# Boosting Expert Ensembles for Rapid Concept Recall

**Achim Rettinger***
Technical University Munich
Department of Informatics (I7), AI/Cognition Group
Boltzmannstrasse 3, D-85748 Garching, Germany
achim.rettinger@in.tum.de

**Martin Zinkevich** and **Michael Bowling**
University of Alberta
Dept. of Computing Science
Edmonton, Alberta, Canada T6G 2E8
maz@cs.ualberta.ca, bowling@cs.ualberta.ca

### Abstract

Many learning tasks in adversarial domains tend to be highly dependent on the opponent. Predefined strategies optimized for play against a specific opponent are not likely to succeed when employed against another opponent. Learning a strategy for each new opponent from scratch, though, is inefficient as one is likely to encounter the same or similar opponents again. We call this particular variant of inductive transfer a concept recall problem. We present an extension to AdaBoost called ExpBoost that is especially designed for such a sequential learning tasks. It automatically balances between an ensemble of experts each trained on one known opponent and learning the concept of the new opponent. We present and compare results of ExpBoost and other algorithms on both synthetic data and in a simulated robot soccer task. ExpBoost can rapidly adjust to new concepts and achieve performance comparable to a classifier trained exclusively on a particular opponent with far more data.

## Introduction

In a career, one often faces a series of jobs (as a grad student, postdoc, faculty member), which have similar properties but are not identical. In a robot (e.g. soccer, rugby, poker) tournament, one plays a sequence of games against a sequence of opponents. Heuristics learned by playing with one opponent are often not directly applicable to the next opponent. This imposes challenging requirements on machine learning techniques like being forced to react instantly to unknown opponents after seeing only a small number of sample situations. Research in the area of inductive transfer refers to this problem of retaining and applying the knowledge learned in one or more tasks to efficiently develop an effective hypothesis for a new task. However we don't focus on elaborate ways of knowledge retention and transfer but simply resort to the predictions of an ensemble of experts. Another well studied problem domain that deals with such setups is on-line learning where the target concept can drift over time. However, the assumptions in on-line learning are fairly weak, making the problem incredibly difficult. In this work, we make stronger assumptions about the data that coincide with playing in a tournament consisting of a sequence of games (of soccer, or chess, or polo) against different opponents. We assume that the game stays the same, thus we can assume that hypotheses learned for previous opponents are still useful. The beginning of a new game indicates a particular time when the concept will change, and thus having an algorithm that can receive the signal that there is a new game can be useful. On the other hand, games can be short and early errors can be catastrophic, so in real tournaments it is important to quickly learn about your new opponent.

In this paper we present **ExpBoost**, a technique that augments AdaBoost with the ability to utilize previously learned hypotheses. The new method is an intuitive extension to AdaBoost and it balances between resorting to using previously learned hypotheses and developing a hypothesis on the new task, relying on the empiricially and theoretically well-established techniques in AdaBoost for combining a set of hypotheses. An empirical evaluation of ExpBoost on synthetic data shows that ExpBoost can adjust quickly to new target concepts. Furthermore we present the scenario of a simulated robot soccer tournament that initiated this research. Applying ExpBoost to the robot learning task clearly outperforms existing experts and almost achieves the performance of an hypothetical optimal classifier that was solely trained on the current task with far more samples.

### The Concept Recall Problem

Research on the concept recall problem was inspired by experiments with teams of agents playing against each other in a tournament.

A tournament can be considered as a learning scenario where every new game a learning algorithm sees a sequence of $T$ labeled examples $\{(x_1, y_1) \ldots (x_T, y_T)\}$. All examples observed during games against the same opponent belong to the same target concept $c : x \to y$. Thus each concept $c^b \in \{c^1, ..., c^B\}$ is represented by samples from games played against $B$ different opponents. At the beginning of a new game the learning algorithm is confronted with a potentially different opponent and can only resort to hypotheses $h^b \in H^b = h^1, ..., h^B$ built with samples corresponding to $c^1, ..., c^B$. Considering the fact that a learned hypothe-

*The author would like to thank all members of the RoboLog project (University of Koblenz) and the RLAI group (University of Alberta). This work was financially supported by the German Academic Exchange Service.

sis $h^b$ is dependent on a specific opponent $b$ and reflects an individual target concept $c^b$ the target concept of a new opponent $c^{B+1}$ is most likely different from the known target concepts.

Situations where the target concept changes over time have been addressed as *concept drift* in the literature (see (Schlimmer & Granger 1986)). In this paper we focus on the ability of a learning algorithm to resort to expert advice from classifiers trained on previous concepts to predict the current concept $c^{B+1}$. In inductive transfer, the entire structure of each previously learned hypothesis is typically utilized; to highlight the fact that we are only using the final hypothesis learned, we call our paradigm **concept recall**.

We focus on the following objectives:

*Increase performance*: Any algorithm should perform better (or at least equally well) on samples from concept $c^{B+1}$ than any expert $h^b$ available in the ensemble.

*Learn rapidly from few samples*: The ability of recalling a concept $c^1, ..., c^B$ is especially crucial if only a few new samples from $c^{B+1}$ are available (for small $T$).

*Incorporate prior knowledge*: It is important that the learner neither only makes use of a general expert, trained on all $c^1, ..., c^B$ together, nor only utilizes data from one certain batch $c^b$, but combine different, potentially conflicting concepts in order to improve performance even if the current opponent is mostly unknown.

*Learn the new concept*: If a system wants to reliably adapt to $c^{B+1}$, it must also be able to learn previously unknown elements of this concept.

## ExpBoost

In this section we introduce a novel algorithm, called **ExpBoost**, based on the popular boosting algorithm *AdaBoost.M1* by (Freund & Schapire 1995). Boosting is a method that combines many weak classifiers in a series of $N$ steps to produce a strong classifier. A *weak classifier* is a classifier that, given a set of samples $S$, returns a hypothesis that does slightly better than random. In practice, it can be any learning algorithm, i.e. we do not require it to be able to handle samples weighted by importance or be able to take "advice" in the form of previously learned hypotheses. Like AdaBoost, we have iterations $i = 1 \ldots N$, and we maintain a distribution $D_i$ over the training set, and as the algorithm proceeds $D_i$ has higher weights on the harder samples. The weak classifier is given a set of training data $S_i$ sampled according to $D_i$ and returns a classifier $h_i^{B+1}$. At this point, we deviate from AdaBoost: we compare the performance of $h_i^{B+1}$ to $h^1 \ldots h^B$ on $D_t$ and define $h_i^{\text{best}}$ to be the hypothesis with lowest error. Then, like AdaBoost, we calculate $\epsilon_i$, the error of $h_i^{\text{best}}$ on $D_i$, and then $\alpha_i$ based upon $\epsilon_i$. Intuitively, $\alpha_i$ measures the importance that is assigned to $h_i^{\text{best}}$. Next the distribution $D_i$ is updated, with the weight on examples $h_i^{\text{best}}$ classifies incorrectly increased and those $h_i^{\text{best}}$ classifies correctly decreased. *This process is restarted and run for multiple iterations each time a new example arrives.* The final hypothesis $h_{final}^T$ is a weighted majority vote of the $N$ weak hypotheses where $\alpha_i$ is the weight assigned to $h_i^{\text{best}}$. A formal definition of all variables can be found in the pseudo code for ExpBoost shown below.

### Algorithm

ExpBoost is very similar to running AdaBoost.M1 whenever a new example arrives. The difference is that not only classifier $h_i^{B+1}$ trained on distribution $S_i$ from $c^{B+1}$ is evaluated in every boosting epoch $i$, but also all previous experts trained on data from concepts $c^1, ..., c^B$. The classifier with the lowest error is stored and used for calculating the new distribution over the training data. This way existing classifiers are incorporated into AdaBoost in an intuitive way.

**Pseudocode**:

When a new example arrives at time $T$, given:

- Training samples: $(x_1, y_1), \ldots, (x_T, y_T)$ where $x_t \in [-1, 1]^m$, $y_t \in \{-1, 1\}$ representing the target concept $c^{B+1}$.
- A pool of experts: $H^B = \{h^1, \ldots, h^B\} : x \to \hat{y}$ trained on samples from one of the concepts $c^1, ..., c^B$ each.
- Initialize $D_1(t) = \frac{1}{T}$

For $i \in 1, \ldots, N$:

- $S_i =$ sample from $(x_1, y_1) \ldots (x_T, y_T)$ drawn according to $D_i$.
- Train hypothesis $h_i^{B+1} : x \to \hat{y}$ using distribution $S_i$.
- $H_i = H^B \cup h_i^{B+1}$.
- Find $h_i^{best} \in H_i$ which minimizes error $\epsilon_i$ according to $D_i$: $\epsilon_i = min(\sum_{t:h(x_t) \neq y_t}^N D_i(t))$.
- If $\epsilon_i = 0$ or $\epsilon_i \geq 0.5$ then stop.
- Calculate $\alpha_i = \frac{1}{2} \ln(\frac{1-\epsilon_i}{\epsilon_i})$.
- Store: $h_i^{\text{best}}$ and $\alpha_i$.
- Update $D$: $D_{i+1}(t) = \frac{D_i(t) \exp(-\alpha_i y_t h_i^{\text{best}}(x_t))}{Z_t}$

Final hypothesis: $h_{final}^T(x) = sign(\sum_{i=1}^N \alpha_i h_i^{\text{best}}(x))$.

## Empirical Evaluation and Application

To empirically evaluate the capabilities of ExpBoost we created two synthetic datasets that both illustrate the concept recall problem. We compared the performance of different learning algorithms including two naive approaches and a technique that is capable of recalling concepts. Finally we present the robot learning task in more detail and demonstrate the performance of ExpBoost on it.

### Synthetic data

**Algorithms**: In both experiments, data from three different concepts is generated. Two concepts $c^1$ and $c^2$ have been previously encountered and on each of the two concepts an expert is trained and added to the ensemble $H^B = \{h^1, h^2\}$. Thus the ensemble that can be used for concept recall is of size $B = 2$. The third is the current concept $c^3$ denoted as $c^{B+1}$. All algorithms listed below are tested on samples from the the current concept $c^{B+1}$:

$h^{b\_best}$**:** The better one of the two experts from the pool of experts $H^B$ if evaluated on test data from the current concept $c^{B+1}$. This expert is not trained with data from the current concept $c^{B+1}$.

$h^{B+1}$**:** A classifier trained on the available data sampled from the current concept $c^{B+1}$ only.

$AdaBoost.M1$**:** AdaBoost.M1 trained on data sampled from concept $c^{B+1}$ only and evaluated on the same concept. Thus, AdaBoost.M1 as well as $h^{B+1}$ have no prior knowledge of the previous concepts $c^1$ and $c^2$.

$WMA+$**:** A modified version of the Weighted Majority Algorithm (WMA) by (Littlestone & Warmuth 1994) that was adapted to the test scenario. WMA+ incorporates ideas from the AddExp algorithm by (Kolter & Maloof 2005). WMA keeps a set of weights over an ensemble of experts and adjusts them according to their performance on the training data by a factor $\beta$. The final predictions are found by adding all predictions by each expert according to their weights. The majority of positive or negative prediction, respectively, decides for the final outcome. In addition AddExp is able to add experts on-line and increment existing experts if new training data is available and hence has the potential to cope with concept recall. The adapted WMA+ does not add a new expert every time it makes a mistake, because we assume that concept drift occures only if the opponent team is exchanged. Thus, one expert $h^{B+1}$ is added for the new concept $c^{B+1}$ only at the beginning of each game. WMA+ is trained on data sampled from concept $c^{B+1}$ and can actually make use of both experts in $H^B$ and $h^{B+1}$.

$ExpBoost$**:** Like WMA+ ExpBoost only learns from the new data corresponding to $c^{B+1}$ and can resort to the ensemble of experts $H^B$ but not to $h^{B+1}$.

**Experiment 1** (separation task)**:**

**Task**: The first experimental setup demonstrates that ExpBoost, as well as AdaBoost, can construct a strong classifier from boosting weak classifiers. Training examples were sampled with uniform distribution from $x \in [-1, 1]^2$. The target concept $h^{B+1}$ of the binary classification task $y \in \{0, 1\}$ was positive if $-0.5 < x_1 < 0.5$ and false otherwise. All experts in were trained with a decision stump algorithm, which finds a threshold in one of the dimensions of the input space that splits the data in two sets and minimizes misclassifications. Obviously one decision stump cannot shatter a set of data points corresponding to this target concept. The two experts $H^B$ were trained each on a batch of data of size $T = 100$ with 20% random class noise and solely sampled from a different half of the space $x_1 < 0$ and $x_1 > 0$, respectively. This concept can be expressed by one decision stump. The parameter for decreasing weights of WMA+ was set to $\beta = 0.5$ as proposed in (Kolter & Maloof 2005) for AddExp. In order to allow Expboost to build a strong $h^T_{final}$ from a weak classifier like decision stump the parameter $N$ was set to 50.

**Results**: Figure 1 shows the accuracy on a noise free test set of 1000 samples averaged over 15 runs. After having seen only 5 examples ExpBoost can already outperform $h^{b\_best}$ and in the end achieves a far higher accuracy than $h^{B+1}$. This establishes that ExpBoost can construct a strong classifier from training and combining weak classifiers, given enough boosting steps. The same applies to AdaBoost but without the ensemble of experts in $H^B$ AdaBoost needs far more samples to reach the same performance. As WMA+ can only combine $H^B$ and $h^{B+1}$ it cannot show significantly better performance than each of them separately.

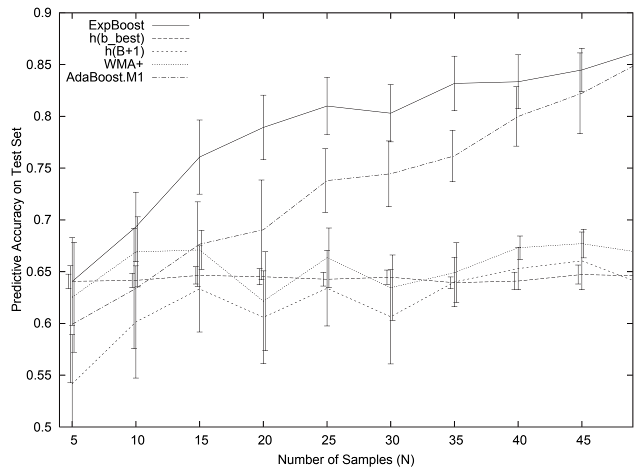


Figure 1: Performance of all experiments on the separation task with decision stump as the base classifier. The error bars show the limits for a 95% confidence interval.

**Experiment 2** (rotating hyperplane task)**:**

**Task**: In the second experiment an already strong classifier, a Support Vector Machine, was used as the base machine learning algorithm. Thus, all classifiers could potentially reach the same accuracy. But this setup demonstrates that another strength of ExpBoost is its ability to rapidly learn from few samples. The input vectors for this task were uniformly distributed in 18-dimensional space $x \in [-1, 1]^{18}$ which was separated by a hyperplane. Every dimensions had an associated weight $r \in [-1, 1]^{18}$ to make the hyperplane rotate around the origin (cmp. (Hulten, Spencer, & Domingos 2001)). A sample is labeled positive if $\sum_m (x_m r_m) \geq 0$ and negative otherwise. The dimension-weights of the hyperplane for the first target concept $c^1$ were $1/9$ for $1 \leq m \leq 9$ and 0 otherwise. The weights for the second concept $c^2$ were $1/9$ for $9 \leq m \leq 18$ and 0 otherwise. Both experts were trained with 20% random class noise on 200 samples. The hyperplane-weights of the current concept $c^{B+1}$ are $1/9$ for $7 \leq m \leq 9$, $-1/9$ for $10 \leq m \leq 15$ and 0 otherwise. The algorithms were trained with an increasing number of samples $5 \leq t \leq 100$ from concept $c^{B+1}$ with 10% random class noise and all classifiers were evaluated on 5000 noiseless samples averaged over 100 runs. Again, the parameter $\beta$ of WMA+ is set to 0.5. The parameter of Expboost was set to $N = 2$ because a SVM already is a strong classifier.

**Results**: Figure 2 compares all different algorithms on the rotating hyperplane task. Already after 20 samples ExpBoost can outperform the best previously trained classifier $h^{b\_best}$. Not before more than 40 samples were used for training the current expert $h^{B+1}$ it can achieve the same performance as ExpBoost. In contrast, WMA+ and AdaBoost focus their predictions almost instantly on $h^{B+1}$. These results confirm that ExpBoost's strength lies in a rapid recall-

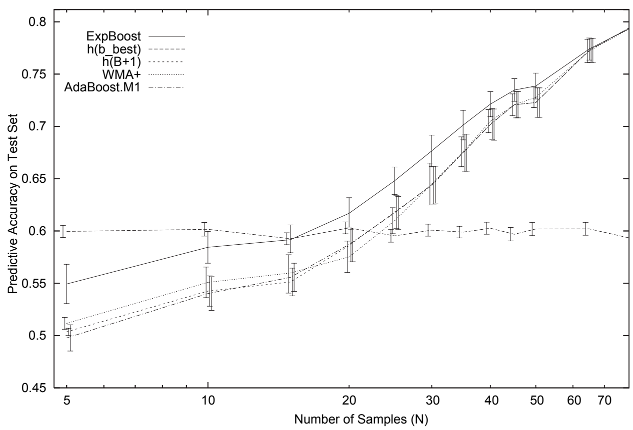


Figure 2: Performance of all experiments on the rotating hyperplane problem with a SVM as the base classifier. The x-axis is logarithmically scaled. The error bars show the limits for a 95% confidence interval.

ing of available experts while additionally learning $c^{B+1}$.

## Application to Simulated Robot Soccer

As mentioned before the learning task that motivated the research on concept recall comes from the domain of simulated robot soccer. The RoboCup 3D Soccer Simulator by (Obst & Rollmann 2005) provided the testbed and 5 different teams from 5 different universities from the RoboCup 2004 competition were used as the opponents to generate data representing concepts $c^1 \ldots c^5$.

**Task:** The learning task that was explored is an extension to the pass-evaluation described in (Stone & Veloso 1996). Passes between simulated soccer robots are evaluated to determine which agent will reach the ball next. This task has the advantage that it occurs very frequently so that the amount of training data that can be observed during one game is large. As passes can also be evaluated analytically and tests showed that knowledge of who received the pass first does not reveal much about whether the situation improved for this team or not, a two-step look-ahead pass-evaluation was considered. Thus, a complex heuristic was constructed that evaluates whether the whole situation has improved after the pass was played. A combination of three heuristics evaluating the position, velocity and possesion of the ball determined the utility of a situations. The utility of the state after the ball has been kicked for the second time was compared to the utility of the state before the ball had been kicked for the first time. If the last kick had improved the situation for the team the binary label $y \in \{0, 1\}$ is set to 1 and to 0 otherwise. The expressiveness of the heuristic was extensively evaluated by a human judge.

After a feature selection step, the initial kick situation when the pass was made was represented as a 15 dimensional vector $x \in [-1, 1]^{15}$, including information like the distance to the goal, ball speed, slope of the pass and velocities and distances of opponents and teammates.

This way a total of 13612 samples were recorded from games played against 5 different opponents. The average number of samples per game was 100.12. Thus, the size of the data set expressing the current concept $c^{B+1}$ is $T \approx 100$ which sets a strict limit on the available data.

**Algorithms:** As mentioned before data from 5 different concepts $c^1, ..., c^5$ corresponding to 5 different opponents was available. By turns 4 concepts were used for training the previous experts $h^1, ..h^4 \in H^B$ The one remaining concept was used as the current concept $c^{B+1}$ to be learned by the algorithms. All experts are based on a SVM classifier. Only $T = 50$ samples from $c^{B+1}$ were used for training which corresponds to the amount of data that on average can be observed during one half of a soccer game. All datasets were balanced before training and testing to make the amount of positive labels equal to the amount of negative labels. The setups used for comparing the performance of ExpBoost are the following:

$h^{b\_best}$**:** The expert from the pool of previous experts $H^B$ that had the best accuracy on predicting the whole data from the current concept $c^{B+1}$. This expert is not trained with data from the current concept $c^{B+1}$. The result is averaged over five runs where each time a different team was used for the current concept $c^{B+1}$.

$h^{B+1}$**:** One SVM classifier trained on 50 random samples from concept $c^{B+1}$ only and evaluated on the remaining data from the same concept. This procedure was repeated 10 times and the result averaged. Furthermore, this result is averaged over 5 runs for each of the five opponent teams and related concepts $c^1, ..c^5$;

$h^{b\_together}$**:** We also tested a setup which is traditionally the most common in simulated robot soccer. One SVM classifier was trained on all data from 4 teams combined except the data used for testing from concept $c^{B+1}$. This way all data available before a game is used for training a single classifier.

$ExpBoost$**:** ExpBoost only learns from 50 samples from the data corresponding to $c^{B+1}$, can resort to the ensemble of experts $H^B$ but not to $h^{optimal}$. The weak classifier $h^i$ that ExpBoost trains in every boosting step is based on the decision stump algorithm.

$h^{optimal}$**:** Like $h^{B+1}$, one SVM classifier was trained on samples from concept $c^{B+1}$ only. But this time all available data (about 2700 per opponent) was used for training and evaluated by 10-fold-cross validation. Again, the result is averaged over 5 runs for each of the five opponent teams. This classifier demonstrates the best possible classifier that can be trained. Of course this setup cannot be applied in the real tournament scenario because a maximum of 100 samples can be observed from the current concept during one game.

**Results:** Figure 3 compares the outcome of all experiments. The good performance of $h^{b\_best}$ shows that at least one of the previous concepts $c^b$ is related to the current opponent's concept $c^{B+1}$. In contrast, the performance of $h^{b\_together}$ is almost random guessing. Training one classifier on all previous concepts $c^b$ tries to combine concepts

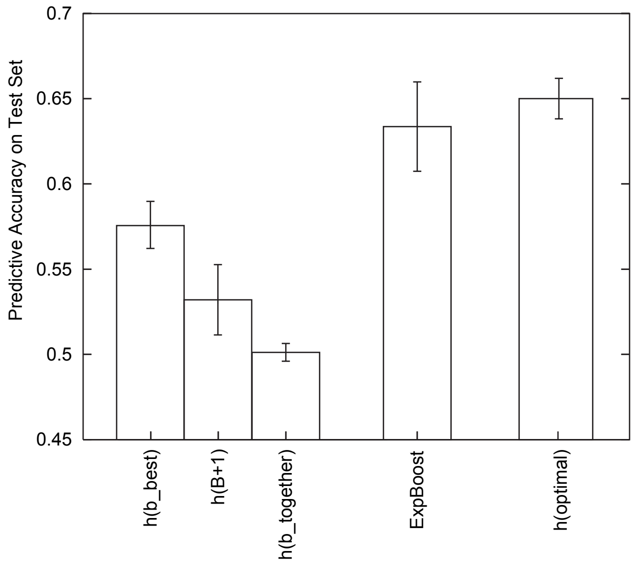


Figure 3: Comparison of all experiments on the pass-evaluation problem. All training and test sets are balanced such that they contain as many positive as negative examples. The error bars show the standard deviation.

that seem to conflict with each other. Thus, the hypothesis $h^{b\_together}$ is not capable of recalling the one concept from $c^b$ that is closest to the current concept $c^{B+1}$. The importance of recalling a known concept is also revealed in the performance of $h^{B+1}$. Training only with 50 samples from $c^{B+1}$ is not enough to achieve good performance in predicting this current concept. To reach a precision of about 65%, as $h^{optimal}$ did, more than 1000 samples needed to be used for training.

But ExpBoost shows that this precision on $c^{B+1}$ can almost be achieved with only 50 samples by finding the best previous concepts and combining them with classifiers trained on the available 50 samples from the current concept $c^{B+1}$. At this point it is interesting to analyze which classifier was picked in each of the $N = 10$ boosting steps. In the first 1-5 steps around 1-3 different experts from $H^B$ and in the remaining steps the decision stumps $h_i$ trained on $c^{B+1}$ were picked. This indicates that ExpBoost focuses on the best existing classifier first and in the end learns the new aspects of the current concept.

Summing up, the analysis of the extracted data shows that the task is highly team dependent. The classical way of learning demonstrated with $h^{b\_together}$ does not give any advantage if the opponent has never been seen before. Thus, concept recall is crucial if only a small number of samples are available.

## Related Work

Work that is related to concept recall can be found under various topics. Life-long learning, transfer learning, multitask learning (see (Caruana 1997)) or meta-learning are just a few domains that deal with retaining and applying the knowledge learned in one or more tasks to efficiently develop an effective hypothesis for a new task. For instance (Marx *et al.* 2005) use a familiar task for a maximum a posteriori elaboration on the logistic regression approach to transfer knowledge to a new task. (Wu & Dietterich 2004) propose a SVM framework that allows successful learning from few training samples by using auxiliary data as data points to constrain the learning process or as candidate support vectors. An example for a multitask learning technique is an MTL network (see e.g. (Silver & Mercer 2001))that uses a feed-forward multi-layer network with an output for each task to be learned. By sharing of the internal representation the network can combine the knowledge of different tasks. However, most of those works have sophisticated ways to store task knowledge (see (Silver & Mercer 1998)) or transfer it to the new task. In our case we restrict the transfer of concept knowledge to the predictions of experts only.

Another area of related research is on incremental online learning and drifting concepts. For instance (Kivinen 2003) or (Klinkenberg 2004) propose Support Vector Machines that can cope with concept drift. Handling concept drift is usually achieved by decreasing the influence of older samples. (Rüping 2001) does this by incorporating the *age* of an support vector in the loss function while (Klinkenberg & Joachims 2000) use a window on the training samples. Even though those algorithms can handle concept drift they have no special capabilities for keeping several potentially conflicting hypotheses obtained from different concepts.

A different area of research that is more targeted at concept recall is on expert prediction algorithms. Most work on expert ensembles can be traced back to aggregating strategies ((Vovk 1990)) and the weighted majority algorithm by (Littlestone & Warmuth 1994). Besides that, there are numerous extensions and related algorithms like *Tracking the Best Expert* by (Herbster & Warmuth 1998). All this work is based on how to minimize the total loss by choosing from the predictions of a set of experts. As discussed before in this paper the approach that is most closely related to the task explored in this paper is presented in (Kolter & Maloof 2005). The proposed algorithm AddExp is especially suited for incremental learning and concept drift.

Regarding robot soccer machine learning techniques have been widely used. Work that is related to on-line learning is for instance: In (Riley & Veloso 2001) the current opponent's action model is recognized from a set of possible models. In (Bowling 2003) a system for adapting the own team to a specific opponent is proposed. An memory based approach to learn on-line whether to shoot at the goal or to pass in a 2-on-1 situation is described in (Stone & Veloso 1996). All of this work is not particularly aimed at the concept recall problem, though.

## Conclusion

In this paper opponent-adaptive learning from the perspective of how to recall prior knowledge from previous games was discussed. We view this as an inductive transfer problem. We proposed the novel algorithm ExpBoost based on AdaBoost that is especially targeted at this type of problem.

ExpBoost applies the principles of inductive transfer, which were developed primarily in the context of neural networks, to boosting techniques. Boosting is a way of taking weak hypotheses and combining them to form a strong classifier. Here we use previously learned hypotheses for related concepts to strengthen the weak classifier. By comparing the performance of ExpBoost to other algorithms on synthetic data we show empirically that ExpBoost can build a strong classifier even if all experts are only weak classifiers. Further experiments confirm that ExpBoost is especially suited for rapidly recalling concepts by showing extraordinary performance on small data sets. This is demonstrated with a more realistic test environment, the RoboCup 3d-soccer-simulator. A task based on predicting the utility of a 2-step look-ahead pass was used for experiments. The analysis of the results confirmed the relevance of concept recall and illustrate that the traditional approach of machine learning in adversarial domains is often not suitable for opponent dependent tasks.

## Future Work

ExpBoost still has a lot of potential for improvements. An extension to regression problems reinforcement learning problems (rather than binary classification) would be interesting. Furthermore, replacing the learning algorithms used so far (SVM and Decision Stump) with algorithms that are suited for incremental and decremental on-line learning should make ExpBoost more computationally efficient and could extend its capabilities to incremental on-line learning. Another direction would be to consider work on dynamically adjusting $N$, the number of boosting iterations. This could reduce the danger of overfitting and make ExpBoost more independent from the learning algorithms used for the experts.

## References


Bowling, M. 2003. Multiagent learning in the presence of agents with limitations. CMU's Technical Report CMU-CS-03-118, CMU.

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

Freund, Y., and Schapire, R. E. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, 23–37.

Herbster, M., and Warmuth, M. K. 1998. Tracking the best expert. *Mach. Learn.* 32(2):151–178.

Hulten, G.; Spencer, L.; and Domingos, P. 2001. Mining time-changing data streams. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 97–106. New York, NY, USA: ACM Press.

Kivinen, J. 2003. Online learning of linear classifiers. In *Advanced lectures on machine learning*. New York, NY, USA: Springer-Verlag New York, Inc. 235–257.

Klinkenberg, R., and Joachims, T. 2000. Detecting concept drift with support vector machines. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, 487–494. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Klinkenberg, R. 2004. Learning drifting concepts: Example selection vs. example weighting. *Learning Drifting Concepts: Example Selection vs. Example Weighting* 8(3):281–300.

Kolter, J. Z., and Maloof, M. A. 2005. Using additive expert ensembles to cope with concept drift. In *Proceedings of the Twenty-second International Conference on Machine Learning*, 449–456. New York, NY: ACM Press.

Littlestone, N., and Warmuth, M. K. 1994. The weighted majority algorithm. *Inf. Comput.* 108(2):212–261.

Marx, Z.; Rosenstein, M.; Kaelbling, L.; and Dietterich, T. 2005. Transfer learning with an ensemble of background tasks. In *Workshop on Inductive Transfer : 10 Years Later held during NIPS 2005*.

Obst, O., and Rollmann, M. 2005. SPARK – A Generic Simulator for Physical Multiagent Simulations. *Computer Systems Science and Engineering* 20(5):347–356.

Riley, P., and Veloso, M. 2001. Recognizing probabilistic opponent movement models. In Birk, A.; Coradeschi, S.; and Tadokoro, S., eds., *Robot Soccer World Cup V*, 453458. Berlin: RoboCup Federation.

Rüping, S. 2001. Incremental learning with support vector machines. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, 641–642. Washington, DC, USA: IEEE Computer Society.

Schlimmer, J., and Granger, R. 1986. Beyond incremental processing: Tracking concept drift. In *Proceedings of the 5th National Conference on Artificial Intelligence*, 502–507. Menlo Park: AAAI Press.

Silver, D. L., and Mercer, R. E. 1998. The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. 213–233.

Silver, D., and Mercer, R. 2001. Selective functional transfer: Inductive bias from related tasks. In Hamza, M., ed., *IASTED International Conference on Artificial Intelligence and Soft Computing (ASC2001)*, 182–189. ACTA Press.

Stone, P., and Veloso, M. 1996. Beating a defender in robotic soccer: Memory-based learning of a continuous function. In Touretzky, D. S.; Mozer, M. C.; and Hasselmo, M. E., eds., *Advances in Neural Information Processing Systems 8*, 896–902. Cambridge, MA: MIT Press.

Vovk, V. G. 1990. Aggregating strategies. In *COLT '90: Proceedings of the third annual workshop on Computational learning theory*, 371–386. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Wu, P., and Dietterich, T. G. 2004. Improving svm accuracy by training on auxiliary data sources. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, 110. New York, NY, USA: ACM Press.