Genetic Programming and Evolvable Machines, 4, 311–331, 2003

Information Immune Systems

DENNIS L. CHAO dlchao@cs.unm.edu
STEPHANIE FORREST forrest@cs.unm.edu
Department of Computer Science, University of New Mexico, Albuquerque, NM 87131, U.S.A.

Submitted November 1, 2002; Revised March 8, 2003

Abstract. The concept of an *information immune system* (IIS) is introduced, in which undesirable information is eliminated before it can reach the user. The IIS is inspired by the natural immune systems that protect us from pathogens. IISs from multiple individuals can be combined to form a group IIS which filters out information undesirable to any of the members. The relationship between our proposed IIS architecture and the natural immune system is outlined, and potential applications, including information filtering, interactive design, and collaborative design, are discussed.

Keywords: artificial immune systems, collaborative design, collaborative filtering, evolutionary art, information filtering

1. Introduction

Information overload is inevitable in a world that produces over an exabyte (one billion gigabytes) of information per year [38]. We will produce and consume increasingly large amounts of information, so we must find innovative ways to manage it. Although the finding and managing of information are both active research areas, much of the effort is directed towards active strategies such as information retrieval. These techniques might help individuals locate desirable information, but they also accelerate the information glut.

In this paper, we outline the features of an *information immune system* (IIS),[1] first described in [10], that could help people deal with the glut of data. The principle contribution of the paper is the expanded and more complete description of this conceptual framework, parts of which are illustrated by two working examples. We draw inspiration from natural immune systems which protect us from a seemingly limitless number of possible invaders such as bacteria, viruses, and parasites. We believe that an IIS can be constructed which eliminates undesired information using methods analogous to those found in nature. Such an IIS would be situated between an individual and a stream of information, as a mediator. Instead of actively bringing more pieces of information to our attention, it would quietly censor unwanted data.

An IIS should be capable of learning what kinds of information a user wants and discarding the rest. The task of distinguishing what is desirable is a difficult one, however. We propose adapting a strategy used by our natural immune systems, which can "remember" a pathogen that infects us so it can eliminate it more quickly in future encounters. An IIS could do this by storing examples of rejected information and then censoring similar data. If the memory of the system is too specific, however, this approach is likely to be ineffective. Pathogens and information can mutate over time, and our immune systems must be able to anticipate these changes. Therefore, both the natural and the information immune systems must also learn to eliminate *related* pathogens while taking care not to harm anything else.

A natural extension of the personal IIS is a group IIS in which the IISs of multiple individuals are applied simultaneously to filter a stream of information. The only information to survive this group filtering will be acceptable to every member of the group—a *consensus solution*. The problem of satisfying groups of people has received little attention in the human–computer interaction community to date, and we believe that the group IIS approach might prove useful. Consensus solutions can be used in shared environments in which media content must satisfy a group of people, such as broadcast music or artistic displays in public spaces.

In addition to filtering out undesirable data, an IIS could be used to generate many instances of desirable patterns. In this application, the IIS would first be trained to learn a class of undesirable patterns. Next, a system would generate random instances, using the trained IIS to filter out those which are similar to previously rejected examples. Under this scenario, a user would be exposed to a broad range of potentially useful data. For example, a single designer could specify a set of unacceptable designs, and the IIS would then explore the space of "unrejected" solutions. This technique could easily be extended to a group IIS to help a group of users find a set of mutually acceptable solutions.

In the remainder of the paper, we first discuss earlier work that is related to the IIS architecture. Next, in Section 3, we outline how the IIS architecture is related to the natural immune system. Section 4 describes several potential applications for the architecture, including information filtering, interactive design, and collaborative design; and Section 5 describes two exploratory projects that illustrate a subset of the overall architecture.

2. Related work

Several areas of research have influenced our conception of an IIS. An IIS must be able to learn from past encounters, and the issues of learning and memory have long been addressed by the fields of machine learning and artificial intelligence. The primary task that we propose for an IIS, information filtering, has been explored by the field of human–computer interaction. The field of collaborative filtering is relevant when IISs are used for data that are difficult to categorize algorithmically. The collaborative filtering systems that make recommendations to groups instead of individuals perform a function similar to a group IIS. In the arts, evolutionary approaches have been used to generate images and music, and some systems allow groups of users to generate artworks. The IIS approach has several advantages over the evolutionary approaches, which we discuss below and illustrate with examples in Section 5. Finally, our IIS design has been informed by earlier work in artificial immune systems. These many influences are briefly reviewed below.

Case-based reasoning is a technique that adapts solutions to past problems to solve similar current problems [57]. Memory-based reasoning [66] and instance-based learning [1] are related schemes that use the solution of the most similar previous problem. Systems using these approaches learn by "remembering" specific past events rather than creating rules or generalizations. Immune memory uses a form of instance-based learning; the particular response that was effective in clearing a pathogen will likely be used in future encounters with related pathogens [8, 40, 56].

Associative memories, often called content-addressable memories, are neurally inspired architectures in which storage and retrieval are performed using approximate addresses. Smith outlines the parallels between Kanerva's sparse distributed memory [34] and the memory of the natural immune system [63]. The memory of the natural immune system is not exact, and exposure to a novel pathogen can elicit the response primed by a related pathogen.

The term "information filtering" refers to a large range of techniques used to remove data from an incoming stream on the basis of user- or group-specified preferences [5]. Early approaches used simple rules [41] or signatures (e.g., keywords) to identify undesirable data to block. These approaches are still popular, and many commercial products, such as Cyberpatrol [41] for web content, Snort [55] for network traffic, and the Realtime Blackhole List [53] and Brightmail [9] for e-mail, come with long lists of rules and signatures, which can be effective at blocking undesirable data but are vulnerable to malicious sources that can craft information to bypass them. To thwart these adaptive adversaries and to personalize filtering, many systems allow users to specify additional rules for accepting and rejecting data. Unfortunately, the specification of such rules is often difficult and error-prone, and therefore not used routinely. An IIS could incorporate reliable signatures of undesirable data as a first line of defense to be supplemented with more adaptive techniques to provide better and more personalized coverage.

Several research systems simplify the filter specification problem by placing the burden of generating rules on software rather than on a user or programmer. Infoscope [21] monitors a user's behavior to create rules for Usenet newsgroup filtering. The system suggests these rules to the user, who can accept, modify, or reject them. Maxims [45], an interface agent for e-mail, also generates filtering rules based on user behavior, but it suggests actions for the user to take when it is confident in its predictions. Rule-based learning schemes such as these often require many examples before they can infer new rules. In contrast, an IIS using an instance-based learning approach could potentially learn to block a class of data upon seeing only a single exemplar.

Collaborative filtering uses the preferences of others to help an individual make choices [24, 42, 54]. A typical example is a system that recommends items to purchase based on individuals with a similar purchase history. By harnessing the collective preferences of many individuals, such systems can infer similarity between items without needing to understand the relationship between them. This approach is useful when it is difficult for a program to quantify similarities between items, such as for art or music. An IIS could incorporate collaborative filtering techniques to determine the similarity between items for its associative memory capabilities.

There are a few systems that recommend items to groups instead of individuals. MusicFX [43] selects music stations that are broadcast to a gym full of people. The members of the gym must rate all the stations beforehand, and MusicFX plays one of the stations with the highest average rating. The system thus attempts to maximize the happiness of the group. One of MusicFX's shortcomings is that it apparently cannot scale to a large number of choices. If the users are not able to evaluate all of the stations, the quality of the system's choices is likely to be degraded. GroupCast [44], developed by the same research group, used a conceptually similar scheme to select content for a public display system. Unfortunately, they found that user profiles would have been too large for any user with a reasonable amount of patience to complete. In addition, without extensive profiles it was difficult to find appropriate intersections of user preferences to put on the GroupCast displays. Instead, they displayed content that was interesting to *one* of the users, hoping that by chance others would have similar interests. Flytrap [13] addresses the profile-building problem by unobtrusively monitoring each user's personal MP3 player to determine the user's musical preferences. It determines what sorts of music to broadcast to groups of these users based on their profiles.

PolyLens [48] recommends movies to small groups of people who watch movies together. This system applies a standard collaborative filtering algorithm to find recommendations for each of the group members, and then it combines the results to make a group recommendation. Unlike MusicFX, PolyLens attempts to satisfy all users without attempting to maximize group satisfaction. PolyLens bases its recommendations on the expected happiness of the *least* satisfied group member. Therefore, a movie that is barely acceptable to each of the group members is recommended over one that one person would hate but everyone else would enjoy greatly.

These group recommendation systems provide insight into the nature of finding solutions for groups. Notably, it is difficult to make recommendations that satisfy all members of a large group. Possible approaches to this problem include pleasing the majority at the expense of a dissenting minority (e.g., MusicFX) or having the majority make concessions to the minority opinion so that the solution accommodates everyone (e.g., PolyLens and the group IIS). In Section 5.2 we discuss a third possibility, partitioning the group into smaller subgroups composed of individuals with similar interests. Obviously, the choice of strategy depends on the particular application domain.

Evolutionary art (evoart) typically relies on humans to provide aesthetic feedback to guide the "evolution" of a work of computer-generated art [6]. In this approach, the user iteratively refines an image by selecting a subset of favorite images out of a small set, which are variants or combinations of the user's favorites from the previous time step. After many iterations, the quality of the image can improve. This process of evolving art is sometimes called *aesthetic selection*. Dawkins was the first to implement evolutionary art on a computer [17]. His system explores the space of images called "Biomorphs" in the iterative manner described above.

Nelson used aesthetic selection to evolve musical phrases called "sonomorphs" [47]. He used an algorithm similar to Dawkins', but the user interface for selection had to be modified to accommodate sound. A population of images can be presented to the user simultaneously, but playing many musical fragments at once is confusing.

The musical phrases must be played sequentially, and Nelson found that listening to a population of sonomorphs sequentially "taxes the memory." To take advantage of a person's ability to evaluate many things at once visually, Nelson's system shows graphical representations of all members of the current population and only plays the musical representations on demand. Using an IIS might be a more suitable approach to exploring sonomorph space because it does not require the user to make comparisons among members of a population so the graphical representation would not be necessary.

A limitation of the interactive evolution approach pioneered by Dawkins is that the final result of many "generations" is a single work of art. Many of his artistic successors have relied on combining examples of good art to generate novel new candidates [60, 64, 68]. This gives the user a way to find new and potentially interesting regions of parameter space. Unfortunately, it is not obvious how to combine works in a sensible way. Many have tried to create hybrids of several desirable works using genetic algorithms or genetic programming [60]. The programmer is then faced with the difficult task of defining operators to mix the "genotypes" of works in a manner such that the offspring's "phenotype" exhibits the desirable traits of its parents. Sometimes, simple averaging of traits works reasonably well, such as for human faces [3, 23]. In other cases, however, the results are unpredictable and displeasing [39], suggesting that more complex combination rules are required. A trained IIS could generate a very large range of aesthetically pleasing art without the use of a difficult-to-define art combination operator.

Collaborative evoart combines the aesthetic judgment of multiple users to generate art. In the simplest version, users vote to determine which work from a randomly generated set will be explored further [46, 61]. Voting is inefficient because many users' votes should be tallied before each round of evolution. A more fundamental problem is that there is no intuitively "fair" voting scheme that best satisfies the preferences of all the participants, a phenomenon known as Arrow's paradox [2].

An alternative approach is to find solutions that combine the judgments of all the users. One could hybridize the various favorite works of the users, but, as mentioned earlier, it is difficult to combine works of art. Another way to combine the preferences of many individuals is to average them. The artists Komar and Melamid averaged preferences to humorous effect by hiring a polling firm to determine the characteristics that would define America's "most wanted" painting [36]. The survey determined the attributes, such as size, color, and content matter, that the majority of people would enjoy in a painting. The artists created a painting based on the results: a tranquil lake scene with a few deer, a hippopotamus, a small group of people at leisure, and George Washington standing stiffly and incongruously in the center.

The Komar and Melamid study highlights a fundamental problem of combining the preferences of a group: it might not even be desirable. As art critic Arthur Danto asserts:

> What is striking about *America's Most Wanted* is that I cannot imagine anyone really wanting it as a painting, least of all anyone in the population whose taste it

> is supposed to reflect. No one who wants a painting of wild animals *or* who wants a painting of George Washington wants a painting of George Washington *and* of wild animals. Komar and Melamid have transformed disjunctions into conjunctions, and the conjunction can be displeasing even if the conjuncts are pleasing, taken one by one. [15]

An IIS does not create such jarring conjunctions because it will never combine disjunctions. If a fan of Salvador Dalí's surrealist canvases were to meet a fan of Norman Rockwell's sentimental and more realistic style, a painting that combined the two painters' tendencies would probably be unsatisfying to both. Perhaps a better solution would be to introduce both to something completely different, like the paintings of Paul Klee. Unless both individuals specified that they dislike Klee, this is the sort of solution an IIS would be capable of proposing.

Immune-inspired algorithms have often been used for anomaly detection. They draw on the metaphor of the adaptive immune system's ability to distinguish between *self*, or normal data, and *nonself*, or anomalous data. One of the first such systems was the negative selection algorithm introduced in [22]. The algorithm generated candidate random detectors (represented as bit strings). In the training phase, the candidate detectors were compared to sequences of bytes in a given computer file (self), and any detectors that matched the self sequences (above a threshold) were eliminated. The surviving detectors were therefore not similar to any in the file. In the testing phase, if one of the detector strings ever matched the contents of the file, then this indicated that the contents had been changed since the training period. The detector strings were used as *negative detectors* to detect novel sequences of bytes, such as those introduced when a virus corrupts or infects a file. The ARTIS framework is an extension of this work that applies negative selection to detect anomalies in streams of data rather than in static data sets [27, 28]. The ARTIS framework was demonstrated on the problem of network anomaly intrusion detection [4, 27, 28] in which any unusual cluster of TCP connections is flagged as anomalous.

Many useful sources of information contain unexpected but interesting data, however, so it might be undesirable for an IIS to reject all novel data. Consequently, the IIS design is inspired by the immune system's ability to remember past encounters with pathogens, rather than on its ability to detect novel foreign molecules. The anomaly detection ability of ARTIS could complement an IIS for certain applications (see Section 3.3), but for many applications we imagine using negative detectors without explicitly censoring them against self, as in negative selection.

Many computer scientists have developed artificial immune systems based on idiotypic network theory [33]. The idiotypic systems focus on the dynamics of interactions among similar antibodies and antigens. Although many do not attempt to reproduce the behaviors seen in the natural immune system, idiotypic systems have useful properties that have been applied to search [7], data classification [30], cluster detection [67], and data mining [18]. The classifications produced by idiotypic artificial immune systems could potentially be used as metadata to enhance the discrimination of an IIS.

3. The immune system as an information filter

We believe that an IIS can fruitfully borrow several pattern recognition mechanisms from the natural immune system. Our natural immune system consists of two components that use different pathogen recognition strategies. The *innate* immune system uses a few reliable signatures of foreignness to identify invaders, which Janeway calls pathogen-associated molecular patterns (PAMP) [31]. An example of a PAMP is the mannose carbohydrate molecule found on many bacteria and other pathogens but not in mammals [65]. These signatures have been stable over evolutionary time and are encoded in the genome of our immune systems. This strategy is used by many of the signature- and rule- based information filtering products mentioned in Section 2. These products could serve as a first line of defense, playing the role of the innate immune system in an IIS. However, not all signatures of pathogens have been (or even can be) anticipated, and evolution will favor pathogens that do not carry the signatures recognized by our innate immune systems. One role of the *adaptive* immune system, discussed below and outlined in Table 1, is to discover the signatures of pathogens not covered by the innate immune system. In the following subsections we describe some issues that an IIS must face and how one can draw inspiration from the natural immune system to address them.

3.1. Negative detectors and shape space

An IIS should be able to remember which pieces of information a user rejected in the past so it can censor them in the future. However, the strategy of rejecting each item individually is ineffective when one is faced with an effectively limitless variety of information. Thus, we need our IIS to generalize beyond each specific piece

Table 1. The immunological analogy made explicit. We list several features from our natural immune systems and their corresponding features in our proposed information immune systems

Natural IS	Information IS
Self	Desirable information
Non-self	Undesirable information
Infection	User exposed to undesirable data
Costimulation	Rejection of information by user
Naïve cells	Implicit (not instantiated)
Active lymphocyte	Detector
Memory lymphocyte	Detector
Cytolytic activity	Dliminate data
Shape space	Parameter space
Cross-reactive radius	Detector radius
Thymic selection	Protecting known desirable information
Cell senescence	Limited detector lifetimes

of information—explicitly rejecting one item should implicitly reject similar items. The natural immune system has this ability.

The adaptive immune system maintains a repertoire of lymphocytes that detect pathogens. Each lymphocyte is specific to a particular antigen, or protein signature, expressed by pathogens. However, pathogens can mutate in ways that subtly change their antigenic profiles, so lymphocytes need the ability to recognize close variants (generalization). A helpful way to visualize the relationship between lymphocytes and mutating antigens is the conceptual framework of *shape space* [50], a high-dimensional space that represents the universe of possible antigens. Every antigen has a location in shape space, and the small mutations that alter its proteins can shift its location in shape space. A lymphocyte covers a large area in shape space so that antigens can not easily evade detection by mutating. The area in shape space that a lymphocyte covers is called its *ball of stimulation* because it is postulated that a lymphocyte can recognize an antigen within a certain radius of its location in shape space.

Returning to the IIS, which relies on negative detection to censor unwanted information from the user, it is clear that we will need to use a similarity measure to implement generalization. As with lymphocytes in shape space, each detector must be able to cover a region of similar patterns, not just a single point. Therefore, it is necessary for an IIS to determine the similarity between two pieces of information, and two similar items would be in close proximity in "information space."

3.2. Costimulation

Because everyone has different informational needs, each IIS should be customized to its user. Most of today's information filters require the user to write rules to customize the filtering, but we believe it more practical to ask users simply to identify exemplars of undesirable information. Once the user rejects a piece of information, the IIS would automatically reject similar information in the future.

In the adaptive immune system, other cells, such as antigen presenting cells, are required to *costimulate*, or activate, lymphocytes in the presence of a novel pathogen. The costimulation signal provides confirmation that the pathogen should be eliminated. This process reduces the chance of immune cells accidentally attacking the body in an autoimmune response. Once costimulated, the effector cell becomes active and attempts to eliminate the pathogen, whether by releasing antibodies in the case of B cells or by killing the infected cells directly in the case of cytotoxic T cells. Some costimulated cells become long-lived memory cells. In future encounters with the same pathogen, memory cells have reduced costimulation requirements [19], allowing them to respond to lower levels of antigen.

In an IIS, the user could provide confirmation signals to the system, an idea introduced in [27] and known in immunology as a *second signal*. The idle cells waiting for costimulation are implicit—only detectors corresponding to active or memory cells would be instantiated by the IIS. When the user rejects a piece of information, a detector specific to that item would be generated. These detectors would filter out similar data in the future. The user's only responsibility would be to inform the IIS when undesirable data are being presented (Figure 1).

3.3. *The addition of negative selection*

When the user has rejected a sufficient amount of information, the space not covered by detectors will approximate the space of useful information (Figure 2).

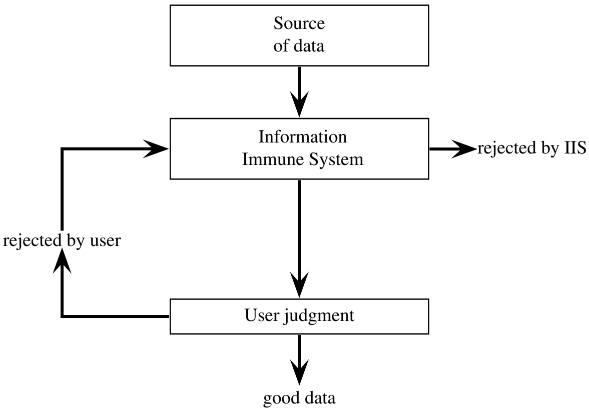

Figure 1. The role of the user in an IIS. The user inspects items from a stream of data, flagging undesirable exemplars. The IIS uses the negative exemplars to generate detectors which subsequently filter out similar data items.

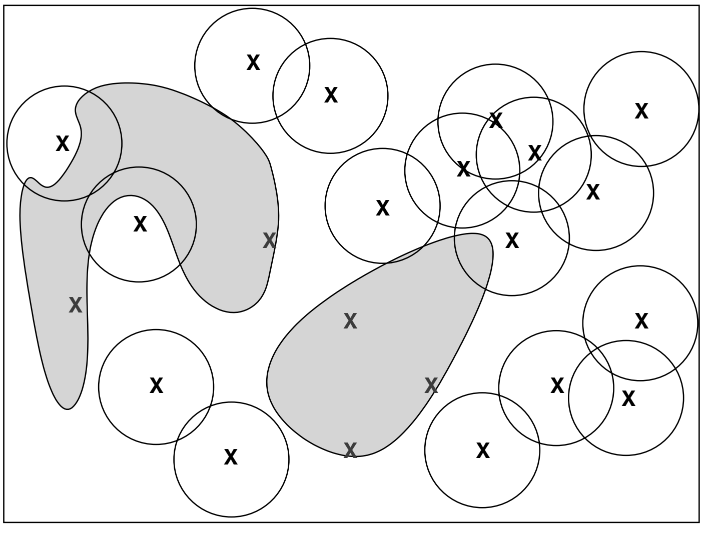

Figure 2. Coverage in the model without negative selection, represented in a 2-dimensional shape space. The shaded regions represent information that is useful. The circles represent the extent of active detector coverage (the ball of stimulation). The Xs without circles represent detectors that will not receive costimulation because they are within a region of acceptable solutions.

Unfortunately, useful information that is too similar to unwanted information runs the risk of being censored by an IIS negative detector (in Figure 2, see the shaded regions that fall within a ball of stimulation). Therefore, we suggest incorporating a technique that the adaptive immune system uses to prevent the immune system from attacking the body's own cells.

The adaptive immune system uses thymic selection to eliminate T cells that might harm the body. Before T cells can enter the repertoire, they are exposed to a large sample of the body's own proteins. Those that bind too tightly to one of the body's proteins are eliminated in a process known as negative selection. Therefore, the T cells that survive are not likely to recognize a self protein.

A similar strategy could be employed by an IIS to protect certain types of information that are known to be useful. These types could be declared "off-limits" to the IIS and would be allowed to bypass the IIS to reach the user. This would be especially useful when the characteristics of certain desirable information are known *a priori*. For example, the IISs of a company's employees might not be allowed to eliminate official company e-mail. When a user costimulates a data item whose detector would cover some desirable information, the system could ignore the costimulation signal because there should be no "implicit" detectors in this region. No information from the "good" regions of information space would ever be censored by the detectors (Figure 3). Alternatively, the IIS could simply ensure that negative detectors will never cover these regions. It could, for example, reduce the generality, or size, of detectors that intersect desirable regions enough to eliminate the intersection. Unfortunately, negative selection does not necessarily protect useful

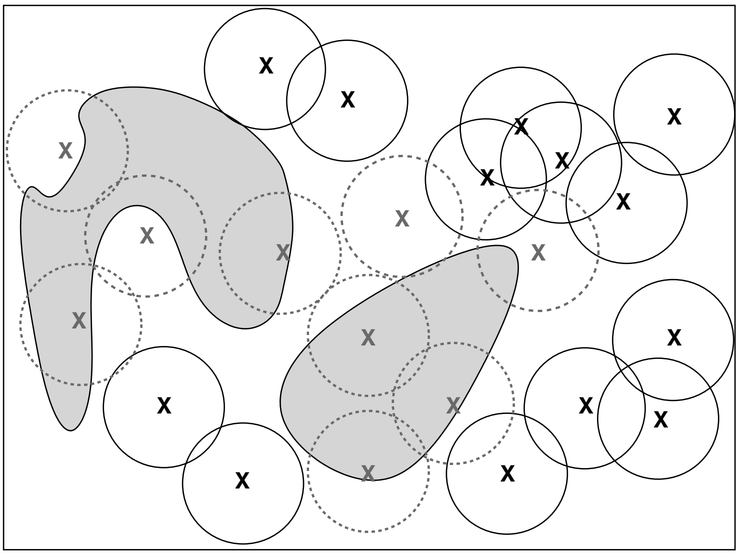

Figure 3. Coverage in the model using negative selection. The dotted circles represent the extent of detectors that are eliminated by negative selection. The solid circles are regions covered by active detectors. Note that none of the useful information protected by negative selection (the shaded regions) can be covered by detectors.

information that is not specified in advance, but we believe that it is impossible to filter information without the risk of eliminating some useful information [26].

3.4. *The role of senescence*

Users might choose to filter out some types of information for only a short period of time. For example, if a radio station plays a song too frequently or if a news story receives too much coverage, a listener may tire of it. These individuals might actually enjoy hearing the song or listening to new developments in the news story at a later date, so the detectors would be counterproductive after their "natural" lifetimes.

Active immune cells are known to have short lifetimes, and memory cells are believed to be routinely eliminated through competition for limited resources [59]. These features would be desirable in the algorithm for two reasons. The first is to provide "rolling coverage" of self. If the fitness function (e.g., the user's tastes) change over time, one could have the lifetime of the active immune cells be finite to reflect the dynamic nature of the user's judgment. The second reason is space efficiency. It is not feasible to store an unbounded number of detectors. One could "age out" old detectors to make room for new ones. Alternatively, the user could manually create memory detectors to cover patterns that he or she never wants to see again.

3.5. *The effect of history*

The order in which an IIS is exposed to information will have impact on its effectiveness. Such phenomena have been observed in the natural immune system, particularly in the case of influenza. Immunologists have discovered that the response to a strain of flu can be dominated by cells that were generated in response to an earlier exposure to a *different* strain [16, 20]. These memory cells are probably most effective against the strain that generated them, but they can respond to related pathogens as well. In shape space, these *cross-reacting* memory cells cover areas that include these similar pathogens and can thus eliminate them. This phenomenon is known as original antigenic sin, and many vaccines take advantage of the effect. For example, if one is exposed to the relatively harmless cowpox bacteria, one is protected against the related but deadly smallpox [32]. Unfortunately, prior exposure to antigens can also work against us [62]. For example, a flu vaccine works by eliciting a mild response to a particular strain's antigens so that an individual will be able to mount an effective secondary response when exposed to it in the future. However, the memory cells generated by a vaccine from a previous year could attack and eliminate subsequent vaccines before they can establish protective immunity. If the first vaccine does not provide protection against the strains corresponding to these later vaccinations, the individual is vulnerable to them (Figure 4). If this same individual had not received the first vaccine, the subsequent vaccines would have been effective.

We could "vaccinate" an IIS by exposing it to undesirable information without necessarily exposing the user. This would allow an administrator to pre-tune the IIS so that it blocks certain kinds of information from a user. For example, a user

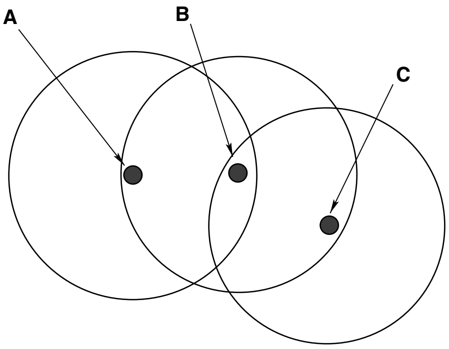

Figure 4. The effect of history. The dots labeled "A" and "B" and "C" represent solutions the user does not like. The circles are the extents of their negative detectors, or "balls of stimulation," in shape space. If solution A is rejected by the user first, the detector that forms around it would reject B before it could be presented. However, C could be presented because it does not fall within the scope of the detector for A. However, if B had been presented first, the coverage would be different. Neither A nor C would be seen after B because its detector would cover all three solutions.

community might prohibit certain kinds of e-mail or web traffic, such as pornography or virus-infected e-mail. The community could vaccinate the IISs of its members with exemplars from the banned categories, and they would not be exposed to these kinds of information. Because the order in which an individual is exposed to undesirable information can affect the coverage of the individual's IIS, the vaccination strategy would need to be planned with care.

4. Applications

The most obvious use of an IIS of the sort described here is information filtering. An IIS could serve as a personalized interface agent that learns a user's preferences for sources of information or for a range of options that is too large or dynamic for a user to evaluate. Because it requires feedback only when the user is exposed to something he or she does not want and it learns without using separate training and testing phases, an IIS could be an unobtrusive addition to many user interfaces. The IIS should gradually become more effective as the user provides feedback over time. In contrast, a scheme that keeps track of what the user *wants* would require separate training and testing phases in which the user first specifies his or her desires then tests the system to determine the system's performance, which might require another explicit training cycle to improve performance. This scheme would also be less likely to explore regions of solution space unknown to the user. An IIS could complement active strategies, such as information retrieval, that search for potentially useful information.

It is straightforward to combine multiple IISs to form a group IIS. If one thinks of each IIS as a sieve that filters undesirable information, then the data that pass through the "sieves" of multiple users are those likely to satisfy all of them (Figure 5). We call these data *consensus solutions*. A group IIS would be useful when the group is exposed to shared information. For example, if a group of people want to listen to

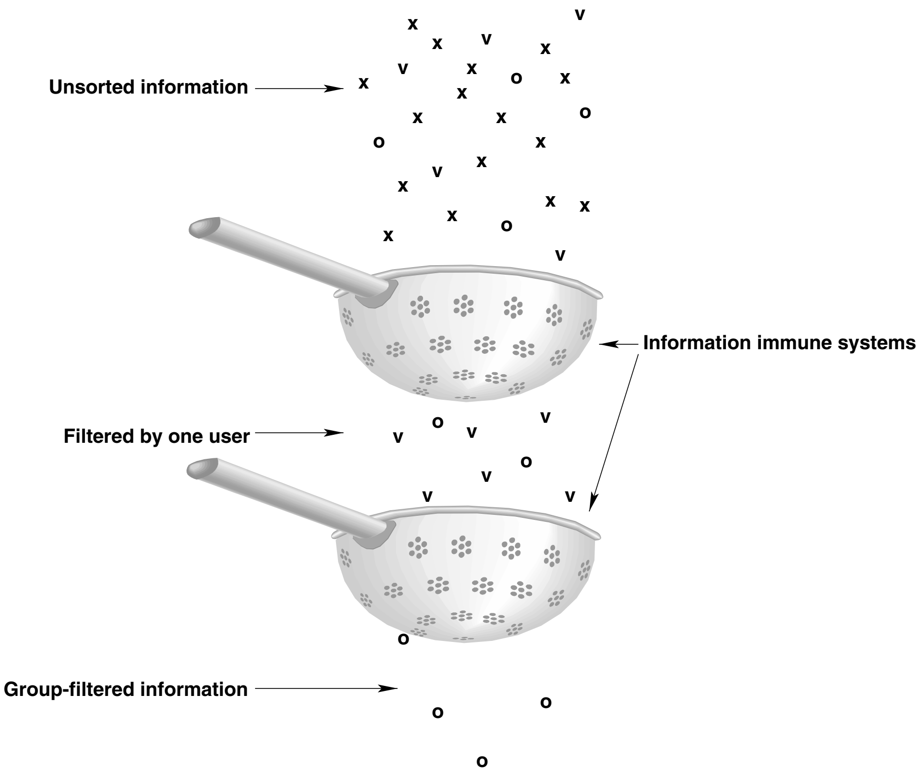

Figure 5. An information immune system as a sieve. The IIS stands between a stream of information and a user, blocking a significant portion of it. Only the information that can pass through the "sieve" actually reaches the user. When the IISs of multiple users are applied in serial, the information that passes all IISs are *consensus solutions*.

music together, they would want to play music that none of the individuals dislikes. It is not clear how well a group IIS would work with a large group of users, and this is likely to depend on the particular application domain. Consensus finding will become more important with the increasing number of intelligent environments which automatically respond to users' needs. For example, in smart home technology, the music, artwork, temperature, and lighting are adjusted to accommodate a building's occupants. Most research on intelligent environments emphasizes the preferences of a single individual [25, 29, 35, 69], but for many environments it will be important to satisfy multiple occupants simultaneously.

Designers and artists could use an IIS for inspiration [11]. If a random source of design solutions or works of art were fed to an IIS, only those designs that were dissimilar from examples rejected in the past would pass through. This could be a useful strategy for design problems in which one is interested in exploring a large range of possible solutions. The solutions that are not rejected could later be refined or optimized using other techniques, such as evolutionary design.

In the case of collaborative design projects, a group IIS could be used to produce consensus solutions to satisfy the entire design team. Although the consensus solutions are not necessarily optimal, they do avoid the problem of combining (hybridizing) multiple solutions, which arises if one tries to combine the favored solutions of each group member (as discussed in Section 2). By using only the known *unacceptable* solutions, an IIS assumes that unexplored regions of parameter space are acceptable until proven otherwise, thus expanding the potential solution space by giving new regions the benefit of the doubt. If instead one used only the known *acceptable* solutions and assumed that the rest of parameter space contained unacceptable solutions, intersections of these solutions for multiple users would be hard to find, if they exist at all (as was found with GroupCast [44]). The group IIS could facilitate "brainstorming" sessions in which a wide range of novel solutions are explored by groups.

5. IIS implementations

We describe two exploratory projects that implement some of the IIS concepts described above, namely negative detectors, costimulation, and group filtering. Both are "aesthetic" information immune systems that shield users from undesirable art. However, they illustrate different capabilities. The first project, which generates simple figures for users to evaluate aesthetically, illustrates the collaborative design potential of IIS-based systems. The second, which chooses music to play to a group, is an information filtering application that could be incorporated into an intelligent environment.

5.1. *A Biomorph information immune system*

We have used the concepts discussed earlier to design a simple IIS that generates computer art, and we reported the results in [11]. We summarize the results here. The IIS characterizes several users' preferences for a particular family of computer-generated images known as Biomorphs [17]. Biomorphs are recursively drawn figures defined by nine parameters. In our experiment, each user was shown a set of randomly generated Biomorphs and instructed to flag those that he or she did not like. For each user, an IIS was created based on the parameters of the rejected Biomorphs (see Figure 6). The IIS filtered out new images that had parameters

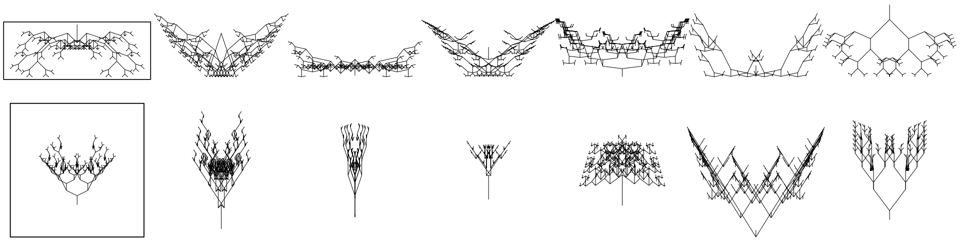

Figure 6. Similar Biomorph parameters result in similar-looking images. The framed Biomorphs are each followed by six variants that have similar parameters, which we believe to look qualitatively similar. If a user were to reject one of the framed Biomorphs, the six variants would also be censored by the Biomorph IIS.

similar to those rejected in the past, and they formed a rough estimate of the parts of Biomorph parameter space that each user wanted to avoid.

We tested whether an IIS could filter a stream of randomly generated Biomorphs to produce an edited stream of acceptable images, based on the subjective judgments of each user. Most users preferred the Biomorph images filtered by their own IISs to the unfiltered ones. We also investigated group IISs which combined the IISs of several users. We were interested in the effect that adding other IISs has on the performance of a single IIS. These effects were measured by asking the users to evaluate three sets of randomly generated Biomorphs: those filtered using no IIS, those filtered with their own IIS, and those filtered by a group IIS composed of the IISs of several users. We tested a subset of three users and a group IIS composed of only these three users' IISs. The images produced by this group's IIS were perceived to be better than unfiltered, and each user found these images to be no worse than those produced using their own IISs, indicating that consensus solutions are possible. This suggests that the IIS approach can accommodate a few users' preferences simultaneously by combining their IISs. The group IIS composed of the IISs of all seven users was not successful, however, producing Biomorphs that were not liked any better than the unfiltered stream.

We propose two explanations for the failure of the Biomorph group IIS to scale beyond three users. First, it is simply more difficult to please a larger number of people. People might have conflicting preferences, making it impossible to find solutions that accommodate a large number of them. Second, the Biomorph aesthetic preference profiles were inaccurate. We used coarse-grained detectors for the IISs in order to reduce the training time for each user, so the resulting profiles were rough. In addition, the users did not appear to have strong or consistent Biomorph preferences. If a user does not have clear and consistent preferences, then it is difficult to construct a reliable profile. More accurate profiles might have allowed the group IIS to satisfy a larger number of individuals.

5.2. Adaptive Radio

Our second IIS application, called Adaptive Radio, is a jukebox that broadcasts songs to a group of listeners who share an office. Users can indicate to the system when they do not like the currently playing song, and the system will record this data to form a musical preference profile for each person. Adaptive Radio avoids playing songs similar to those that have been rejected by any of the users who are currently listening, resulting in a song playlist that should please all users listening to the broadcast. Thus, if there is only one listener, the system will play music that this person will like. As more people arrive, the selection of music will narrow to accommodate the listening preferences of the new users. Because we don't currently have a reliable automated method to determine the similarity between songs, similarity between songs was assessed by the developers in advance. The Adaptive Radio prototype is still under development, but the basic functionality is in place and we are currently using it informally in our research group.

We believe that the Adaptive Radio project will better illustrate the scaling properties of group IISs than the Biomorph project because we expect to obtain more accurate user preference profiles. People generally have clearer musical preferences than Biomorph preferences, so the Adaptive Radio user profiles should be more accurate and consistent. People are also more likely to invest time building their musical preference profiles, so the detectors can be finer-grained. We are optimistic that the finer preference profiles will allow the Adaptive Radio system to support a larger number of users than MusicFX [43]. Users of the MusicFX system could only choose from ninety-two music stations, while Adaptive Radio listeners are able to rate hundreds of songs individually.

Adaptive Radio is a demonstration of IIS principles in a human–computer interaction context. Although we have not conducted formal user studies, we have made informal observations of its use in our office. The users quickly became comfortable with the user interface, which allows them to reject songs with little conscious effort. Registering disapproval became a nearly automatic reaction to undesirable songs, as evidenced by the "channel-surfing" behavior during which a user would quickly reject several consecutive songs without interrupting his or her work. We believe that people find it more natural to reject songs than to provide positive feedback to a music selection system. When Adaptive Radio is playing desirable music, the listener should not need to think about the system. When undesirable music intrudes upon a listener's consciousness, he or she can quickly register disapproval.

Users who had seemingly different musical tastes discovered that they enjoyed the music of their coworkers. These serendipitous newfound musical preferences would be difficult to discover using a MusicFX-like approach that preferentially plays what the listeners already know they like. Other users with little obvious overlap in musical tastes have noticed that Adaptive Radio only plays Simon and Garfunkel songs when they are in the room together. We soon realized that fast or loud songs are prone to rejection by people trying to work. The songs that are least likely to be rejected are slow, quiet, and familiar. It appeared that Adaptive Radio was quickly censoring all interesting songs and leaving only "elevator music." Although the term "elevator music" is usually used pejoratively, Paul Simon would not object to this characterization of his music, claiming that "it's nice to have any song that you write played in an elevator" [58].

Our *passive* musical preferences can be quite different from our *active* ones. While we might enjoy dynamic and challenging music in a concert setting, at work we might prefer something more soothing. Background music can subconsciously elevate our moods and increase productivity, and music that calls attention to itself could be detrimental. In a workplace with broadcast music, *everyone* must be accommodated, even if compromising seems unsatisfactory to the majority. A Muzak executive describes what can happen when employees try to choose their own music:

> In an office for a garment factory outside of Atlanta, the workers got tired of the Muzak and used a radio for their background music. If they turned on rock, 25 percent of the people in the workplace didn't like it. So they got a committee

> together and took a vote. They played the classical station, and only 10 percent of the people ended up liking it. So they tried a country station, and 60 percent didn't like it. They had another meeting. They decided on one day for each format: country one day, classical the next, disco for maybe half a day. But the 10 percent who liked whatever was playing got tired of people glaring at them. Finally the office manager called us and asked if they could have the Muzak back. It proved what I was doing was working. Muzak proved the least of all possible evils. [37]

The advantage that Adaptive Radio has over the often-ridiculed fare of the Muzak corporation is that it can cater specifically to the occupants of a room. The users are likely to appreciate the fact that they have control over the music being broadcast [43]. If they happen to share musical tastes, the variety of acceptable music can be large; if not, the range is likely to be small and possibly less than satisfactory. As one Muzak programmer explains, "There are literally 90 million people listening to Muzak per day. It's a real challenge to put something together that's going to please everyone ... Since we have so many people listening at once, we are forced to amalgamate" [37].

Even if Adaptive Radio does not scale to accommodate large numbers of diverse listeners, we believe that people could be automatically partitioned into smaller subgroups that would have good consensus solutions. Adaptive Radio could be used to create non-intuitive groupings of individuals, which would be useful if there are only a limited number of broadcast channels to accommodate a large number of listeners. Normally, radio stations specialize in genres of music, and listeners must choose among them based on these predetermined categories of music. If Adaptive Radio were to partition the listeners automatically based on their preferences and cater to each group's collective preferences, it could generate novel playlists that cross established genre boundaries. For example, one could use a greedy algorithm to assign users to groups in such a way that maximizes the overlap of musical dislikes within the groups. The broadcaster could then choose music for each group that would reflect the preferences that the listeners in each have in common.

6. Conclusions

Information immune systems provide a way for individuals to filter out undesirable information and to explore parameter spaces. A successful IIS would reduce information load and make existing strategies for finding and processing information more effective. The IIS approach could easily be applied to groups of users, and it represents a novel approach to combining different sets of preferences.

IISs, however, should be fielded with caution. As filtering strategies become more sophisticated, the producers of unwanted information will themselves adapt, creating a kind of information arms race. We see this already in the adaptation of magazine advertisements designed to resemble content articles and "junk mail" packaged in official-looking envelopes. Even more insidious techniques embed advertising in content in which people are interested. Advertisements can be wrapped around e-mail before the user can receive it [12], corporate logos and products can be digitally edited into films and television programs [71], and some shows integrate their sponsors' products into the plotlines [49]. Even in the absence of adaptive adversaries, information filtering technology will drive a selective process that minimizes the differences between desirable and undesirable information. As our filters gain efficacy, undesirable information will evolve to evade them. The filters must constantly co-evolve or else they will rapidly lose effectiveness. When we deploy IISs, we must be prepared to live in a dynamic information ecosystem in which our defenses must adapt as quickly as the abilities of unwanted information to penetrate them [70].

We must also examine the nature of consensus solutions. Consensus solutions might encompass a wide variety of acceptable candidate solutions, but they do not necessarily include the best solutions. A group IIS could be an aid to human creativity, or even be a source of creativity itself, by finding novel and innovative solutions that would not have been discovered otherwise. On the other hand, it could find bland and inoffensive solutions that are barely acceptable to anyone. We have not yet determined which scenario is more likely, and we believe that this will largely depend on the particular application.

Acknowledgments

This paper is a revised and expanded version of [10]. We thank Marc Millier of Intel Corporation for suggesting the term "information immune system" and Steve Cayzer for introducing us to a reference that improved the paper greatly. The authors gratefully acknowledge the support of the National Science Foundation (ANIR-9986555), the Office of Naval Research (N00014-99-1-0417), the Defense Advanced Research Projects Agency (AGR F30602-00-2-0584), the Intel Corporation, the Santa Fe Institute, and the ICARIS program committee.

Notes

[1] The term "information immune system" was introduced by Neil Postman (in [51] and [52, page 63]).

References

1. D. Aha, D. W. Kibler, and M. K. Albert, "Instance-based learning algorithms," Machine Learning, vol. 6, pp. 37–66, 1991.
2. K. Arrow, Social Choice and Individual Values, Wiley: New York, 1951.
3. A. L. Austin, Letter to Charles Darwin, 1877, reprinted in Galton, 1879.
4. J. Balthrop, S. Forrest, and M. Glickman, "Revisting LISYS: Parameters and normal behaviour," in Proceedings of the 2002 Congress on Evolutionary Computation, D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Shackleton (eds.), IEEE Press: Piscataway, New Jersey, 2002, pp. 1045–1050.
5. N. J. Belkin and W. B. Croft, "Information filtering and information retrieval: Two sides of the same coin?" Communications of the ACM, vol. 35, no. 12, pp. 29–38, 1992.
6. P. J. Bentley and D. W. Corne (eds.), Creative Evolutionary Systems, Morgan Kaufmann Publishers: San Francisco, California, 2002.

7. H. Bersini and F. J. Varela, “Hints for adaptive problem solving gleaned from immune networks,” in Parallel Problem Solving from Nature, H. Schwefel and R. Männer (eds.), Springer-Verlag: Berlin, 1991, pp. 343–354.
8. J. A. Borghans, A. J. Noest, and R. J. de Boer, “How specific should immunological memory be?” J. Immunol., vol. 163, no. 2, pp. 569–575, 1999.
9. Brightmail Solution Suite, Brightmail Inc., San Francisco, California, 2002. http://www.brightmail.com.
10. D. L. Chao and S. Forrest, “Information immune systems,” in Proceedings of the First International Conference on Artificial Immune Systems, J. Timmis, and P. J. Bentley (eds.), University of Kent at Canterbury Printing Unit: Canterbury, England, 2002, pp. 132–140.
11. D. L. Chao and S. Forrest, “Generating biomorphs with an aesthetic immune system,” in Artificial Life VIII: Proceedings of the Eighth International Conference on the Simulation and Synthesis of Living Systems, R. Standish, M. A. Bedau, and H. A. Abbass (eds.), MIT Press: Cambridge, Massachusetts, 2003, pp. 89–92.
12. N. Cochrane, “Mobile entrepreneur rapt in wireless e-mail advertising,” The Age (Melbourne), vol. 24, July 2001, p. 7, 2001.
13. A. Crossen, J. Budzik, and K. J. Hammond, “Flytrap: Intelligent group music recommendation,” in Proceedings of the 7th International Conference on Intelligent User Interfaces, ACM Press: New York, 2002, ISBN 1-58113-382-0, pp. 184–185.
14. Cyberpatrol, SurfControl plc, Westborough, Massachusetts, 2002
15. A. C. Danto, “Can it be the ‘most wanted painting’ even if nobody wants it?” Painting by Numbers: Komar and Melamid’s Scientific Guide to Art, J. Wypijewski (ed.), Farrar, Straus & Giroux: New York, 1997, pp. 124–139.
16. F. M. Davenport, A. V. Hennessy, and T. Francis, “Epidemiologic and immunologic significance of age distribution to antibody to antigenic variants of influenza virus,” J. Exp. Med., vol. 98, pp. 641–656, 1953.
17. R. Dawkins, The Blind Watchmaker, Longman Scientific and Technical: Harlow, UK, 1986.
18. L. N. De Castro and F. J. Von Zuben, “aiNet: An artificial immune network for data analysis,” Data Mining: A Heuristic Approach, Chap. XII., H. A. Abbass, R. A. Sarker and C. S. Newton (eds.), Idea Group Publishing: Hershey, Pennsylvania, USA, 2001, pp. 231–259.
19. C. Dubey, M. Croft, and S. L. Swain, “Naive and effector CD4$^+$ T cells differ in their requirements for T cell receptor versus costimulatory signals,” J. Immunol., vol. 157, no. 8, pp. 3280–3289, 1996.
20. S. Fazekas de St. Groth and R. G. Webster, “Disquisitions of original antigenic sin. I. Evidence in man,” J. Exp. Med., vol. 124, no. 3, pp. 331–345, 1966.
21. G. Fischer and C. Stevens, “Information access in complex, poorly structured information spaces,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1991), S. P. Robertson, G. M. Olson, and J. S. Olson (eds.), ACM Press: New York, 1991, pp. 63–70.
22. S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, “Self-nonself discrimination in a computer,” in Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press: Los Alamitos, California, 1994, pp. 202–212.
23. F. C. Galton, “Composite portraits, made by combining those of many different persons into a single resultant figure,” Journal of the Anthropological Institute of Great Britain and Ireland, vol. 8, pp. 132–144, 1879.
24. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.
25. S. R. Hedberg, “After desktop computing: A progress report on smart environments research,” IEEE Intelligent Systems, vol. 15, no. 5, pp. 7–9, 2000.
26. S. R. Hiltz and M. Turoff, “Structuring computer-mediated communication systems to avoid information overload,” Communications of the ACM, vol. 28, no. 7, pp. 680–689, 1985.
27. S. A. Hofmeyr, An immunological model of distributed detection and its application to computer security, Ph.D. thesis, University of New Mexico: Albuquerque, New Mexico, 1999.
28. S. A. Hofmeyr and S. Forrest, “Architecture for an artificial immune system,” Evolutionary Computation, vol. 8, no. 4, pp. 443–473, 2000.

29. House_*n* Living Laboratory, School of Architecture and Planning, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2001.
30. J. E. Hunt and D. E. Cooke, "Learning using an artificial immune system," J. Network Comp. Appl., vol. 19, pp. 189–212, 1996.
31. C. A. Janeway Jr, "The immune system evolved to discriminate infectious nonself from noninfectious self," Immunol. Today, vol. 13, no. 1, pp. 11–16, 1992.
32. E. Jenner, An Inquiry into the Causes and Effects of the Variolae Vaccinae; a Disease Discovered in some of the Western Counties of England, Particularly Gloucestershire, and Known by the Name of the Cow Pox, 1798.
33. N. K. Jerne, "Towards a network theory of the immune system," Ann. Immunol. (Inst Pasteur), vol. 125C, pp. 373–389, 1974.
34. P. Kanerva, Sparse Distributed Memory, MIT Press: Cambridge, Massachusetts, 1988.
35. C. D. Kidd, R. J. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. Mynatt, T. E. Starner, and W. Newstetter, "The aware home: A living laboratory for ubiquitous computing research," in Proceedings of the Second International Workshop on Cooperative Buildings (CoBuild'99) of Lecture Notes in Computer Science, N. A. Streitz, J. Siegel, V. Hartkopf and S. Konomi (eds.), Springer: Berlin, 1999, pp. 191–198.
36. V. Komar, A. Melamid, and J. Wypijewski, Painting by Numbers: Komar and Melamid's Scientific Guide to Art, Farrar, Straus & Giroux: New York, 1997.
37. J. Lanza, "Elevator Music: A surreal history of Muzak, easy-listening, and other moodsong," St. Martin's Press: New York, 1994.
38. P. Lyman and H. R. Varian, "How much information?" 2000, retrieved from http://www.sims.berkeley.edu/how-much-info on 1 June 2002.
39. P. Machado and A. Cardoso, "All the truth about NEvAr," Applied Intelligence, vol. 16, pp. 101–118, 20022002.
40. C. R. Mackay, W. L. Marston, L. Dudler, O. Spertini, T. F. Tedder, and W. R. Hein, "Tissue-specific migration pathways by phenotypically distinct subpopulations of memory T cells," Eur J Immunol, vol. 22, no. 4, pp. 887–895, 1992.
41. T. W. Malone, K. R. Grant, and F. A. Turbak, "The Information Lens: An intelligent system for information sharing in organizations," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1986), M. Mantei, and P. Orbeton (eds.), ACM Press: New York, 1986, pp. 1–8.
42. T. W. Malone, K. R. Grant, F. A. Turbak, S. A. Brobst, and M. D. Cohen, "Intelligent information sharing systems," Communications of the ACM, vol. 30, no. 5, pp. 390–402, 1987.
43. J. F. McCarthy and T. D. Anagnost, "MusicFX: An arbiter of group preferences for computer supported collaborative workouts," in Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work, ACM Press: New York, 1998, pp. 363–372.
44. J. F. McCarthy, T. J. Costa, and E. S. Liongosari, "UniCast, OutCast & GroupCast: An exploration of new interaction paradigms for ubiquitous, peripheral displays," in Workshop on Distributed and Disappearing User Interfaces in Ubiquitous Computing at the SIGCHI Conference on Human Factors in Computer Systems (CHI 2001), ACM Press: New York, 2001.
45. M. Metral, A Generic Learning Interface Agent, B.Sc. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology: Cambridge, Massachusetts, 1992.
46. J. Mount, S. N. Reilly, and M. Witbrock, "International Genetic Art II site," http://robocop.modmath.cs.cmu.edu:8001/htbin/mjwgenformII, 1994, (site no longer accessible).
47. G. L. Nelson, "Sonomorphs: An application of genetic algorithms to the growth and development of musical organisms," in Proceedings of the Fourth Biennial Art & Technology Symposium, Connecticut: New London, 1993, pp. 155–169.
48. M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl, "PolyLens: A recommender system for groups of users," in Proceedings of the Seventh European Conference on Computer Supported Cooperative Work, W. Prinz (ed.), Kluwer Academic: New York, 2001, pp. 199–218.
49. G. Pennington, "Just try zapping these ads," St. Louis Post-Dispatch, vol. 14, April 2002, p. F1, 2002.
50. A. S. Perelson and G. F. Oster, "Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-non-self discrimination," J. Theor. Biol., vol. 81, no. 4, pp. 645–670, 1979.

51. N. Postman, “Informing ourselves to death,” Speech delivered to the German Informatics Society (Gesellschaft für Informatik), 1990.
52. N. Postman, Technopoly, The Surrender of Culture to Technology: Vintage Books, New York, 1992.
53. Realtime Blackhole List, Mail Abuse Prevention System LLC, Redwood City, California, 2002, http://www.mail-abuse.org/rbl/.
54. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An open architecture for collaborative filtering of Netnews,” in Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work, R. K. Furuta and C. M. Neuwirth (eds.), ACM Press: New York, 1994, pp. 175–186.
55. M. Roesch, “Snort—lightweight intrusion detection for networks,” in Proceedings of Thirteenth Systems Administration Conference (LISA ’99), The USENIX Association, Berkeley: California, 1999, pp. 229–238.
56. F. Sallusto, D. Lenig, R. Forster, M. Lipp, and A. Lanzavecchia, “Two subsets of memory T lymphocytes with distinct homing potentials and effector functions,” Nature, vol. 401, no. 6754, pp. 708–712, 1999.
57. R. C. Schank, Dynamic Memory: A theory of reminding and learning in computers and people, Cambridge University Press: New York, 1982.
58. T. Schwartz, “Interview: Paul Simon,” Playboy, vol. 31, no. 2, pp. 49–51, 163–174, 1984.
59. L. K. Selin, K. Vergilis, R. M. Welsh, and S. R. Nahill, “Reduction of otherwise remarkably stable virus-specific cytotoxic T lymphocyte memory by heterologous viral infections,” J. Exp. Med., vol. 183, no. 6, pp. 2489–2499, 1996.
60. K. Sims, “Artificial evolution for computer graphics,” Computer Graphics, vol. 25, no. 4, pp. 319–328, 1991.
61. K. Sims, Genetic Images, 1993.
62. D. J. Smith, S. Forrest, D. H. Ackley, and A. S. Perelson, “Variable efficacy of repeated annual influenza vaccination,” Proc. Natl. Acad. Sci. USA, vol. 96, no. 24, pp. 14001–14006, 1999.
63. D. J. Smith, S. Forrest, and A. S. Perelson, “Immunological memory is associative,” in Workshop Notes, Workshop 4: Immunity Based Systems, Intnl. Conf. on Multiagent Systems, 1998, pp. 62–70.
64. J. R. Smith, “Designing biomorphs with an interactive genetic algorithm,” in Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA’91), R. K. Belew and L. B. Booker (eds.), Morgan Kaufmann Publishers: San Francisco, California, 1991, pp. 535–538.
65. P. D. Stahl and R. A. Ezekowitz, “The mannose receptor is a pattern recognition receptor involved in host defense,” Curr Opin Immunol, vol. 10, no. 1, pp. 50–55, 1998.
66. C. Stanfill, D. Waltz, “Toward memory-based reasoning,” Communications of the ACM, vol. 29, no. 12, pp. 1213–1228, 1986.
67. J. Timmis, Artificial immune systems: A novel data analysis technique inspired by the immune network theory, Ph.D. thesis, University of Wales, 2000.
68. S. Todd and W. Latham, Evolutionary Art and Computers, Academic Press: London, 1992.
69. “The user in control,” Philips Research Password, vol. 3, pp. 10–13, 2000.
70. L. Van Valen, “A new evolutionary law,” Evolutionary Theory, vol. 1, pp. 1–30, 1973.
71. “Virtual ads, real problems,” Advertising Age, vol. 70, no. 22, p. 30, 1999.