Solving Mastermind using Belief Propagation

Jeremy Thorpe, Robert McEliece

November 12, 2001

Abstract

We demonstrate the versatility of the Belief-propagation (BP) algorithm by applying it to the game of Mastermind. Furthermore, we argue that Mastermind may provide a reasonable model for the robot colony inference problem. We also propose that Mastermind could provide a basis for comparison of different inference alogrithms.

Although simulation results show an appreciable gap between BP and optimal inference, the BP algorithm is feasible for large problems. The challenge now is to find better performing, yet practical algorithms.

1 Background

In the future, spacecraft that scientists send to extraterrestrial planets to collect scientific data may be filled with small, inexpensive robots.

Each of the robots would be equipped with a small number of sensors with which to sense its own environment. We may imagine that one robots own measurements would be insufficient to accurately determine its environment. The robots could overcome this by communicating with their neighbors estimated probabilities

The robots should be able to remove as much of their uncertainty about the environment as possible while using as little energy in communication as possible.

[Below] is an illustration of one possible scenario. A colony of robots has landed on Mars. The robots will together form a coherent idea of their environment and then communicate that idea back to an orbiter and eventually back to earth for consumption by JPL scientists. Another compelling scenario is that the robots must use their observations to act autonomously, locating prime samples for analysis or for transport back to the earth, all without the intervention of earth-bound decision-makers.

2 Mastermind

Popularized by Hasbro, Mastermind is a popular childrens game. The object of the game is to guess the colors of a certain row of pegs hidden from the players view. The player obtains information through making guesses and receiving replies which tell him how close his guess is to actual colors of the pegs. The game ends when the guess matches the actual vector of colors.

In this analysis, we simplify the game considerably. First, we suppose that the elements of the hidden vector X comes from the binary alphabet 1,-1. The elements of each guess G come from the alphabet 1,-1,0. The reply, Y, is equal to the inner product between X and G. Note that a zero component of G is equivalent to not guessing at that position. The game is also simplified in that the guesses are made at random before the game begins, instead of being selected by the player after knowing the result of the previous guess.

An instance of the game is parameterized by n, the length of X and G, m, the number of guesses, and s, the number of non-zero components per guess.

A player is given the values of all of the Gs and Ys and must return an estimate of the probability of each of the Xs individually. The distortion [1] between this estimate and the true value of X, along with the complexity of the algorithm determine its fitness.

3 Concepts

3.1 Communication Network

The communication network represents the ability and cost of communication among the robot colony members. If the robots are dropped from a spacecraft, the communication network is likely to be an ad hoc network-that is, a network whose topology cannot be known in advance. If the robots are mobile, then the topology of the network may also be time-varying; connections become available when two robots come in contact with each other.

3.2 Inference Problem

Technically speaking, an inference problem is to compute the a posteriori distribution PrX—Y=y, when some a priori distribution PrX,Y is given. The solution is given by Bayes Rule and is always PrX,Y=y/PrY=y. More generally, we may be interested in an approximate distribution, or just the value of X which is most probable given the value of Y.

Examples of inference problems include decoding of error-correcting codes received over noisy channels, detection of airplanes from radar data, and the childrens game of Mastermind.

3.3 Output Goodness Measure

We think of the robot colony as a producer of soft, or probabilistic, information. Robots may also consume probabilistic information since they may make decisions about where to go and where to point their sensors.

We choose a goodness measure [1] that reflects the need for accurate soft information. Briefly, it is minus the logarithm of the estimated probability of the event that happens.

4 Belief Propagation Algorithm

Belief Propagation is a powerful and general algorithm which solves, exactly in some cases and approximately in others, an inference problem. The BP algorithm solves an inference problem in the case where the a priori probability distribution can be represented by a Bayesian Network.

If the Bayesian Network is characterized by a tree-like graph whose edges represent the dependencies between variables, the BP algorithm solves the inference problem exactly. However, if the Bayesian Network has cycles, then BP solves it only approximately.

BP is an iterative algorithm which works by making progressively better estimates of probability functions (or beliefs) about each of a set of basic variables. The iterations work in two parts:

1) In the first half of an iteration, probabilities of each random variable are generated from a priori beliefs and likelihoods generated in the second half of an iteration. On the very first iteration, only a priori beliefs are used. The estimated probabilities are passed in one direction on each edge in the graph.

2) In the second half of an iteration, likelihoods of each random variable are generated (the conditional probability PrY—X is called a likelihood of X). Likelihoods are passed in the opposite direction on the graph.

5 Robot Colony/Mastermind analogy

There is meant to be an analogy between the robotic colony inference problem and Mastermind. The vector variable X represents all of the environmental variables which the robots wish to determine.

Each of the guesses G represent the way in which each sensor is dependent on the environment–sensitivity can be positive or negative to different environmental variables. We imagine that each one of the sensors is physically located on a different robot from the other sensors. Naturally, the variable Y represents the value that the sensor observes.

6 Mastermind BP Decoder

The Belief-Propagation decoder for the game of Mastermind is shown in the figure to the left. The yellow circles form a graph which can be seen to be a Bayesian Network since the Xis are chosen independently of each other and then the Yis are stochastic functions only of the Xs with which they share an edge (deterministic functions, in fAact).

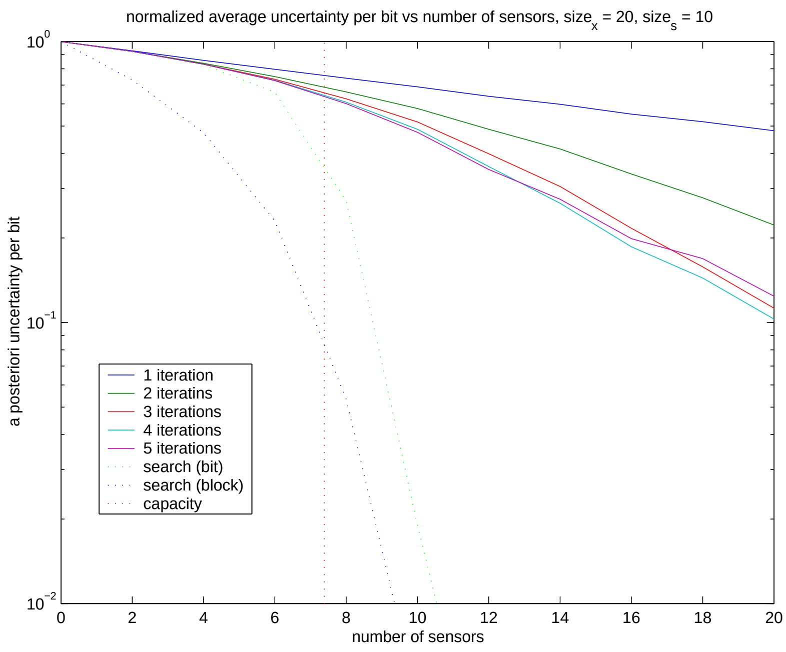
This is a bipartite graph, and the variables which we estimate are on the right side of the graph. The evidence nodes are on the left. In this graph, probability functions are passed from right to left; likelihood functions are passed from left to right.

The performance of the BP algorithm is shown in the two slides to the right. The game parameters are 10 non-zero elements per guess, and the length of X 20 and 50. The independent variable is the number of guesses, and the dependent variable is the distortion as measured by our defined distortion measure. Lower on the graph represents better performance.

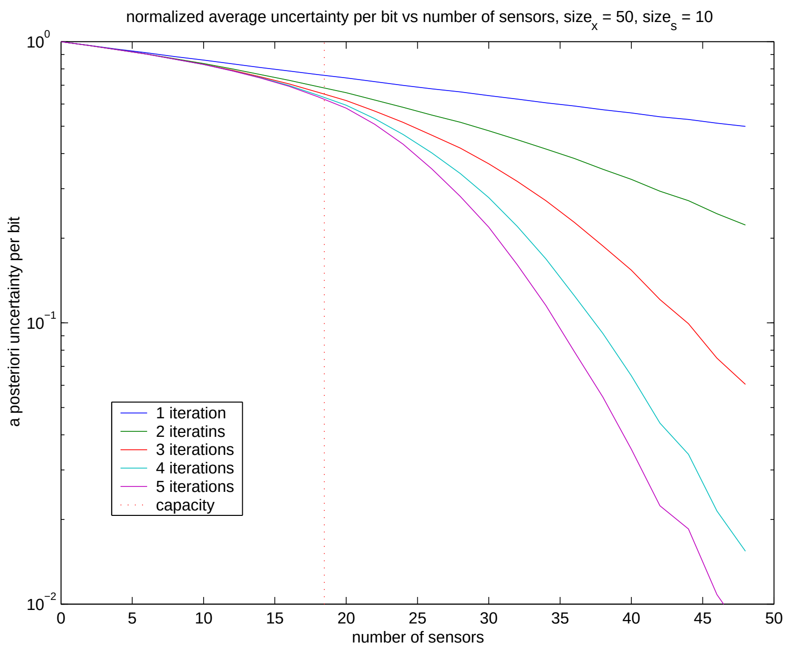
In the upper figure, it takes 20 sensors to reduce the uncertainty to approximately 1/10 of a bit for each of 20 bits in X, compared to about half that number for exhaustive search. However, in the lower figure, it takes only about 34 sensors to reduce the uncertainty of 50 components to 1/10 of a bit.

Another point of comparison is marked by the blue vertical line, which represents the number of sensors whose total entropy is equal to the vector X which we are trying to determine. To the left of this line, it is impossible to drive the uncertainty to zero.

It is also noteworthy that the number of messages which must be passed (a measure of complexity relevant to robot colonies) is linear in g, s, and the number of iterations used. The complexity of exhaustive search is exponential in n, although the number of messages passed is just proportional to g



;



;

References

[1] J. Thorpe "A Goodness Criteria for Probabilistic Estimates" unpublished (and unwritten).