Thomas Dietterich, Wolfgang Maass,
Hans-Ulrich Simon, Manfred Warmuth
(editors):

Theorie und Praxis des Maschinellen Lernens

Dagstuhl-Seminar-Report; 91
27.06.-01.07.94 (9426)

Report
of the First Dagstuhl Seminar on
Theory and Praxis of Machine Learning
June 27th – August 1st, 1994

The first Dagstuhl Seminar on Theory and Praxis of Machine Learning was organized by Thomas G. Dietterich (Oregon State University), Wolfgang Maass (Technical University in Graz), Hans U. Simon (Universität Dortmund), and Manfred K. Warmuth (University of California at Santa Cruz). The 34 participants came from 13 countries, 25 came from europe, 5 from north america, 2 from Israel, 1 from the GUS, and 1 from Japan.

32 lectures were given, 8 of them were related to the pac-learning model of Leslie Valiant, 6 to the on-line learning model of Nick Littlestone (which is roughly equivalent to the EQU-query model of Dana Angluin), 5 to neural nets, 4 to inductive logic programming, 4 to direct learning applications, and 5 covered still other topics in machine learning.

Two introductory tutorials were given on neural nets (on monday morning by Thomas G. Dietterich and Andreas Weigend) and on the pac-learning model (on thursday morning by Shai Ben-David and Nick Littlestone). The tutorials were adressed to participants who are not experts in these fields (which was quite helpful because the seminar brought together experts from different areas in machine learning). A special tutorial was held by Manfred K. Warmuth on monday night. He compared additive and multiplicative updating schemes for agnostic on-line learning of an unknown target function by linear functions. He presented quite new results concerning this topic, thereby probably opening a new area of research.

On Tuesday night, a panel discussion about the proper choice of models for learning and opportunities for cross-fertilization between theory and practice of machine learning took place. The discussion shed light on the gap between the needs of practically inclined people and that what can be done by theoreticians. Wednesday night, an open problem session was held.

The seminar was intended to provide a meeting place for computer scientists who explore from various points of view the possibility for computing machinery to 'learn'. 'Machine learning' is a fast growing research area that attracts researchers from Theoretical Computer Science ('Computational Learning Theory'), from Artificial Intelligence, and from various other areas such as Pattern Recognition, Neural Networks, and Statistical Physics. Unfortunately, these areas tend to organize separate conferences, and opportunities where researchers in Machine Learning from different ares can meet are very rare. One main goal of the seminar was to fill this gap.

Second, it should be pointed out that Computational Learning Theory has become by now a well-established discipline of Theoretical Computer Science within the USA, with a very succesful annual conference (COLT). On the other hand, the active researchers in Computational Learning Theory in Europe have no regular common meetings, and they rarely collaborate across borderlines. The second main goal of the seminar was to offer for European researchers an opportunity to get to know each other and to exchange ideas. This may facililitate the creation a more permanent organisation for this research community in Europe.

The above-mentioned classification of the 34 talks shows that the seminar was indeed a meeting place for experts from different areas although there was a bias towards the theory side. The gap between these areas was certainly not filled. However, the seminar helped to understand more thoroughly 'how people from the other area are thinking'. A major problem for future meetings of this kind will be to build upon earlier meetings and, simultaneously, to enable new researchers 'to jump on the train'. For instance, for this meeting the introductory tutorials were quite helpful. Presenting them again at a second meeting will be helpful for new participants, but probably boring for the other-ones. This problem will be taken into account within an application for a second meeting.

Berichterstatter: Stefan Pölt

Participants

Foued Ameur, Universität Paderborn, Germany
Martin Anthony, London School of Economics, Great Britain
Peter Auer, Technische Universität Graz, Austria
Shai Ben-David, Technion, Haifa, Israel
Andreas Birk, Universität des Saarlandes, Germany
Ivan Bratko, University of Ljubljana, Slovenia
Nicolo Cesa-Bianchi, Universitá degli Studi di Milano, Italy
Antoine Cornuejols, Université Paris-Sud, France
Thomas Dietterich, Oregon State University, USA
Paul Fischer, Universität Dortmund, Germany
Ricard Gavaldà, University of Barcelona, Espain
Marko Grobelnik, Josef Stefan Institute, Ljubljana, Slovenia
Tibor Hegedues, Comenius University, Bratislava, Slovak Republic
Robert Holte, University of Ottawa, Canada
Jörg-Uwe Kietz, GMD, St. Augustin, Germany
Miroslav Kubat, Technische Universität Graz, Austria
Rupert Lange, Universität Mannheim, Germany
Ruquian Lu, Universität Bremen, Germany
Nick Littlestone, NEC Research Institute, Princeton, USA
Wolfgang Maass, Technische Universität Graz, Austria
Manfred Opper, Universität Würzburg, Germany
Wolfgang Paul, Universität des Saarlandes, Germany
Stefan Pölt, Universität Dortmund, Germany
David Sanchez, DLR, Oberpfaffenhofen, Germany
Michael Schmitt, Technische Universität Graz, Austria
Hans-Ulrich Simon, Universität Dortmund, Germany
Birgit Tausend, Universität Stuttgart, Germany
Naftali Tishby, Hebrew University, Jerusalem, Israel
Gyoergi Turan, University of Illinois at Chicago, USA
Volodya Vovk, Research Council for Cybernetics, Moscow, GUS
Manfred Warmuth, University of California, Santa Cruz, USA
Andreas Weigend, University of Colorado at Boulder, USA
Gerhard Widmer, University of Vienna, Austria
Thomas Zeugmann, Kyushu University, Japan

Contents

Jörg-Uwe Kietz
Learnability of Relational Knowledge
Marko Grobelnik
Induction of First Order Theories in Inductive Logic Programming
Gerhard Widmer
The Importance of Explicit Domain Knowledge for Learning: Two Case Studies
Birgit Tausend
Language Bias in Inductive Logic Programming
Gyoergi Turan
On Learnability and Predicate Logic
Antoine Cornuejols
Analogy as a Minimization Principle
Paul Fischer
Learning from Random Walks
Stefan Pölt
Time-Efficient Learning of Nearly Optimal Decisions
Tibor Hegedues
The History of Computational Learning Theory
Shai Ben-David
Off-line Versions of On-line Learning
Nick Littlestone
Choosing Learning Algorithms
Nicolo Cesa-Bianchi
Models of Learning with Noise
Martin Anthony
Function Learning from Interpolation
Foued Ameur
Space-Bounded Learning of Axis-Parallel Rectangles
Wolfgang Paul
Schemas and Genetic Algorithms
Thomas Zeugmann
Across the Boundaries of Learning Recursive Languages: A Survey

Abstracts

Connectionist Supervised Learning – An Engineering Approach

by Thomas Dietterich and Andres Weigend

We present a tutorial reviewing current methods for applying multilayer sigmoid networks to supervised learning tasks. We began by discussing network architectures and output representations. We then discussed learning algorithms (objective functions, optimization algorithms, and overfitting-avoidance techniques). We spent the most time describing various methods for avoiding overfitting including cross-validated early stopping, regularization, pruning of weights, and growing of networks. There is no consensus on which of these methods (or which combination of them) gives the best result. Next, we described three successful applications of sigmoid nets to automatic steering of an automobile, handwritten digit recognition, and predicting sun spots. Finally, we summarized the strengths and weaknesses of these algorithms.

Decision-Tree-Based Neural Nets

by Miroslav Kubat

This paper presents a new learning system and reports its first successful application to a medical domain. The system builds on the idea of generating a decision tree and translating it to a neural network architecture that is trainable by the backpropagation algorithm. The network has unambiguously defined topology and initial weights, and its subsequent training is very fast. The contribution of the system's individual aspects to the overall performance is studied. The achieved classification accuracy compares very favorably with other learning systems.

Learning Problems for Simple Neurons

by Michael Schmitt

We investigate the complexity of restricted consistency problems for single neurons with binary weights. Two types of restrictions are defined by imposing conditions on the permitted example sets: coincidence and heaviness. The first one is defined as the maximum inner product of two elements, the second one is the maximum hamming-weight of an element.

The consistency problem is shown to be NP-complete when the example sets are allowed to have coincidence at least 1 and heaviness at least 4. On the other hand, we give linear-time algorithms for solving the complementary cases.

For the maximum consistency problem (minimizing disagreement problem) we show that NP-completeness occurs at coincidence 1 and heaviness 2. This result also holds for neurons with arbitrary weights. In contrast, consistency for neurons with arbitrary weights is known to be solvable in polynomial time by Linear Programming.

On the Teacher-Student Problem for Multilayer Perceptrons

by Rupert Lange

Key idea: Target is a Neural Net (NN). Experiment: How does performance of pupil degreed on training set size, for different complexities of the teacher (target)?

Setup:

1. select n, initialize a NN with n inputs, n hidden layers, and n outputs to random weights $\in [-1, +1]$. This is the teacher (fixed architecture and weights)
2. generate samples by applying binary random input $\in \{-1, +1\}$ to the teacher
3. train a pupil with the same architecture as the teacher on the training set of size p
4. measure training and generalization error
5. statistics over weight initializations

Results: There exists a critical training set size! Below we have poor generalization, severe overfitting, and local minima. Above we have perfect generalization, no overfitting, and no local minima. There is a sharp transition between these two regions! The critical training set is proportional to the number of weights!

Further results: This transition also occurs in n-n-m, n-n-n-n, and n-m-l architectures. Backpropagation has a critical learning rate, scales with network complexity.

Predicting Impredictability - How Sure Are We about the Future?

by Andreas Weigend

It's not hard to generate forecast with neural networks. But how sure are we about them? Three methods are presented, addressing different sources of uncertainty

1. error bars through maximum likelihood. $y(\bar{x})$ is the conditional mean of the forecast. $v(\bar{x})$ is the conditional variance of the forecast. The cost function is the negative logarithm of a Gaussian; note that the variance is not treated as a constant but kept in the calculation.

2. predicting the density of the next step. We use a "fractional binning" method as representation that allows us to predict multimodial distributions.

3. uncertainty due to splitting of the data into training, cross-validation, and test sets. By bootstrapping the splits, we find that the effects of the split can be much larger than the effects of the specific initial conditions of the net.

Corresponding papers are available from ftp.cs.colorado.edu: Time-Series

1. error-bars.ps

2. prob-density.ps

3. bootstrap.ps

Statistical Physics of Learning Nets

by Manfred Opper (joint work with David Haussler)

Methods from Statistical physics have been successfully applied to the calculation of exact average case learning curves of neural nets. For this distribution dependent scenario one often finds a generalization error which decays like d/m asymptotically. Here m is the sample size and d an effective "dimension". It is tempting to relate d to the VC-dimension of the problem. But, as can be seen for special hypothesis classes (computable by neural nets), d can be much smaller than d_{VC}. To explain the scaling behaviour of the learning curves, we derive new upper and lower bounds for the expected risk of a learner which uses a Bayes strategy to predict a new label. This quantity is related to the cumulative error of the so called Gibbs algorithm which randomly selects a hypothesis from the version space and which represents the typical learning strategy studied in the statistical mechanics approach. The asymptotics of Bayes risk can be expressed by the scaling of the volume of all hypotheses in a small ϵ-ball around the target, where the distance between two hypotheses is defined as a squared averaged Hellinger distance (making the bounds applicable to noisy models as well). The case $Vol \sim \epsilon^d$ explains the $\sim d \ln m$ scaling of the cumulative error.

Issues I Wish Theory Could Address

by Thomas Dietterich

My talk described six areas where there are opportunities for mathematical analysis to provide insight and assistance to experimental machine learning.

- Multiple instance problem. In some supervised learning applications, each training example is ambiguous: it can be represented by multiple feature vectors: $< (V_1, V_2, \ldots, V_k),\ f(x) >$ where the V_i are the feature vectors and $f(x)$ is the class of example x. A case that arises in drug design is that for positive examples the target concept must label at least one of these features as positive. For negative examples, the targetconcept must label all of these features as negative.

- Feature-manifold problem. In other supervised learning applications, we are given a feature-extraction function $V(x, p)$, where x is a training example and p is a set of pose parameters. $V(x, p)$ is a vector of features describing x in pose p. Again, a case that arises in drug design is to find a hypothesis $g()$ such that the maximum value of $g(V(x, p))$ over all poses p is equal to the target function $f(x)$.

- Overfitting avoidance. When should pruning (and other overfitting avoidance techniques) be used?

- Out-of-sample error rate. What would need to be true of the target class, the hypothesis class, and the learning algorithm such that the out-of-sample performance of a learning algorithm would improve monotonically with sample size?

- Reinforcement learning. In reinforcement learning, one typically attempts to learn a function $Q(s, a)$ that predicts the expected long-term reward of performing action a in state s and then following an optimal policy thereafter. Recently, people have started using neural networks to learn $Q(s, a)$, because they want the function to generalize over various states where the right action is identical. Under what conditions will this succeed? How does this interact with exploration strategies?

- Learning multiple concepts. Consider two target concepts c_1 and c_2 and suppose that they are related in some way (e.g., c_1 and c_2 are very similar, c_1 and c_2 are very different, c_1 is a subset of c_2). During learning, an algorithm can request examples labelled according to c_1 or according to c_2. What is the most effective algorithm for PAC-learning c_1 and c_2?

Learning Stochastic Models for Man-Machine Communication

by Naftali Tishby (joint work with Dana Ron and Yoram Singer)

We consider the problem of extracting statistical structure in complex sequences coming from natural human communication (e.g. text, handwriting, or speech). We propose and analyze a learning algorithm for a class of sequence distribution induced by variable memory Markov process. We show that this class is learnable with respect to the

Kullback-Leibler divergence, in time and samples polynomial in the number of states and maximal memory of the source model and in its mixing time, as well as in the accuracy and confidence parameters. This is in contrast to other models, e.g. hidden Markov models, on which there are known hardness results on such learning.

We demonstrate the power of the algorithm in efficient learning of English text, with applications to error correction using a viterbi decoding scheme. We then apply a similar algorithm to the learning of a Probabilistic Acyclic Finite Automata and demonstrate its use for cursive handwriting modeling and recognition. We further discuss the ability of learning hierarchies of such models and their possible role in a complete language understanding system.

For further reports: see COLT '94 and NIPS '94

Automatic Knowledge Acquisition by Understanding Pseudo-Natural Languages

by Ruquian Lu

This talk describes a series of experiments done at the Institute of Mathematics, Academia Simica, towards automatic construction of knowledge-based systems. The key point of these experiments is to acquire knowledge automatically from texts by understanding pseudo-natural languages. Our main idea is to divide the development process of knowledge-based systems into two stages.

In the first stage, the knowledge engineer translates a book (recommended by domain experts) into a pseudo-natural language (let's call it PNL). Since PNL looks very similar to written languages used in technical texts, it is easy for the user to do the translation (much easier than to translate English into Chinese). The translated text is already a computer program which can be compiled and analysed by the computer. The computer extracts the knowledge from the program and classifies and recognizes it into a knowledge-base which, combined with a preexisting inference engine, forms a prototype expert system.

In the second stage, the domain expert can

1. use PNL to enrich the knowledge-base
2. put in his own experiences
3. use a set of well-justified cases which, processed by the knowledge refiner, will refine the above obtained knowledge base

Based on this idea, experiments in fields like expert systems, ICAI systems, MIS and computer-aided animation systems have been done with quite satisfying results.

A common framework for deriving on-line learning algorithms and for proving worst-case loss bounds

by Manfred Warmuth

We give a framework for deriving on-line learning algorithms. Assume the hypothesis of the on-line algorithm is specified by a parameter (weight) vector. The algorithm sees one example at a time and incurs some loss measuring the how badly the current hypothesis fits the example.

After each example the algorithm updates its parameter vector. In making an update, the algorithm must balance its need to be "conservative", i.e. retain the information it has acquired in the preceding trials by staying close to the old parameter vector, and to be "corrective", i.e. to make certain that if the same example was observed again, then the loss on this example would be smaller. An update is derived by approximately minimizing the following sum: the distance between the updated and the old parameter vector plus eta times the loss of the updated parameter vector on the current example. Eta is a positive parameter representing the importance of correctiveness as compared to the importance of conservativeness.

The loss function is usually determined by the learning problem. However the algorithm designer may chose a distance function (typically not a metric). When using the squared Euclidean distance then the above framework gives the usual gradient descent update. Many of the standard neural network algorithms belong to the gradient descent family.

When entropy based distance functions are used then this leads to a new family of update algorithms. Whenever there is an algorithm derived by the gradient descent heuristic there is now a competitor algorithm belonging to the entropic family. For example there is a simple alternate to the standard backpropagation algorithm. Both algorithms calculate the same gradients, but they are used differently in the updates of the weights.

In the above framework a distance function motivates an update algorithm. But the distance function is then also used in deriving worst-case loss bounds for this algorithm. Such bounds are proven using an amortized analysis where the distance function serves as a potential function.

So far worst-case bounds have only been obtained for simple one-neuron settings. The loss bounds show incomparable learning performance between the gradient descent and the entropic family of algorithms. The theoretical analysis is supported by experiments. The algorithm from the entropic family typically outperforms the corresponding gradient descent based algorithm when the parameter vector is sparse. This was first observed by Nick Littlestone when he analyzed his algorithm WINNOW for learning Boolean disjunctions (WINNOW belongs to the entropic family.)

We give reasons why the gradient descent family typically does not converge until it has seen a number of examples that is proportional to the number of parameters. This is sometimes called the "curse of dimensionality". The entropic family to avoid this curse when the best parameter vector is sparse.

We believe that the radically different behavior seen in the one neuron case will carry over to more complicated settings such as the comparative performance of the two backpropagation algorithms.

Efficient Learning with Virtual Threshold Gates

by Wolfgang Maass (joint work with Manfred Warmuth)

As a new approach towards the design of efficient learning algorithms we introduce "virtual threshold gates" (as a tool for designing suitable hypothesis spaces, in combination with the learning algorithm WINNOW due to Littlestone). These threshold gates are "virtual" in the sense that their size exceeds by far the acceptable complexity bounds. However, one can nevertheless work with such gates in an "efficient" manner, since their weight-pattern has various regularities so that one can perform an on-line compression.

In this way one can design more efficient learning algorithms for a number of well-studied classes of target concepts such as rectangles and halfspaces in any fixed dimension. The worst case number of mistakes of our new on-line learning algorithms can be shown to be optimal (resp. almost optimal) for these classes, even if one compares their performance with that of algorithms which may use substantially more resources (such as arbitrary hypotheses instead of computable ones, additional types of queries, and arbitrary computational power).

In addition, these new learning algorithms inherit the quite favorable noise-tolerance behavior of WINNOW, and also yield new PAC-learning algorithms that are with regard to sample-complexity and noise-tolerance superior to those algorithms that were previously known.

Binomial Weights for On-line Learning

by Manfred Warmuth

We study the problem of deterministically predicting boolean values by combining the boolean prediction of several experts. Previous on-line algorithms for this problem predict with the weighted majority of the experts' prediction. These algorithms give each expert an exponential weight β^m where β is a constant in $[0,1)$ and m is the number of mistakes made by the expert in the past. We show that is better to use sums of binomials as weights. In particular, we present a deterministic algorithm using binomial weights that has a better worst case mistake bound than the best deterministic algorithm using exponential weights. The binomial weights naturally arise from a version space argument. we also show how both exponential and binomial weighting schemes can be used to make prediction algorithms robust against noise.

On-line Learning for Finite Automata: Decision Theoretic Approach

by Volodya Vovk

We consider the problem of optimal control of a finite-state plant in a finite-state environment. This problem has two sides: the control side (the main interest of this talk) and the learning side. With the help of Martin's theorem on the determinacy of Borel games we prove the existence of a "control structure" on the plant in the case that optimal control is possible. Given such a control structure, we can control the plant efficiently using for learning the environment either Littlestone and Warmuth's Weighted Majority Algorithm or Rivest and Schapire's reset-free algorithm for exact learning of finite automata with membership and equivalence queries. The advantage of the control strategy based on the Weighted Majority Algorithm is its robustness and better performance, and the advantage of the control strategy based on Rivest and Schapire's algorithm is its computational efficiency. A control structure (when it exists) can be found in time $O(N^4)$, where N is the number of states of the plant.

On-line Learning of Decision Lists and Trees

by Hans-Ulrich Simon

The following results are shown:

1. Boolean decision lists over a base of N functions can exactly identified after making at most $(N-1)(N+2)/2$ mistakes.

2. For Boolean decision trees of rank r the mistake bound becomes $O(n^{2r})$.

Both mistake bounds are achieved by polynomial time algorithms. The talk finally discusses the problem of constructing the shortest representation of the target concept (final hypothesis must be short; preliminary hypotheses may be long). If the equivalence-test for 2 hypotheses is in P, then learning the shortest representation is computationally equivalent to minimizing a given representation. If the equivalence-test is NP-hard, then

- the computational power of the learning algorithm is bounded by $\Delta_2 = P[NP]$
- learning the shortest representation is unfeasible if minimizing a given representation is Σ_2-hard ($\Sigma_2 = NP[NP]$).

These insights can be applied to k-decision lists.

$k = 1$: Learning shortest representation is in P

$k \geq 3$: Learning shortest representation is not in P (modulo an unproven assumption in structural complexity)

$k = 2$: – open – but in P for the special case of 2-CNF or 2-DNF

Computational Complexity of Neural Nets: A Kolmogorov Complexity Characterization

by Ricard Gavaldà (joint work with José Balcázar and Hava Siegelmann)

The computational power of neural networks depends on properties of the real numbers used as weights. We focus on networks restricted to compute in polynomial time, operating on boolean inputs. Previous work has demonstrated that their computational power happens to coincide with the complexity classes P and p/poly, respectively, for networks with rational and arbitrary real weights. Here we prove that the crucial concept that characterizes this computational power is the Kolmogorov complexity of the weight, in the sense that, for each bound on this complexity, the network can solve exactly the problems in a related nonuniform complexity class located between P and p/poly. By proving that the family of such nonuniform classes is infinite, we show that neural networks can be classified into an infinite hierarchy of different computing capabilities.

Simulating Access to Hidden Information while Learning

by Peter Auer (joint work with Phil Long)

We presented a general technique and some applications how a learner without the access to some hidden information can learn nearly as well as a learner which has access to this hidden information. The most striking application is the comparison of a learner which uses only equivalence queries with a learner which uses equivalence and arbitrary boolean queries. (In this case the hidden information are the answers to the boolean queries.) It turns out that the performance of the learner without boolean queries is of by at most a constant factor of 2.41 from the performance of the learner with boolean queries.

Automated Ecological Modelling through Machine Learning

by Ivan Bratko

Machine learning techniques can be used as tools for inducing models from measured data in a domain of exploration. In this talk, automated modeling of an ecological process, the growth of algae in the Lagoon of Venice, is described. The problem is to determine the dynamics of the biomass depending on the nutrients and weather conditions in the lagoon. Two Machine Learning tools were used in these experiments: RETIS (Karalic) that induces regression trees, and LAGRANGE (Dzeroski and Todorovski) that induces differential equations from timed data. It is argued that the interpretability, by domain experts, of induced hypotheses is essential and probably more important than the prediction accuracy. To improve interpretability, constraints should be imposed on the hypothesis language. Experience from this study suggests ways of constraining the hypothesis language so that only such hypotheses are generated that have natural interpretation in terms of prior domain knowledge.

Occam's Razor in Theory and Practice

by Robert Holte

A brief survey of the use and study of Occam's Razor principle in the theory and practice of machine learning, including recent large-scale experimental studies by Murphy & Pazzani. Reflection on the interaction between theory and practice on this principle.

Learnability of Relational Knowledge

by Jörg-Uwe Kietz

The talk gives on overview of recent theoretical results in the rapidly growing field of inductive logic programming (ILP) [1,2,3]. The ILP learning situation (generality model, background knowledge, examples, hypotheses) is formally characterized and compared to the PAC-model of learnability. As the general problem isn't efficiently learnable, various restrictions of it are discussed in the light of their impact on learnability. Several learnability results for logic programs are then presented, both positive and negative. These results precisely characterize the class of polynomial learnable relational knowledge.

[1] Jörg-Uwe Kietz: Some lower Bounds for the Computational Complexity of Inductive Logic Programming, In: Proc. Sixth European Conference on Machine

Learning (ECML-93), 1993.

[2] Jörg-Uwe Kietz and Sazo Dzeroski: Inductive Logic Programming and Learnability, SIGART Bulletin, Vol. 5, Nr. 1, 1994.

[3] William W. Cohen: Learnability of Restricted Logic Programs, In: S. Muggleton (Ed.): Proc. of the Third International Workshop on Inductive Logic Programming (ILP-93), Jozef Stefan Institute, Ljubljana, Slovenia, pp. 41-71, 1993

Induction of First Order Theories in Inductive Logic Programming

by Marko Grobelnik

Inductive Logic Programming (ILP) is an area within Machine Learning which uses Horn clause logic as a hypothesis language. Given background knowledge B, positive examples of the concept E and negative examples N, the system must find a hypothesis H such that $B \wedge H \vdash E$ and $B \wedge H \nvdash N$.

The ILP research field started 1990 s. Muggleton. Because of the powerful hypothesis language it is possible to learn very strong concepts, otherwise hardly expressible. On the other hand, because of the very powerful language, there is no efficient algorithm to find more complex theories.

The Importance of Explicit Domain Knowledge for Learning: Two Case Studies

by Gerhard Widmer

Prior knowledge is an important source of bias in learning. Humans rarely learn without taking into account what they already know about the world. Knowledge is especially important when little training data is available.

We present two case studies that illustrate two approaches to introducing explicitly formulated domain knowledge into a learner. The target concepts to be learned are general rules of musical expression (expressiveness performance); training examples are expressive performances by human musicians. An explicit qualitative model of human music understanding is formulated and provided to the learner as background knowledge.

In the first learning system that is presented the knowledge is used to guide and constrain a heuristic search-based generalization algorithm. The effect is a strong bias towards musically sensible generalizations. In the second system, the musical knowledge is used to transform the original learning problem to a more abstract level where relevant regularities become apparent. In both cases, this enables the learner to extract useful regularities form very few examples. This is demonstrated in various experiments with performances of music of different styles.

Language Bias in Inductive Logic Programming

by Birgit Tausend

Controlling and adapting Inductive Logic Programming systems is an important task. In particular, the powerful hypothesis language needs to be controlled in order to search efficiently and to exploit knowledge about applications. In contrast to other biases, the language bias is not very well investigated. The reason is that there is a great variety of language biases in ILP, and they are neither uniquely represented nor their basic constituents are easy to identify. Moreover, many language biases in ILP are not explicitly declared but "hidden" in the operators.

In this talk, we present MILES-CLT, a representation for language bias that aims to overcome these problems. The basic idea of the representation is to extend the scheme-based approach that represents sets of hypothesis clauses by schemes. A scheme for a hypothesis claus called clause template in MILES-CLT includes schemes for each literal in the clause. Similar as in a record data type, a literal template consists of several identifiers followed by a scheme variable or a constant. In addition, the domain of a scheme variable in a literal template may be further restricted by conditions. Since the items in a literal template describe the set ov covered literals, they have to include at least an item for the predicate name and the arguments. Other items describe the arity, the number of new variables, the argument or the predicate types or the depth of the covered literals, for example. A declaration of a hypothesis language in MILES-CLT consists of a set of schemes T, the vocabulary Σ including predicates, functors, and types, and an instantiation function I. Given a set T and Σ, I constructs hypotheses by instantiating the scheme variables in a clause template.

In MILES-CLT, the declarative representation covers the language biases used in ILP, the declaration of a hypothesis language is concise but understandable and it is suitable for large hypothesis languages as well as hypothesis languages consisting of several subsets with different biases. Other advantages are that schemes support the search through the hypothesis space while specialising hypotheses, and they enable the shift of bias.

On Learnability and Predicate Logic

by Gyoergi Turan

We consider an approach to concept learning in predicate logic. Given a model M

and a formula $\phi(x_1, \ldots, x_n)$, the concept $C_{\phi,M}$ defined by ϕ in M is the set of k-tuples satisfying ϕ. The concept class $C_{\Phi,M}$ consists of the concepts $C_{\phi,M}$ for $\phi \in \Phi$. The learning models considered are PAC-learning, learning with arbitrary equivalence queries, and learning with equivalence queries. A class of formulas Φ is called *easy* for a learning model if there is a constant upper bound for the learning complexities of the concept classes $C_{\Phi_{\mathrm{M}},M}$ for all M. Here Φ_M is the class of formulas obtained from Φ by substituting arbitrary constants for the constant symbols, assuming that there is a constant for each element of M. For example, $\Phi = \{R(c, x)\}$ is hard for PAC-learning, as the corresponding concept classes can have arbitrarily large VC-dimension. It is shown that the class of quantifier free formulas of bounded complexity containing unary predicates and functions is easy for learning with equivalence queries, and the class of formulas of bounded complexity containing unary predicates and a single unary function is easy for PAC learning. The proof of the latter result uses some results from model theory. Examples such as the one mentioned above indicate that more general classes of formulas defined in a similar way are already hard.

Analogy as a Minimization Principle

by Antoine Cornuejols

Analogical reasoning viewed as a psychological phenomenon presents a set of properties of which some viz. non symmetry, non transitivity and learnability, have not been addressed and accounted for in the past. This had led to a new examination of analogy.

Analogy making involves the interplay between two processes : (i) perceiving the two ”analogues” as similar, and (ii) finding a relevant ”explanation” for the source case that can be usefully transferred to the target case.

Both processes can be analyzed in the light of Kolmogorov complexity measure and of Occam's principle. However, in order to plainly account for both the presence of background knowledge and the interdependencies between processes (i) and (ii), it is necessary to modify Kolmogorov complexity by introducing building blocks (concepts or abstractions) shared in both (i) and (ii). This can be done by using a form of the Minimum Description Length principle. Thence, we get a new complexity measure that takes into account prior knowledge and we obtain a characterization of analogy that explains the properties observed.

Learning from Random Walks

by Paul Fischer

An on-line approach to passive learning is considered. In contrast to the classical PAC model we do not assume that the examples are independently drawn according to an underlying distribution, but that they are generated by a time-driven process. We define deterministic and probabilistic learning models of this sort and investigate the relationships between them and with other models. The fact that successive examples are related can often be used to gain additional information similar to the information gained by membership queries. We show that this can be used to design on-line prediction algorithms. In particular, we present efficient algorithms for exactly identifying Boolean threshold functions, 2-term RSE, and 2-term-DNF, when the examples are generated by a random walk along the edges of the Boolean cube.

Time-Efficient Learning of Nearly Optimal Decisions

by Stefan Pölt

Learning scenarios in pattern recognition and statistical decision theory are analyzed within a formal learning framework. It turns out that Valiant's PAC-learning model does not cover such scenarios. The main reason for this is that in the PAC-model the classification labels are assumed to be a function of the objects. A decision-theoretic generalization of the PAC-learning model, called PAB (probably almost Bayes) model and introduced by Haussler, is presented. Within this model Haussler proves upper bounds on the sample complexity but disregards computational aspects. In fact, his learning algorithm can be proven to be not time-efficient even for simple applications, since it contains NP-hard subproblems. Here another very simple and general learning algorithm is analyzed. Upper bound on the sample and time complexity are proven. But the price one has to pay for gaining time-efficiency is loosing robustness. While Haussler's results are distribution independent, for our results the learning algorithm has to know the parametrical form of the distribution of the objects in advance.

The generalized learning model is compared to other related learning model which are also extensions of the basic PAC-model. It is shown that results in the model of probabilistic concepts (introduced by Kearns and Schapire) and in the random classification noise model (Angluin and Laird) might help to make progress in the PAB-model.

The History of Computational Learning Theory

by Tibor Hegedues

We give a survey of the main results achieved within the framework of two learning theory related research directions: the fault detection research (testing) and the function deciphering research. While testing is closely related to topics like teaching (exact specification) and learning with membership queries, the function deciphering model is completely equivalent to learning with membership queries. We mainly focus on the results given by Korolkov, Hansel and others on the complexity of learning monotone

Boolean functions with membership queries, and on the combinatorial arguments the algorithms and lower bounds were based on.

Off-line Versions of On-line Learning

by Shai Ben-David (joint work with Eyal Kushilevitz and Yishay Mansour)

We present an off-line variant of the mistake-bound model of learning. Just like in the well-studied on-line model, a student in the off-line model has to learn an unknown concept from a sequence of "guess and test" trials. in both models, the aim of the learner is to make as few mistakes as possible. The difference between the models is that, while in the on-line model only the *set* of possible queries is known, in the off-line model the *sequence* of queries (i.e., the identity of the queries as well as the order in which they are to be presented) is known to the learner in advance.

We give a combinatorial characterization of the number of mistakes in the off-line model. We apply this to solve several natural questions that arise for the new model. First, we compare the mistake bounds of an student to those of a student learning the same concept classes in the on-line scenario. We show that the number of mistakes in the on-line learning is at most a $\log n$ factor more than the off-line learning, where n is the length of the sequence. In addition, we show that if there is an off-line algorithm that does not make more than a constant number of mistakes for each sequence then there is an on–line algorithm that does not make more than a constant number of mistakes.

The second issue we address is the effect of the ordering of the queries on the number of mistakes of an off-line student. It turns out that there are sequences on which an off-line student can guarantee at most one mistake, yet a permutation of the same sequence forces him to err on many queries. We prove, however, that the gap, between the off-line mistake bounds on permutations of the same sequence of n-many queries, cannot be larger than a multiplicative factor $\log n$, and we present examples that obtain such a gap.

Choosing Learning Algorithms

by Nick Littlestone

This talk compares the on-line learning algorithm Winnow with the perceptron training algorithm and with a Bayesian prediction algorithm based on the assumption of conditionally independent attributes (sometimes called the naive Bayes algorithm). All of these algorithms make linear threshold predictions, but have different performance in the presence of redundant and irrelevant attributes.

Winnonw and the perceptron algorithm update their staet only when a mistake is made. These algorithms cope much better than the naive Bayesian algorithm with redundant attributes. Interestingly, a mistake-driven version of the Bayesian algorithm also copes well with redundant attributes in simulations.

Winnonw copes with large numbers of irrelevant attributes much better than the perceptron algorithm, and also much better than both versions of the Bayesian algorithm in simulations.

When running a mistake-driven algorithm, making predictions using a tested hypothesis instead of the algorithm's current hypothesis can give better results, and gives one way to use information that would otherwise be ignored by mistake-driven algorithms.

On-Line Learning with Malicious Noise

by Nicolo Cesa-Bianchi (joint work with Peter Auer)

We investigate a variant of the on-line learning model for classes of $\{0,1\}$-valued functions (concepts) in which the labels of a bounded fraction of the input instances are corrupted by adversarial noise. We propose an extension of a general learning strategy, known as "Closure Algorithm", to this noise model, and show a worst-case mistake bound of $m/(1-\eta(d+1))$ for learning an arbitrary intersection-closed concept class $\mathcal{C}$, where η is the noise rate, d is a combinatorial parameter reasoning $\mathcal{C}$'s complexity, and m is the worst-case mistake bound of the Closure Algorithm for learning $\mathcal{C}$ in the noise free model. For several concept classes our extended Closure Algorithm is efficient and can tolerate a noise rate equal to the information-theoretic upper bound.

Function Learning from Interpolation

by Martin Anthony (joint work with Peter Bartlett, Yuval Ishai, and John Shawe-Taylor)

Suppose that H and C are sets of functions from a domain X to the real numbers. We say that H validly generalises C from approximate interpolation if for each positive constant η and each ϵ, δ between 0 and 1, there is an integer $m_0(\eta, \epsilon, \delta)$ such that for any function t in C and for any probability distribution P on the set X, if $m \geq m_0$ then with probability at least $1-\delta$, if $\bar{x} = (x_1, \ldots, x_m) \in X^m$ and h is in H and $|h(x_i) - t(x_i)| < \eta$ for i between 1 and m, then

$$P(|h(x) - t(x)| \geq \eta) < \epsilon \ .$$

We find conditions that are necessary and sufficient for H to validly generalise C from approximate interpolation and we obtain bounds on the sample length function m_0. The key result is that a necessary and sufficient condition for H to generalise from approximate interpolation the set of *all* functions from X to the real numbers is that H has finite pseudo-dimension.

A relaxed condition is discussed, which has a strictly weaker necessary and sufficient condition.

Space-Bounded Learning of Axis-Parallel Rectangles

by Foued Ameur

We consider the standard model for on-line learning of Angluin and design a d-space-bounded learning algorithm for the concept class of axis-parallel rectangles that has learning complexity $O(d^2 \log^2 n)$ (which could easily be improved to $O(d^2 \log n)$).

This result improves the best known from Chen and Maass.

Schemas and Genetic Programming

by Wolfgang Paul (joint work with Andreas Birk)

With the help of schemas and genetic programming we describe systems which

- interact with the real world
- make theories about the consequences of their actions and
- dynamically adjust inductive bias

We present experimental data about the learning of geometric concepts and of moving a block in a microworld.

Across the Boundaries of Learning Recursive Languages: A Survey

by Thomas Zeugmann

The present paper deals with with the learnability of indexed families of uniformly recursive languages *from positive data* as well as from both, *positive and negative data.* We consider the influence of various monotonicity constraints to the learning process, and provide a thorough study concerning the influence of several parameters. In particular, we survey results concerning learnability in dependence on the hypothesis space, and concerning order independence. Moreover, new result dealing with the efficiency of learning are provided. First, we investigate the power of *iterative* learning algorithms. The second measure of efficiency studied is te number of *mind changes* a learning algorithm is allowed to perform. In this setting we consider the problem whether or not the monotonicity constraints introduced do influence the efficiency of learning algorithms.

The paper mainly emphasis to provide a comprehensive summary of results recently obtained, and of *proof techniques* developed. Finally, throughout our guided tour we discuss the question how a natural language learning algorithm might look.