AI: Expert Systems Pioneer Meeting
Session 8: Artificial Intelligence and Machine Learning in the 1990s

Moderators:
Burt Grad
David C. Brock

Recorded May 14, 2018
Mountain View, CA

CHM Reference number: X8652.2019

Table of Contents

Participants:

Name	Affiliation
David Brock	Moderator, CHM, Director Center for Software History
Burton Grad	Moderator, SI SIG
Brad Allen	Inference
Denny Brown	Teknowledge
Ed Feigenbaum	Intelligenetics/Intellicorp; Teknowledge
Peter Friedland	Intellicorp
Paul Harmon	Newsletter
Peter Hart	Syntelligence
Gary Hendrix	Symantec
Fritz Kunze	Franz, Inc
Doug Lenat	MCC/CyCorp
Brian McCune	Advanced Decision Systems
Alain Rapaport	Neuron Data
Herb Schorr	IBM/ISI
Reid Smith	Schlumberger
Stephanie Dick	Historian, University of Pennsylvania (attended remotely using the Beam Robot)
Colin Garvey	Historian, Rensselaer Polytechnic Institute
David Grier:	Historian, George Washington University
Thomas Haigh	Historian, University of Wisconsin at Milwaukee
David Hemmendinger	Historian, Associate Editor IEEE Annals
Hansen Hsu	Historian, CHM, Center for the History of Software
Paul McJones	CHM, Special projects
Len Shustek	CHM, Chairman

Ed LaHay	SI SIG, Meeting Project Manager

Burton Grad: This last segment is sort of a segue, a bridge from all the stuff we've been talking about here. We started with the AI stuff in the 1950s and 1960s, which at the time didn't go anywhere, but it bridged over. Now, they're using the same words at least; whether it's the same meaning is a whole different discussion, but it's certainly the same words. AI and machine learning are where they started at the beginning; and they talk in the 1990s about AI and machine learning again, whether it's the same stuff or not, but certainly the technology, the hardware, things have changed dramatically, in terms of price and capability.

What we want to talk about now is in effect two different things: One, we've already talked about some of the expert systems that still exist, but what aspects have kept on and are still in the world today, whether by that name or not?

Secondly, what changed in the 1990s. What changed then that made AI capable, and what kinds of things was it starting to be used for in the 1990s? Not so much what happened afterward although that is certainly an interesting story.

Expert Systems Today

Grad: Let's finish up on the expert systems stuff. You have alluded to a number of the things there. Is the word expert systems still in use? Does anybody use that terminology anymore? Do they use the terminology knowledge-based systems? Is that still in use?

Brad Allen: There's still expert systems conferences. The ACM has proceedings up to last year, which I assume means it's still active.

Grad: Who comes?

Allen: I just told you there are conferences. It doesn't mean I've gone to the conferences. Do you want me to look it up?

Grad: Do any of you go to those conferences? Did any of you know they existed? Is there anything else that you know of in which expert systems as a group or the name is still a topic?

Fritz Kunze: Is AAAI an example of a place where people still talk about expert systems?

Ed Feigenbaum: Ask Reid. He goes to those things. I'm retired.

Reid Smith: Sure, you are. Not that I know of, but the sessions are more specialized than that on Burt's terms. Academic side, perhaps I could say that. On the innovative side, they're about specific solutions to specific problems. Those are the foci every time, whether it's Ford's scheduling for manufacturing or whether it's still FinCEN as Allen pointed out or the follow-on to that which was for ADS, or securities fraud, or scheduling. The areas are more specific than a particular approach to transferring knowledge into systems in my experience.

Grad: Do they talk about tools, methodologies, systems, or is it specific solutions only?

Smith: The foci again are specific solutions to specific problems. I've been on the Program Committee for 30 years, so I've seen a few of these things. Along the way, what authors are required to talk about is what technologies they use, and there's quite a variety of technologies that they use.

As you can imagine, over the years, the amount of space devoted to talking about expert systems has gone down because that's no longer innovative. I mean everybody does it so it just, “Psshh.” You know? They don't talk about whether they wrote it in Fortran either. You know what I'm saying? Other technologies have come up, but those are all sort of on the side. Not a sideshow but they're ancillary to the actual foci, which is the solutions to specific problems.

Peter Friedland: There still is an active journal. I just have it here. This is an LSphere journal, expert systems with applications. The most recent issue had papers with topics we would all recognize. ArchCam is a real-time expert system for suspicious behavior detection in ATM sites. Let's see another one: a network-based classification framework for predicting treatment response of schizophrenia patients—possibly useful here.

Academic versus Business Applications

Grad: Are they all academic papers, or are they by companies who are using it for business?

Friedland: Well, the first one I read, real-time expert systems for behavior detection in ATM sites, seems like a business application.

David Grier: I'm going through mine in the ACM and the IEEE. They all sound like real-time applications, but they're all by academics.

Smith: IAAI, I really would point you there because the focus is explicitly not academic only. These have to be deployed applications and used by real people. We could play this game if you like. I'll read you if you want titles of papers.

Grad: So, there' are two streams still; one is an academic stream and the other is business applications stream.

Smith: That use technologies that we know and love.

Grad: Is there any further academic development going on in expert systems, developing new concept structures, things like this that any of you would know?

Friedland: I think there is because of the fact that there is still at least one journal in the area with a bunch of academic papers in it. When I visit universities overseas, particularly in Asia, there's considerable work in papers labeled as expert systems technology. I don't think the term has died away. Certainly, the term, knowledge-based systems, has not died away at all.

Transition to Knowledge-Based Systems

Feigenbaum: Yes, that's the key thing. We transitioned somewhere in the late 1980s and early 1990s. We transitioned even in the mid-1980s from calling things expert system things to calling them knowledge-based system things. That's the terminology that stuck, including in our own laboratory, which is called Knowledge Systems Laboratory.

Grad: Is there a set of changes in the technologies or concepts that are being used in the knowledge-based systems that are significantly different than what we've discussed here as expert systems?

Feigenbaum: I tell you, I'm just not up to date enough to know.

Allen: There is a lot of work I think going on now and this is going to be slightly from a personal perspective. All of the work that has gone on recently with the real capitalization on machine learning techniques that often go back to the times that we were talking about, they hit their stride when certain kinds of technical events happen: the emergence of GPUs for acceleration, the discovery that we could actually build neural networks in multi-layers and get them to be performant and they would be able to really do a good job in certain tasks mainly around perception. People are capitalizing on those.

As they attempt to take those kinds of systems, which are much better than the ones that we were able to build even back in the 1990s with image recognition, speech recognition, speech production, and so forth, they're having to kind of fall back on techniques that were pioneered by the expert systems community or the knowledge-based systems community or even going back to the general problem-solving kind of era to build things that actually are able to function to solve problems that involve a certain amount of reasoning.

AlphaGo Systems

Allen: The kind of symbolic AI is creeping back in for people to be able to build systems that have a certain level of performance in areas that they can make real strides on. An example of this, which is not a commercial example but is kind of emblematic is what happened, is the AlphaGo systems.

Friedland: Ah, good, I'm glad you mentioned that.

Grad: What is that?

Allen: This is an application that was built by an outfit called DeepMind, which had been acquired by Google. Over the history of AI, the notion of game playing has been a challenge problem that drew a lot of attention notably with people like Simon and Newell and so forth. Go was always viewed as an even further reach than chess. Solving chess, not solving in the technical sense but getting to the point where you can beat a Garry Kasparov, was an achievement for AI in a very focused way which combined rule-based reasoning in certain areas plus really powerful search and other kinds of heuristic techniques. But Go was seen as something that was orders of magnitude off.

DeepMind was able to take some of these technologies in the area of deep learning and apply them to perception of the Go game board to build things that essentially were a faster, simpler way to acquire the knowledge that had been handcrafted into the chess systems of the late 1990s. It was subsequently able to learn those things from data and by self-play in essence over the internet and then get to the point where it could basically achieve superhuman levels of performance.

Friedland: It could beat a world champion, but it's not strictly a deep-learning application as Brad was saying. There was a very good talk about that at recently. It was a victory for knowledge combined with machine learning to solve the problem.

Grad: It was a combination, a hybrid?

Allen: There's been a revolution in our ability to build things that can be, essentially, learned off of data which correspond to either rules or sets of rules. To be able to work from, “Here's a situation, here's a conclusion or an action you can bring,” but orchestrating those things into meaningful behavior is something that's just now beginning to borrow from the toolkit of traditional AI.

<u>Neural Networks and Decision Making</u>

Grad: I see this as a chess game, but by studying games that are recording Go, for example, seeing the outcomes. Is that how they built up the knowledge, or was it by saying, "Here's how the experts play the game"?

Allen: No, in subsequent iterations, the system effectively learned from scratch. One of the things that happened from my perspective in the 1990s and into the 2000s has been a considerable improvement in our ability to do the essentially data-driven creation of things that you can think of as rules or rule sets.

Essentially, you're representing these narrow networks that aren't the same kind of decision-making systems that we were talking about in expert systems. We're solving a big problem that Ed alluded to in terms of the difficulty of knowledge elicitation, by going to a model where you just put a lot of data together of examples. Here's a situation, here's a conclusion or an action to take, and then have the system does its best job of inducing out that using this neural network technology. You could build a lot of the things that were handcrafted in the expert system.

Grad: Let me ask a question then: if we had had that kind of capability in the 1970s, 1980s, that timeframe, could we have built up the logic, the rules that you created by talking to the experts by having examined what was done and what the outcomes were and inductive reasoning? Could we have come up with that kind of logic? Would that have been possible?

Feigenbaum: I don't know. The reason I say I don't know is because the theory of why multilayered neural networks work is not really worked out. It's a huge empirical enterprise, and you can really not tell at the start of it what it's really going to do.

<u>Supervised Classification and Discovery</u>

Grad: Denny, you have a comment on that?

Deny Brown: Yes. Again, a story for the non-techies in the room: There's a task called chicken sexing in which you take baby chicks and decide whether they are male or female. People who do that well can look at the chicks and can sort it. The way they teach somebody to do that is not by trying to explain what it is that they're looking for because they don't know. The way they do it is the same way that machine learning works, which is they'll take the person that knows how to do it and they'll take the newbie. The guy will go, "Just guess." Then, “Okay, and you go.” "Female." "No, wrong." "Okay, female." " Right." That person learns to do the chicken sexing without knowing how he's doing it. That's a perceptual task. The machine learning guys do that very, very well.

Friedland: That’s supervised classification.

Brown: Yes, right. It's this is yes, this is no.

Grad: Within the defined scope.

Brown: If you have a big enough data source so that you can do that training automatically, you get to good behavior. You get to good behavior not necessarily knowing what's underlying. What Ed said a minute ago about not really knowing how those layers work: for chicken sexing, even the expert can't give you the explanation of why he decided that was male or female.

Grad: If the machine with the visual capabilities we talked about before looks at those chickens, makes a guess, looks at the chicks, makes a guess, can it in effect deduce what it is it's seeing that's different?

Brown: Currently, no. Currently, they don't do that. They can get the answers, but they don't necessarily know why those answers are there.

Grad: Interesting.

Brown: Now, when you add some of the multilayer stuff and some of the combinations you can start with the perceptual. For chicken sexing, there's an economic advantage of being able to do this, whether you understand it or not. The point is that there are a lot of applications for the machine learning stuff that is interesting but not so interesting to those of us who are at the knowledge level and want to be able to write systems that explain themselves that say why this was going on.

Paul Harmon: Let me say something for the machine learning. The Go system that mastered Go not only beat the world's champion at Go but introduced plays that previous Go players had not used. We're talking about plays that previous Go players had considered and rejected.

After the Grand Master from China played the system, he then adopted the system's moves. We're talking about creativity, whatever that means. We're talking about not just learning but learning completely new things that nobody could possibly have taught them because the Go Masters didn't know it. In other words, we're talking about a system that will start to introduce new knowledge that it's acquired, and that knowledge will be used by human experts.

Grad: Is that creation, creative thinking?

Friedland: For the example you're giving, it was actually a Korean Grand Master, not Chinese. The Grand Master was Korean, which actually created a revolution in spending, a new Korean fund for a billion dollars in new AI spending in Korea.

Feigenbaum: Yes, but I think there was an argument between the Chinese guy and the Korean guy as to who was the real champion.

Friedland: Anyway, to answer your question about the reason it discovered new moves though, I don't know whether you'd call it creative or not. One of the things it could do which is the same thing with chess was look deeper into the plays, more levels of after this, this, this, this than any human being. It was sort of the combination of things, of being able to look deeper as well as having the way to do the kind of things you were doing that led finally to the new moves.

Grad: Enough about Go.

Advances in Computing Power

Smith: I wanted to go back to the question you posed to Ed and reflect a little more personally. In the work that I have seen and participated in, we never had the volume of labeled training examples that are required to do this kind of magic. You saw more applications than I did. I wonder. Let's imagine for a moment that the empirical experiment kind of works out. Did you ever see the number of examples labeled that would be required to make it work even today? I did not.

Feigenbaum: In our days, back then, the answer is no. We didn't have the grand cosmic network that Linkletter was talking about, really operating then linking everyone to everyone with a culture that wants to share everything.

A paper was published in *Nature* on dermatology: take a picture of your skin lesion with your iPhone and send it in to the doctor. It turns out that the AI system learned and could diagnose the nature of the dermatological lesions better than doctors. That was published in *Nature* a year ago right now.

Grad: How did it get the information? Did it just look at 10,000, 500,000 pictures and someone said, "This is this, this is that"? That's how it learned?

Feigenbaum: Yes, it labeled. It labeled, absolutely. It's a granularity issue. When we were going after knowledge from a doctor, we wanted a gold bar; we didn't want gold dust. We wanted it all packaged up with your expertise and all your rules of good judgment, all packaged together. What you get now is gold dust, and you need 100,000 of them.

Grad: That's the point.

Feigenbaum: You need a tremendous amount of computer power because you're computing in 1,000-dimensional space.

Grad: To get to your numbers on your pulmonary thing, you looked at 400 cases.

Feigenbaum: No, no, no, 400 rules.

Grad: Four hundred rules. How many cases?

Feigenbaum: We looked at 215 cases, let's say.

Grad: So, it's a small sample.

Friedland: No, you're missing the point. If only there had been the 215 cases classified as this or this, that wouldn't have been enough. This was 215 cases with an explanation of why the doctor had called them this syndrome or this syndrome or this syndrome. In the case of the dermatology, it was simply the pictures labeled with melanoma or eczema—good examples.

Smith: I just want to build on something Brad said. The question writ large is what changed over the years. GPUs [graphics processing units] is one thing that changed. The amount of computing power went up four orders of magnitude. Then there were specialized machines, and it went up way more than that and there was the storage.

There are two other things I think: one is the internet. Remember, there was no internet, practically speaking, except among a small of number of sites. What that meant was there are thousands of examples of almost everything on the net.

Then to me the last thing, to speak to your point about sharing, was the whole open source movement. Nobody starts from the ones and zeros today. You start from pretty good libraries for almost anything you can imagine.

Hybrid Systems

Kunze: From evolutionary software, there's software that simulates evolution with the ability to make offspring that are selected from to create brand new electrical circuits. No one's ever seen before, so it's very original. Basically, it's done by creating things that are evolving, some kind of selection function. They do things that you couldn't figure out how to do but they're brand new.

I wouldn't be surprised at all if this Korean Go Master story that came out actually involved some of that because I understand they had the Go player playing against himself in multiple iterations.

Grad: Is there anything further on expert systems, how they're used today or how they've been propagated?

Harmon: I don't know so much about this, but Peter [Norvig] from Google told me that the self-driving car that they're developing uses neural networks to scan the area, but that they use a knowledge system to make decisions about moves on the map. In other words, turn right, and turn left. He said it is a kind of a mess as a technical issue, but he's got a neural network and a knowledge-based system working together to make the car work.

Allen: I think that's the point. The progress right now that we're making is in building these hybrid systems that are bringing together the best of the things that we learned how to do in the symbolic area with the power that's emerged out of building increasingly powerful perception systems.

Grad: Are these creating business opportunities for companies in the software business?

Allen: Absolutely, absolutely.

Feigenbaum: A good example is the Israeli company, Mobileye. This hybrid philosophy is most beautifully articulated by the CEO of Mobileye, who makes these cameras for these self-driving cars but has told the world that that's not good enough for safe driving. You have to have a rule-based system in there as well. At one point, he said he had 600 rule writers working for him now, and he was going to go up 1,000.

Grad: He has a market.

Feigenbaum: He was bought by Intel.

Grad: Because he had a very large market opportunity if he could succeed in doing what he was doing, is that correct? That's a good one. Other examples?

Okay, I'm going to finish then. It looks like the future of knowledge-based systems is really in a hybrid world, not standalone, but mixing something to do with this machine learning. We'll talk a little bit more about here and the old concepts that we started with at the beginning with AI.

Deep Thinking and Learning

Feigenbaum: I would just change one word in what you said, not to just say a future.

Grad: Okay, I'll accept that.

Feigenbaum: There's still a lot of problems that are pure symbolic problem solving. If we want to construct a scientific theory, for example, or write a novel or something like that, those really symbolic tasks, they're not recognition tasks. It's not like we're going to recognize what a great novel is or what a great theory is. We're going to construct it with deep thinking. I invented this term deep thinking to match deep learning.

Grad: So the question is if I really read 10,000, 100,000 novels and I had them ranked by experts as to how good the quality was on these things in 10 categories, would I be able to build a system that would then be able for a publisher to look at the novels proposed and say, "Aha, that's going to be great. It's going to sell 100,000 copies, and be good, or sell 20,000. It's going to be like my book."

Friedland: That's a different application than what Ed talked about. I think what Ed was saying is if you looked at those 10,000 and ranked them, would you then be able to create one that was one of the best? That's another step forward.

Grad: That's another forward. I'm not quite there yet.

Brown: That's the symbolic…

Perception and Mental Modeling

Friedland: There was a really good keynote talk given a little while ago. I forget who gave it. It pointed out that our generation, the 1970s, 1980s, and so on was the symbolic AI knowledge. That went through that hype cycle and now has leveled off. Because of the success of the recognition systems, statistical machine learning systems, this generation has done remarkably well on certain types of recognition problems. Those people are being hired for $250,000 or more, as fresh outs in the Valley. But many of the really hard problems in real life are going to require both forms, both the symbolic and the statistical, the recognition things. There will be pure recognition problems, and there will be pure symbolic problems, but the grand synthesis of AI is probably coming from the merger.

Brown: One quick one, this is something I learned from the Hearsay Project 30 or 40 years ago or something: There's not enough information in the speech wave to really understand human speech. You actually require the prediction capability of what's coming next in order to understand what we're saying. That's one of the reasons why you can't understand somebody with a heavy accent sometimes because you can't predict the next phoneme.

I'll give you an example that I learned from Raj Reddy a long time ago. I'm going to say an English sentence, and if you haven't heard this English sentence before, you won't understand it. It's going to be in grammatical English, and I'm going to speak at the same rate of speed. The sentence is, "In mud eels are, in clay none are." And you didn't get it. If I slow it down now and I say, "In mud, eels are, in clay, none are." You now know it. Now if I say it again at the same rate, "In mud eels are, in clay none are," you'll get it.

The issue there is it's the combination of the perception part of hearing and processing. Whatever the hardware is in here that processes the speech wave, it's the combination of that and your mental model that says, "What's the next word that's going to come out? What's the next phoneme that's going to come out to predict what's going to be next?"

That example I think really takes down to nitty-gritty this business of the combination of perceptual. Now you got to blow it up to real problems, but perceptual issues combine with some intellectual issues if you will and knowledge-based issues.

Grad: Let me ask you a question. In 1990, 1989, somewhere the first of the new AI applications, the new machine learning applications start to get built. Is the Kurzwell thing one of the first ones of those, or were there others before that?

Friedland: What do you mean by the Kurzwell thing?

Grad: It's language translation stuff.

Friedland: That wasn't Kurzwell.

Grad: Who was it?

Feigenbaum: He was OCR [optical character recognition].

Grad: OCR was an earlier time period you were saying.

Friedland: Kurzwell wasn't known for that.

Grad: What was Kurzwell known for?

Brown: The reading machine.

Friedland: Yes, wasn't it?

Grad: Tell me what you think were the first.

Feigenbaum: He also did the Kurzwell synthesizer, the music synthesizer.

Friedland: Yes, that's what he made his money on.

Grad: Something else that he did; I thought it was interpretive, language interpretation, maybe I'm wrong. Speech recognition, wasn't that his?

Gary Hendrix: Kurzwell had a speech system, but it predated that. It was early 1980s I think he had something. It wasn't very good, but it evolved, of course.

Grad: We talked about OCR earlier as being one early mechanism. What is one of the earliest? What do you know of the earliest applications where they were primarily using machine learning as against expert systems?

Allen: Going back into the late 1980s there was a company called Hecht-Nielsen. Does everybody remember that?

Feigenbaum: Yes, I do.

Allen: Okay. They had some hardware components that accelerated, but they had a platform for neural networks that they principally sold into the financial and the services industry dealing with things like risk assessment and so forth. That was kind of a play that had some software and some hardware associated with that. They were certainly used by Fair Isaac, but I don't know where they went as a company.

The sort of first initial phase of—I don't know if you would call it commercialization—certainly the realization of machine learning technology in a way which was easy for people to get at and absorb and use, where you were doing things for example like inducing decision trees, which was a very natural representation for expert systems, which had the property of explainability to some extent in a lot of that, other sorts of things, bubbled along into the late 1990s. It wasn't until you saw the emergence of hard-core neural network approaches that leveraged GPUs; that was in the mid to late 2000s.

Language Translation

Grad: What applications? What was it used for?

Allen: I think some of these simple kinds of expert system tasks were like decision support for diagnostics at point of care, some kinds of diagnostic applications in medicine; there was a smattering.

Feigenbaum: Face recognition in cameras.

Allen: Yes, those sorts of things where they're leveraging that technology. It was based on training off of data.

Feigenbaum: One of the biggest ones was this language translation.

Allen: Oh, right.

Feigenbaum: All the computational linguists that we had working AI couldn't solve that problem. It got solved by the fact that they were a heck of a lot of Estonians communicating with Germans and a lot of Germans communicating with Estonians. In the billions of things that they've said, Google has it all.

Feigenbaum: They went through all the documents in the literature of Canada because it all has to be in French and English and they had the exact translations. You need about a million examples in 1,000-dimensional space to make these statistical judgments. They finally were able to get that amount of data.

Grad: What's an example of natural language translation?

Feigenbaum: That's Google Translate.

Grad: But this is words. This is pictures, not my spoken speech. This is written, right?

Brock: Text.

Feigenbaum: All I do is feed the URL of a Japanese website to Google Translate, and it translates the whole page for me.

Grad: But that's today. I'm talking about the late 1990s and so forth. Was that being done then?

Feigenbaum: No. Google Translate was about 2010.

Allen: Well, actually getting it to scale.

Herb Schorr: Yes, we had the French-Canadian thing done in the late 1980s.

Allen: Yes, Peter Brown.

Thomas Haigh: I think in the late 1990s, before Google had it, there was a thing called Babel Fish. I don't know if someone acquired them, but the point is I think there were online translation services by the late 1990s.

Grad: Was that voice translation or text?

Haigh: Text translation.

<u>Harold Cohen's Intelligent Painter</u>

Kunze: I just want to say one thing since this thing is almost over. In 1973 there was a guy named Harold Cohen who made an intelligent painter. You guys should actually do some research on that.

Brock: We have Aaron downstairs.

Feigenbaum: It was probably the longest-lived expert system in the history of expert systems. Harold died two years ago. It was a 15,000-rule system representing the rules that he got out of his own head.

Kunze: I mean the thing was amazing. I was a programmer once, but this is one of the few things you could see, and you couldn't imagine, I couldn't imagine how he did it. Gordon Bell and I acquired his art.

Grad: He actually created paintings. Is that what you're saying?

Brock: The system did.

Kunze: He created a robot that would actually create a painting.

Grad: I see. Based upon these rules.

Feigenbaum: The robot part of it wasn't essential. Kurzwell was entranced by this, so he gave Harold a lot of money to make this available as free screen savers for people all over the world. With that money, Harold bought an industrial scale inkjet printer, so he got beautiful output that way without robotizing the thing. One of the robot things is downstairs.

Then Harold invented a process by which he could fix the inkjet spray so that it would last forever. These paintings were exhibited all over the world in museums and art galleries.

Grad: Is that an expert system?

Kunze: It's an expert system.

Feigenbaum: Oh yes. It's an expert, absolutely.

Kunze: It's a very early example of one—1973, according to Wikipedia.

Feigenbaum: That's when he started, but I invited him to this Stanford AI Lab for a year, and he stayed for two years.

Machine Learning Unsupervised Classification

Grad: Let's go back to AI. What were the earliest AI machine learning things that were done in the 1990s? We've mentioned the beginning work here. Anything else besides language?

Friedland: Well, it depends what you mean. If you're talking about purely statistical machine learning, that's a more specific question. There were lots of very successful machine learning systems of various types dating from the 1970s, 1980s, and 1990s.

For example, one of the people from yesterday, one of his mentors at NASA, Peter Cheeseman, developed a system called AutoClass. It was a Bayesian learning system which was spectacularly successful in analyzing space science databases.

In fact, an early observational satellite, the Infrared Astronomical Satellite, produced images of infrared objects in space, and the scientists themselves had produced a classification catalog. We tried using Peter's system on that same database. It produced a new set of classifications that the astronomers agreed was better than theirs and resulted in a new catalog. I could bring it in some day to show you, a whole new catalog done by that system.

Following onto that, scientists at JPL [Jet Propulsion Laboratory] built a machine learning system called Skycat that did a classification on the entire Palomar whole Sky Survey that was better than any humans had done on that. There were all types of machine learning methodologies that were used.

Grad: Are these mostly statistical stuff, you say, using Bayesian?

Friedland: Well, no. When people talk about machine learning right now, they usually mean Bayesian supervised classification, huge databases that are labeled, like Ed was describing. The machine looks at all those, and then after throwing all those into its filters, it is able to take objects that haven't been classified yet and say, "This is melanoma. This eczema."

The system I'm talking about was an unsupervised system. It basically decided from looking at the data where the lines of classification should be. It wasn't a neural net system; it simply analyzed probabilities, Bayesian probability rules to decide where the lines should be drawn and how many classes there should be. It wasn't statistical because statistical means based on lots and lots of prior labeled data. This was trying to find in the data what the patterns were.

Intersection of Symbolic and Statistical Technologies

Brock: I have a question. When do either the symbolic or the statistical technologies start? If they do, when do they start to intersect with things like algorithmic trading or high-frequency trading in the financial world?

Friedland: I guess when David Shaw probably started his company was the first time.

Brown: Yes, David Shaw was another Stanford student that runs a hedge fund now: David Shaw & Associates.

Friedland: Not anymore, he made so many billions he now runs a place that tries to use the same techniques for the good of humanity like medical diagnostics and stuff.

Friedland: When did David Shaw start D.E. Shaw Company? Do you know?

Brown: He was a grad student when he started.

Brock: When was that?

Brown: He was my roommate in the Serra House trailer.

Feigenbaum: What year was that?

Brown: 1972, 1973, 1974.

Feigenbaum: Yes, so he came to our place in the mid-1970s and left in 1980 and went to Columbia to teach. Well, he was following the normal path that he got taught, which is don't work in the abstract. Get a sandbox to play in. If you live in New York, it's either theater or finance.

Friedland: There's more money in finance.

Feigenbaum: You knew that he was working on high-speed computers. He did finance, so he went down to Wall Street and talked to those people. When he realized what they were doing, he said, "I can do that better than you." Then he started D.E. Shaw & Company in maybe 1985.

Friedland: That became multi-billions.

Brock: That was an expert system symbolic type approach?

Feigenbaum: No.

Brock: Was it the stock ticker?

Friedland: It was computational, but it certainly used the same learning type of methodologies we're talking about.

Feigenbaum: I just don't know enough about it to confirm or deny.

Friedland: Burt or David was asking, when did we start to put together in a real way the symbolic and the statistical?

Unidentified Speaker: D.E. Shaw started in 1988.

Grad: Let me switch questions. When did handwriting recognition pick up? We have OCR, but we didn't do handwriting until later.

Feigenbaum: We know that Jerry Kaplan's company Go failed. It was a handwriting recognition company. That was in the 1990s?

Friedland: Graffiti in the Palm world and the Apple Newton all attempted to do that.

Feigenbaum: And didn't do it well.

Friedland: It never crossed the chasm. There were fanatic early adopters of Graffiti. There may even be some in this room. I wasn't one. Maybe some other people were, but it never crossed the chasm into mainstream.

Grad: Were there any other major new applications introduced in the 1990s that were a result of using pattern recognition, machine learning, or any of these kinds of things?

Paul McJones: Well speech recognition had gotten good in that timeframe. Somebody that I know who has worked in that field said the more they fired their people with actual linguistics background and just went to machine learning, the better it worked.

Feigenbaum: I was using the smartphone Dragon app, which they were giving out for free. I talked to Raj Reddy about that. Dragon was originally started by one of his students, Jim Baker. He said, "Oh, they're giving it out for free because they just need all those samples. Nuance needs all those samples."

I think I can give you a path that I think is going to be exploited. We're entering the world that some people call the Internet of Things. I think that's a lousy name, but let's just call it smart things. Everything has computation in it and about it, and things in the world are not that simple. For example, I just bought a new car. I actually wrote to them and asked them if I can get a paper copy of the manual because the manual for it is 600 pages, and you can't imagine how complicated these systems in cars are now.

One of the ways to conceptualize what AI does, AI: is the ultimate on the computational spectrum. It's the ultimate translator. You express what it is you want the problem solver to do for you, and it translates it all the way down to exactly how a computer's going to do that. How many times have I wished that when I was in PowerPoint, I could just talk to it and tell it what I want PowerPoint to do or what I want Word to do for me and not have to understand all that stuff? How many times would I just like to talk to this car and tell the car what I want and not know which switches to press and which buttons and which sequences and all that?

Things are complex. I mean even dishwashers are complex. What we're going to see is that AI is going to move into that space; it's going to do small problem-solving jobs in which to interface humans to complex machines. Machine functionality can be as complicated as we want. The problem is people can't use it if it's too complicated, but they can if you provide them with intelligent assistance.

Hansen Hsu: All right. I wanted to ask a question related to what Doug said this morning about CyCorp throwing out the general problem solver because it was too slow and choosing between specialized problem-solving techniques. That seemed to imply that the move from general AI techniques to these more specific solutions was more of an implementation reason.

But then, Ed, you responded that you didn't think that there was such a thing as general intelligence. That seemed to be something that you said as a principle or a theory rather than a mere implementation detail. Was that correct?

Feigenbaum: Oh no, let's call it an empirical fact, **a**n empirical observation. You remember that joke that ends with “it's turtles all the way down”? Well, it's sort of like that, except its turtles almost all the way down. It's knowledge almost all the way down until you get to Aristotle. Then you get *modus ponens* and you get backward chaining and forward chaining and maybe a little bit of extra stuff from John McCarthy's work, but that's all you need for the reasoning engine. Maybe you need a blackboard framework that the two of us have used, but otherwise it's knowledge all the way down.

Feigenbaum: Look at the general problem software of Newell, Shaw, and Simon in the early 1960s. Then look today and ask, are there any general problem solvers? I mean, means-ends analysis, did that get anywhere?

END OF THE INTERVIEW

