Predicting water pump failure in Tanzania and optimising maintenance routes

Sam Kim
Harvard College
Cambridge, MA 02138
samuelkim@college.harvard.edu

Gareth Haslam
Harvard Extension School
Cambridge, MA 02138
haslam.gareth@gmail.com

Abstract

We describe MCMC techniques to model the functionality of water pumps in Tanzania and suggest an optimised route for their maintenance. Using field data describing various attributes of each pump, such as construction year, installer, and location, we describe a method to predict the current status of the pump from a set of three possibilities (functioning, functioning needs repair, and not functioning). We also optimize the route that a maintenance crew could take to repair damaged pumps in a version of the well–known, NP–hard, traveling salesman problem. We find acceptable solutions using stochastic metaheuristics. Code and more information are available at http://ghaslam.github.io/AM207/.

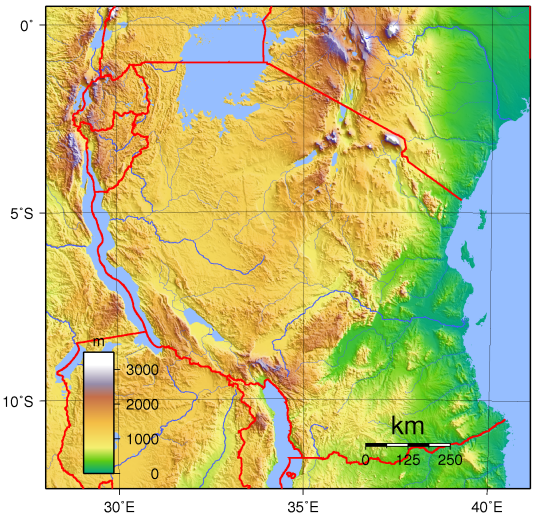
1 Introduction

Access to clean water is a fundamental need for people all over the world. Around 11% of the world's population currently lack such access due to either lack of natural resources or lack of infrastructure [**?**]. In Tanzania, despite the availability of water, only 33.5% have access to a piped source [**?**]. Many others depend on pumps for their supply. When these pumps fail, that can create serious problems for the people who depend on them. Previous work on predicting the failure of mechanical devices such as pumps has benefitted from the use of both statistical analysis as well as data obtained from direct observation of the machines, for example diagnostic vibration sensors [**?**]. As the popularity of machine learning and 'big data' continues to grow, researchers are also studying how failure can be predicted based on training data from large historical datasets, for example for rod pumps in the oil industry [**?**]. Liu's research used subject matter experts to first identify whether a pump was operating normally or about to fail and then trained their model to recognise the features in the data such as daily usage, that predicted the label.

We aim to use Bayesian Methods to assign one of three categorical labels indicating the functional status each pump (functioning, functioning needs repair, and not functioning). Then, we treat the repair of the damaged pumps as a constrained optimisation problem over the space of all possible routes that the maintenance crew could take to reach the damaged wells. We ignore the existing road infrastructure, terrain, height etc. and assume that all locations are equally accessible.

2 The Data

The data for this project was provided as part of a data science challenge run by DrivenData.org [**?**]. DrivenData aims to encourage data scientists from around the world to take part in online competitions to develop predictive statistical models that can address important problems in the fields of health, international development, and the environment. The data for the Tanzanian water pump competition was aggregated from the Tanzanian Ministry of Water by Taarifa, an open source platform for tracking infrastructure related issues [**?**]. The dataset contains records of 59,400 water



(a) Physical geography of Tanzania (courtesy Wikimedia).

(b) Distribution of water pumps.

Figure 1: Geography of Tanzania and distribution of water pumps.

pumps with information on a range of 39 parameters, such as pump construction year, installer, type of pump, water quality, population around the well, cost of water, quantity and others. These include a mix of categorical and numerical data, and there are also many missing values. Each pump is assigned a unique ID as well as precise coordinates for its locations. A subset of the parameters are shown in Table 1. The final column in Table 1 indicates the current status of the pump which the model will attempt to predict. The optimisation algorithm will then attempt to find the route that reaches the damaged pumps in the shortest distance.

Table 1: The DrivenData Tanzania Dataset

ID	Installer	GPS Height	Longitude	Latitude	Construction_Year	Source_Type	Quantity_Type	Status_Group
69572	Roman	1390	34.93	-9.86	1983	Spring	Enough	Functioning
8776	Grumeti	1399	34.69	-2.15	2002	Rainwater	Insufficient	Functioning Needs Repair
34310	World Vision	686	37.46	-3.82	1967	Dam	Enough	Functioning Needs Repair
67743	UNICEF	263	38.49	-11.15	1997	Machine	Dry	Non-functioning
19728	Artisan	0	31.13	-1.83	2007	Rainwater	Seasonal	Functioning

In addition to the parameters given in the original dataset, we calculate some additional parameters such as age (based on the construction year), and relative remoteness (based on the average distance of the 5 nearest neighbours). Figure **??** shows a map of the physical geography of Tanzania and Figure **??** showing the locations of the pumps. The high density of pumps appears as a solid blue colour but actually represents many individual points.

3 Modeling water pump functionality

3.1 Determining a baseline

To give a measure of the effectiveness of our model, we first determine a naive baseline based on simply predicting that any given pump follows the most popular outcome. The distribution of the pumps functional status is shown in Figure **??**. Excluding records which have no data for latitude and longitude, most pumps in the dataset are in the functional category (31389). The number of pumps which are either non-functional or need repair is 22268 + 3931 = 26199. The accuracy of the baseline is thus:

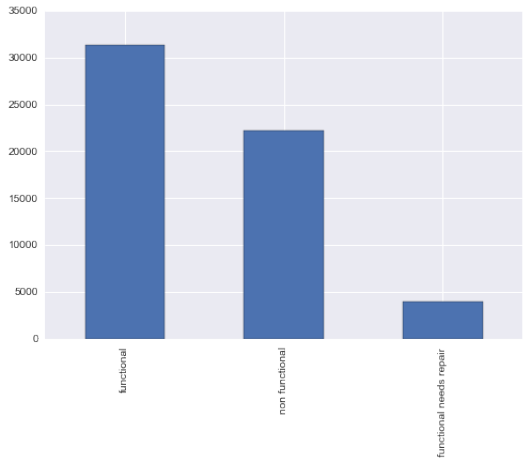


Figure 2: The distribution of pump functional status.

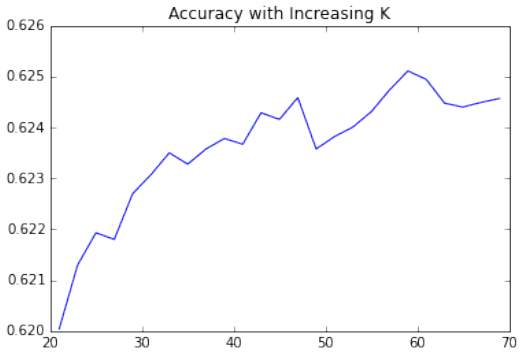


Figure 3: The effect of increasing number of nearest neighbors on pump functional status prediction accuracy.

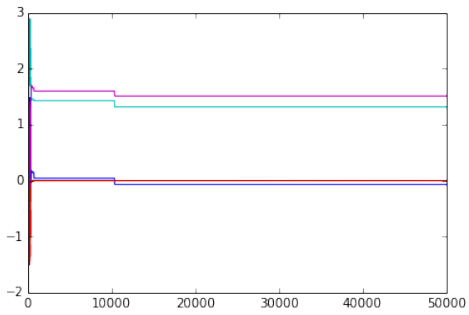
$$ACC = \left(\frac{TP + TN}{P + N}\right)$$
$$ACC = \left(\frac{31389}{31389 + 26199}\right) = 0.545$$

3.2 Machine learning methods for prediction

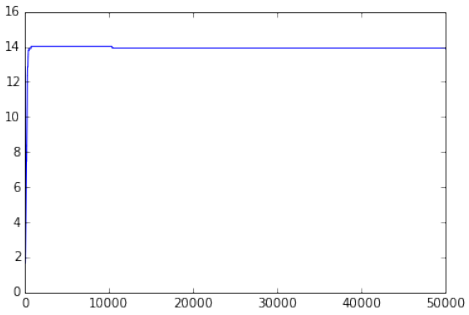
There exist a wide variety of methods for classifying categorical data such as linear and logistic regression, random forests, and K-nearest neighbours (k-NN), and an overview of their details is given by Murphy [**?**]. Python contains many libraries which can quickly and easily implement these methods and we use them for initial exploration of the data, for example identifying which parameters are likely to have most predictive power. An example of using k-NN to label unknown pumps based on the status of an increasing number of nearest neighbours is shown in Figure **??**. Above 21 nearest neighbors, the model is able to outperform the naive baseline by at least 8%.

3.3 Constructing our model

To try to improve our predictions, we propose to model the data as a logistic regression (logit), but with the values of the parameters chosen probabilistically by Monte Carlo Methods. The logit is often used to predict binary variables such as pass or fail. In our dataset, we have three possible



(a) Diagnostic plot for beta sampling



(b) Diagnostic plot for sigma sampling.

Figure 4: Trace plots for parameter sampling.

outcomes, so we choose a modified version of the function, known as the ordered logit, which is able to assign to two or more categories. The general form of our model is thus:

$$y_i = \frac{1}{1 + e^{\theta_i}} + \sigma\epsilon_i$$
$$\theta_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + ... + \beta_{n+1} x_{i,n}$$

where y_i is our predicted pump status and values for β are chosen by sampling from the posterior distribution:

$$p(Y, \Theta) = p(Y|\Theta)p((\Theta)$$

The sampling is conducted using a Metropolis-Hastings sampler. Given that many of our parameters are categorical variables with many levels, then to avoid overfitting and to reduce computation time, we choose an initial model that includes parameters identified in the data exploration stage to have significant impact on the outcome.

We calculated the Geweke statistic, which is used to indicate convergence. A value between -2 and 2 indicates convergence and for our draws the parameters fell within this range. We can also examine convergence by looking at the trace plots for the samples. As we see in Figure **??** these appear stationary.

We could also show the Geweke plots and any other diagnostics.

3.4 Other samplers? Or Other comments?

4 Optimizing maintenance crew route

In the second part of this project, we aim to suggest an optimised route by which a maintenance crew could visit the damaged pumps. The choice of locations can either be taken from the originally labeled pumps or based on the outputs of our model's labels.

4.1 The traveling salesman problem

Having hopefully identified a method for accurately predicting which pumps will fail, we can imagine that the Tanzanian Ministry of Water will want to task a maintenance crew with visiting each of these locations. We may also assume that the maintenance crew are based in the commercial capital of Dar Es Salaam. The crew wishes to visit all N locations in as efficient manner as possible. This problem is equivalent to the traveling salesman problem (TSP), first described in 1932 by Karl

Figure 5: Visiting all conflicts without reloading.

Menger [**?**]. A compact mathematical formulation of the problem is given by Miller et. al. [**?**] as follows:

$$\text{Find variables } X_{ij} \text{ and } U_i i, j = 1, 2, \cdots, n \text{ that minimize : } Z = \min \sum_{i=0}^{n} \sum_{j \neq i, j=0}^{n} C_{ij} X_{ij}$$

subject to

$$\begin{aligned}
&\sum_{i=1}^{n} X_{ij} = 1 && j = 1, 2, \cdots, n \\
&\sum_{j=1}^{n} X_{ij} = 1 && i = 1, 2, \cdots, n \\
&U_i - U_j + n X_{ij} \leq n - 1 && i, j = 2, \cdots, n i \neq j \\
&x_{ij} \in \{0, 1\} && i, j = 0, \cdots, n
\end{aligned}$$

As is clear from the constraints, this is an integer linear program (ILP) where:

- x_{ij} is a binary decision variable indicating whether we go from location i to location j.
- c_{ij} is the distance[1] between location i and location j.
- The objective function is the sum of the distances for routes that we decide to take.
- The final constraint ensures that all locations are visited once and only once.

The problem, of course, is that brute force solution of the TSP is $\mathcal{O}(n!)$. Traditional, deterministic algorithm approaches such as branch-and-bound or branch-and-cut are still impractical for larger numbers of nodes. In many cases, exhaustive search for global optimality is not even particularly helpful as long as the solution found is good enough. We will use simulated annealing (SA) to get acceptable solutions to the TSP.

Figure **??** shows a sample draw of conflict data (the blue points), and a near-optimal TSP route found through 50,000 iterations of simulated annealing.

4.2 Packing the aid truck — the Knapsack Problem

We extend the TSP into a multi-objective optimization problem where *the contents of the aid trucks* also have an optimization component. Therein lies the knapsack problem: subject to a volume or weight constraint, and given that different locations might have very different needs such as food, vaccinations, or emergent medical supplies, *which supplies do we pack on the trucks*?

Here's the unbounded[2] version of the knapsack problem:

[1] In our application, we deal with geospatial data on a large enough scale that the Euclidean distance is actually very imprecise. In order to model distances over the planet's surface, we use the Haversine formula.

[2] Often, this problem is formulated such that you can only bring one of each item, but that does not make sense in our application. Rather, we want to be able to bring as many types of each type of aid as we think necessary, and we'll assume that as many as desired are available to load on the trucks before starting out from HQ.

$$
\begin{aligned}
&\max \sum_{i=1}^{n} v_i x_i \\
&\text{s.t.} \\
&\quad x_i \in \mathbb{Z} \\
&\quad x_i \geq 0 \\
&\quad \sum_{i=1}^{n} w_i x_i \leq W
\end{aligned}
$$

In this formulation:

- x_i is a zero or positive integer decision variable indicating how many units of item i we load on the truck.
- v_i is the utility we get from bringing along item i.
- w_i is the weight of item i.
- W is the maximum weight the truck can carry.

4.3 A brief detour for modeling assumptions

Before we can optimize this aid delivery mechanism, we will need to decide a way to model humanitarian aid needs at a given conflict.

Let us assume that there are K distinct types of humanitarian aid to be delivered. (Without loss of generality, we will use three categories for all of our examples — perhaps we can think of them food aid, first aid supplies, and medicines for concreteness.) We can model each conflict's aid needs as

$$\boldsymbol{x} \sim \text{Dir}(\boldsymbol{\alpha})$$

where $\boldsymbol{\alpha}$ parameterizes the distribution to generate vectors of length K representing the relative proportions of needs.[**?**] For example, in our three category example we might draw the vector $(0.11,\ 0.66,\ 0.23)$ for a certain conflict, meaning that 11% of the aid needed at this conflict is food aid, 66% is first aid supplies, and 23% is medicines. Now that we know the proportions for the given conflict, how might we turn this unitless vector into absolute amounts?

For that reason, let's assign each conflict a scaled size $s \in [1, 10]$ based on the number of casualties (a proxy for the severity of the conflict). We can use this size scalar to turn our proportion vector into a vector of absolute needs.

It should be noted that **both of these modeling methods for proportions and size are "plug-and-play"** — because of purposely designed loose coupling in our model, these methods could trivially be replaced by a different method of calculating or predicting the needs of each conflict. For example, if an independent model was used to calculate each of K needs based on the features of each conflict, those quantities could easily be plugged in to this model. Ultimately, the only quantities that our TSP/Knapsack model needs is an $n \times K$ matrix of aid needs for n cities and K categories of aid.

4.4 A new objective function to integrate TSP and Knapsack

For the vanilla TSP, we simply try to minimize the total distance. Now that we are adding a new objective, we will need to integrate the two into a coherent **loss function**. Here is the function we will actually try to minimize in the combined TSP/Knapsack:

$$L(\boldsymbol{x}) = \text{total distance} + \text{sum of squared aid shortfalls}$$

The effect of squaring aid shortfalls acts as a weight, causing greater importance to be placed on minimizing this aspect of the problem first. Proposals wherein aid shortfalls occur are heavily penalized. As we will see in later graphs, once the SA algorithm is able to avoid all shortfalls and the concurrent massive loss function penalties, a much slower descent begins to take place wherein the distance is slowly optimized. See figure **??** for a depiction of this phenomenon.

4.5 Implementing the Knapsack aspect

Figure **??** shows the same draw of cities as in figure **??**, this time factoring in limited carrying capacity for aid supplies on the aid delivery mechanism and using our new loss function. As we can see, the huge penalty incurred when supplies run out quickly induces the simulated annealing algorithm to converge on a solution with multiple stops at HQ to reload.

Figure 6: Routing with reloading from capital city Kampala.

Figure 7: Loss function acceptances over 100,000 iterations.

Figure **??** uses some uniformly distributed points on the $[0, 50]$ plane to demonstrate how the proposed TSP/Knapsack routes converge as the number of iterations increases.

(a) After 5,000 iterations. **(b)** After 20,000 iterations.

(c) After 100,000 iterations.

Figure 8: Example routing for the TSP/Knapsack hybrid using uniformly distributed points.

4.6 Finding the optimal site for the resupply location

Our initial assumption was that the HQ was located in the capital city of Kampala. However, we should ask whether our HQ could be more conveniently located. We can answer this question by treating the reload location as another parameter and continuing to sample HQ locations using SA. Figure **??** shows the TSP/Knapsack optimized once again, this time using a the optimal HQ location, while **??** compares the loss function as each method converges to its best possible configuration.

Conclusions

In each part of this problem, analytical solutions either do not exist (e.g. in the distribution of events) or are computationally infeasible (e.g. in the TSP/Knapsack optimizations). We found that using a metaheuristic such as SA converged on robust solutions in relatively short order. In the future, we would like to formulate our loss function based on real world data based on refugee locations and aid distribution requirements; our methodology would not change, but the solutions would be more useful for predictive tasks. Additionally, the separate models could be fit and incorporated which realistically model how much of each type of aid is needed at each conflict location. Future research might also include adding many more constraints or twists to the problem, and trying different stochastic optimization techniques such as genetic algorithms, Tabu search, or ant colony optimization.

(a) Routing with reloading from optimal HQ.

(b) Example loss function optimization convergence for $n = 50$.

Figure 9: Optimizing aid delivery routing with reloading.

References

[1] UN Water, "World Water Day Day 2013 - Facts and Figures," *www.unwater.org*, Checked 8th May 2015.

[2] J. Morisset, "Tanzania: Water is life, but access remains a problem," *blogs.worldbank.org*, September 2012.

[3] J. Nakamura, "Predicting time-to-failure of industrial machines with temporal data mining," Master's thesis, University of Washington, Dept. of Computing and Software Systems, Seattle, WA, 2007.

[4] Y. Liu, *Applying A Data Driven Approach To Failure Prediction For Rod Pump Artificial Lift Systems*. PhD thesis, USC, Dept. of Electrical Engineering, Los Angeles, CA, 2013.

[5] *http://www.drivendata.org/about/*, Checked 8th May 2015.

[6] *http://taarifa.org*, Checked 8th May 2015.

[7] "Introducing ACLED-armed conflict location and event data," *Journal of Peace Research*, vol. 47, no. 5, pp. 1–10, 2010.

[8] G. A. and D. Rubin, "Inference from iterative simulation using multiple sequences," *Statistical Science*, vol. 7, p. 457?511, 1992.

[9] K. Menger, "Das botenproblem," *Ergebnisse eines Mathematischen Kolloquiums*, vol. 2, pp. 11–12, 1932.

[10] W. Winston, *Operations Research: Applications and Algorithms*. Thomson Brooks/Cole, 2004.

[11] K. P. Murphy, *Machine Learning: a Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012.