

5 Probabilistic Computation

Traditional computers often seem brilliant and simple minded at the same time. On the one hand, they can perform billions of high-precision numerical operations per second with perfect repeatability. On the other hand, they fail catastrophically when their inputs are incomplete or ambiguous. These strengths and weaknesses flow from their mathematical foundations in deductive logic and deterministic functions.

Probabilistic computation systems is working to change all this, by building the world's first natively probabilistic computers, designed from the ground up to handle ambiguity, make good guesses, and learn from their experience. Instead of logic and determinism, its hardware and software are grounded in probability distributions and stochastic simulators, generalizing the mathematics of traditional computing to the probabilistic setting. The result is a technology as suited to making judgments in the presence of uncertainty as traditional computing technology is to large-scale record keeping.



360°
thinking.

Deloitte.

Discover the truth at www.deloitte.ca/careers

© Deloitte & Touche LLP and affiliated entities.



As an alternative paradigm to deterministic computation, it has been used successfully in diverse fields of computer science such as speech recognition (Jelinek 1998), natural language processing (Charniak 1993), and robotics (Thrun 2000). Its success lies in the fact that probabilistic approaches often overcome the practical limitation of deterministic approaches. A trivial example is the problem of testing whether a multivariate polynomial given by a program without branch statements is identically zero or not. It is difficult to find a practical deterministic solution, but there is a simple probabilistic solution: evaluate the polynomial on a randomly chosen input and check if the result is zero.

Probabilistic automaton

In mathematics and computer science, the probabilistic automaton (PA) is a generalization of the non-deterministic finite automaton; it includes the probability of a given transition into the transition function, turning it into a transition matrix or stochastic matrix. Thus, the probabilistic automaton generalizes the concept of a Markov chain or subshift of finite type. The languages recognized by probabilistic automata are called stochastic languages; these include the regular languages as a subset. The number of stochastic languages is uncountable.

The probabilistic automaton has a geometric interpretation (Rabin 1963): the state vector can be understood to be a point that lives on the face of the standard simplex, opposite to the orthogonal corner. The transition matrices form a monoid, acting on the point. This may be generalized by having the point be from some general topological space, while the transition matrices are chosen from a collection of operators acting on the topological space, thus forming a semiautomaton. When the cut-point is suitably generalized, one has a topological automaton. An example of such a generalization is the quantum finite automaton; here, the automaton state is represented by a point in complex projective space, while the transition matrices are a fixed set chosen from the unitary group. The cut-point is understood as a limit on the maximum value of the quantum angle.