The Challenge of Feature Engineering in Programming for Moving Bodies

LOUIS MCCALLUM

Department of Computing, Goldsmiths University of London

REBECCA FIEBRINK

Creative Computing Institute, University of the Arts London

The design of bespoke human movement analysis and control systems by end users and other people without programming or signal processing expertise presents great opportunities for the arts, accessible interface design, games, and other domains. In this paper, we describe the challenge of feature engineering that confronts many people wishing to build such systems. We have conducted three studies exploring approaches to supporting feature engineering and investigated how such approaches may impact on system accuracy, user experience, and design outcomes. We briefly outline study outcomes that are most relevant to the workshop themes.

CCS CONCEPTS • Computing methodologies → Feature selection; • **Human-centered computing →** *Gestural Input*; Interface design prototyping.

Additional Keywords and Phrases: Interactive machine learning, feature engineering.

ACM Reference Format:

Louis McCallum and Rebecca Fiebrink. 2020. The Challenge of Feature Engineering in End-User Programming for Moving Bodies. 2020.

1 CHALLENGE: BESPOKE MOTION ANALYSIS SYSTEMS MAY REQUIRE FEATURE ENGINEERING

Many people—including artists and musicians, disabled people, designers, gamers, and educators—could benefit from a greater ability to create bespoke human movement analysis and control interfaces with common sensors (e.g., accelerometers and gyroscopes built into smartphones, watches, and fitness trackers) as well as video. Such devices can be used to create new gesturally-controlled digital musical instruments [4][2], bespoke and customisable interfaces for people with disabilities [11], personalized interfaces for rehabilitation [13], and novel games [9], among other applications. Yet enabling a truly broad range of people to create and configure motion analysis and control systems requires design approaches and tools that are accessible to people without deep expertise in programming and signal processing.

Demonstration-based approaches, including interactive machine learning (IML), can enable non-programmers to build and customize gesture recognisers and movement control systems by iteratively providing examples of movements that are then used to train statistical models. IML software tools have been developed for creating new interactions with sensors and/or video, finding real-world use in music [2][5], games [1], accessible interface design [11], and elsewhere. However, recognition of all but the simplest of human movements using motion sensors also requires applying extensive processing to raw sensor data to produce a data representation ("features") that can be used by machine learning (or a human programmer) to create an accurate model. In the case of video, off-the-shelf tools such as PoseNet [7] can compute locations of limbs and joints from raw pixels, yet further processing of such representations is often still necessary for systems that analyse or recognise actions or temporal qualities of movement rather than static poses. Modern feature learning techniques can, in principle, automatically learn good representations from raw data [8], but these typically require very large datasets (e.g., tens of thousands of examples or more), making these inappropriate for most end-user designers who must generate or curate examples themselves. Likewise, transfer learning techniques [12] may be used to leverage representations learned from large datasets containing examples of human movements that are related but not identical to the target movements, and captured with a similar sensor, but such datasets often do not exist for many potential applications described above.

Experts in signal processing, sensing, and programming who build analysis and control systems from such data therefore must typically use analytical and experimental approaches to derive good features for a given task. This feature engineering process may involve applying smoothing filters, frequency domain analysis, first- and second-order derivatives, or quaternion estimation, amongst other techniques [10]. Experts may draw on knowledge about these features (e.g., which properties of motion they capture, which features have been used in similar analysis tasks) as well as experimentally compare candidate feature sets (e.g., by training statistical models on alternative representations and comparing their performance on test data).

We are unaware of prior work exploring how to support feature engineering for human movement analysis or control by end users or others without deep technical expertise. Existing systems for end-user design of gesture recognisers typically "bake in" features chosen by developers (e.g., [6]), limiting the movements that can be recognized, or require users to configure feature extraction and processing outside the tool (e.g., [4][2]), limiting the ability of people without programming/signal processing expertise to create truly novel systems).

2 EXPLORING NEW TECHINQUES TO SUPPORT END-USER FEATURE ENGINEERING IN IML DESIGN OF BESPOKE MOVEMENT ANALYSIS AND CONTROL SYSTEMS

We have recently conducted studies exploring how computational and interaction techniques might better support feature engineering by people designing bespoke motion analysis or control systems with motion sensors, who lack signal processing or sensor analysis expertise. We investigated how new techniques may impact on system accuracy as well as user experience and design processes. Our work has focused on feature engineering in the context of design using IML: not only does IML enable design by non-programmers, but research has also shown that—when end-user designers are working with suitable feature representations—IML brings a number of benefits to the design process, compared to designing using programming. These include encouraging rapid prototyping and exploration of alternative designs, and enabling users to naturally build systems that recognise embodied expertise—due to embodied engagement in the design process through the live demonstration of training examples and, often, real-time evaluation of trained systems through embodied experimentation. We are interested in how to support feature engineering in such contexts, in which designers are already interacting bodily with design tools (and therefore might take advantage of interactively demonstrating data to inform automated techniques, and/or interactive data visualisations to learn about new features), and in which design processes are often characterised by fast iteration and exploration (e.g., including rapid cycles among providing training examples, training models, evaluating model behaviours, and changing training examples in order to attempt to correct model mistakes and/or reflect changing design goals).

We conducted three studies with people who had experience with IML using Wekinator [4], and who were studying or teaching creative computing, music computing, or digital arts computing. Some had worked with sensors before but few had much signal processing knowledge or feature engineering experience. In a first lab study, participants had 20 minutes to build an accurate gesture recognizer for a hand-held iPhone (using its accelerometer and gyroscope sensors), for a set of 4 bespoke dynamic gestures chosen by the participants at the beginning of the study. We empirically compared classification accuracy and computation time of several automated feature selection methods with features selected by participants using a new feature engineering interface added to Wekinator. This interface provided implementations of 202 features previously identified by experts as relevant to a wide variety of human motion analysis tasks [10]. Its GUI enabled users to browse, visualize, and select/deselect features as well as read high-level descriptions of each. We also used a poststudy questionnaire and semi-structured interviews to investigate participants' experience with the interface and to learn how they used it to reason about features, and augmented this with log data analysis to discover how they employed feature engineering within the larger IML workflow. In a second, two-hour study, participants used an updated feature engineering GUI (informed by the first study outcomes, and shown in the accompanying video) and a hand-held, six-degrees-of-freedom inertial measurement unit (IMU) to build a classifier for four bespoke dynamic gestures, which participants were allowed to change as they wished. Participants used these movements to control a pre-built drum machine, animation, or game. Using empirical analysis and participants' subjective evaluations, we compared the quality of classifiers built from user-engineered features with those built by automated methods, and we used survey and log data to investigate how people used different capabilities of the software interface (e.g., for feature and data visualization, automatic feature ranking and thresholding) as well as what they learned about feature engineering through using the interface. In a third study, two undergraduates spent six weeks building gesturally controlled musical instruments with the IMUs. Weekly semi-structured interviews and analysis of student logs and completed projects enabled us to discover how the students used and thought about feature engineering within the larger design process, how this changed over time, and whether/how the ability to use more complex features influenced what they ultimately built.

2.1 Highlighted Outcomes

We provide below an overview of the most important study outcomes, which we hope will inform thinking in the workshop and beyond about how end-user feature engineering can be supported, relevant to the workshop themes of "What new tools and methods would be needed for programming with and for movements?" and "How to work with bodily data as a design material when programming with and for bodies?" Specifically:

- **Feature engineering was necessary** to realise most participants' designs. Raw sensor data alone usually lead to worse empirical accuracy and lower subjective ratings than user- or automatically-selected features.
- **Leveraging a large number of features identified by domain experts as relevant to human motion sensing facilitated a number of useful approaches to support participants' designs.** These include (i) enabling the use of all such "expert" features (which performed surprisingly well for a number of tasks); (ii) implementing lightweight automated methods based on information gain for selecting from these features (this also often performed well, while reducing computational load compared to using all features); and (iii) enabling a number of interactive approaches to supporting human feature selection and reasoning, e.g., enabling users to add features that met certain criteria, and enabling users to visualise all features ranked by information gain and to select features above an experimentally tuneable threshold on this ranking.

- **It was still often challenging for participants to interactively select appropriate features** using our interfaces**,** and participants' final selected features were often outperformed on both empirical and subjective measures by both our lightweight automated methods and/or by using all available features. **Participants sometimes reacted to the difficulty of choosing good features for a given task by changing the task** (i.e., choosing different gestures). We observed that participants used knowledge they had about data representations—whether raw sensor data or other features they understood well—to choose target gestures that were easily learnable using those features, suggesting that, in practice, **teaching users about features may be important** in facilitating designs capable of recognizing a richer array of motions.
- **Many participants appeared to have difficulty in effectively evaluating and comparing alternative feature sets**: there were many inconsistencies in their stated preferences for different feature sets, and often a large mismatch between their stated preferences and empirical classifier accuracies on user-provided test data. We therefore hypothesise that **it may be helpful to explore how interfaces can better scaffold structured empirical experimentation** with candidate features; simply supporting non-experts by providing implementations, explanations, and GUIs for interactively exploring and selecting features (as we did) may be inadequate. There are open questions about what such experimentation should look like in movement analysis applications in which conventional experimental approaches (e.g., comparing accuracy on held-out data) and embodied and subjective approaches to evaluation (e.g., moving in different ways and observing how a classifier responds) may both be relevant, possibly providing contradictory results [3].

ACKNOWLEDGMENTS

This research was supported by the UK Engineering and Physical Sciences Research Council grant "Supporting Feature Engineering for End-User Design of Gestural Interactions" (grant number EP/P024327/1).

REFERENCES

[1] C. G. Diaz, P. Perry, and R. Fiebrink. 2019. Interactive machine learning for more expressive game interactions. IEEE Conf. on Games.

[2] R. Fiebrink. 2011. Real-time human interaction with supervised learning algorithms for music composition and performance. PhD thesis, Princeton University.

[3] R. Fiebrink, P. R. Cook, and D. Truman. 2011. Human model evaluation in interactive supervised learning. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 147–156.

[4] Fiebrink, R., D. Trueman, and P. R. Cook. 2009. A meta-instrument for interactive, on-the-fly machine learning. In Proceedings of New Interfaces for Musical Expression.

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

[6] B. Hartmann, L. Abdulla, M. Mittal, and S. R. Klemmer. 2007. Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 145-154.

[7] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In IEEE Conference on Computer Vision.

[8] H. Lee, R. Grosse, R. Ranganath, and A. Y Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the International Conference on Machine Learning. 609-616.

[9] P. Perry and S. Katan. 2016. User-defined gestural interactions through multi-modal feedback. In Proceedings of the TEI 2016 BodySenseUX Workshop on Full-Body and Multisensory Experience.

[10] 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 Symp. on Artificial Neural Networks, Comp. Intelligence and Machine Learning.

[11] S. Parke-Wolfe, H. Scurto, and R. Fiebrink. 2019. Sound Control: Supporting custom musical interface design for children with disabilities. In Proceedings of the International Conference on New Interfaces for Musical Expression.

[12] 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.

[13] H. Zhou and H. Hu. 2008. Human motion tracking for rehabilitation—A survey. *Biomedical signal processing and control* 3(1): 1–18.