Supporting Feature Engineering in End-User Machine Learning

Louis McCallum
Goldsmiths University of London
London, United Kingdom
l.mccallum@gold.ac.uk

Rebecca Fiebrink
Goldsmiths University of London
London, United Kingdom
r.fiebrink@gold.ac.uk

ABSTRACT

A truly human-centred approach to Machine Learning (ML) must consider how to support people modelling phenomena beyond those receiving the bulk of industry and academic attention, including phenomena relevant only to niche communities and for which large datasets may never exist. While deep feature learning is often viewed as a panacea that obviates the task of feature engineering, it may be insufficient to support users with small datasets, novel data sources, and unusual learning problems. We argue that it is therefore necessary to investigate how to support users who are not ML experts in deriving suitable feature representations for new ML problems. We also report on the results of a preliminary study comparing user-driven and automated feature engineering approaches in a sensor-based gesture recognition task.

CCS CONCEPTS

• **Computing methodologies** → **Feature selection**; • **Human-centered computing** → *Gestural input*; Interface design prototyping;

KEYWORDS

Interactive machine learning, human-centred machine learning, feature engineering.

HCML/CHI'19, May 2019, Glasgow, Scotland

ACM Reference format:
Louis McCallum and Rebecca Fiebrink. 2019. Supporting Feature Engineering in End-User Machine Learning. In *Proceedings of Human-Centered Machine Learning Perspectives Workshop at CHI'19, Glasgow, Scotland, May 2019 (HCML/CHI'19),* 6 pages.
https://doi.org/10.475/123_4

INTRODUCTION

Machine learning (ML) can be a powerful tool even in the hands of people who lack expertise in computer science, mathematics, programming, and other skills normally associated with ML practice. Given appropriate user interfaces and ML workflows, end users are capable of employing ML to perform tasks such as creating animal behaviour recognisers, clustering medical images, filtering nuclear facility surveillance data, and creating new user interfaces for music and accessibility [4].

Yet little research considers how to support end users in choosing and reasoning about the feature representations employed by ML systems. In this paper, we argue that more work is needed to understand how to design user interfaces and ML workflows to support effective end-user ML in applications where the choice of representation is non-trivial, cannot be learned directly from the data, and cannot be determined in advance for the user by an ML expert. We also report results from a user study that suggest even when implementations of potentially relevant features are made available, users may struggle to select good features using a GUI. Further, some light-weight automated methods may select better-performing feature sets in short enough time to accommodate an IML workflow.

INTERACTIVE ML AND FEATURE REPRESENTATIONS

Interactive machine learning (IML) approaches are often useful in enabling end-user ML. Amershi et al. describe IML as characterised by more "focused, frequent, and incremental" iterations of users supplying information (e.g., new training examples) to a learning system, then examining the behaviour of the system [1, p.107]. This "allows users to interactively examine the impact of their actions and adapt subsequent inputs to obtain desired behaviours. As a result of these rapid interaction cycles, even users with little or no machine-learning expertise can steer machine-learning behaviors through low-cost trial and error or focused experimentation with inputs and outputs" [1, p.106].

While IML is sometimes used to support end-user training of systems whose structure (e.g., feature representation, choice of learning algorithm, usage of trained models) has been pre-defined by experts, IML can also be used to support end-user design of entirely new and unique systems. For instance, tools such as Wekinator [5] and GRT [6] place few constraints on the type of data being modelled or the ways model outputs are used; rather, they merely provide ML algorithm implementations and interfaces for users to interactively train supervised learning models. Users must find or implement code to extract features from their chosen data source and send these to the IML tool. Users must

Examples of novel systems designed with IML

- Using light sensors to control a new musical instrument: "MARtLET" by Michelle Nagai (https://vimeo.com/19980514, Figure 1)
- Using computer vision to recognise and augment shadow puppets: "Machine learning shadow play" by Isabella Ong and Marianna Chrapana (https://vimeo.com/253056211, Figure 2)
- Using audio of live musicians to drive visualisations: Analema Group's "KIMA: The Wheel" (https://www.youtube.com/watch?v=yGBPjv2Sgbk)
- Using game controllers to build musical interfaces that can be played by a wide variety of physical motions, including by people with physical disabilities [7]
- Using Kinect to create full-body games controlled by "low power" and "high power" poses: Nightmare Kitty by Perry and Fox [9]

Sidebar 1

Figure 1: The MARtLET instrument, created with IML, uses gestured sensed with photoresistors to control sound.

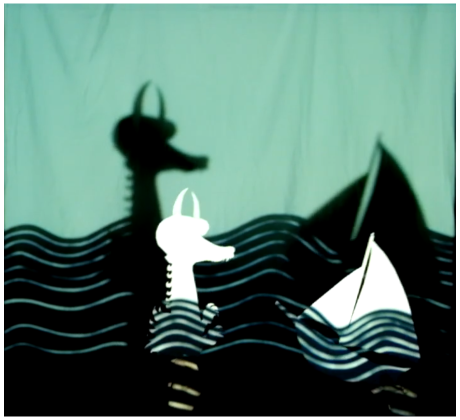


Figure 2: "Machine learning shadow play" uses IML to recognise shadow puppets and augment them with digital content.

also implement logic (potentially in a high-level or domain-specific programming environment) for receiving model outputs and using them in some way (e.g., to influence a game, music, or smart-home device). These tools have enabled users to design systems as diverse as those listed in Sidebar 1. To build each of these systems, a designer iteratively employed IML to build models from their own training examples of sensor features paired with the classification or regression output(s) that they desired in response to those feature values.

Feature Representation Challenges Facing Non-Experts

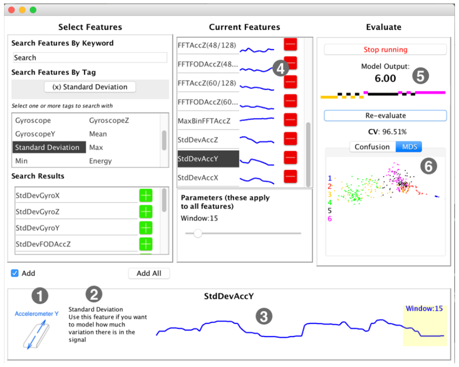
Choosing and reasoning about feature representations can be a monumental challenge for end users applying ML to novel problems such as those in Sidebar 1. In many domains, including audio and image analysis, it may be impossible to create useful models directly from the raw data. (Certain deep neural networks can learn from raw audio and images, though these require many more examples than the dozens to hundreds often supplied by IML users creating datasets from scratch for each learning problem.) In other domains, models may be accurately built from raw data for certain problems but others require feature engineering. For instance, a user working with simple hand-held accelerometers might build an accurate model for sensing tilt from the raw data, but a model for classifying actions based on shaking speed would require additional features to be extracted.

A suitable choice of features can dramatically increase the ability to create an accurate model for a given problem; a poor choice can prevent anything useful from being learned. Moreover, in many end-user ML design applications, users have considerable flexibility in defining the learning problem: for instance, they must decide what gestures should be recognised in a new game, or how a live visualisation should change in response to perceptual properties of musical audio. Understanding what characteristics of the data are captured in a given feature representation should be useful in helping a system creator reason about which definitions of the learning problem will likely be feasible.

People with expertise in domain-specific data and signal processing draw on this expertise to choose or implement appropriate features, and to reason about what will likely be learnable for a given set of features. Yet many people—e.g., many artists and musicians, disabled people, athletes, gamers, educators—who would benefit from the ability to build novel systems with machine learning and sensors or other complex data sources lack this expertise.

Automated Methods

In some applications, automated feature learning has drastically reduced the need for manual feature engineering [13]. However, this typically requires large datasets (thousands of examples or many more); no such datasets are available for many of the niche data analysis tasks of interest to many IML users, and IML users themselves may not be able to easily create new large datasets. Certain IML users may be aided by transfer learning (e.g., [13]) using features learned from existing data sets, but this does not help users designing with sensors or data sources for which such datasets do not exist. Further, transfer learning is only effective when the new problem being learnt will benefit from the features generated by the larger dataset. For example, the representation learned from a neural network trained on the ImageNet dataset can give good results for new everyday object classification tasks (e.g., mugs, shoes), but less so on tasks involving medical or astronomical imaging. Similarly, large benchmarking datasets of everyday gestures may be of limited use in learning features that are relevant for modelling the motions of a dancer in a particular piece, or of a disabled person making a personalised alternative gaming controller. Automated feature selection methods (e.g., wrapper selection [8] or filtering by metrics such as information gain [2]) have been used to search for an appropriate choice of features for problems for which a potentially suitable set of features can be enumerated, though these can be computationally expensive and may be prone to overfitting [10]. Of course, even when automated methods can feasibly derive or select a suitable feature representation, it may be difficult to communicate to non-expert users what information is captured in this representation and for what types of analysis problems this representation is suitable.



Standard Deviation
Use this feature if you want to model how much variation there is in the signal

Figure 3: The feature selection and evaluation interface employed in our study: (1) Diagram illustrating the feature currently highlighted with the mouse; (2) Text description of this feature; (3) Plots the value of this feature in real-time; (4) Plots all features currently selected for use in classification; (5) Plots real-time output of the trained classifier; (6) Shows confusion matrix for current classifier or 2D projection of current training examples.

.

Open Questions: Human-Centred Approaches to End-User Engineering and Understanding of Features

While much research in IML has investigated mechanisms for leveraging domain- and task-specific information elicited from users—and has shown practical benefits to doing so—very little research has investigated mechanisms for eliciting information from users that can directly inform the choice of features for a new problem. The main exceptions appear in domains with features that are understandable to non-expert users (e.g., words chosen as relevant to a text modelling task [12]). Little research has investigated how to help users understand what may be learnable from a given feature representation, or how to give users feedback about the suitability of a particular feature representation for a target problem. Little work has explored how automated or user-driven approaches to feature selection should be used in IML contexts in which other aspects of the problem (e.g., the training data) may be frequently changing, and users rely on rapidly switching between adjusting and re-evaluating the ML system. Nor has research investigated trade-offs between automated and user-driven approaches in terms of system accuracy and efficacy, or user satisfaction or cognitive load.

COMPARING FEATURE SELECTION METHODS IN A GESTURE RECOGNITION TASK

We have conducted a short study investigating how to support amateurs creating gesture recognisers. We asked 17 participants (students and faculty at our university) to pick 5 dynamic gestures that could be performed holding an iPhone, and then had 20 minutes to use a modified version of the Wekinator user interface (Fig. 3) to build a classifier using IML. This interface allowed users to choose from over 200 common motion features (from [11]) computed from the 6 inputs of the 3-axis accelerometer and

Feature selection methods chosen for comparison (computation and comparison performed after the user study)

- *Raw*: Just the raw sensor data.
- *All*: All 202 available features.
- *User*: The final feature set chosen by the participant using the GUI.
- *EarlyInfo*: A small feature set chosen using information gain. All features are ranked by information gain and a limited number of increasingly large sets are chosen. We choose the smallest such feature set whose cross-validation accuracy is better than that of *All*.
- *BestInfo*: Identical to above, except the set with the very best cross-validation is chosen.
- *Wrapper*: A set selected by forwards wrapper selection with 5-fold cross-validation evaluation and a search termination of 10 [8].

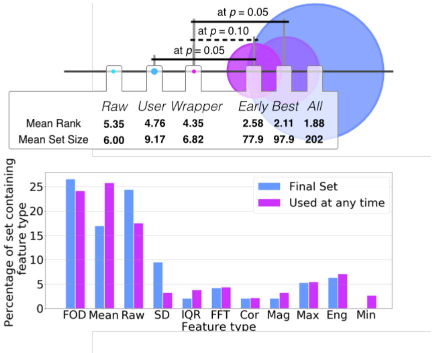


Figure 4: Top: Average within-user accuracy ranking and feature set size for each feature selection method. Bottom: Proportion of sets containing each feature type for each user’s final set and the set of features used at any time.

gyroscope on the phone. Each feature had an accompanying explanatory description and diagram, and participants could view its output plotted in realtime. After 20 minutes, we asked users to record a separate test set for their 5 gesture types and conducted a short interview.

Empirical Comparison of User-Selected and Automated Feature Sets

Following [3], we used a Friedman Test and a Nemenyi post-hoc test to compare, for each user, the the test-set accuracy of the user-selected features with a number of automated feature selection methods (enumerated in the sidebar). Shown in Fig 4, we found that the *User* and the *Raw* sets had significantly worse accuracy ($p = 0.05$) than *EarlyInfo*, *BestInfo* and *All* features, and *Wrapper* had significantly worse accuracy ($p = 0.10$) than *EarlyInfo*, *BestInfo* and *All* features ($p = 0.05$).

Another clear difference between the *User* and *Wrapper* selection and the sets that returned higher test set accuracy is their size. The size of *BestInfo* feature set for each participant is significantly larger than the *User* set ($t(7.5) = 7.97$, $p < 0.05$) and *Wrapper* set ($t(7.5) = 8.21$, $p < 0.05$).

With an average execution time of 16m 32s (using a mid 2015 MacBook Pro, 2.2 GHz Intel Core i7), *Wrapper* selection took far too long to reasonably fit into a standard IML workflow. Calculating *BestInfo* was much faster at 64.4s, and *EarlyInfo* faster again at 48.6s.

Analysis of User-Driven Feature Selection

Empirical test set accuracy rates contrast strongly with participants’ own perception of their success. When asked “How well were you able to complete the task?” 10 out 17 participants reported that they had been able to complete the task to their satisfaction, whilst a further 5 were happy with performance on some gestures but not others. This either demonstrates that these users are poor judges of accuracy, have low thresholds for what they consider good accuracy, or evaluated the quality of the classifier using criteria not closely correlated with accuracy.

In the interview, we asked participants how they decided which features to use. 8 out of 17 reported evaluating new features by selecting them in the GUI, re-training the classifier, then moving the sensor and observing the model output graph. This graph was also rated as the most useful interface item in the exit questionnaire (mean 4.0 of 5).

When asked what motivated their choice of features, 7 participants mentioned choosing features they had knowledge of previously; most often, these were means and first-order differences. Fig 3 shows these were present in 26.8% and 25.1% of all user-chosen feature sets, respectively, in comparison to less than 5% for interquartile range, FFT, minima, standard deviations, magnitudes and correlations. When we expand the analysis to the set of features each participant explored at any point in the study, we see most participants barely experimented outside of their comfort zone (Fig 4, bottom). This demonstrates that users may avoid using features whose function is not immediately clear, at least initially.

Questions Informed by the Study

- How might alternative interfaces for end-user-driven feature selection discourage users from picking feature sets that are too small, inaccurate, and limited to familiar features? (For instance, can a UI encourage users to experiment with larger numbers of features, as well as unfamiliar features, by encouraging the choice of features in subsets rather than one at a time?)
- How might feature selection (automated or user-driven) be better integrated into the iterative IML process (e.g., by applying the faster information gain-based approaches each time new training data is added, and by enabling users to experimentally adjust information gain thresholds)?
- How might information about feature quality be communicated to users (e.g., by showing users information about features' information gain according to the current training set)?
- Can we design mixed-initiative workflows that take advantage of users' task knowledge and reasoning abilities to inform the choice of feature representation, in order to improve accuracy beyond both simplistic approaches to user-driven and automated feature selection?

Sidebar 2

DISCUSSION

We found that, given a simple interface and a short task, people were bad at choosing features, but they did not realise they were bad at it. They also picked small sets with low accuracies and—being initially reticent to explore new features—focused on features they thought they understood. Larger features sets, chosen using automated methods, proved more accurate.

This study has informed our current work exploring the questions in Sidebar 2. Our current work also seeks to empirically investigate our intuition that giving users some control over feature engineering and/or insight into the problem characteristics captured by a given feature set is likely to be helpful in informing IML users' design decisions about what to model and how to adjust the modelling problem and/or training examples in order to improve model performance.

REFERENCES

[1] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. 2014. Power to the People: The Role of Humans in Interactive Machine Learning. (Dec. 2014), 1–16.

[2] R Battiti. 1994. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks* 5, 4 (July 1994), 537–550.

[3] J. Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research* 7, Jan (2006), 1–30.

[4] John J Dudley and Per Ola Kristensson. 2018. A Review of User Interface Design for Interactive Machine Learning. *ACM Transactions on Interactive Intelligent Systems* 8, 2 (July 2018), 1–37.

[5] R. Fiebrink, P.R. Cook, and D. Trueman. 2011. Human model evaluation in interactive supervised learning. In *Proc. CHI'11*. 147–156.

[6] N. Gillian and J.A. Paradiso. 2014. The Gesture Recognition Toolkit. *Journal of Machine Learning Research* 15 (2014), 3483–3487.

[7] S. Katan, M. Grierson, and R. Fiebrink. 2015. Using interactive machine learning to support interface development through workshops with disabled people. In *Proc. CHI'15*. 251–254.

[8] R. Kohavi and G.H. John. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97, 1 (Dec. 1997), 273–324.

[9] P. Perry and S. Katan. 2016. User-Defined Gestural Interactions Through Multi-Modal Feedback. In *Proc. of the TEI 2016 BodySenseUX Workshop on Full-Body and Multisensory Experience.*

[10] J. Reunanen. 2003. Overfitting in Making Comparisons Between Variable Selection Methods. *Journal of Machine Learning Research* (2003).

[11] J.L. Reyes-ortiz, R. Ghio, D. Anguita, X. Parra, and J. Cabestany. 2013. Human activity and motion disorder recognition: Towards smarter interactive cognitive environments. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.*

[12] P. Simard, D. Chickering, and A. Lakshmiratan et al. 2014. ICE: enabling non-experts to build models interactively for large-scale lopsided problems. *arXiv preprint arXiv:1409.4814* (2014).

[13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems.* 3320–3328.