TECHNICAL REPORT
YR-2007-2

PREDICTION GAMES IN INFINITELY RICH WORLDS

Omid Madani
Yahoo! Research
3333 Empire Ave
Burbank, CA 91504
{madani@yahoo-inc.com}

June 15, 2007

Santa Clara, California • Berkeley, California • Burbank, California • New York, New York
Barcelona, Spain • Santiago, Chile

Yahoo! Research Report No. YR-2007-2

Yahoo! Research Report No. YR-2007-2

PREDICTION GAMES IN INFINITELY RICH WORLDS

Omid Madani
Yahoo! Research
3333 Empire Ave
Burbank, CA 91504
{madani@yahoo-inc.com}

June 15, 2007

ABSTRACT: How can a system acquire millions of inter-related categories? The process of playing *prediction games* may make this massive learning possible. In this paper, we describe the games and discuss the constraints and desiderata on solution techniques and some of the challenges.

1. Introduction

Categorization is fundamental to intelligence [36, 48, 26]. Without categories, every experience would be new, and one couldn't make sense of one's world. Categories are the means of advanced pattern recognition and generation, and categorization allows for proper subsequent actions and informs intelligent decisions. Furthermore, an entity needs large numbers of categories, millions and beyond, to exhibit increasingly sophisticated intelligent behavior. Generally speaking, the higher the intelligence, the more categories are required. A fundamental question is then, *how so many categories may be obtained?* The process of acquiring entails *operationality*. Operationality includes efficient and robust classification (categorization). New categories are to join an operational system, and furthermore increase the system's overall capability. Categories inter-relate in a number of ways and can be complex. For example, categories can be compositions of other categories. Thus, the *face* category in the familiar visual domain is a composition of several subcategories such as the *eye* and the *nose* categories. Categories enjoy various types of relations (spatial, temporal, functional, ..) and many such relations. These dependencies present opportunities as well as considerable difficulties. For example, such relations may help in easing or accelerating categorization. However, apriori, it is very hard or impossible to anticipate and program them[1], and one can imagine numerous domains that are unfamiliar to humans or different to various degrees.

We conjecture that an approach based primarily on learning can meet the challenges of acquiring myriad categories. In that case, the above considerations suggest that much learning, of different kinds, but in an integrated fashion, is required. The major question is then *how this massive integrated learning can be achieved?*

In this paper we propose and explore an avenue that we call *prediction games in infinitely rich worlds*. In these games, the world is a source of an unlimited stream of information. The games are played by a *prediction system* that in effect repeatedly experiments with its world and learns from its experiments. The system converts its input stream from the world into a sequence of learning episodes for itself. Each learning episode consists of the system hiding parts of the input, guessing (predicting) them using the remainder of the input (context derived in part from local information in the stream), and updating itself based on comparing its observations with its predictions. The goal is to improve predictions.

The games enjoy two general properties that we want to emphasize:

- Ample learning opportunity is available, and there is much to learn.
- The world is rich in regularities that make efficient scalable learning possible.

A major attractive aspect of these games is that the number of learning trials is regarded as unbounded. Abundant learning experience is a necessity for learning millions of richly interacting categories.

[1] These important considerations are further discussed in Section 4.9 (limits of human involvement).

Just as important, we point to the fundamental role of predictions in learning. We explain how the goal of improving predictions can drive the learning of various kinds.

A thesis of this work is that we should explore the space of *systems*, multiple parts working in concert, in thinking about the general task. In the course of playing predictions games, the system acquires new categories to be predicted and to help predict. The two processes of acquiring categories and predictions go hand in hand.

We aim to provide a useful *specification* of the task and some of the needed processes in this paper. We explore the many challenges and indicate promising research avenues throughout the paper. We do not propose or investigate specific algorithms here, but we provide evidence for the feasibility of scalable learning in the face of a complex world. We hope that this work can serve as a useful guide and framework for thinking about large scale learning systems. We believe that now is a good time to build and experiment with prediction systems.

We describe infinitely rich worlds and prediction games in Section 2. We describe our notion of categories and their fundamental role as building blocks. In Section 3, we discuss several aspects of "game design". In particular, we explain and motivate our choice of objectives or evaluation criteria. We explain what we mean by a *systems* approach, and motivate that choice. We discuss desiderata on solution techniques and some of the challenges we see, such as scalability and prevalence of noise. In Section 4, we give the wider context and motivation for prediction games, including considerations of early learning in higher animals. We situate prediction games with respect to various learning formalisms and tasks, and discuss the scope of the games and some limits, as well as potential applications. This paper substantially expands on our earlier work [29].

2. Prediction Games

Prediction games are proposed to make massive learning possible. By *massive learning*, we mean learning in the order of millions of categories and beyond, and estimating related parameters, nonzero connection weights that serve different functions such as prediction, in the order of billions, and beyond. The learning processes that we explore involve aspects such as extended or long term (or life-long) learning, cumulative learning, and autonomous development [50, 38]. Prediction games consist of a world and a system that plays the games in it.

We will use prediction over a long stream of text, for example from all the available web pages, as our main source of examples. In particular, consider the following fill-in-the-blank(s) game: every segment of text such as a sentence or passage in the online text can serve as a source of several learning episodes. In each episode the system hides a portion of an input sentence, say a character or a word, and sees how well it can predict it, using the context derived from the rest of the sentence and possibly the broader context such as the passage, the page, and so on. For example, in the sentence, "I rode my bike to school", the word "bike" could be covered, and then the question posed by the system to itself is what can replace "?" in "I rode my ? to school". The answer may be in the form of a single phrase,

or several candidate phrases ranked (*e.g.*, "bike, vehicle, motor bike, car, horse, table,..") or assigned probabilities. More generally, the candidates are *categories*. In typical statistical language modeling tasks [43, 19, 13], the game is *predict-the-next-word*: words are predicted and the context is often limited to the words occurring before. Prediction games are meant to extend the scope significantly. In particular, the categories learned exhibit structure, as we explain next.

2.1. Categories

We borrow the notion of a category (concept[2]) and some of its associated properties from cognitive psychology. Categories are fundamental to intelligence [36, 48, 26]. If they are effectively learned and used, the problems of sparsity (rare events), ambiguity, and invariance in domains such as natural language processing and vision are addressed [33, 14]. Categories are the building blocks (the cells!) of intelligence.

We will develop this framework by focusing on only some aspects of categories that we see fundamental. These aspects can be seen as especially useful for pattern recognition and generation tasks. For us in this paper, categories begin from the very low level of directly observed patterns. Example such categories can be a single bit, or a single character in text. The ability to detect the lowest level categories are provided to the system from the outset, *i.e.*, they are hardwired or programmed, meaning that their detection in a learning episode is achieved by some component, say a sensor, that is part of the system from the beginning, or given to it at some point. In prediction in text say, if we start the games at the level of characters, then any observed single character such as *j*, <, and "." (the period character) are categories.

Many other categories are synthesized or built[3] by the system in the course of playing the games. These categories can be abstract and composite. We will refer to them as *high level* categories. The system builds these categories out of other categories already in the system. Example high level categories are *phone number*, *i.e.*, a sequence of characters that corresponds to a phone number, phrases such as *new york*, and types of web pages such as *home page* and *news article*. Note that the prediction system simply assigns ids to categories. As may be expected, to communicate the categories to humans, in domains that are familiar to humans, the categories need to be examined and named explicitly by humans. Not all the acquired categories, though useful, may be human understandable or have a concise description. Also, one can imagine a foreign language or an unfamiliar domain.

Categories correspond to recurring patterns. Within the system, one or more *nodes* in a graph or network may be allocated for the category. Conditions for the *activation* of a node

[2]In cognitive psychology, a distinction is made between a category in the external world and the concept of it, as represented internally in the mind [36]. This is an important distinction, but in this paper, to simplify the presentation, we don't make the distinction.

[3]We don't mean necessarily that a central process in the system intentionally or explicitly builds categories or predicts with them. Various parts of the system, algorithmic processes, achieve these functionalities.

include when the corresponding category is present in the input and when the category is predicted or anticipated (see Figure 3). We leave out the details of possible implementations and algorithms in this paper, but later discuss the nature of the solution approach that we expect is necessary.

The low level categories may elsewhere be referred to as (low level) features (or patterns). However, we call them uniformly categories: it will be difficult to draw a line. More importantly, we expect that the role of categories will be, at least roughly, uniform whether they are at the low or at higher levels: we expect similar processes to work on categories regardless of their level. We refer to them as categories, instead of features, nodes, classes, or patterns, to emphasize the relations that they enjoy with one another. The use of the term category (or concept) connotes this aspect best.

2.1.1. Grouping and Composition The following are two principle themes that we see particularly useful and feasible for building higher level categories:

- Composition (roughly, constrained conjunctions, also "part_of" relations)
- Grouping (roughly, disjunctions, also "IS-A" relations)

Composition of categories leads to the discovery of "bigger" categories that the system has to identify and predict in its input stream. Composition for us is a fairly general task that includes learning subclasses of probabilistic finite automata, such as strings. More generally composition involves learning spatial or temporal relations between categories in order to form new categories. For example the category *new* is built out of the three categories *n*, *e*, and *w*, and the category *new jersey* is composed of the categories *new*, " " (blank space), and *jersey*. The category *resume page* is composed out of several subcategories corresponding to say *contact information* (region of the page that has person contact information such as address), *education*, *skills*, and so on. The category *volleyball spike* as a visual category in the game of volleyball includes visual sequences of movements that correspond to hitting the ball in a certain form. Composition corresponds to the *part-of* relation: *n* is part of *new*. Composition necessitates detection of a *conjunction* of subcategories in the following sense: a sufficient number of the subcategories should be present in the input stream in some constrained configuration (satisfying a mixture of spatial and temporal constraints).

Grouping may be viewed as a kind of a generalized disjunction. For example, a useful category in text may be one that corresponds to the presence of any of the digit characters *0* through *9* (a disjunction of all the digits). We may call it the *digit* category. The days of the week (*Monday*, *Wednesday*, ..) and the set of the to-be verbs (*is*, *am*, *are*, ..) are other examples of groupings. Groupings correspond to the *IS-A* relation [11] (or *type-of* or *member-of*). Thus *Monday* is a *weekday*. Groupings, when effectively discovered and used, allow for abstraction and address some of the challenges posed by invariance and sparsity.

New categories can form by using the above operations in a nested or recursive fashion. The *phone number* category corresponds to a composition of *digit* categories and other

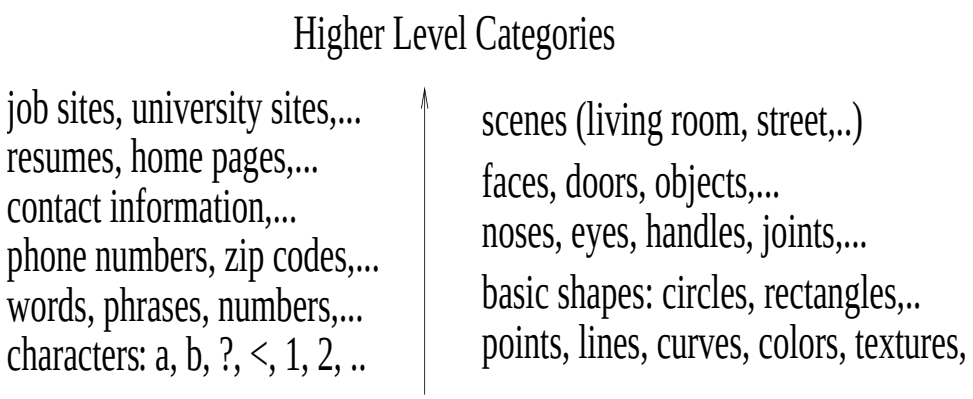

Figure 1: The hierarchical nature of categories in two familiar domains (text on the web and images). Here, some example categories from the domains are shown. Rich worlds exhibit hierarchical structure.

categories, occurring according to certain sequential constraints. The *digit* category is in turn a category formed by grouping, as explained earlier. Similarly, the *human face* category in vision is a composition. By a *higher level* category, we mean a category that involves at least one of the operations of composition or grouping, though this does not necessarily imply a particular hierarchical or leveled representation of such categories within the system. Higher categories are in general more abstract and bigger (composed of more parts) than lower level categories. We also note that a category can participate in many groupings and compositions.

Categories thus exhibit a hierarchical or recursive structure via use of composition and grouping. Figure 1 gives some examples in familiar domains. Such structural patterns are not a coincidence: these categories originate from complex but modular and hierarchical worlds/systems [8, 44]. We expect that this meta level regularity (across domains) is an important aspect of categories, and an important factor in making efficient massive learning possible. This aspect deserves further study, in particular, how such properties can aid efficient learning and in general interact with learning, and how to more precisely characterize and formalize such worlds.

2.1.2. Discussion There is an infinity of ways in which to carve up a rich world into various categories. The prediction system uses limited but efficient means to build the higher level categories out of categories that it can already detect. Candidate categories for the most part should be efficiently discoverable and should have a utility for the system. Here, the main purpose is improving overall prediction ability (Section 3.2). Note that categories that involve grouping are never directly observed. This poses challenges (Section 3.5.2), though efficient effective algorithms likely exist (considering nature appears to have solved the problem), and the value of such categories (abstractions) is immense.

One can conceive of other sources of new categories. We have mentioned conjunction and disjunction, but how about the negation of an existing category? A negated category is not present in the current input. Uncontrolled negation leads to loss of sparsity in description of the context[4], and thus inefficiency. It is possible that limited forms of negation may be achieved efficiently. Another candidate useful method of acquiring new categories is via refinement or specification. The category *party*, corresponding to the word "party", has multiple meanings including one related to politics and others related to gatherings for celebration. Its detection may be split into two or more internal categories corresponding to its various meanings. Ambiguity is pervasive and an important challenge: all directly observed categories in our familiar rich domains (text, vision) exhibit some form of ambiguity, *i.e.*, they can have different meanings or serve different uses in different contexts. Ambiguity may be addressed in part by means such as effective use of context including use of higher level composed categories. For example, the category *"party"* as part of the category *party cake* is no longer as ambiguous. In this work, we will focus on category construction via composition and grouping.

2.2. Infinitely Rich Worlds

The world is an integral participant in the equation of massive learning. An infinitely rich world enjoys the following properties:

1. A source of unlimited information or experience.
2. Rich in myriad regularities, in the form of categories.
3. Numerous complexities and challenges (*e.g.*, the scale of learning, ambiguity, noise, variety, ..)
4. Properties that make efficient learning possible (*e.g.*, the hierarchical or modular nature of categories).

Ample learning experience is a prerequisite for achieving massive learning, and we will emphasize this point repeatedly throughout the paper. The regularities in the world are perceived by the system in terms of recurring patterns, and many of these recurring patterns form recognizable *categories* within the system. An aspect that makes learning categories feasible is the special regularities or structure that the categories enjoy, in particular their hierarchical form, as described.

Infinitely rich worlds are complex and present many problems on one hand. On the other hand, they provide ample experience and enjoy much regularities, *e.g.*, at the cross-domain (meta) level of enjoying similar hierarchical structures, which makes massive learning feasible. Together, these opposing aspects make things interesting.

[4]Sparsity here is a desirable attribute, and means that the number of active categories at any given time point is relatively small (manageable), which is important for efficiency.

Good examples of rich worlds include our every day sensory worlds (visual, tactile, audio, ..), and the character streams from all or a subset of web pages. Probably significantly less rich in regularities is a time series obtained from the stock market. Much less interesting are an infinite string generated by a regular expressions or a purely random sources (simple probabilistic finite-state generators).

2.3. An Abstraction of the Games

A picture of the interaction between the system and the world is shown in Figure 2. The prediction system begins with capacity for detecting the lowest level categories, thus "sees" the world in those terms. It plays the games and gets better at predictions, in part since it learns to predict and recognize higher level categories. It learns to see the world, the stream of information that it gets, at higher level categories (bigger chunks)[5].

Each prediction episode consists of constructing and using a context vector in order to predict. The vector consists of the categories that comprise the mostly local context. The mapping from such vectors to the outcomes is learned. Initially, when playing the game at the character level, for the question mark in the input "n?w", the categories *n* and *w* can form the context, and *e* or *o* may be among the top predictions. Later high level categories are formed, and the learning trial may be "? is the time.", and a good prediction might be *Now*. As soon as a category is formed, it can serve as a target to be predicted whenever it occurs, and as a feature for predicting other categories. When categories serve as features in constructing context vectors, they may be distinguished based on factors such as the position they occupy in the episode.

Thus categories serve as both classes (to predict) as well as features (attributes or predictors): Features and classes are basically two sides of the same coin here, and prediction games may be viewed as a process for feature induction. Processes within the system build new higher level categories. Thus the set of categories grows and can easily exceed millions. Even in our familiar domains, such as natural language, the number of useful categories can greatly exceed our explicit linguistic vocabulary: the larger context serves to specify and disambiguate, and in general ease communication.

There are many challenges here, and we explore some of them in the next section. There are also details that we leave out, such as whether lower and high level categories may participate in the same context vectors, the choice of ordering of hiding the various categories or how the input is scanned, and whether one or multiple categories are hidden in an episode, whether only the preceding history forms the context and in general the extent of the context and what it may include, and so on. Some of these choices can make good

[5]It is possible that at the lowest levels (say bits or characters), the world appears random, while at some higher level, the world is indeed regular. Not all types of regularities can be captured. Conversely, what may seem a regularity during the lifetime of the system may turn out to be coincidental and impermanent. These touch on fundamental limits of induction itself as well as induction methods (see Section 4.12.1). We assume the system is given adequate sensors and the world, or an "important" part of it, is *modular*.

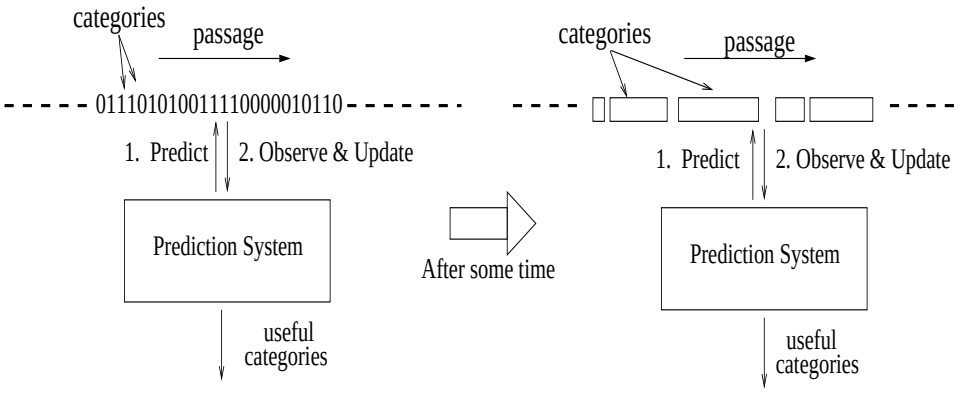

Figure 2: The world as an infinite stream of bits. The prediction system begins by "seeing" the world at low level categories, 0's and 1's here. With experience (learning), the prediction system sees the world as a stream of "bigger" categories (bigger chunks) as well, and plays the game at multiple levels. Seeing means predicting the next category and then (often) determining whether the prediction was true by observation (matching) and updates.

research problems, and others can depend on the domains and the end uses. The major point is that these games provide ample learning episodes.

3. Game Design

In the last section, we described the games and the worlds, in terms of learning categories coupled with learning their inter-relations, in particular in terms of how they help predict each other. We claimed that the process of playing the game makes possible the desired phenomenon of learning higher level categories. We also claimed that learning higher level categories can help improve predictions.

In this section, we explore several aspects that may be referred to as "game design". This includes determining the appropriate specification for the entity that plays the game, or the nature of the solution approach: is it a single algorithm or algorithmic process, or should it be best thought about as multiple processes? We discuss properties of the algorithms and systems that we think are appropriate and possibly necessary for the required scale (*e.g.*, online, incremental, sequential), and we identify further challenges. We also explore evaluation criteria or objectives for playing the games so that the desired outcome, in particular the learning of higher level categories, may be achieved.

3.1. Why Systems?

We have used the term system several times. Here we describe what we mean by a system and why we see a systems approach as necessary. Systems often enjoy the following

properties, with the first being a defining property:

1. Multiple interacting parts, working in concert, in our case: numerous categories, algorithmic processes, control mechanisms, different types of memories, and so on.
2. Persistence: operationality for long or indefinite durations.
3. Exhibiting complexity.

The diversity of categories and relations that could hold between them, and the evolving nature of the world as well as the internals of the system, makes statistical modeling challenging and likely inadequate. Similarly, a single algorithm cannot do all the important tasks, such as building categories as well as adjusting weights for improved predictions. A prediction system does not try to model the world, in the sense of building some kind of replica of it internally, nor estimate parameter values for a model within some constrained family of statistical models. Exploring the space of *systems* may be the best strategy for finding effective solutions. In this mode of research, various tasks need to be identified and adequate algorithms developed for them. We have mentioned for example the task of prediction and two major category building tasks as candidates, and each of these may translate to one or more problems that require their own algorithmic processes. Our intentions is for other functionalities to be added as they are identified and assessed useful. In general we must develop and abide by system designs that allow for such extensibility. The various parts should work in concert. An important aspect in this regime is that one should keep in mind how solving the different subproblems may contribute to the performance of the overall system. This constraint can be a good guide to problem discovery and formulation, and contrasts somewhat with most traditional research that involves isolated problem formulation.

A system, for example the operating system of a computer, consists of a number of interacting components and is driven by a number of (online) algorithms responsible for tasks such as paging, job scheduling, and I/O. Similar to an operating system, issues of operationality or maintaining some level of service, long time persistence in general, also holds for prediction systems. The difference from an operating system is that the prediction system learns and grows internally and in its functionality over time. Finally, we expect that the patterns of interactions with the world and within the system may lead to interesting complexities that often follows such interactions.

There are many challenges here. For example, how can we design algorithms and systems that remain robust for relatively long durations? There will be many errors, for example in building high level categories (Section 3.5.2). How do we ensure, in designing the systems and algorithms, that the errors remain in check? We see the learning to be primarily online and incremental (Section 3.3). Then, should we worry about converging to regions of poor performance or dead-ends? How should the system validate itself? Immediate feedback from predicting and observing may allay some of these concerns. Would the feedback from observing the world suffice? In general, typical issues in learning with finite data, such as

model selection and issues of sample complexity, may take a different form or change in their relative importance in this setting of unlimited data and long-term learning. Numerous new problems will surface. We explore several challenges in this paper, but it is important to build predictions systems in order to better identify the problems.

3.2. Performance Assessment

Operationality includes efficiency, robustness, and accuracy. We will touch on efficiency and robustness later in Sections 3.3 and 3.4. Here, we explore the evaluation criteria that may drive a prediction system into learning high level categories and higher order regularities, and that may serve for assessing a system's prediction performance.

Why should a system learn higher level categories? We explore the benefits first, which then leads to evaluation criteria. The benefits can be broken into two camps.

1. External to the system: the "high order bits", corresponding to predicting higher level categories, can be more valuable for various tasks than lower order ones.

2. Internal to the system (independent of outside uses): learning various regularities improves the system's predictions.

For a biological organism, it is very valuable to detect/predict a bit corresponding to the presence of one's mother or the bit corresponding to the likely presence of some dangerous predator, regardless of what it is. These are high level categories or regularities, and they are useful for reasons external to the prediction system. However, external use has drawbacks as an evaluation criterion. For example, it is not clear how it may drive the system to learn the category of interest in the first place, especially when the number of useful categories is huge, many of which are unlikely to be known to the system or to the system creator before hand (but see Section 3.2.2 for how it may influence what information is seen/processed). How can a biological prediction system know that learning to detect one's mother is important before being able to detect it[6]? One may say the system will simply be designed to discover/learn higher and higher order regularities, with the faith that such discovery will be useful. Ultimately, that is our underlying motivation:[7] a "blind" massive "search" for discovering regularities is a good thing, and furthermore feasible. Still, this leaves something to be desired: in comparing systems or algorithms, we would need to have an external use. We want to compare two systems after they have processed say 10^{12} bits of information from the same world, and subject to similar constraints, independent

[6] In biological systems such as mammals, it may be argued that say evolution programmed the desirability of learning certain categories (or furthermore programmed their detection). It is unlikely that evolution could encode substantial functionality of this type in the genes when the number of categories is large (especially, in the light of the very limited number of genes). In any case, it may be possible to have useful learning/driving criteria independent of the external use of the system. We explore such possibility here.

[7] However, this is not the only conceivable avenue. Another approach is for a more directed search for learning categories that would be useful could take place. However, we don't see how such an approach is feasible for the learning scale we see necessary.

of external uses and applications. We would also like to measure the progress of a single system as it learns. Let us explore "internal" desiderata or goals. How can we define the performance in terms of prediction quality?

At the lowest level, we could just set a measure of accuracy as a prediction criterion. Each lowest level category (*e.g.*, bit or character) is predicted then observed, and we could measure the average fraction of a bit correctly predicted per prediction action. But why should high level categories be learned?

Predicting that a sequence of numbers corresponds to a *phone number* is probably important for detecting that a region of text corresponds to *contact information* (a higher level category than *phone number*), while knowing the actual digits (accurate predictions at the lower levels) is not as useful. This kind of utility is not entirely satisfactory as a prediction criterion either, as it is the use of a lower level category for predicting a higher level category (*contact information*, which includes *phone number* as a part).

3.2.1. Internal Criteria and Category Benefits What is the use of higher level categories in terms of predictions at the lower levels? Does learning to predict higher level categories improve predictions for the lower categories? If so, how? The potential benefits that we see from successfully predicting higher level categories may be listed as:

1. The system that predicts bigger portions of the world, has captured the regularities in the world better.

2. The predict-observe-update cycle for bigger chunks takes less overall time and system resources than the predict-observe-update cycle for an equivalent number of smaller chunks.

3. Higher level categories can help define the *context* better and disambiguate, for predicting lower level categories, than only using lower level categories for predictions.

4. Higher level categories can allow for compression (space savings), and generalization

Benefit 1 may be viewed as a principle ideal of learning: the more you can predict into the "future" (what will be seen next) the better, as it is a sign that you "know" your world better. Higher level categories can be "bigger" than lower level ones, as they involve compositions, therefore learning to predict them (in one shot or single prediction action) amounts to capturing more regularity. This benefit is related to learning as compression. But this benefit as it stands appears somewhat circular, and still does not yield an explicit criterion.

Benefits 2 and 3 are more practical. Each learning cycle, consisting of prediction, observation, and update, takes time. If accurate larger predictions are possible, it would be more economical to predict at such levels, as the total number of memory accesses needed (for recalls and updates) are fewer.

Here, we are making an important assumption: categories are assumed to be *atomic*, for operations such as memory access and prediction. We need not assume this for matching. This simplifying assumption may be valid. Consider the situation in which the string "New York" has occurred and is to be predicted. A system that predicts piecemeal one character at a time, first predicts *N*, observes and verifies that it is correct, updates the context and possibly its various internal parts, then predicts *e*, and so on. This can take more time and memory accesses than the system that needs to make the context once and predicts *New York* in one prediction action[8].

Benefit 3 directly addresses the "bottom line" (prediction at the lowest level): Assume the system is correctly predicting that it will see *area code* next, which is a high level category involving both grouping and composition (US area codes are three digits long, may have parenthesis around them, the first digit is not a zero, so on). Then the likelihood that a digit follows the first digit is more likely than the background likelihood of predicting a digit after seeing a digit (*i.e.*, only using the lowest level information for context). On the other hand, after seeing the third digit, it is unlikely that a digit will follow immediately. Similarly, the category *new* (composition of *n*, *e*, and *w*) better constrains the category to come next, than the sum of the predictions of the individual categories[9] *n*, *e*, and *w*. Thus higher level categories, in particular categories involving compositions, can improve predictions by providing better context or constraining the context for predictions at the lower levels.

Higher level categories, and in particular those involving grouping (abstractions), allow for space savings. For example, if every digit predicts basically the same set of next categories with similar prediction weights, then a single category, corresponding to the set of the 10 digits, can be instead used for prediction. That is, a particular digit first activates the general group digit, which then predicts the next category (Figure 3). The difference in number of edges can be linear versus quadratic. While this transformation may lead to some loss of prediction performance, at least to what has been previously processed, it can help *generalization* by removing unimportant variations that can due to noise or can be insignificance to the larger application the system is being used for. Prediction accuracy on streams that may have somewhat different distributions can also improve (versatility).

Considering benefits 2 and 3, the following prediction criterion appears to be a good candidate for capturing what would guide our design and evaluation of prediction systems

[8]With higher level categories, the number of possible categories may be higher, so a look up of memory may take longer. But this memory look up may not increase significantly with increasing number of categories (see Section 3.5.1). Our experience with scalable categorization provides some evidence for the assumption of atomic categories.

[9]We are making a linearity assumption here: prediction is achieved by a linear function of the predicting categories (Section 3.5.1). The creation of higher level categories achieve the "non-linear" effects.

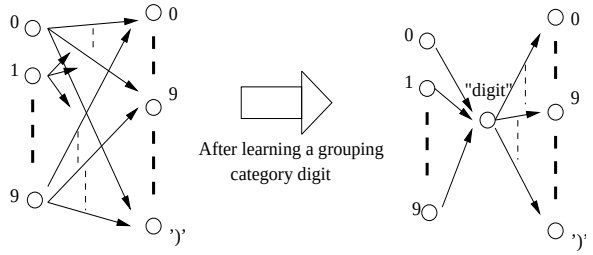

Figure 3: Grouping can lead to space savings. Here, we assume all digits basically predict the same categories to appear next (mostly digits, but also, periods, dashes, etc), with similar prediction weights. Then the different digits can first activate a single node, corresponding to the *digit* category (discovered via grouping operations), and that node can predict the same categories. If the out-degree was about $|E|$ per digit before, the number of required edges goes down from $10|E|$ to $10 + |E|$.

and algorithms:

Maximize the number of correctly predicted bits per unit time.

We may assume each prediction action has roughly the same cost, therefore a simplifying substitute is:

Maximize number of bits predicted reliably per prediction action.

These objectives are to be achieved subject to appropriate constraints, and in particular efficiency constraints (space and time). There can be close variations that may be more useful for evaluation and measuring progress, such as relaxing the matching of the predicted category against the actual category. One may compute the proportions of bits that match via an appropriate matching procedure. For matching candidates, one may consider all the top k predictions, for some small $k > 1$, instead of merely the top most. Finally, different bits to predict can differ in their importance (see next section). The question is whether the above evaluation criteria adequately account for learning of categories (regularities) of various kinds, for example at different levels of granularity, and in various complex domains, such as the visual. This remains an open question and may perhaps be best settled by experience in building prediction systems. See also Section 4.11 for a discussion of the relations of the above proposed evaluation criterion to evaluation criteria commonly used in language modeling, such as (lowering) perplexity and word error rate.

One aspect and motivation for a good objective is to identify a clean problem and detach the prediction task from other relevant or complementary tasks. The subject of how

a prediction system may work with other systems is another promising research area that we touch on next.

3.2.2. Control of the Input Stream What if the system just focuses on a very regular (easy) part of the world, a part that corresponds to a string concatenated ad infinitum? In the visual domain, imagine the camera of the system being set to point to a corner of a wall. In such scenarios, the prediction system can score very high according to any of our proposed evaluation criteria.

The choice of the input stream is external to the system. In vision for instance, this is the problem of how to scan the scene, for example in which order and where to focus. Therefore, if the system is part of a larger biological organism say, survival is very important, and the stream that is fed into the system, the choice of input, should be a function of survival utility. The prediction system's outputs however can certainly influence how it is controlled. The information provided by the system, or derived, includes:

- No more improvements (via learning) appear feasible.
- The prediction confidence is sufficiently high, or more accurate predictions are not useful.
- Movement is required for disambiguation and verification.

The interface between a prediction system and other systems that use its outputs or control its inputs should offer a fruitful area of research. Topics such as attention, active learning, and seeking of novelty are relevant. See also Section 4.6 for the related subject of modularity.

3.3. Scalable but Capable

Scalability is paramount. The system is to operate in an information rich environment, and what is to be learned, an operational predicting system, requires by design information hungry processes. We expect that algorithms that are incremental, sequential and online play a major role in the functionality of a prediction system, due to the following considerations:

- The utility, necessity, and availability of ample data (in general experience).
- Memory efficiency
- Time efficiency
- The evolving and changing nature of the learning
- Continual availability of a functioning system

We briefly clarify what we mean by online systems and algorithms here, and then argue why such properties are important. Online algorithms basically perform their computations in an streaming manner, *i.e.*, the data points in the stream are not revisited. Some algorithms compute certain important statistics online, but may yield a solution only after the input sequence is over and perhaps some post-processing is performed. Here, we also have in mind the "any-time" property: at any given time point, the product of the algorithm, for instance a classifier, is available. The algorithmic processes work within a functioning system, incrementally changing it. Note that the former type of algorithm can perhaps be easily converted into the latter type.

3.3.1. Data Ample data is necessary due to the breadth of what is to be learned. Other factors such as noise also contribute to the amount of data required. See also Section 4.12.2 on the substantial data requirements of higher level categories.

In some scenarios, for example learning to predict by processing the online text, there is much data available, and the data is relatively static. It is the system's (algorithms') speed that determines how fast it can learn from this abundance. In real-time scenarios such as in vision, the system is inundated with information, and may have to ignore (drop) much incoming and possibly valuable information due to its processing limits.

There is a trade-off here: on one hand the system can spend much time on the current instance (learning episode), or it can memorize and revisit past instances, perhaps optimize some measure of accuracy over them, or, on the other hand, to spend the computational resources for further exploring the world and acquiring possibly new useful information. Recent work in large-scale text mining has raised some of these issues [40, 4], and has pointed to the utility of ample data. We expect that the benefit of online algorithms and systems in being able to process abundant streaming data overwhelms the drawbacks.

3.3.2. Time Efficiency Batch learning or optimization can provide significant improved accuracy in typical learning problems, when compared to online methods. However, batch techniques are designed inherently with finite data in mind, and prediction games are about unbounded data. Researchers have pointed to both the theoretical and empirical advantages of properly designed online classification algorithms over batch, in terms of accuracy achieved, when training data is abundant (*e.g.*, [7]). While one can imagine techniques such as subsampling and instance selection (*e.g.*, via some measure of instance utility) to keep things sufficiently small, and using incremental or staged batch learning and optimization, this approach appears complex, and may not be the best utilization of learning time (see also Section 3.6 on code complexity).

A quick calculation demonstrates the considerable potential advantages of light linear time online learning. Consider algorithms A, B and C taking respectively n, $n \log_{10} n$, and n^2 steps to learn from n instances. Then in the time that algorithm B takes to process a million instances, algorithm A can learn from six millions instances, while algorithm C has only processed a 1000 instances. Of course, the advantages of linear time processing grows

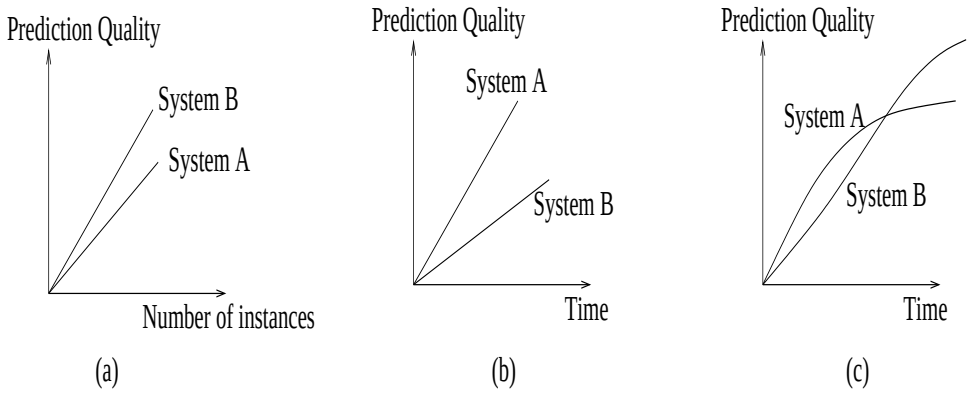

Figure 4: The tale of two systems, in three scenarios.

with increasing n. One has to ask whether the accuracy benefits of a more time consuming algorithm is worth the opportunity loss in learning from more data. One such opportunity can be discovering better features that improve prediction accuracy.

Let x denote the feature vector for the current learning episode. By linear time learning we mean processing that takes basically $O(|x|)$ time per instance[10]. Even within linear time processing settings there are choices, and one should keep the tradeoffs in differential processing time in mind. Imagine a system B taking 10 times longer per instance than a simpler system A as system B uses more sophisticated feature extractors or many more newly discovered concepts or deploys sophisticated inference processes. Note that there is a choice in constructing the feature vector, such as how large a context to use. The lighter system A may still exhibit superiority since it can learn from ten times as many training instances in the same period. More sophistication, for example in improved feature extraction or in inference, is likely needed at some point to extend the reach of the prediction system, but a challenging problem may be determining a good balance. For best results, a system may have to learn to deploy increasingly sophisticated processing only as needed, and perhaps involving pipelined, distributed, and parallel processes. The field of perceptual leaning in cognitive science has investigated learning of new features for improved recognition as well as learning to speed up task accomplishment [25].

3.3.3. Space Efficiency In general, finite memory will also be a major constraining factor, informing the design of systems and algorithms. The significant advantage of online algorithms here is that one need not store the instances: instances are ephemeral. An instance (a learning episode) is extracted in part from the portion of the stream currently

[10]There can be dependencies on other parameters, but the expectation is that other costs are tolerable. For instance, a $\log |C|$ factor (a processing time of $O(|x| \log |C|)$), where $|C|$ denotes the number of categories, may be feasible.

being processed[11], used for learning, and then discarded (its assigned memory recycled), and onto the next instance. Most of the memory will be in terms of the categories acquired and the connections among them. Thus considerations need to be made regarding for example the number of (nonzero) weights that will be learned, and the memory consumption and its growth rate during learning. Research on online computations and streaming algorithms share some of the objectives and challenges, and insights and algorithmic and analysis techniques developed there are relevant [15, 6].

Note that space constraints is not necessarily a negative aspect, and does not imply degraded accuracy performance. Space constraints also motivate grouping (abstraction) processes that address rare events and promote generality and versatility (Section 2.1.1 and 3.2).

3.3.4. Capacity for Continual Change and Adaptation Some online algorithms consist of an underlying process that basically converges to a single point, say a single model within a family of models. An example is estimating the parameters of a beta distribution for a coin, by observing the stream of the toss outcomes, and keeping track of the number of heads and tails.

Prediction games are fundamentally about change and development. In general, category connections of various kinds may be added, dropped, strengthened or weakened repeatedly during the course of the games, in processing that achieve grouping and composition. For example, categories may change their meaning, including what they tend to predict, over time. This can be due to the outside world, but also due to the inside dynamics of the system. The category *new* may often be followed by the category *york*, but when the category *new york* is formed, the category *new* may no longer predict *york* (as *new york* may be primarily processed as a whole). Likewise, many categories may be created and used for some time, then discarded due to lack of use.

Thus, we desire online algorithms and systems that can adequately adapt to changing needs and circumstances, or nonstationary distributions. This can require moving fairly fluidly from one model to another, behaving very differently at different times. A basic processing ability that can support a change of course or relearning, non-stationary distributions, and adaptations to the most recent observation appears to be crucial. This may be achieved in some cases by imposing in effect limited memories for the algorithmic processes. Some online classifier learning algorithms such as the perceptron and winnow enjoy the any-time and "adaptability" properties [42, 28]. Note that some level of stability is also desired. Moreover, learning and plasticity incur costs and complete plasticity may not remain desirable forever. In general, there will be a trade-off between fluidity or adaptability of an online system versus stability and fidelity to the past.

3.3.5. A Functioning System In some domains, such as in robotics, a system that is always functioning may be a necessity. However, we expect that the requirement of always

[11] Parts of the system, such as active categories, may also contribute to defining the instance.

functioning is more fundamental. The primary mode of learning of a prediction system as we see it is by *functioning*, that is by actually performing predictions and adjusting oneself via observing the outcomes. We see that a number of learning tasks benefit from this model of learning from interaction.

3.3.6. Local Distributed Streaming Computations Time constraints in various applications may require different degrees of distribution of computation (parallelism or concurrency) and pipelining, although we do not know whether the necessity for these attributes is more fundamental. Efficiency, and in particular the basic requirement of roughly linear or $O(|x|)$ time per instance of length $|x|$, may imply that the computations, such as prediction and building new categories, at least in principal, should be carried out mostly locally by nodes corresponding to the categories. However, not all nodes simultaneously process. This is not a massively parallel computation. The system "routes" the stream of bits appropriately. This routing is also learned and adapted over time. Thus, each node, can be viewed as receiving a subset of the full stream of bits. Here, we are also assuming that a node (or perhaps even the connections) can do fairly elaborate computations that may involve state changes, within a limited number of states, keeping and updating a number of statistics, and so on. In this respect, we agree with Valiant [51] that, if implemented in neural network model, it appears that we need nodes with more programmability (functionality) than the nodes studied in typical neural network literature.

3.3.7. The Tasks' Inherent Sequentiality Prediction games involve learning tasks that appear to be inherently sequential. Composition and grouping of subcategories only apply when the base categories have been learned, at least to some extent (the prerequisites met). One cannot reduce the order of the learning speed by adding more processors to substantially parallelize the task. This aspect of learning one concept in terms of others already learned has been referred to as cumulative and layered learning [50, 38].

3.3.8. Summary The constraints of efficiency as well as desired functionalities such as a basic capacity for change and relearning imply that the major parts of learning should be carried out by online incremental and possibly distributed (locally computed) means. In addition to prediction and acquiring new categories, various candidate useful tasks include management of memory consumption, learning speed ups in doing prediction or recognition, and learning to parse (*i.e.*, imposing a nested structure on segments of input stream). For whichever (learning) functionality that appears useful, we should ask whether/how to achieve it efficiently and primarily online.

One possible exception to the online processing paradigm is the potential need for "non-learning" processes useful for maintenance or certain book-keeping operations, for example process that are akin to garbage collection (Section 3.5.2). These tasks may involve processes that run concurrently online, or may require offline periods for the system. Note that there is a distinction between online versus offline processing of the learning instances,

versus other "management" tasks that may require offline times, to work on the internals of the system and in particular on what has been acquired (*e.g.*, the categories and their connections, but not instances[12]).

3.4. Robustness to Imperfections, Uncertainty, Variety

There are numerous sources of imperfections and uncertainty or noise. In prediction in text, features include phrases or categories discovered, and their relations. But phrases are ambiguous and there can be misspellings, or missing values, or imperfect segmenters, or inaccurate passages. The newly discovered categories are also imperfectly recognized (poor precision or recall). In vision (and other perceptual domains), sensors are noisy and objects may occur in different lighting conditions, or may be partially occluded or be viewed in differing orientations and distances. There may be considerable variety within the same object class. Therefore, uncertainty and variability may be even more problematic. Prediction games are played for developing robustness in a variety of conditions, to detect/predict categories in many of their guises. Therefore effective solution algorithms will yield a system that is robust in numerous ways. The prediction system should learn from the sum of all its experience, rather than putting too much weight on any single instance. A learning strategy may include ignoring (in effect) those instances for which the target category to be predicted, the observed outcome, is very uncertain. A similar strategy goes in using feature values. Ignoring and in general down weighing difficult episodes, at least temporarily, may be an effective partial strategy. Of course, this may require that the system knows when it doesn't know, at a satisfactory level (calibration): it has access to accurate confidence values. This is also related to a problem we may refer to as the *grounding problem*. The first time a system starts out, how could it be sure of any thing? We assume that the system is equipped with sensors or feature (category) extractors that work adequately, and they are sufficient to start the games.

3.5. Dealing with Categories

There are many challenges with predicting and acquiring categories. Two that we highlight and discuss briefly are:

- Efficient use of myriad categories both as predictors (features) and targets of prediction.
- Effectively acquiring categories that are not directly observed.

3.5.1. Large Dimensionality Categories serve both as predictors (features) and predictables (classes). This immediately raises the challenges of having to face huge and

[12]There can be exceptions, at least for the larger intelligent system, such as the need for storing some kind of episodic memory, *i.e.*, remembering explicit events with relatively high fidelity. This is beyond the scope of this work. We have focused on what may fall under semantic as well as implicit memory.

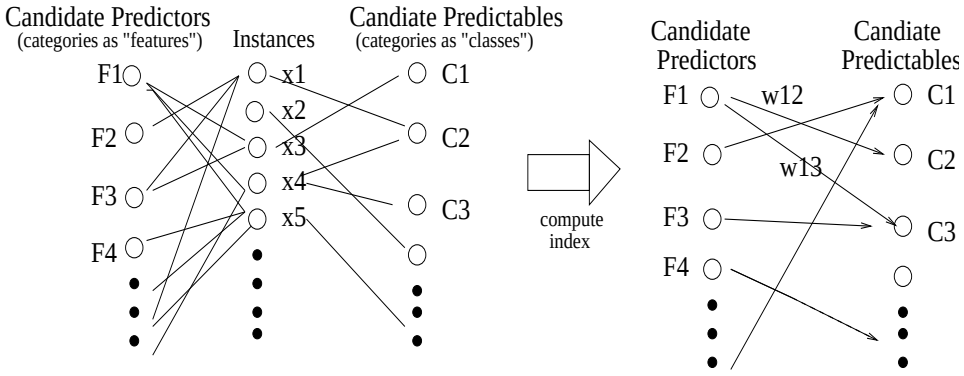

Figure 5: Computing (learning) indices, sparse bipartite graphs, for fast prediction in the face of many categories. Instances are ephemeral and function as "intermediates", leading to connections being formed, strengthened, or dropped, between the predictors and the predictables.

growing dimensionalities. In addressing large dimensionalities, one proven approach is to only use the active categories (predictors) in each instance, at least for the most part (a so-called "positive" representation). We expect that the predictions, in terms of efficiency and noise robustness, are best carried out by simple linear operations: aggregations of the predictions of the predictors active in a context in a linear manner. Existing classifier-based learning algorithms, several online ones such as perceptron and Winnow, have proved their value on numerous large feature set problems [42, 28, 13]. A new challenge is that the number of categories to predict can also be very large. Thus in prediction in text, while the number of single letter categories can range in the 10s (*a*, *b*, *)*, $\cdots$) , the number of two-letter categories can range in the 100s (*en*, *st*,*ci*, $\cdots$), and the number of three letter categories can range in the 1000s, and so on. While the growth is far from exponential, it is substantial.

How can a system efficiently categorize or predict given a context, and given thousands of categories that it has learned? It seems that this problem has found good solutions in nature [49, 20]. How can a system *efficiently learn* to do quick categorization? Or how may the wiring or routing of features to categories be achieved?

We require scalable sample efficient online algorithms that can handle large dimensionality in both spaces of predictors and predictables, as well as being tolerant of substantial noise/uncertainly and drifts in distribution, for example, in the feature-to-category mappings. Our work in this direction offers good prospects [31, 32, 30]. We have designed online memory efficient algorithms that, on problems such as large scale text categorization and text prediction, are competitive in accuracy with traditional multi-class methods, while enjoying advantages in efficiency. These algorithms can be viewed as learning an index, or a sparse weighted bipartite graph (or a sparse matrix, instead of a single dimensional

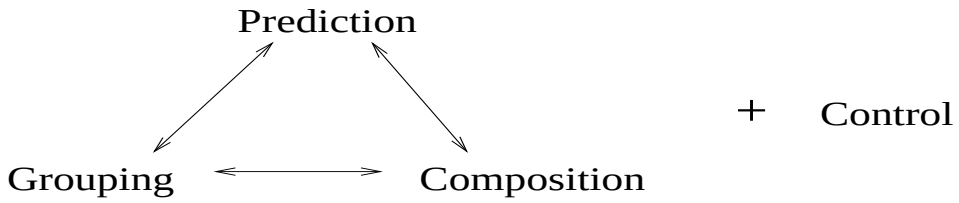

Figure 6: A depiction of the inter-dependence of the learning processes of prediction, composing, and grouping, and the need for proper control (*e.g.*, in timing the concept activations, defining and updating the context vector(s), and the order in which a chunk of the input stream is processed).

weight vector), mapping features to categories (Figure 5). The index is empty (no edges) at the start, and learning from the stream of instances leads to features connecting to categories that they tend to predict best, in a manner so that overall prediction performance is increased. We referred to a system that is quick in categorization (prediction) in the face of myriad categories a *recall system*. The functionality achieved is related to content addressable and associative memory tasks [3], though at a very large scale.

3.5.2. Unobserved (Internal) Categories Robust learning of high level categories that involve grouping operations is challenging as these categories are never directly observed. However, we expect that a large number of most useful categories in various rich domains would involve groupings! These unobserved categories are in a sense *hidden* in the data stream, or in another perhaps controversial view, they are not really there: it is only that these categories have much utility, for example, in serving to improve system predictions, that the corresponding patterns lead to formations of "concepts" inside the prediction system.

The problem here is also somewhat similar to the challenging problems of learning new nodes or their weights in hidden layers of neural networks. Here, the "nodes", corresponding to higher level categories, are created in a data driven online way, and their use (existence) will rely on the difference they make in predictions. Moreover, we expect that prediction and in particular co-occurrence of predicted categories is utilized in generating the grouping concepts. For instance, the digit category may be formed due to the system predicting the digits simultaneously (with roughly equal confidences or activation weights). Thus the prediction process not only benefits from higher level concepts, but contributes to forming them. A depiction of this interdependence between prediction, grouping, and composition, and the need for appropriate control processes (such as timing of category activations, and properly defining the context), is given in Figure 6.

Even if statistical tests (filters) are used for acquiring categories, there will be some percentage that will be faulty for various reasons. For example, a digit category may

be learned that also includes the extra category *b*. Other types of potential errors and difficulties include categories whose presence, while statistically significant, may be very short-lived, and (near) duplications . Thus a question is whether these problems could simply be ignored, or otherwise whether such imperfect categories can be efficiently detected and modified or removed. Here, we may in part need a "garbage collection" scheme for recycling unneeded categories and reuse the allocated memory. The garbage collection scheme may have to be run offline and periodically. The issue of acquiring categories is akin to clustering, though there are differences (see Section 4.8). We are currently investigating ways that higher level categories may form and evolve, and how they may affect predictions.

3.6. Program Simplicity

Ideals of simplicity, such as low code complexity and uniformity of architecture, should also serve as important guidelines to abide by in designing prediction systems and their algorithms. Perhaps most of the program complexity and diversity may be concentrated into the preprocessing and raw feature extraction components. The system will be learning a variety of regularities to improve overall prediction ability (in breadth, depth, and speed), but we hope that the number of distinct core algorithmic ideas utilized will be relatively few (perhaps in the order of 10s!). Much of the functionality and diversity in the system should be the result of learning and experience, as opposed to explicit programming. Simpler designs and algorithms can have a better chance of being efficient, and may be better understood. "Keep it simple, as much as possible" should inform the system design and the search for algorithms.

4. Discussion and Related Work

Prediction games liberate us from the training ("labeling") data bottleneck necessary in typical supervised learning, *i.e.*, explicit human supervision. The world serves as the "teacher", the source of feedback (validations and invalidations). Here, we briefly review considerations and work that motivated or influenced ours, as well as relations to various learning formalisms and tasks. The related literature is diverse, and we cannot be exhaustive. We hope, at a minimum, to touch on a sample of work that is significantly related.

4.1. Early Learning

Considerations of early learning, in infants and babies, is specially valuable. This stage provides the crucial foundation for learning and development in later years. Understanding how this foundation is developed from a computational point of view, *i.e.*, the nature of the major algorithms and organizing principles at play, is very important. Considering how infants and babies develop, in animals as well as in humans, one may conclude that:

1. There is much (massive) learning taking place during first months and years.

2. The learning does not involve (explicit) supervision.

We assume the above two statements are true[13]. See for example Ballard [3] on the enormous plasticity of the brain just before birth and during the first few months after birth. This massive learning includes recognizing myriad categories in various conditions (*e.g.*, objects, such as faces), learning the dynamics of the physical world, and becoming adept at bodily movements and control. The infant, in the first few months of its birth, may have indeed mined its world, including its developing internals, very effectively!

There is much that remains unclear regarding the nature of this learning stage, *i.e.*, what problems are being solved and what kind of algorithms are at play. Since at a minimum early learning appears not to be (explicitly) supervised, a major question is then how is this learning taking place? We expect that research on processes similar to prediction games is a promising avenue for pointing to ways on how massive learning of different kinds may take place in the young brain. We imagine such learning processes to be taking place largely under unconscious control, although specifying what is meant by unconscious remains challenging [26, 41, 9]. The outcome of this process may ultimately be a system that has developed a feel or a sense of a world that was once very unfamiliar and uncertain. We expect that this mode of prediction-driven learning and operation would continue its importance for the rest of life. We touch on this further next.

4.2. Brain as a Prediction Machine

The brain has numerous aspects and implements a variety of algorithms, but it has also been referred to as a predicting machine. The phenomenon and importance of predictions (expectations or anticipations) and feedback has been noted [3] (and see the popular reading books [35, 22]). In particular, Hawkins raises the role of prediction to another level and claims that making predictions is central to all of human intelligence [22]. For example, he states that the nature of understanding or knowing may be explained by the ability to predict. He gives many useful examples of how predictions could occur in different sensory pathways, and claims that similar mechanisms are responsible for generating physical actions. He describes at a high level how prediction-related processes may be taking place, by pointing to the circuitry and the organization of the hierarchical neocortex [22]. He provides valuable references to the relevant neuroscience literature. Assuming the premise, finding the robust algorithms that could achieve such functionality remains open, and we next touch on some of the related literature on computational approaches.

4.3. Algorithms and Architectures

There is significant work on online learning, hierarchical or layered architectures, algorithms that learn parameters for and/or hierarchies of various form, such as hierarchical

[13] Given the first assumption, the second appears very plausible, as babies' communication capability in humans and animals is limited.

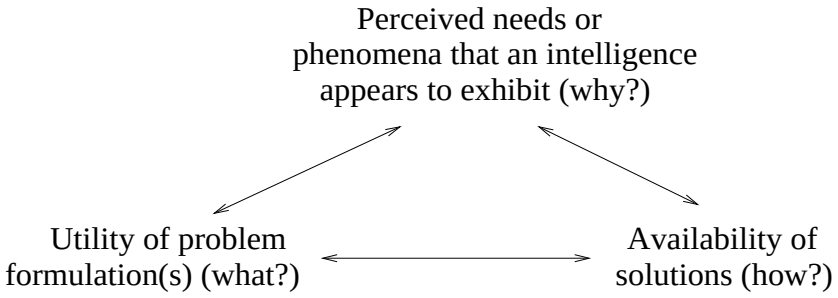

Figure 7: The inter-relationship between needs or observed phenomena (the why questions), choice of one or more problem formulations (what questions), and solutions in terms of algorithms, processes, or systems (how questions) in the process of research and discovery. Beliefs or expectations about one node affects beliefs and expectations about another. In approaching complex puzzles, it is important to visit all the nodes periodically.

Bayesian models, the potential benefits of hierarchies in learning systems, and the common phenomenon of hierarchies (compositional and taxonomic) in the world (complex systems) [18, 50, 16, 38, 5, 44, 3]. The proposed approaches share in some of the goals and consideration with our work here. As we have been focusing on an specification of the learning processes, we leave comparisons of solution techniques to future work. We note however that the uniform functionality, in prediction and category construction within a single learning system, together with the scale and flexibility that we see essential, appears not to have been addressed successfully before.

The problem of large scale ongoing learning is greatly simplified if one or a few unifying learning tasks or processes are identified. Prediction games as we have described provide a candidate. Stepping back, we note however that the question of what tasks or problems are useful is entangled with an understanding of what is feasible, *i.e.*, knowledge of (likely) existence of adequate solutions (processes, algorithms, systems, architectures,..) in addition to insights on how the world may be (to know what is most useful). Figure 7 is a depiction of this entanglement in a why-what-how triangle. The process of discovery does not necessarily follow the straight path from "Why?" to "What?" to "How?" although it may be useful to present it that way in hindsight. Frequently, it is the knowledge of or an expectation in existence of solutions (perhaps implicit) that leads to precise problem formulations. Marr [34] stressed the importance of asking the Why's and the What's in vision research. In general, in solving a complex puzzle say, it's important to revisit all three nodes periodically.

4.4. Psychology of Concepts

Categories are essential to our working minds [36, 26]. A category (concept) is a very general notion and the properties that categories exhibit, in the words of a cognitive sci-

entist, can be "maddeningly complex" [36]. In fact, in part it is due to this complexity that we conjecture that categories are acquired and deeply "internalized" through years of exposure and experience. This motivates long-term learning processes such as prediction games. We have focused on only some aspects of categories, ignoring others such as functional relations and the complex inference patterns exhibited with categories (conceptual combination, prototype reasoning, analogy and metaphor, ...). This work may serve as a starting point for capturing more complex phenomena.

4.5. Resource Constraints

Valiant explores a network model of the neocortex with locally programmable elements [51]. He stresses the importance of paying attention to resource constraints[14], such as constrained graph connectivity and processing speed, and shows how a number of existing learning algorithms, for various tasks such as unsupervised and supervised memorization and inductive learning, could be implemented on his model. He views the model and the algorithms as providing a candidate substrate for higher level cognitive computations, and believes also that much information acquired by the brain is through learning. A difference of our approach with that work is that we are content with the accepted high level programming model (higher level than Valiant's neuronal level), as long as the given algorithms are sufficiently efficient.

The field of bounded rationality [17], in studying human behavior, has also emphasized the roles of uncertainty, the issues of resource constraints, and the variety of implicit frequently competing objectives shaping observed human behavior, although in the context of high level human decision making. In playing prediction games, a system has to contend with significant resource constraints and uncertainty. We also expect that achieving some kind of optimization of the overall prediction objective subject to resource constraints is not the best place to put one's efforts on. A more useful goal is in understanding what determines satisfactory operationality. In particular, we expect that building successful prediction systems requires breaking down the task into sub-problems, each of which could then be approached somewhat independently but keeping in mind that the different parts should work together (Section 3.1). Very importantly, we think building preliminary large-scale experimental systems is feasible, and it is very useful to do so to help better see the challenges.

4.6. Role within a Larger System

Modularity in its various incarnations manifests itself in diverse complex systems such as the physical and biological systems, and in organizations and societies [8]. Software is written in a modular form by design, due to the benefits of modularity. An issue that has been subject of research is how and in what ways complex systems and organizations exhibit modularity properties, and how and why do they evolve into such configurations. According

[14] And in fact in a positive light, as such considerations can guide one to fruitful research avenues.

to H. Simon, modular systems are ones that exhibit *hierarchical* and *nearly decomposable* organizations [8, 44]. We already mentioned the modularity of categories (Section 2.1.1).

Pylshyn [39] cites much evidence and makes strong arguments that the vision system in humans, while achieving sophisticated tasks, *i.e.*, not merely feature extraction but high level category detection (*e.g.*, "table", "human", "my face"[15]), is a separate module, at least conceptually. In particular, he argues that it is not "cognitively penetrable"[16]. In this view, the control points and interface to the system are limited and well defined. They are mainly in the choice of the stream to feed to the system (attentional mechanisms), and how to perform final interpretations of the outputs of the system.

It is conceivable that the vision system is an instantiation of a prediction system (and the same for auditory and other sensory systems). The vision system may be the product of a great deal of learning, but at least after learning, it may be cognitively impenetrable. In general, it may be useful to view the prediction system within a larger system as a module with well-defined interfaces and control points. This mode of usage can be likened to using a camera (the main control point is where to point it to). System modularity has a number of benefits in terms of simplifying the overall organization and interactions and allowing for efficiency. Feasibility constraints motivate such an organization, but there can be some loss in flexibility.

4.7. Density Learning

Prediction games involve unsupervised learning, and in particular the outputs of the components may be interpreted as confidence values for or probabilities on variables (categories). In this sense, they are akin to density estimation or distribution learning. After learning a model of a distribution, one can in particular query it as follows: given the values or probabilities for certain variables, infer the values or probabilities for other variables of interest. Prediction games are played to achieve efficiency and accuracy in answering many such queries (presence or absence of categories). We believe this prediction capability can be achieved by efficient learning of sparse and large networks (see Section 3.5.1). Prediction systems are however restricted, *i.e.*, will perform best, for the type of queries (learning episodes) experienced in the world.

The use of graphical models has made density estimation and inference tasks more efficient, due to the focus on utilizing constraints on the type and number of relations that the variables of interest may actually have (*e.g.*, the actual dependencies in case of Bayes networks) [37]. Most graphical models (such as Bayesian networks) may turn out to be too constraining for learning the myriad categories and the variety of relations between them. Efficiently learning the nodes and connections may be an issue as well. On the other hand, graphical models allow powerful types of inference. The benefits of sophisticated inference versus its computational costs is a subject under the theme of computation versus

[15] But not naming the categories.

[16] Pylshyn explains that by "cognition" he means general purpose reasoning.

information: the tradeoff between extensive computation on the current situation, or learning instances accumulated so far, versus foraging for further information via new learning episodes.

4.8. Clustering

Prediction games involve both supervised tasks and techniques, in the sense of learning connections between predictor features and existing categories, as well as unsupervised tasks and techniques, in the sense of forming new categories out of existing ones. A major differentiating factor from the typical use of clustering is that we seek dynamic operational categories: these categories are to serve to improve predictions, and they interact with other categories to make new categories. The online nature of prediction games and scalability issues have consequences too. In prediction games clusterings of various kinds are achieved on *categories*, not instances (prediction episodes) as is often the case with traditional clustering. The basis for clustering will be co-occurrence and co-activation patterns of the nodes that correspond to categories, as opposed to instance similarity computations.

4.9. Limits of Human Involvement

It is very hard or impossible to anticipate and program the many relations that exist between categories. Human consciousness appears to be sealed off (for good reason) from the internal steps involved in what is made available to it. Significant portions of human knowledge (its effective storage and use) appears to be difficult to access or inaccessible. Thus, it is very doubtful that manual encoding of say a reasoning system is feasible, except for relatively constrained domains. These consideration have motivated the learning approach to building intelligence.

Supervised learning reduces human involvement significantly, and has enjoyed much success, but the issue of obtaining training ("labeled") data has always been a significant bottleneck. This issue has motivated much research on topics such as query learning, active learning, and semisupervised learning, in order to reduce the need for human/teacher involvement [1, 10]. Researchers have even developed social games to encourage humans to indirectly provide training data via playing games [2]. Approaches similar in nature include efforts to encode common sense knowledge via mass participation [45].

The goal of prediction games is to make possible systems that acquire in the order of billions of parameter values (nonzero connection weights). We believe that the amount of training information required to yield sophisticated capable systems makes classic explicit supervision infeasible. We also conjecture that massive learning is required to build the foundation for various levels of common sense. Finally, there is much to be gained in learning at this scale in numerous domains that are not the every-day familiar domains of humans.

Note that we are making a distinction between supervised techniques, *i.e.*, techniques that use a source of feedback to learn and improve (feedback-driven or prediction-driven or

discriminative learning), which is utilized in playing prediction games, and explicit supervision or training signal via a human teacher or some other costly source.

4.10. Reinforcement Learning

Reinforcement learning is also a kind of unsupervised or environment driven learning [47, 24], though the focus is not on learning regularities per se, but on learning good behavior, including actions or plans, for improved success in obtaining short and long term rewards or goals. In one view, in reinforcement learning, the whole problem of intelligent acting is addressed directly in a single shot (an end-to-end approach). However, the difficulties involved in complex environments may hinder this direct approach, and such obstacles motivate a more *modular* approach of breaking the problem into pieces. Thus, we see that prediction games can complement reinforcement (in general value or goal driven) learning. Acquiring prediction ability serves the goal of familiarization to one's complex world, thus to an extent separating the problem of "knowing" ones world from behaving well in it. Learning regularities directly and indirectly is geared toward helping the intelligent agent predict and obtain rewards, and in general to more successfully navigate its complex world.

Learning from prediction games can occur at a larger scale than reinforcement learning, as typical reinforcement learning requires taking action and attaining (possibly delayed) rewards. In the physical world, actions take time and energy. Prediction games involve mostly observations and information processing. However, from time to time, depending on the application domain, they may also require information seeking actions such as moving the camera (see Section 3.2.2).

4.11. Statistical Language Modeling

Statistical language modeling (SLM) also attempts to capture regularities in streams and in particular in natural language [43, 19, 23, 12]. It finds applications in various language technology tasks. A core problem is to assign a probability to the next word (in general, a symbol from a vocabulary of symbols), given a sequence of observed words so far, *i.e.*, given some history. Probabilities for larger units such as sentences and documents can then be derived.

A variety of techniques including neural networks, decision trees and linear classifiers and max-entropy models have been explored, but n-gram (Markov) models are the preferred method due to their scalability and simplicity. SLM also involves very large scale learning, and performance goals are also roughly similar. A main difference is that our interest in supporting different kinds of learning in an integrated fashion, and in particular in learning new categories from playing prediction games. The set of categories that a prediction system acquires is the vocabulary in the SLM sense, but this vocabulary is structured. The requirement of learning new and higher level categories is a major difference that we expect impacts the nature of solution algorithms significantly. The objective given in Section 3.2 is different from though related to the entropy or perplexity measures often

used in SLM evaluations, in part due to the fact that categories can be composed of other categories (*e.g.*, prefixes). For example, a system can simultaneously predict *n*, *new*, and *new york*, by ranking them or assigning probabilities. Assume all are correct. How should it be rewarded depending on its ranking or probability assignments, as all three answers are correct? Roughly, it should be reward by the number of bits that match, when its chosen single candidate (however the system picks its sole candidate) is compared against the input. This criterion is somewhat similar to the word-error rate, except that we are not just taking into account the identity of the category (the symbol in SLM) that is matched, but also how big it is. The issue of proper objectives is not settled, and see Section 3.2 for further discussion.

Prediction games are intended to substantially widen the scope and applicability of SLM and call for more powerful and flexible algorithms. Of course, prediction systems, incorporating whatever algorithms should still remain adequately competitive with traditional SLM techniques on the more restricted SLM tasks.

4.11.1. Understanding Language In playing prediction games in text, for example on all the information available on the web, our immediate goal is not obtaining a system that understands language or generates it the way humans do. The regularities available (say in the web pages) may not contain the needed common sense knowledge that was used to generate the text. More importantly, the nature of processes required for understanding remain very poorly understood. Work on prediction games can be an important step toward that. We do think the text available on the web is sufficiently rich to allow for significant massive learning of high level categories. This can lead to the creation of very capable pattern recognition and generation systems, with a number of applications (Section 4.13).

4.12. Scope and Limits of the Games

There are of course limits to what prediction games can achieve. In practical scenarios, the richness of the environment(s) that the system operates in, the finite capacities of the system and the powers of its algorithms, and the lifetime of its operation all set limits on what is learned. Higher level processes such as planning and reasoning, and predicate and first order logic, do not fall under the scope of the games, at least as far as we have described the process. However the products of prediction system and the prediction system itself may be used in service of such higher level processes [22, 26], and we may better see how such advanced functionalities are achieved once we gain experience with prediction systems. Furthermore, prediction systems may primarily learn concepts that have *finite extent*. These concepts may best correspond to a subclass of probabilistic finite automata, those automata without loops. In one view, we are focusing on breadth (the number) of things to learn, rather than the depth. The extent of the representation power of prediction systems is open, and efficiency and utility constraints will clarify these aspects.

We next discuss the connection with compression and the fundamental limits on induction, and then discuss practical limits imposed by sample size requirements, especially for the categories at the higher levels.

4.12.1. Description Complexity Capturing regularities allows for compression, and the fundamental connection between compression and induction has been extensively studied [46, 21, 27]. The Chaitin-Kolmogorov-Solomonff complexity or the minimum description length of a finite string is the length of the smallest Turing machine that outputs that string. An immediate consequence to note is the fundamental limit that this puts on induction capability: Computing the minimum description length is an undecidable problem. In our model of interaction between the system and the world, the string is infinite, and this relaxation does not alter the fundamental limits on induction capability. We cannot hope to design a general purpose prediction system that achieves the best compression outcome for the string it has processed so far, *i.e.*, captures all the regularities in that string in a way that yields a minimum description of the string[17]. Changing the setting to an infinite string and requiring online computations has not made the compression objective as stated easier. On the more practical side, the so-called "no free lunch" results also point to the futility of looking for general-purpose learning algorithms [52].

We have said the worlds we are interested in, while challenging in many respects, enjoy simplifying properties, such as the hierarchical nature of categories, that make the learning scheme we propose, which is in part "bottom up", likely feasible. In particular, at different levels there exist *local* regularity patterns that can be discovered by algorithms that essentially only look for and can only learn limited types of regularities. As higher level (larger) categories are acquired, the regularities discovered, local with respect to the high level categories, are no longer local with respect to the lowest level categories.

4.12.2. Sample Complexity Constraints and Limiting Behavior Higher level categories require more training data (experience) than lower levels, and possibly exponentially more with each additional level of composition, as higher level categories are simply bigger. If we assume each higher level category on average is a composition of b categories at the level below ($b \geq 2$), then instances belonging to the category at level l require $\theta(b^l)$ ("raw") bits to be specified fully, *i.e.*, each instance of such a category takes $\theta(b^l)$ space in the input stream, and requires at least the same amount of data proportionally, $\theta(b^l)$, to learn. However, in general, one of the say two subcategories may be at a lower level, thus the increase in category size may not be as fast as exponential. The number of useful categories can also increase with each level, and since category frequencies are likely to be far from uniform in various domains, improving prediction on an adequate number of categories at a given level

[17]The rough argument is: Turing machine that effectively used the prediction system together with a sequence of hints and corrections to the systems predictions, all internally, could reconstruct and output the original input string. The better the prediction system is at striking a balance between prediction accuracy and its size (at worst, it could just memorize the string), the smaller the Turing machine.

may again require substantial increased experience with level[18]. The presence of noise also always adds to the needed training experience.

First, these considerations underline the importance of access to abundant data. They also show that learning for higher level categories occurs at a significantly lower rate (perhaps exponentially slower pace) than learning for the lower ones. This is reminiscent of the slower interaction properties at higher levels in complex systems [8, 44]. Finally, this also draws attention to the limits that the life time constraints of a prediction system (if any) together with its finite speed and memory put on the scale of the patterns that a prediction system (which solely plays prediction games) could learn. This aspect also motivates naming categories and communication among intelligent organisms, to help accelerate learning and gaining knowledge.

4.13. Possible Applications

A major objective from playing these games is to acquire categories (recurring patterns) and, simultaneously, to learn about their interactions, such as co-occurrences. At the same time, another outcome achieved via this process is learning to predict in ones world better, via use of the many categories acquired. We see great scientific value in exploring this avenue. We summarize some of the possible applications in terms of current domains and tasks.

In the vision domain, playing the games could lead to acquiring visual categories, *i.e.*, robust object classification of numerous object classes in a variety of conditions. A similar possibility holds for other sensory modalities. Thus a prediction system, after adequate learning, may be viewed as an advanced pattern recognition machine. In general, we may envision a robot that is situated in an unfamiliar domain. Its own parts and how they interact with each other and with the external world may be initially substantially unknown to it as well. Through playing the games, the robot develops competency and familiarity with its world, internal as well as external.

In text, prediction games can lead to learning common phrases and expressions as well as their usage patterns, patterns corresponding to phone numbers and product numbers, and so on. Therefore, tasks such as language modeling and information extraction may benefit.

The techniques developed for acquiring categories may lead to new feature induction methods, useful for instance in improvements in playing computer games such as chess and GO via repeated play. In many unfamiliar domains, such as monitoring online activity of transactions, improved prediction can aid tasks such as anomaly detection.

[18] It is also conceivable that the average number of episodes required to learn a category at a given level significantly changes (increase or decrease) as a function of level.

5. Conclusion: Lets Play the Game!

We proposed prediction games for the goal of acquiring and effectively using myriad inter-related categories. Prediction games are about having much to learn and plenty to learn from. We highlight some of the ingredients of the games:

1. A world providing unlimited information and enjoying rich regularities.
2. Regularities in terms of categories, their hierarchical nature, and predictable relations among them.
3. A systems that continually experiments with its world, in particularly by playing prediction games, in order to acquire and use many categories.
4. Playing the games has to incorporate a number of (learning) tasks, in an integrated fashion. In particular, we see prediction, composition and grouping as fundamental components.
5. Scalability with capability: the system consists of processes that are memory and time efficient and robust to many forms of uncertainty. Learning has a large online and sequential component.

We touched on various aspects of the games, such as the role of the system within a larger intelligent system, proper evaluation criteria, and limiting behavior. We raised many challenges, and identified a number of research directions. We hope that this work can serve as as a useful starting point and framework, by pointing to goals and identifying constraints and challenges, for thinking about large scale learning and for building persistent or long-term, autonomous, and massive learning systems. We believe that a basic knowledge of scalable learning algorithms is available now to build preliminary systems in order to better see the possibilities. We expect domains such as the online text and vision provide the richness that would enable playing prediction games in infinitely rich worlds.

Acknowledgments

Many thanks to Yoshua Bengio, Curt Burgess, Michael Connor, Dennis DeCoste, Mark Gluck, Russ Greiner, Wiley Greiner, Rosie Jones, John Langford, Michael Littman, Gregory Murphy, Russell Poldrack, and Rajesh Rao, for valuable input, discussions and pointers, contributions to related research, or general encouragement.

References

[1] D. Angluin. Comp. learning theory: survey and selected bibl. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 351–369, 1992.

[2] L. V. Anh and L. Dabbish. Labeling images with a computer game. In *ACM CHI*, 2004.

[3] D. H. Ballard. *An Introduction to Natural Computation*. The MIT Press, 2000.

[4] M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *ACL*, 2001.

[5] Y. Bengio and Y. LeCun. Scaling learning algorithms towards AI. In *Large-Scale Kernel Machines*. MIT Press, 2007 (to appear).

[6] A. Borodin and R. El Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[7] L. Bottou and Y. L. Cun. Large scale online learning. In *NIPS*, 2003.

[8] W. Callebaut and D. Rasskin-Gutman, editors. *Modularity: Understanding the Development and Evolution of Natural Complex Systems*. The MIT Press, 2005.

[9] A. Cleeremans. Implicit learning. In *Encyclopedia of Cognitive Science*. Macmillan Publishes, 2002.

[10] D. A. Cohn, L. Atlas, and R. E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[11] A. M. Collins and M. R. Quillian. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8, 1969.

[12] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[13] Y. Even-Zohar and D. Roth. A classification approach to word prediction. In *Annual meeting of the North Amercian Association of Computational Linguistics (NAACL)*, 2000.

[14] D. A. Forsyth and J. Ponce. *Computer Vision*. Prentice Hall, 2003.

[15] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Mining data streams: A review. *SIMOD Record*, 34(2), June 2005.

[16] D. George and J. Hawkins. A hierarchical bayesian model of invariant pattern recognition in the visual cortex. In *International Joint Conference in Neural Networks*, 2005.

[17] G. Gigerenzer and R. Selten, editors. *Bounded Rationality*. The MIT Press, 2002.

[18] F. Gobet, P. Lane, S. Croker, P. Cheng, G. Jones, I. Oliver, and J. Pine. Chunking mechanisms in human learning. *TRENDS in Cognitive Sciences*, 5, 2001.

[19] J. T. Goodman. A bit of progress in language modeling. *Computer Speech and Language*, 15(4):403–434, October 2001.

[20] K. Grill-Spector and N. Kanwisher. Visual recognition, as soon as you know it is there, you know what it is. *Psychological Science*, 16(2):152–160, 2005.

[21] P. D. Grunwald, I. Myung, and M. A. Pitt, editors. *Advances in Minimum Description Length*, chapter 1 and 2. The MIT Press, 2005.

[22] J. Hawkins and S. Blaskeslee. *On Intelligence: How a New Understanding of the Brain will lead to Truly Intelligent Machines*. Owl Books, 2004.

[23] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 2001.

[24] L. P. Kaebling, M. Littman, and A. Moore. Reinforcement learning: a survey. *Artificial Intelligence Research*, 1996.

[25] P. J. Kellman. *Handbook of Experimental Psychology*, chapter 7: Perceptual Learning. NY, Wiley, 2002.

[26] G. Lakoff and M. Johnson. *Philosophy in the flesh: the embodied mind and its challenge to western thought*. Basic Books, 1999.

[27] M. Li and P. Vitanyi. *An Introduction to Kolmogrov Complexity and its applications*. Springer-Verlag, 2nd edition, 1997.

[28] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[29] O. Madani. Prediction games in infinitely rich worlds. In *Utility Based Data Mining Workshop (UBDM at KDD)*, Aug. 2006.

[30] O. Madani and M. Connor. Ranked Recall: Efficient classification by efficient learning of indices that rank. Technical report, Yahoo! Research, 2007.

[31] O. Madani and W. Greiner. Learning when concepts abound. Technical report, Yahoo! Research, 2006.

[32] O. Madani, W. Greiner, D. Kempe, and M. Salavatipour. Recall systems: Efficient learning and use of category indices. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, (to appear) 2007.

[33] C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[34] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, 1982.

[35] J. McCrone. *Going Inside*, chapter 7 and 8. Fromm International, 1998.

[36] G. L. Murphy. *The Big Book of Concepts*. MIT Press, 2002.

[37] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kauffman, 1988.

[38] K. Pfleger. *On-Line Cumulative Learning of Hierarchical Sparse n-grams*. PhD thesis, Stanford, 2002.

[39] Z. Pylshyn. *Seeing and Visualizing: It's not what you think*. The MIT Press, 2003.

[40] D. Ravichandran, P. Pantel, and E. Hovy. The terascale challenge. In *KDD Workshop on Mining for and from the Semantic Web*, 2004.

[41] A. S. Reber. *Implicit learning and tacit knowledge: An essay on the cognitive unconscious*. Oxford, UK: Oxford University Press, 1993.

[42] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[43] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *IEEE*, 88(8), 2000.

[44] H. A. Simon. *The Sciences of the Artificial*. The MIT Press, third edition, 1996.

[45] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Z. (2002. Open mind common sense: Knowledge acquisition from the general public. In *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems*, 2002.

[46] R. Solomonoff. A formal theory of inductive inference, part 1 and part 2. *Information and Control*, 7, 1964.

[47] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

[48] P. Thagard. *Mind: Introduction to Cognitive Science*. The MIT Press, 2nd edition, 2005.

[49] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381:520–522, 1996.

[50] P. E. Utgoff and D. J. Stracuzzi. Many layered learning. *Neural Computation*, 14, 2002.

[51] L. G. Valiant. *Circuits of the Mind*. New York: Oxford University Press, 1994.

[52] D. H. Wolpert. The lack of apriori distinctions between learning algorithms. *Neural Computation*, 8(7), 1996.