse performances on the Boston Housing data set of uncoupled (left boxplots) and coupled (right boxplots) ensemble models after 100 randomizations. Part *a* shows the performance on the NOX prediction and part *b* shows the performance on the MEDV prediction. In both cases we see the improvement of the coupled ensembles with a lower mean and a smaller variance. These experiments are done for a fixed value of $\nu = 1$.

As coupling set we use either 10 % of the total training data, randomly chosen, without the y-values or we use the test set as coupling set. In the latter case we have a transductive learning scheme. In both cases we used the Gaussian radial basis function kernel. The used hyperparameters are found by cross-validation for the individual models. For the NOX we used the hyperparameters $(\gamma, \sigma^2) = (20.67, 15.44)$ and for the MEDV $(\gamma, \sigma^2) = (81.19, 12.19)$. The coupling parameter is kept constant at $\nu = 1$.

To test the difference in performance between the methods, we computed the mean squared error (mse) on a test set of both the MEDV and NOX estimations. We compared the distributions for 100 randomizations (see Figure 4) based on a Wilcoxon ranksum test. This indicates a statistically significantly difference between the averages of the two groups.

The results are summarized in Table 1. In this table we show the mse performances on a test set. We use ensemble models based on collections or learning algorithms of either size 8 or 16 (the number of models is indicated by the second number in Table 1, example NOX16 is a ensemble model based on 16 individual models on the NOX prediction). We give the mean (mea), median (mea) and variance (var) of these distributions after randomization. In the last two columns the result of the Wilcoxon ranksum test are given: h indicates the test that the two distributions are different and p is the corresponding p-value. If h equals 1 there is a statistically significant difference with 95% probability.

From these results we can conclude that the coupling of the models with transductive inference leads to statistically significantly better performance in all the shown cases. If the coupling set is randomly chosen, the performances are not significantly better for the coupled ensembles. But notice that only in one of the shown tests, the NOX prediction with 16 models, the performance of the coupled ensemble is worse.

	Uncoupled Ensemble			Coupled Ensemble				
	mea	med	va	mea	med	va	h	p
NOX,8	1.20	1.16	4.76e-2	1.31	1.20	6.78e-2	0	6.74e-2
NOX,8,T	1.12	1.16	4.49e-2	1.01	1.00	1.33e-2	1	8.82e-11
NOX,16	1.92	1.59	2.85e-2	2.49	1.78	7.17e-2	1	3.20e-2
NOX,16,T	2.03	1.61	7.24e-2	1.05	1.02	1.74e-2	1	0
MEDV,8	1.30	1.21	7.20e-2	1.34	1.23	8.27e-2	0	7.26e-1
MEDV,8,T	1.28	1.23	6.00e-2	0	1.00	3.02e-2	1	4.37e-11
MEDV,16	2.11	1.58	5.61e-1	2.23	1.55	4.71e-1	0	6.04e-1
MEDV,16,T	2.28	1.74	9.17e-1	1.14	1.10	4.41e-2	1	0
NOX, LS-SVM	1.84	1.76	1.19e-1					
MEDV, LS-SVM	1.33	1.31	6.08e-2					

Table 1: This table shows the mse performances (Boston housing data problem) on a test set. We use ensemble models based on collections of learning algorithms of either size 8 or 16 (the number of models is indicated by the second number in Table 1, example NOX16 is an ensemble model based on 16 individual models on the NOX prediction). We show the mean (mea), median (mea) and variance (var) of these distributions after 100 randomization. In the last two columns the result of a Wilcoxon ranksum test are given: h indicates the test that the two distributions are different and p is the corresponding p-value. In the last two rows the mse performance of a standard LS-SVM is given. Again the mean (mea), median (mea) and variance (var) of these distributions are given after 100 randomization.

In Figure 4 we show the distributions of the mse performances of ensemble models created on 8 individual models for both the NOX prediction as the MEDV. One sees the performance of the uncoupled and coupled ensembles. Notice that besides the fact that the mean mse of the coupled ensemble is always better, also the variance is smaller. We observe that this property holds for the majority of experiments in Table 1. An intuitive explanation of this behavior is the following.

By coupling of the submodels within the ensemble, we prevent the fact that some of the models give a very bad prediction. These *outlier models* can have a very disturbing influence on the overall performance of the averaging models. This is indicated by a big variance. In statistics it is also known that taking an average is very sensitive to outliers. By coupling of the individual models, we prevent that some of the models of the ensemble have a performance that deviates too much from the other models. As long as the majority of the elements of the ensemble give good predictions, they will correct the outlier models. As we can notice this results into a smaller variance.

Although it is not our purpose in this paper to make an extensive benchmarking with other learning models, we have also tested the performance of a standard LS-SVM with the same hyperparameters (γ, σ^2) trained on the whole data set of 400 points and tested on a test set of 106 points. After 100 randomizations we get the results gives in the last two rows of Table 1. We observe that the ensemble methods that we use here, give performances

comparable to the standard LS-SVM. For the coupled ensemble models with transductive inference, we notice that the performances are better.

5.1.3 The Role of the Coupling Parameter

In order to show the effect of the coupling parameter ν, we have done the following test. We computed the mse on a test set for different values of ν while keeping the other hyper-parameters constant. To overcome worst-case effects we randomized the test and training set 20 times for each value of ν.

As we explained previously, by the coupling parameter ν we can adjust the coupling strength. If we put $\nu = 0$ we end up with a model similar to (3); $\nu > 0$ indicates a synchronization coupling between the models; $\nu < 0$ indicates that the models will be *de-synchronized*. This last option may cause numerical problems because the convexity of the problem formulation may be violated.

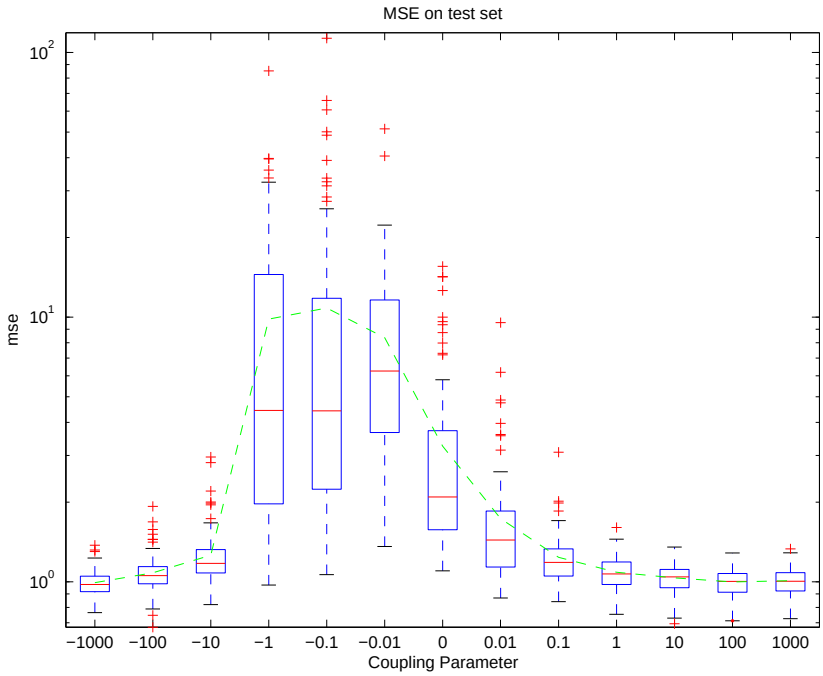

Figure 5: This figure shows the mse performance on a test set for the Boston Housing data set with different values of the coupling parameters ν, after randomization. The dotted line is a fitting between the means of each randomization experiment. The mse values are plotted on log-scale.

As we can observe in Figure 5, the coupling with $\nu > 0$ improves the performance of the ensemble models in comparison with the uncoupled situation. This indicates that the submodels are learning from each other. The inductive bias of the individual algorithms improves the overall performance. Notice also that there is a clear reduction in the variance of the models.

For small negative coupling $\nu < 0$, the experiment shows that the performance improves, not only in average performance but also in variance. Surprising is the behavior for larger negative values of ν. Apparently the performance can also improve when we the models are *de-synchronized.* We do not have a clear explanation for this behavior. But, we want to stress that high value of ν should be used carefully, as we will explain.

Remember that the main goal is to create averaging models based on individual models that are data driven. Each individual model is fitted to its corresponding data set. By synchronizing them, each of them is slightly modified by the information exchange, also mentioned as the inductive bias. Now, if we increase the size of ν, the coupling will gain on influence. In the limit, the coupled ensemble models will focus more on synchronizing the individual models than creating data driven submodels with a small correction achieved by coupling. Therefore, in the focus of this paper, we advice the take the coupling not too large.

5.2 Classification

Although all the models discussed in the previous Section are intended for regression tasks, we will also use them for classification. This approach works well as was shown for LS-SVMs (see Suykens et al. (2002)). Because the output of a classification task is a binary variable, we will have to adapt our models. Therefor we use for the evaluation of new inputs:

$$\hat{f}\left(\mathbf{x}\right) = \text{sign}\left(\sum_{j=1}^{\text{SM}} \beta^{(j)} \text{sign}\left(\sum_{p=1}^{g} \alpha_p^{(j)} k(\mathbf{x}_p^{(j)}, \mathbf{x})\right)\right).$$

Training of the first layer of the uncoupled as well as the coupled ensemble remains the same. But for the training of the second layer we will always use the sparse formulation (7), with number of support models $\text{SM} \leq q$. This sparse solution also improves the performance as observed in the experiments.

5.2.1 Tic-Tac-Toe Data Set

The Tic-Tac-Toe Endgame database is a UCI (Blake and Merz (1998)) data set contributed by Aha, encoding the complete set of possible board configurations at the end of the tic-tac-toe games. The target concept is "win for x" where "x" is assumed to have played first. The data set consists of 958 observations with 9 attributes. Note that the 16 records that contained missing values were removed from the data set. The first 638 observation were used as training set while the last 320 as the test set. This data set is known to be a separable by a highly nonlinear model. We employ the Gaussian RBF kernel with the hyperparameters $(\gamma, \sigma^2) = (9.46e1, 9.96)$. In all tests on this data set, we used an ensemble consisting of 11 submodels and the coupling set is chosen as test set (transductive setting).

As a first test we studied the effect of the coupling parameter ν in this classification setup. We randomized the test and training set 20 times for a whole range of values of ν. For each of those values we measure the misclassification rate on a test set. The result of this experiment is shown in Figure 6. In comparison with the regression case we now see a clear minimum in the region 0.01 and 1. Since this is for $\nu > 0$, this indicates that coupling as synchronization between the models has the best performance here.

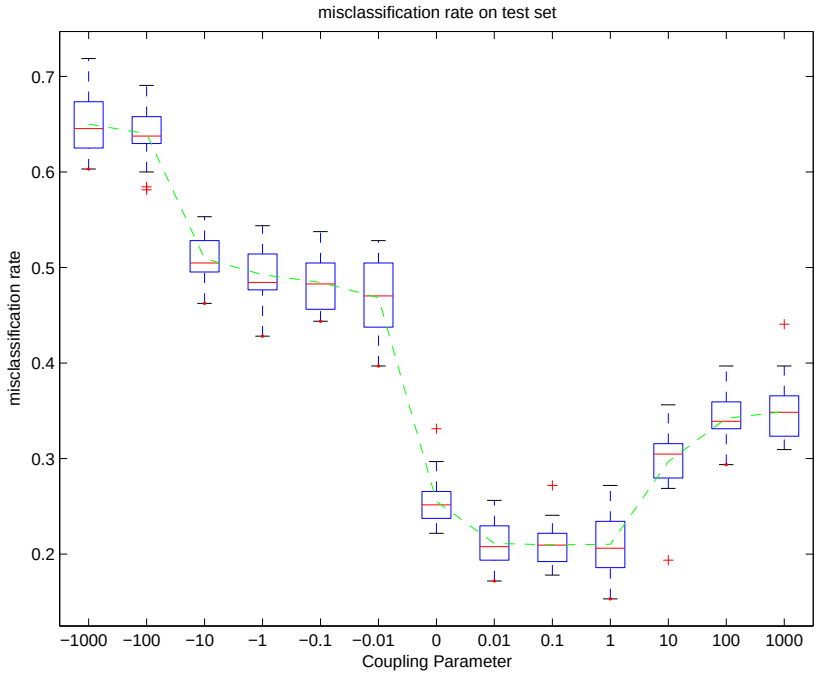

Figure 6: This figure shows the effect of the coupling parameter on the misclassification performance of the Tic-Tac-Toe classification example. For each of the values of ν we have done 20 randomizations. The dotted line is a fitting between the means of each randomization experiment.

If we now compare the uncoupled and the coupled ensemble models constructed based on 11 models, with $\nu = 0.1$ (corresponding to the minimum of the previous test) and with the same hyperparameters, we see the following results. There is a significant improvement of the coupled ensemble compared to the uncoupled ensemble. This again is shown via a Wilcoxon ranksum test after 100 randomizations. The mean, median and variance of these tests together with the p-value is given in Table 2. In a last test we show the performance of the ensemble model in relation to the individual submodels. Therefor we have taken one of the models as trained in the previous randomization experiment with the optimal hyperparameter and coupling parameter as indicated. We constructed the receiver operating characteristic (ROC) curves for the submodels of the ensemble together with the ROC curve of the ensemble model. In Figure 7 the results are shown where it is clear that the ensemble performance is better than the mean performance of the elements of the ensemble. If we compare the area-under-the-curve (AUC) of the ensemble model, we see that for the uncoupled ensemble the AUC is 0.77 with standard error $SE = 0.03$, for the coupled ensemble this is 0.86 with standard error $SE = 0.02$. This is an improvement in performance of nearly 10% for the coupled ensemble.

In a further study of this experiment we show the weights of the individual submodels of the ensemble as found from (7). In the previous chapter it was already explained that by using (7) a sparse solution for the second layer is obtained. In Figure 8 we show the

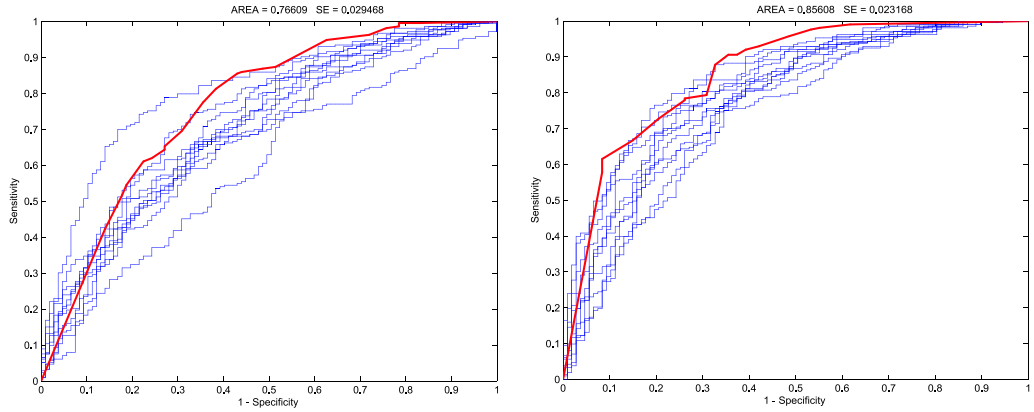

Figure 7: This figure shows the ROC performance of the Uncoupled (left) and the Coupled (right) ensemble on the Tic-Tac-Toe data set. Each figure represents the ROC curve of each of the submodels of the ensemble (thin lines) together with the ROC curve of the corresponding ensemble model (thick line). At the top of each figure the area-under-the-curve together with the standard error is indicated. One can see an improvement of nearly 10% by taking the Coupled ensemble.

	Uncoupled Ensemble			Coupled Ensemble				
	mea	med	va	mea	med	va	h	p
TTT,11,T	0.25	0.25	7.09e-4	0.21	0.21	8.37e-4	1	0
ACR,10,T	0.16	0.16	5.08e-4	0.15	0.14	5.08e-4	1	9.8e-5
ADULT,100,T	0.27	0.28	7.92e-4	0.24	0.24	8.34e-4	1	2.7e-3

Table 2: Misclassification rates on a test set (Tic-Tac-Toe (TTT), Australian Credit Card Data Set (ACR) and the Adult Data Set (ADULT)). The number of models is indicated by the second number in Table 1, example TTT11 is an ensemble model based on 11 individual models on, the TTT prediction. We give the mean (mea), median (mea) and variance (var) of these distributions after 100 randomization. In the last two columns the result of a Wilcoxon ranksum test are given: h indicates the test that the two distributions are different and p is the corresponding p-value.

values of the weights as a result of this optimization problem. In the uncoupled as well as the coupled ensemble case only 4 out of 10 models have a weight different from zero.

5.2.2 Australian Credit Card Data Set

The Australian Credit Card data set is a UCI data set donated by Quinlan and is one of the Credit Approval Databases which were used in the Statlog project. There 690 observations in this data set with six numerical and eight attributes. The optimal hyperparameters for the Gaussian RBF kernel are $\left(\gamma, \sigma^2\right) = (9.03e2, 12.15)$. All the ensembles were based on 10 submodels. Similar to all the previous experiments did we check the influence of the

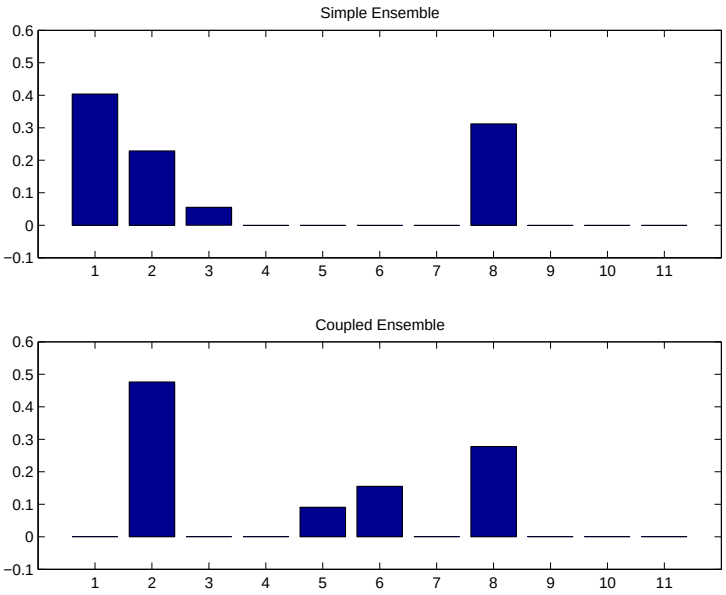

Figure 8: Sparseness of the solution for the parameter vector of the second layer (Tic-Tac-Toe data set). In the uncoupled (top) ensemble as well as the coupled ensemble (bottom) we see that only 4 out of 10 models have a weighting different from zero.

coupling parameters ν by a randomization test for different values of ν. The behavior was similar to the Tic-Tac-Toe experiment with a minimum in $\nu = 1$.

For this optimal value of ν we then did a randomization test to check the differences between a coupled and uncoupled ensemble. The results are given in Figure 9 and the numerical values of the distributions together with a result of the Wilcoxon ranksum test are given in Table 2. Again we may conclude that the performance of the coupled ensemble is significantly better compared with the uncoupled ensemble. If one compares the differences between the mean and median of both distributions, one sees that it is small. However, there is a difference in the distribution as indicated by the ranksum test. If we take a closer look at Figure 9 we see that the uncoupled ensemble counts some outliers with a bad performance. It are these results of the misclassification tests that cause the difference. This behavior was often noticed in our experiments (see also Figure 4). The coupling between the elements of the ensemble causes the models to be more robust against outlier performances.

5.2.3 Adult Data Set

The Adult data set is a UCI data set donated by Kohavi. It involves the prediction whether income exceeds 50,000 dollars a year based on census data. The original data set consists out of 48,842 observations each described by six numerical and eight categorical attributes. All the observations with missing values were removed from consideration. To show the use of our method towards larger data set we took a subset of 8,000 points. We used another 500 points as test set. The optimal hyperparameters for the Gaussian RBF kernel are

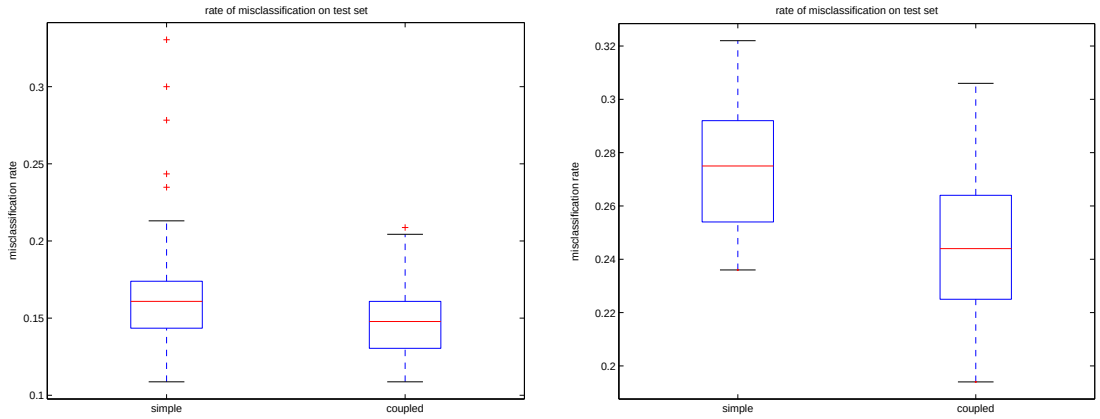

Figure 9: This figure shows the distribution of the misclassification rates on the Australian Credit Card data set (figure-left) of uncoupled (right) and the coupled (left) ensemble models after 100 randomizations. One sees the improvement of the coupled ensemble with a lower mean and smaller variance. Both experiments are done with a $\nu = 1$. (Figure-right) similar experiments and results on the Adult data set.

$(\gamma, \sigma^2) = (1.11, 3.00e1)$ and the coupling parameters $\nu = 1$. All the experiments used the transductive setting in which the test set is used as coupling set for the coupled ensemble model. To study the behavior of our models with large ensembles we constructed the ensemble on 100 submodels. In this way, the needed memory complexity is $O(27N^2/100)$ which means a reduction by a factor 4 compared to the memory usage of a traditional kernel model (in a worst case scenario).

In a first experiment on this data set, we used (7) to compute the weights of the second layer. As explained in the previous Section, this results into a sparse solution. Only about 5% of the submodels had a weight different from zero. Computationally, this means a big advantage towards evaluation at new points with this ensemble model. However, after a randomization experiment, we did not observe a statistically significant difference of the misclassification performance between the coupled and uncoupled ensemble models on this problem. The explanation behind this is the following. In the previous Sections we mentioned that coupling results into a reduction of the 'outlier' models. Those submodels of the ensemble who intent to have a bad performance are corrected by the other models. In this way, the overall performance of the ensemble model increases. But, the computation of weights of the second layer based on (7) does a similar thing. Also here the submodels with a bad performance are pruned away. If, like in this case, only 5% of the submodels have a weight different from zero, the effects of the coupling is less obvious. To check this we did another experiment.

In a second experiment, we computed the weights of the second layer based on (6). In this case the ensemble model is built on all 100 submodels, so there is no pruning. In this case we see a clear distinction between the performances of the uncoupled and the coupled ensemble as is shown in Table 2. This was again verified by a Wilcoxon ranksum test after

a randomization experiment of which the results are also shown in Table 2. In Figure 9 the distributions of the misclassification performance after the randomization experiment are shown for the uncoupled and coupled ensemble. In this case where the ensemble models is built on a larger set of submodels the effect of the coupling becomes more clear.

6. Conclusions

In this paper we have shown the effect of coupling between the submodels within ensemble models. We introduced the concept of a coupling set and showed that this can be used to achieve a new way of transductive learning for regression as well as classification problems. We explained the links between multitask learning and coupling and showed that coupling can be regarded as a group regularization imposed on all the elements of the ensemble. We also demonstrated different adaptations of the learning scheme towards large data sets. In the experiments we have illustrated how the coupling between the elements of the ensemble almost always leads to improved test set performance, in comparison with uncoupled ensembles.

Acknowledgments

This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven. It is supported by grants from several funding agencies and sources: Research Council KU Leuven: Concerted Research Action GOA-Mefisto 666 (Mathematical Engineering), IDO (IOTA Oncology, Genetic networks), several PhD/postdoc & fellow grants; Flemish Government: Fund for Scientific Research Flanders (several PhD/postdoc grants, projects G.0407.02 (support vector machines), G.0256.97 (subspace), G.0115.01 (bio-i and microarrays), G.0240.99 (multilinear algebra), G.0197.02 (power islands), research communities ICCoS, ANMMM), AWI (Bil. Int. Collaboration Hungary/ Poland), IWT (Soft4s (softsensors), STWW-Genprom (gene promotor prediction), GBOU-McKnow (Knowledge management algorithms), Eureka-Impact (MPC-control), Eureka-FLiTE (flutter modeling), several PhD grants); Belgian Federal Government: DWTC (IUAP IV-02 (1996-2001) and IUAP V-10-29 (2002-2006) (2002-2006): Dynamical Systems and Control: Computation, Identification & Modelling), Program Sustainable Development PODO-II (CP/40: Sustainibility effects of Traffic Management Systems); Direct contract research: Verhaert, Electrabel, Elia, Data4s, IPCOS. Bart Hamers is a PhD student supported by the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT). Johan Suykens and Bart De Moor are an associate and full professor at KU Leuven Belgium, respectively.

References

B. J. Bakker and T. M. Heskes. Clustering ensembles of neural network models. *Neural Networks*, 2003. (in press).

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Irvine, CA: University of California, Dept. of Information and Computer Science.

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

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of SVMs for very large scale problems. *Neural Computation*, 14(5):1105–1114, 2002.

N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, 2000.

A. Elisseeff and M. Pontil. Leave-one-out error and stability of learning algorithms with applications. In J.A.K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, editors, *Advances in Learning Theory: Methods, Models and Applications*, volume 190 of *NATO Science Series III: Computer and Systems Sciences*, pages 111–125. IOS Press Amsterdam, 2003.

T. Evgeniou, L. Perez-Breva, M. Pontil, and T. Poggio. Bounds on the generalization performance of kernel machine ensembles. In *Proc. 17th International Conf. on Machine Learning*, pages 271–278, San Francisco, CA, 2000a. Morgan Kaufmann.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000b.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore MD, 1989.

B. Hamers, J.A.K. Suykens, V. Leemans, and B. De Moor. Ensemble learning of coupled parmeterised kernel models. In *Supplementary Proc. of the International Conference on Artificial Neural Networks and International Conference on Neural Information Processing (ICANN/ICONIP)*, pages 130–133, Istanbul, Turkey, 2003.

T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 200–209. Morgan Kaufmann Publishers, San Francisco, US, 1999.

D. G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *J. Chem. Metall. Mining Soc. S. Africa*, 52(6):119–139, 1951.

D. J. C. MacKay. Introduction to gaussian processes. In C. M. Bishop, editor, *Neural networks and machine learning*, volume 168 of *NATO-ASI Series F*, pages 133–165. Springer, computer and systems sciences edition, 1998.

D. Pavlov, J. Mao, and B. Dom. Scaling-up support vector machines using boosting algorithm. In *Proceedings of the International Conference on Pattern Recognition, ICPR-2000*, 2000.

M. P. Perrone and L. N. Cooper. When networks disagree: Ensemble method for neural networks. In R.J. Mammone, editor, *Neural Networks for Speech and Image processing*, pages 126–142. Chapman-Hall., 1993.

T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

R. E. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

A. Schwaighofer and V. Tresp. The Bayesian committee support vector machine. In *Proceedings of ICANN 2001*, number 2130 in Lecture Notes in Computer Science, pages 411–417. Springer Verlag, 2001.

J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

J. A. K. Suykens, J. Vandewalle, and B. De Moor. Intelligence and cooperative search by coupled local minimizers. *International Journal of Bifurcation and Chaos*, 11(8):2133–2144, 2001.

V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

M. Yamana, H. Nakahara, M. Pontil, and S. Amari. On different ensembles of kernel machines. In *European Symposium on Artificial Neural Networks (ESANN)*, pages 197–201, Bruges, Belgium, 2003.