Resource-Efficient Feature Gathering at Test Time

Gavin Gray[*] **Amos Storkey**[†]
Institute for Adaptive and Neural Computation
The University of Edinburgh
Edinburgh, EH8 9AB

Abstract

Data collection is costly. A machine learning model requires input data to produce an output prediction, but that input is often not cost-free to produce accurately. For example, in the social sciences, it may require collecting samples; in signal processing it may involve investing in expensive accurate sensors. The problem of allocating a budget across the collection of different input variables is largely overlooked in machine learning, but is important under real-world constraints. Given that the noise level on each input feature depends on how much resource has been spent gathering it, and given a fixed budget, we ask how to allocate that budget to maximise our expected reward. At the same time, the optimal model parameters will depend on the choice of budget allocation, and so searching the space of possible budgets is costly. Using doubly stochastic gradient methods we propose a solution that allows expressive models and massive datasets, while still providing an interpretable budget allocation for feature gathering at test time.

1 Introduction

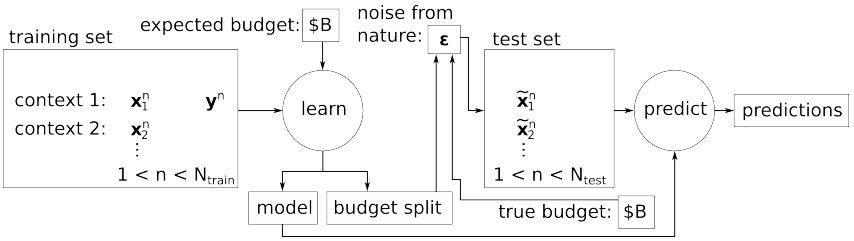


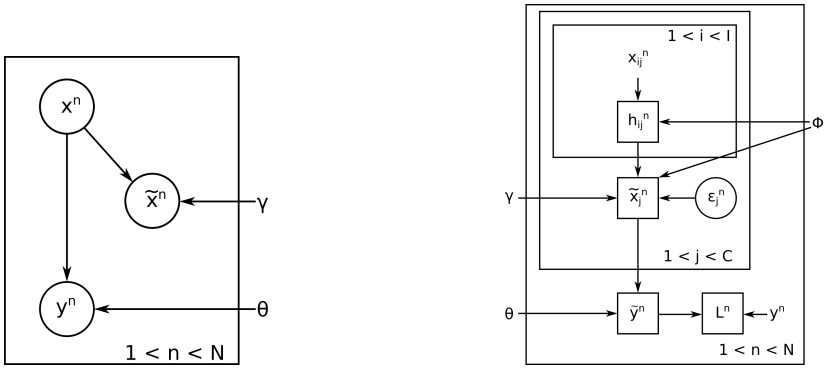
Figure 1: Given a training set and proposed budget, we wish to produce a trained *model*, and at the same time a *budget split*, such that we can then apply this at test time to modulate the noise applied by nature to our features in different *contexts*.

In the real world information costs money. For example, in a sensor network, we may use aggregate measurements from each of several groups of sensors; each group would cover a similar location. The more sensors in the group the more accurate the aggregate measurement. Similarly, in social science we could survey in different locations or different categories of people; the more people surveyed the more accurate the sample for that location/category. In invasive or time-limited medical imaging (e.g. MRI) we may wish to focus the scanning location on one region over another. In attentional models, we may choose to focus attention and hence accuracy in one region over others.

We call each variable that we are aggregating a sample over a *context*. Returning to the sensor network example, given prior measurements from the sensors, such as from a simulation, and a

[*] `https://gra.ygav.in/`
[†] `https://bayeswatch.github.io/`

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.



(a) (b)

Figure 2: (a) A supervised task with noisy inputs $\tilde{\mathbf{x}}^n$, noise-free true inputs $\mathbf{x}^n$ and targets $\mathbf{y}^n$. At training time we observe $\mathbf{x}^n$ and $\mathbf{y}^n$, while at test time we have only $\tilde{\mathbf{x}}^n$. (b) A stochastic computation graph [Schulman et al., 2015] used when incorporating optimisation of statistics as described in section 2.1.

budget to be spent adding sensors to contexts, there is at least one best way to allocate this budget to each context to maximise reward. Allocating the budget reduces the noise added by nature as shown in Figure 1. In general, if the features in a context are statistics [Richman and Mannor, 2016], the variance of the added noise will be inversely proportional to the budget allocated.

By simulating the noise addition by nature at training time, it is possible to obtain an estimate of our performance at test time. However, any change of noise settings would also involve a change in the optimal model. Hence the search over noise settings would involve training the model at each iteration. This inner loop could be extremely time consuming when we wish to use expressive models and massive datasets; we would prefer to train this model rarely.

In this paper, we propose using gradient-based algorithms, including the expected noise on the test set as a component of the model. Using the reparameterization trick [Kingma and Welling, 2013, Bonnet, 1964, Price, 1958, Salimans and Knowles, 2013] to sample these noise variables, we can update the parameters of the budget directly as we learn our model. If the model were linear we could apply the closed form expressions in Richman and Mannor [2016], but the nonlinear models we focus on are not compatible with these.

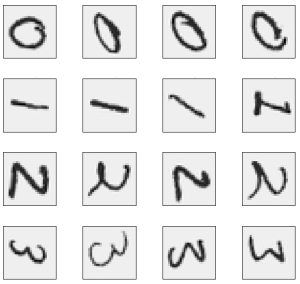
2 Methods

This paper addresses the problem of optimising a model under budget constraints controlling the accuracy of the features that are used to make a prediction. In general, we assume that the features are statistics obtained by *aggregating samples*, so the cost of accuracy is linear in the number of samples that need to be collected to achieve that accuracy. Under IID assumptions, a central limit argument means the variance of each statistic is inversely proportional to the number of samples.

At training time we expect to have a supervised problem defined by a dataset $\mathcal{D} = \{\mathbf{x}^n, \mathbf{y}^n\}_{n=1}^N$. We have a prior that the test set will differ from the training set by the addition of noise. This noise will depend on the budget allocation. After adding noise we have a variable $\tilde{\mathbf{x}}^n$, leading to the belief network shown in Figure 2a.

On this supervised problem we assume a predictive function $f_{\boldsymbol{\theta}}$ and a differentiable noising function $g_{\boldsymbol{\gamma}}$, parameterised by $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$. Each example produces a loss defined by our loss function l. Taking the expectation of the sum of these losses gives our expression to minimise:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \mathbb{E}_{p(\{\tilde{\mathbf{x}}^n | \mathbf{x}^n\}_{n=1}^N)} \left[\sum_{n=1}^{N} l(f_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}^n), \mathbf{y}^n) \right] \tag{1}$$



105 °

Figure 3: Rotational MNIST input (left) and output (right). Each context is a digit class, and at test time the budget decides how many images from each class we receive.

To obtain a cheap Monte Carlo estimator of the gradients of this expression we use a noising function $g_{\boldsymbol{\gamma}}$. According to the parameters $\boldsymbol{\gamma}$, the inputs $\mathbf{x}^n$ are combined with sampled noise $\boldsymbol{\epsilon}$ [Kingma and Welling, 2013].

$$\tilde{\mathbf{x}}^n = g_{\boldsymbol{\gamma}}(\mathbf{x}^n, \boldsymbol{\epsilon}) = \mathbf{x}^n + \boldsymbol{\sigma}_{\boldsymbol{\gamma}} \odot \boldsymbol{\epsilon} \tag{2}$$

For example, in our experiments we use the common reparameterization of a conditional Gaussian with mean $\mathbf{x}$ and variance $\boldsymbol{\sigma}^2_{\gamma}$. In this case $\boldsymbol{\epsilon}$ is a standard Gaussian with mean zero and variance one.

To express the constraint of a budget, the $\boldsymbol{\sigma}_{\gamma}$ variable is dependent on the allocation of a portion of the budget B to each of its dimensions. To split the budget, an unconstrained $\boldsymbol{\gamma}$ variable is passed through a softmax function. The result can then divide down a preset initial variance v_i:

$$\sigma^2_{i,\gamma} = \frac{v_i}{B \times \text{softmax}_i(\boldsymbol{\gamma})} \tag{3}$$

2.1 Simultaneous Statistic Optimisation

To continue with the sensor network example, we may be gathering multiple samples at each location to then combine into a statistic that can be used for the overall prediction problem. In this case, the network can incorporate this process and build the statistic into the architecture prior to adding noise [Edwards and Storkey, 2016]. In Figure 2b, h^n_{ij} represents the hidden activations prior to taking the mean and adding noise to produce $\tilde{x}^n_j$. This process is differentiable and opens the whole network to gradient-based optimisation as shown in our experiments with rotational MNIST in the following section.

3 Experiments

We introduce rotational MNIST as a synthetic problem of inferring the rotation angle of a set of MNIST digits [LeCun et al., 1998]. An input/output example is illustrated in Figure 3. This dataset is used to illustrate the optimisation of the statistic gathering procedure, test time performance, and robustness to required budget variation.

All of the following experiments were run using Theano [Bergstra et al., 2010] and Lasagne [Dieleman et al., 2015], with GPyOpt [The GPyOpt authors, 2016] for Bayesian Optimisation. All Figures were produces using Holoviews [Stevens et al., 2015].

In Figure 4, the performance of a budget found using our method is compared to that of other methods. The histogram illustrates the various attempts by a Bayesian optimisation algorithm [The GPyOpt authors, 2016], while lines illustrate the performance of a uniform budget and a budget based on the mean L2 of weights connected to a context.

The experiments on rotational MNIST illustrated in Figure 4 are performed with an induced sparsity on the MNIST images: according to a probability associated with each context, an image will be

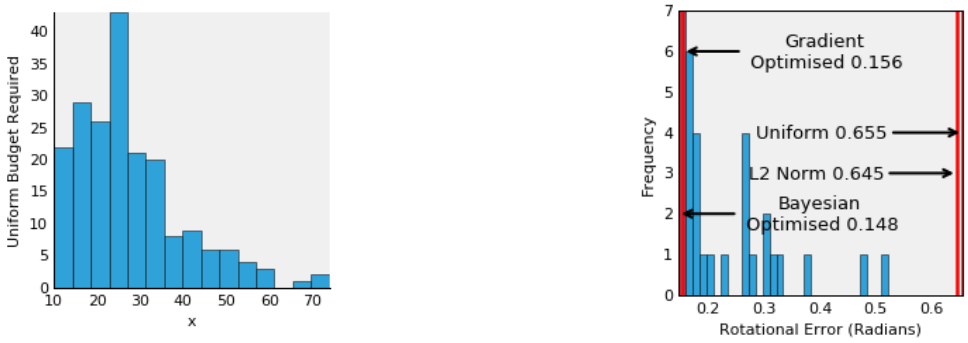


Figure 4: Left: comparing over different sparsity settings and total budget of 10, the distribution of total budget required by a uniform budget to match the performance of a gradient optimised budget. Right: histogram of Bayesian optimised budgets with vertical lines for competing methods to allocate budgets.

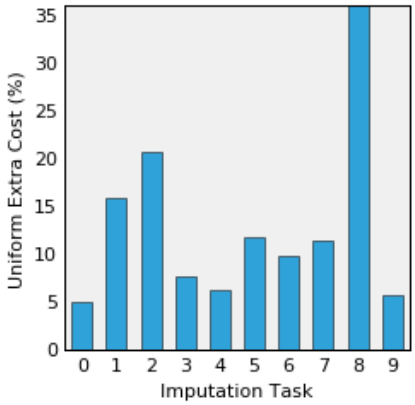


Figure 5: The extra cost of using a uniform budget allocation over one found by gradient optimisation. Each imputation task is the problem of inferring the observations at any buoy. Buoys 2 and 8 are from proximal sensors, and learn budgets preferring data from each other.

randomly zeroed. As these probabilities are varied, the required budget to perform well will also vary. Over this variation in budget, we found that on average a uniform budget would require a 173.3% +/- 12.6% greater budget to obtain the same performance as an optimised budget.

Using data from the Tropical Atmospheric Ocean (TAO) sensor array, which focuses on the El Niño event [Lichman, 2013], we try to infer the observations at each buoy given the observations at all other buoys. We assume that the resources allocated to each buoy will reduce the noise at each location according to equation 3. Due to the small size of the dataset and the noise in the predictions, the relative benefit of using an optimised budget over a uniform budget is much less than on the rotational MNIST problem. The average extra cost of using a uniform budget was 13%, but varied by the imputation task as illustrated in Figure 5.

4 Conclusion

The main conceptual difference in this work is the assumption that the test set will differ from the training set, and we can anticipate how. Once we've defined this prior, we are able to build the probabilistic model and define a loss function. As with most applications of probabilistic modelling, we end up with an expectation that is difficult to compute. Using modern stochastic methods, we have shown that we can deal with this expectation and obtain a method that is efficient and expressive.

References

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

G. Bonnet. Transformations des signaux aléatoires a travers les systemes non linéaires sans mémoire. *Annals of Telecommunications*, 19(9):203–220, 1964.

Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, Diogo Moitinho de Almeida, Brian McFee, Hendrik Weideman, Gábor Takács, Peter de Rivaz, Jon Crall, Gregory Sanders, Kashif Rasul, Cong Liu, Geoffrey French, and Jonas Degrave. Lasagne: First release., August 2015. URL `http://dx.doi.org/10.5281/zenodo.27878`.

H. Edwards and A. Storkey. Towards a Neural Statistician. *ArXiv e-prints*, June 2016.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, December 2013. URL `http://arxiv.org/abs/1312.6114`. arXiv: 1312.6114.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

R. Price. A useful theorem for nonlinear devices having Gaussian inputs. *IRE Trans. on Information Theory*, IT-4:69–72, June 1958.

Oran Richman and Shie Mannor. How to allocate resources for features acquisition? *arXiv preprint arXiv:1607.02763*, 2016.

Tim Salimans and David A. Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 12 2013. doi: 10.1214/13-BA858. URL `http://dx.doi.org/10.1214/13-BA858`.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3528–3536, 2015.

Jean-Luc R Stevens, Philipp Rudiger, and James A Bednar. Holoviews: Building complex visualizations easily for reproducible science. In *SciPy Conference Proceedings*, 2015.

The GPyOpt authors. GPyOpt: A bayesian optimization framework in python. `http://github.com/SheffieldML/GPyOpt`, 2016.