Distributed computing is the field in computer science that studies the design and behaviour of systems that involve many loosely-coupled components. The components of such distributed systems may be multiple threads in a single program, multiple processes on a single machine, or multiple processors connected through a shared memory or a network. Distributed systems are unusually vulnerable to no determinism, where the behaviour of the system as a whole or of individual components is hard to predict. Such unpredictability requires a wide range of new techniques beyond those used in traditional computing.

Like other areas in computer science, distributed computing spans a wide range of subjects from the applied to the very theoretical. On the theory side, distributed computing is a rich source of mathematically interesting problems in which an algorithm is pitted against an adversary representing the unpredictable elements of the system. Analysis of distributed algorithms often has a strong game-theoretic flavour, because executions involve a complex interaction between the algorithm's behaviour and the system's responses.

Michael Fischer is one of the pioneering researchers in the theory of distributed computing. His work on using adversary arguments to prove lower bounds and impossibility results has shaped much of the research on the area. He is currently actively involved in the study of security issues in distributed systems, including cryptographic tools and trust management.

James Aspnes' research emphasizes the use of randomization for solving fundamental problems in distributed computing. Many problems that turn out to be difficult or impossible to solve using a deterministic algorithm can be solved if processes can flip coins. Analysing the resulting algorithms often requires using non-trivial techniques from probability theory.

Distributed systems research at Yale includes work in both programming language support for distributed computing, and in the use of distributed systems techniques to support parallel programming. Such work is designed to lift some of the burden of understanding complex distributed systems from the shoulders of distributed system designers by letting the compiler or run-time libraries handle issues of scheduling and communication.

Zhong Shao's FLINT project focuses on developing a new mobile-code architecture to support efficient data and program migration on distributed and heterogeneous computing platforms. FLINT uses a common typed intermediate language to support safe execution of code written in multiple programming languages such as Java, C, and ML.

David Gelernter's work on developing the Linda coordination language and related tools is an example of using distributed system techniques to support parallel programming. Linda provides a virtual "tuple space" through which processes can communicate without regard to location. His current Life streams project similarly simplifies information-management tasks, by freeing the user from many of the clerical duties imposed by traditional file systems. Avi Silberschatz specializes in transaction management techniques as they relate to both distributed database systems and multidatabase systems.

Recent improvements in computers and communication technology have made staggering amounts of information available to us. Fortunately there has been a concomitant increase in processing power. One of the major challenges for computing lies in finding ways to organize the available processing power in order to extract maximum benefit from the available information. Different aspects of that problem are addressed by machine learning, knowledge discovery and data mining.

Every field of human endeavor is affected by these issues; therefore a few examples will suffice. In bioinformatics, there are opportunities to use information about the human genome to design more effective medications, information from medical records to track drug side-effects or emerging diseases, and information from medical imaging to understand the structure and function of the brain and other organs. In robotics, major challenges lie in making use of visual and other sensed information to allow a robot to adapt flexibly to its environment. In the domain of e-commerce, all but the simplest artificial agents designed to gather information and act on behalf of human users will have to be equipped with the ability to adapt their behaviour successfully to unexpected conditions. The very large number of documents and databases available on the web has fostered the development of tools for abstracting, extracting, and combining sources in a variety of ways to make the information more usable to humans.

Machine learning is driven by the goal of making programs or agents that exhibit useful learning behaviour, autonomously or in cooperation with teams of other agents, either human or artificial. One goal is software that is easier to use, e.g., a word-processing program that can guess from an example or two what text transformation a user wishes to make. Machine learning can be used in the design of a team of soccer-playing robots to allow the team to adapt its tactics to the observed behaviour of an opposing team. At the level of hardware, sophisticated processor designs can adapt to characteristics of the program currently being run to achieve increases in speed.

As learning components are increasingly included in software and embedded systems, it is important to have a clear understanding of their theoretical and practical strengths and limitations. Computational learning theory – a theoretical branch of machine learning–develops and studies algorithmic models of learning, using tools from analysis of algorithms, theory of computation, probability and statistics, game theory, and cryptography. Fundamental models and results in computational learning theory have established the learnability or non-learnability of a number of classes of concepts from various kinds of information. Many exciting questions and areas remain open, including learnability of probabilistic models, learnability in games and other cooperative and competitive situations, and aspects of learnability in natural language. As one example, in the context of e-commerce it is important to study how a population of agents can learn and use information about the trustworthiness of the other agents in the population.

Knowledge discovery and data mining address the question of extracting usable information from very large and heterogeneous sources of information. In this context, even a linear-time algorithm can be too slow. One promising approach to a collection of text documents is to treat it as a matrix of the occurrences of distinct words in the documents, and to approximate the information in the matrix using a low-rank singular value decomposition. This has motivated the search for fast provable approximations of important matrix-vector computations based on weighted sampling.

Machine learning tools are increasingly controlling various aspects of modern society: from social interactions (e.g., Facebook, Twitter, Google, and YouTube), economics (e.g., Uber, Airbnb, and Banking), learning (e.g., Wikipedia, MOOCs), to governance (Judgements, Policing, and Voting). These systems have a tremendous potential to change our lives for the better, but, via the ability to mimic and nudge human behaviour, they also have the potential to be discriminatory, reinforce societal prejudices, and polarize opinions. Moreover, recent studies have demonstrated that these systems can be quite brittle and generally lack the required robustness to be deployed in various civil/military situations. Our goal is to develop a principled and thoughtful approach to the redesign of machine learning tools for society.

Many procedures in machine learning and nature at large—Bayesian inference, deep learning, protein folding—successfully solve non-convex optimization and sampling problems that are NP-hard, i.e., intractable on worst-case instances. Moreover, often nature or humans choose methods that are inefficient in the worst case to solve problems in P. Our goal is to develop a rigorous theory to resolve this mismatch between reality and the predictions of worst-case analysis. Such a theory could identify structure in natural inputs that helps sidestep worst-case complexity, and guide the choice of ML algorithms and their parameters.

Faculty members with interests in this area include Dana Angluin, Amin Karbasi, Smita Krishnaswamy, John Lafferty, Sahand Negahban, Dragomir Radev, Brian Scassellati, Daniel Spielman, Marynel Vázquez, and Nisheeth Vishnoi.

At the Language, Information, and Learning lab at Yale (LILY), we are working on the following cutting-edge research in natural language processing (NLP).

Multilingual information retrieval – We collaborate with researchers from Columbia University, the University of Maryland, the University of Edinburgh, and the University of Cambridge to build search engines for English users to query documents written in other languages including Swahili and Tagalog. This cross-lingual information retrieval system improves our capability of understanding and processing different low-resource languages and it offers users a reliable access to foreign documents.

Resources for learning NLP and AI – We aim to make dynamic research topics more accessible to the public by generating surveys of topics, discovering prerequisite relations among topics and recommending appropriate resources based on a given individual's education background and needs. We host a search engine, AAN (All about NLP) and tool which is available at http://aan.how.

Medical NLP –We work with the Centre for Outcomes Research & Evaluation (CORE) at the School of Medicine at Yale, investigating the use of NLP on electronic health records. The tasks include abbreviation disambiguation, patient digital pairs, and patient history record summarization. We are interested in how to transfer general knowledge to the medical domain and how to take good advantage of the limited high-quality data. Besides, we are also developing an NLP toolkit for the medical domain to perform tasks like named entity recognition and relation extraction.

Semantic parsing, Natural language database interfaces and dialogue systems – The goal of this project is to allow users with any background to talk to relational databases directly using human language. In this way, anyone can easily query and analyze a vast amount of data. You can check out our current work here. Moreover, we also aim to build conversational interfaces for even more natural information access, where the users participate in a conversation and the system takes the responsibility of choosing data sources and developing queries.

We also work on text summarization, question answering, and graph methods for NLP, question answering, etc.

Computers have dramatically changed the practice of many disciplines including engineering, medicine, and science. For example, it is now possible to test thousands of product designs and run thousands of trials without first building a prototype for each product or conducting an elaborate experiment for each trial. The impact of this new ability, this power to simulate the real thing, is revolutionizing the practice of engineering and science. Reliability, flexibility, efficiency, and (often attractive) costs have placed scientific computation as the keystone between theory and applications.

Basic research in scientific computing conducted at Yale is being applied to a wide range of applications. Currently the emphasis is on problems originating in the biomedical sciences. These range from high throughput genomic search engines to simulations of biological cells. Active collaborations are in place with several researchers in the Yale Biology Departments and the Yale Medical School. It is clear that high performance scientific computing is an essential component of the “genomic revolution.”

Scientific computing research at Yale emphasizes algorithm development, theoretical analysis, systems and computer architecture modelling, and programming considerations. Algorithm development is concerned with finding new, fast and/or parallel methods. Theoretical analysis evaluates such questions as rates of convergence, stability, optimality, and operation counts. Systems modelling research examines the performance implications of the interactions between computationally intensive algorithms, operating systems, and multiprocessor machines.

Programming considerations include coding efficiency, numerical accuracy, and generality of application, data structures, and machine independence.

One focus of work in scientific computing at Yale today is the adaptation of fast serial algorithms to parallel multiprocessor environments. Clusters or LANS of workstations and PCs are commonly used as virtual multiprocessors.

Underlying scientific computing are applied mathematical techniques for modelling physical systems. Mathematical models are widely used throughout science and engineering in fields as diverse as theoretical physics, bioinformatics, robotics, image processing, and finance. In spite of the broad range of applications, there are only a few essential techniques used in attacking most problems. Research in applied mathematics at Yale comprises mathematics and its applications in computer science, statistics, engineering, and other sciences. The area is conveniently divided into two general areas: discrete mathematics (such as discrete algorithms, combinatorics and combinatorial optimization, and graph algorithms), and continuous mathematics (comprising many traditional areas such as linear and nonlinear partial differential equations, numerical analysis, harmonic analysis, geometric algorithms, and so on).