Representation, Reasoning, and Learning for a Relational Influence Diagram Applied to a Real-Time Geological Domain

Matthew Dirks*, Andrew Csinger†, Andrew Bamber†, David Poole*
mcdirks@cs.ubc.ca

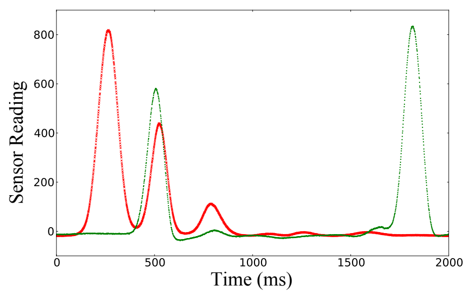
*University of British Columbia (UBC), †MineSense Technologies Ltd.

Abstract. Mining companies typically process all the material extracted from a mine site using processes which are extremely consumptive of energy and chemicals. Sorting the rocks containing valuable minerals from ones that contain little to no valuable minerals would effectively reduce required resources by leaving behind the barren material and only transporting and processing the valuable material. This paper describes a controller, based in a relational influence diagram with an explicit utility model, for sorting rocks in unknown positions with unknown mineral compositions on a high-throughput rock-sorting and sensing machine. After receiving noisy sensor data, the system has 400 ms to decide whether to divert the rocks into either a keep or discard bin. We learn the parameters of the model offline and do probabilistic inference online.

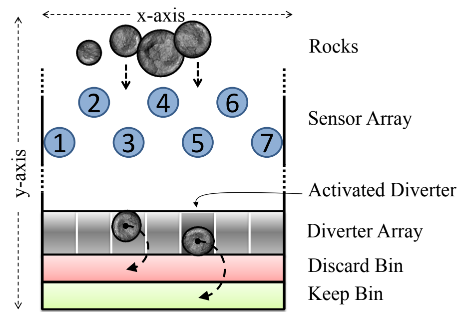
1 Introduction

This paper considers the problem of sorting rocks, separating valuable, high-grade rocks from low-grade rocks as they pass over an array of electromagnetic sensors. By sorting more effectively ahead of the mill, we reduce costs and help preserve the environment because the amount of material sent to further downstream mining processes is reduced[1][7]. The rock sorting machine, known as SortOre™, has been deployed in field pilot situations in Ontario, Canada and Guatemala in addition to the 60 tonnes per hour unit available in lab.

We have developed Rock Predictor Sorting Algorithm (RPSA) based on a rock sorting machine on which we have performed training and evaluation. The machine, with schematics shown in Fig. 1(b), passes rocks on a conveyor belt, moving downward on the y-axis, over a **sensor array** of 7 electromagnetic (EM) coils. From each sensor we read the change in voltage caused by the rock disrupting the magnetic field above the EM coil every millisecond (ms), which we call **sensor readings**. The rocks travel for about 400 ms after the sensor array to the end of the conveyor belt where they fall off onto a **diverter array** which may have one or more diverters **activated**, displacing rocks into a **keep bin** or else leaving them to fall into a **discard bin**. Both the sensing and diverting are imperfect.



(a) Sample of sensor readings from 2 neighboring electromagnetic coil sensors.



(b) A schematic of the sorter.

Fig. 1. Rock sorting machine.

An example of sensor readings from 2 neighboring sensors of an unknown number of rocks is shown in Fig. 1(a). We can see there is a rock passing over the red sensor at around 250 ms, and one over the green sensor at 1800 ms. At 500 ms, most likely a rock passed between the sensors, slightly closer to the green sensor, or there were two rocks. If there was only one rock, because it passed between the sensors, its signal is lower than if it had passed directly over a sensor, and we should adjust for this. We need to be able to reason about the inherent uncertainty about the world based on the sensor readings.

Much research has been done in electromagnetic (EM) sensing for other applications, such as locating unexploded ordnance beneath the ground[5]. Lowther simulates electromagnetic devices in a full 3D setting in which he very accurately approximates electromagnetic physics[11]. However, computation time ranges from 1 second to 1 hour per simulation typically – whereas our work requires a very fast response. In a lab setting, Mesina et al. have explored the use of EM for sorting scrap metal, using simple thresholds to classify particles[12]. Their work differs from ours in that they do not attempt to locate the particles or determine how much metal is present. Our work appears to be the first to apply artificial intelligence to the task of sensing and sorting individual rocks.

2 The Model

The system can be described using a relational influence diagram, as seen in Fig. 2. Influence diagrams, developed by Howard and Matheson[8], are an extension to belief networks and consist of 3 node types: random variables (ovals), decision variables (rectangles), and utility nodes (diamonds). Arrows going into random variables represent probabilistic dependence: the value of a random variable is probabilistically dependent on its parents. Arrows going into decision variables represent the information available when making a decision. Arrows going into a utility node represent deterministic dependence on the value of its parents.

We extend influence diagrams further with the idea of relations between variables as in relational probabilistic models[13]. A similar technique is used under the name relational decision diagram[9]. We use plate notation[3] to avoid redundancy when a node, or set of nodes, is repeated more than once. A plate is drawn as a box around one or more nodes, and is labelled with the name of a class. For instance, we have zero or more rocks which are represented by the plate labelled "Rock (≥ 0)". The nodes within a plate are duplicated, along with any arrows connected to them, for each instance of the class. We chose to use a relational influence diagram because it explains the model and reveals explicitly what is and isn't being modeled, the assumptions made, and the areas where future work may need to take place.

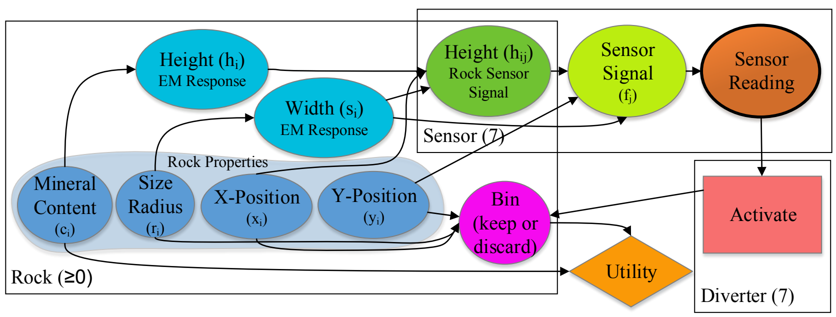


Fig. 2. Relational Influence Diagram

There is an unknown number of rocks, denoted with an i subscript (1 through n), which may or may not be overlapping. We assume rocks are circles and we do not model mineral type (iron versus copper bearing minerals, etc), instead we focus on a scalar mineral content which is tailored to the rocks we are looking for. We model each rock as a 2-dimensional symmetric Gaussian function, with height and width. Together these form the **EM response** of the rock (see Fig. 2 and Table 1). The rocks do not necessarily pass directly over a sensor, but may pass between sensors or over multiple sensors. Sensors are denoted with a j subscript. The **rock sensor signal** node is within two plates (rock and sensor) in Fig. 2 because at each time, for each rock, there is one rock sensor signal for each of the 7 sensors. A rock's sensor signal is the signal of the rock over a particular sensor, which we model as a 1-dimensional Gaussian. The height of this Gaussian depends on the EM response of the rock. For each sensor, the **sensor signal**, $f_j(t)$, sums each rock sensor signal from the rocks passing (partially or wholly) over the sensor, where t is continuous time. Finally, a **sensor reading** is a noisy sample from the sensor signal at discrete times (milliseconds). Given only the sequence of sensor readings of all 7 sensors, a decision is made to activate or not activate each diverter for each millisecond in time by comparing each rock's mineral content to a threshold (T in Table 1). The diverters guide each rock into either the keep bin or the discard bin. However, rocks can end up in the wrong bins because the diverter array is a chaotic system, with rocks tumbling and colliding, and there are delays in activating the diverters. The **bin** node simulates this chaotic behaviour. Rocks can also pass between sensors unnoticed and can be close together or even on top of another. The bin and mineral content of each rock determine the value of the **utility** function which takes into account misclassified rocks, misclassification costs, and class distributions. The controller optimizes the normalized expected cost (NEC), as shown at the end of Table 1. Misclassification costs depend on various factors outside of the model (e.g. environmental costs, the cost of transportation, and the price of minerals). C_{FN} is the cost of each false negative (a good rock diverted to the discard bin) and C_{FP} is the cost of each false positive (a bad rock diverted to the keep bin). Class distributions are the probability of a good and bad rock. A mine operator will input the misclassification costs and class distributions under which the mine operates which affect how the sorting algorithm behaves and the performance therein. In Table 1, FN_r is the false negative rate ($\frac{FN}{TP+FN}$), and FP_r is the false positive rate ($\frac{FP}{FP+TN}$). FN is the count of false negative rock classifications, and similarly for FP, TN, TP.

Node Name	Definition
Height – EM Response	$h_i = c_i m$
Width – EM Response	$s_i = ar_i + b$
Height – Rock Sensor Signal	$h_{ij} = h_i \mathrm{e}^{-1/2 \frac{(x_j - x_i)^2}{s_i^2}}$
Sensor Signal	$f_j(t) = \sum_{i=1}^{n} h_{ij} \mathrm{e}^{-1/2 \frac{(t-y_i)^2}{s_i^2}}$
Sensor Reading	$reading \sim \mathcal{N}(f_j(t), \sigma)$
Activate	$\begin{cases} yes & \text{if } c_i \geq T \\ no & \text{otherwise} \end{cases}$
Utility	$NEC = \dfrac{FN_r \cdot P(+) \cdot C_{FN} + FP_r \cdot P(-) \cdot C_{FP}}{P(+) \cdot C_{FN} + P(-) \cdot C_{FP}}$

Table 1. Definitions of relation influence diagram nodes.

3 Inference

There are two computational problems. First is the online reasoning that occurs in the approximately 400 ms between the time a rock passes over the sensors and the time it reaches the diverters. The second is the offline computation to learn the parameters of the model.

Online Inference At runtime RPSA samples rock parameters – the number of rocks, and the position, size, and mineral content of each rock – then generates a sensor signal for each sensor conditioned on the rock parameters and compares the sensor signal to the observed sensor readings. Sampling repeats until time runs out, at which point the maximum a posteriori (MAP) hypothesis is chosen. Each rock's mineral content from this hypothesis is compared to a threshold parameter (T in Table 1). If the mineral content exceeds this threshold, the rock is classified as good and the appropriate diverters are activated. In most cases only one diverter is activated, but if the rock position is predicted to be on the boundary between two diverters then both diverters are activated.

MAP takes into account both the prior and the likelihood. The prior information we use is an exponential distribution over the number of rocks and also reflects that it is unlikely that rocks overlap (they tend to fall off each other). The likelihood specifies how well the hypothesis fits the observed data. RPSA is an anytime algorithm[15]; the search can be stopped at any time in which case the algorithm returns the best result it has found. We experimented with a number of search algorithms including exhaustive search, MCMC[4], gradient-based methods[14], and random search[2]. An exhaustive search over all possible rock properties cannot be done in a reasonable amount of time. MCMC and gradient-based methods are also too computationally intensive for our timing requirements, plus they have a tendency to get stuck in local optima. What worked best is a mix between random and exhaustive search with a coarse discretization of some of the parameters.

Offline Learning A number of random restarts of the whole online procedure ensure RPSA has searched enough of the space. The number of restarts were automatically chosen offline, as well as the values for a, b, m, σ, and T from Table 1. We learn the model offline using recorded sensor data and video. We ran about 480 rocks through the machine while diverting using our algorithm (with hand-picked parameters since the model was not optimized at data collection time). Generating data and manually labelling every rock's position, approximate size, and mineral content (either good or bad) is expensive, but it allows us to train and evaluate the sorting machine in a simulated setting. With this data we automatically configure our model's parameters using a state-of-the-art algorithm configuration method called sequential model-based algorithm configuration (SMAC)[10]. SMAC searches over the space of parameters optimizing a given cost function — in our case we optimize for our utility function, NEC.

Conclusion

Sorting rocks using electromagnetic sensors is a challenging real-world planning problem where there is an unknown number of objects and we have to act in real-time. We created a new sorting algorithm, RPSA, by modeling the problem using a relational influence diagram and automatically configuring its parameters offline. RPSA does online inference and beats the previously used algorithm, VBSA, by 9% on average (see [6]), while still being able to make decisions online within the real-time constraint.

References

1. Bamber, A., Klein, B., Pakalnis, R., Scoble, M.: Integrated mining, processing and waste disposal systems for reduced energy and operating costs at xstrata nickel's sudbury operations. Mining Technology 117(3), 142–153 (2008)
2. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. The Journal of Machine Learning Research 13, 281–305 (2012), http://dl.acm.org/citation.cfm?id=2188395
3. Buntine, W.L.: Operations for learning with graphical models. Journal of Artificial Intelligence Research 2, 159–225 (1994)
4. Christian Robert, G.C.: A short history of Markov chain monte carlo: Subjective recollections from incomplete data. Statistical Science 26(1), 102–115 (2011), http://www.jstor.org/stable/23059158
5. Das, Y., McFee, J.E., Toews, J., Stuart, G.C.: Analysis of an electromagnetic induction detector for real-time location of buried objects. Geoscience and Remote Sensing, IEEE Transactions on 28(3), 278–288 (1990)
6. Dirks, M.: Sensing and Sorting Ore Using a Relational Influence Diagram. Master's thesis, University of British Columbia (2014), http://circle.ubc.ca/handle/2429/49998
7. Hitch, M., Bamber, A., Oka, P.: Presorting of high grade molybdenum ore–a case for enhanced small mine development. Journal of Engineering and Applied Science 2(5), 129–135 (June 2015)
8. Howard, R.A., Matheson, J.E.: Influence Diagrams. Decision Analysis 2(3), 127–143 (September 2005), http://pubsonline.informs.org/doi/abs/10.1287/deca.1050.0020
9. Hsu, W., Joehanes, R.: Relational decision networks. In: Proceedings of the ICML Workshop on Statistical Relational Learning. p. 122 (2004)
10. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Proceedings of the 5th International Conference on Learning and Intelligent Optimization. pp. 507–523. LION'05, Springer-Verlag, Berlin, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-25566-3_40
11. Lowther, D.: The development of industrially-relevant computational electromagnetics based design tools. IEEE Transactions on Magnetics 49, 2375–2380 (May 2013)
12. Mesina, M.B., de Jong, T.P.R., Dalmijn, W.L.: Improvements in separation of non-ferrous scrap metals using an electromagnetic sensor. Physical Separation in Science and Engineering 12(2), 87–101 (2003), http://dx.doi.org/10.1080/1478647031000139079
13. Poole, D.: Logic, probability and computation: Foundations and issues of statistical relational AI. Logic Programming and Nonmonotonic Reasoning (2011), http://link.springer.com/chapter/10.1007/978-3-642-20895-9_1
14. Snyman, J.A.: Practical mathematical optimization : an introduction to basic optimization theory and classical and new gradient-based algorithms. Applied optimization, Springer, New York (2005), http://opac.inria.fr/record=b1132592
15. Zilberstein, S.: Using anytime algorithms in intelligent systems. AI Magazine 17(3), 73–83 (1996), http://rbr.cs.umass.edu/shlomo/papers/Zaimag96.html