

Fundamental limits in computing functions over networks

Ardhendu Tripathy

Many applications such as parallel processing, distributed data analytics and sensor networks often need to compute functions of data that are observed in a distributed manner over a network. A network can be modeled as a directed graph, each vertex of which denotes a node that can carry out computations and communicate with its neighbors. The edges of the graph denote one-way noiseless communication links. A subset of nodes - called sources - observe independent messages, and a possibly different subset of nodes - called terminals - wish to compute a particular demand function of the messages. The information transmitted on the edges are specified by a set of functions, one for each edge, called a network code. We are interested in network codes that allow each terminal to compute the demanded function with zero-error.

In the first part of the talk, we assume that the message random variables are independent and uniformly distributed over a finite field. The demand function is set to be the finite field sum of all the messages observed in the network. A valid network code for this sum-network problem allows each terminal to compute the sum, and has an associated computation rate. We wish to find the best possible computation rate for a given sum-network; this value is called its computation capacity. Finding the computation capacity of a sum-network is known to be a difficult problem. Here we are able to evaluate it for certain systematically constructed sum-network problem instances. The construction procedure uses incidence structures, whose combinatorial properties allow us to be able to evaluate the computation capacity of the constructed sum-networks. An important aspect of the problem that we uncover is the strong dependence of the computation capacity on the finite field over which the sum is to be computed. This is shown by a sum-network, whose computation capacity is 1 over a finite field and close to 0 over a different finite field. We also construct sum-networks whose computation capacity can take on arbitrarily many different values over different finite field alphabets.

In the second part of the talk, we focus on a particular directed acyclic network that has four nodes and four edges. It is the simplest instance of a network that does not have a tree structure. Three of the nodes are sources that observe independent messages that are uniformly distributed over a finite discrete alphabet. The fourth node is a terminal which wants to compute a demand function of the three messages. The demand function is an arbitrary discrete-valued function. We focus on network codes that have different rates on each of the four edges, thus we have a rate tuple associated with every valid network code. We obtain inner and outer bounds for the rate region that contains all valid rate tuples for function computation over this network.