Exploring the role of Forward and Inverse models in HCI

Roderick Murray-Smith
University of Glasgow
Glasgow, Scotland
Roderick.Murray-Smith@glasgow.ac.uk

John Williamson
University of Glasgow
Glasgow, Scotland
JohnH.Williamson@glasgow.ac.uk

ABSTRACT

We outline the role of forward and inverse modelling approaches in the design of systems to support human–computer interaction. Causal, forward models tend to be easier to specify and simulate, but the inverse problem is what typically needs to be solved in an HCI context.

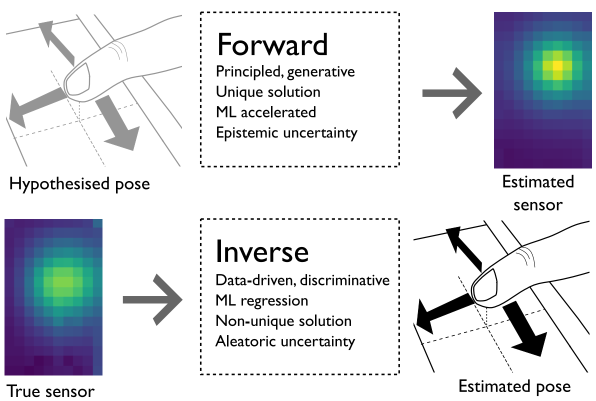

Figure 1: Forward and inverse models in a finger sensing task

INTRODUCTION

Interactive systems must be able to sense and interpret human actions to infer their intentions. HCI research continually explores novel sensors for novel forms of interaction, but lacks a coherent, consistent framework for characterising this process with incrementally improving precision for different sensors and different human behaviours. We argue that for the field to make consistent incremental progress, we need a more general, formal framework for characterisation of the pathway from human intent to sensor state. This pathway can include formal, computational models of human elements such as cognition and physiological processes, as well as purely technical elements such as the characterisation of the physical processes of the sensor, and eventually the forward model could include the anticipated impact of the feedback from the interface on the forward process.

Forward models and inverse problems in HCI

Scientific theories let us make predictions. If we have a complete mathematical model of a physical system we can predict the outcome of measurements of states of that system. The *forward problem* is this problem of predicting measurements. Solution of the *inverse problem* uses measurements to infer hidden parameters that characterise the system or its inputs [18]. In many cases, scientists can simulate a system better than they can practically observe it.

Computational Modelling workshop, SIGCHI2019, May 2019, Glasgow, Scotland.
.

The forward problem tends to have a unique solution, but the inverse problem does not.[1] For example, take something as simple as a nonlinear saturation effect in the forward model 2. We can predict the forward values with precision, but there are infinitely many possible solutions to the inverse problem in the saturated areas. This means that solution of inverse problems requires explicit use of *a priori* information about the system, and careful consideration of uncertainty in the data.

[1] If our model parameters, structures or inputs are uncertain, then we have a distribution of solutions, but for each sample of the uncertain variables we can generate a deterministic prediction.

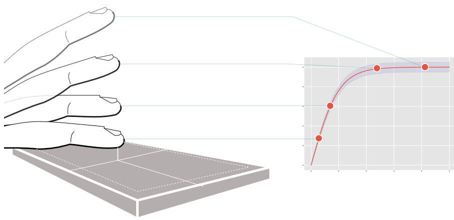

Figure 2: A saturation effect is one potential cause of ill-posedness. Distinguished world states (finger poses) are projected to identical sensor states.

We propose a *dual approach* where we model both the forward and the inverse problems, and fuse them consistently via a probabilistic framework. The core ideas is that there are two types of uncertainty that pervades interaction: **epistemic** uncertainty where a lack of knowledge means we unsure as to whether a model of the user, sensor or world is valid, and **aleatoric** uncertainty where measurements are noisy and subject to random variation [14]. First-principles models are an attractive way of approaching HCI problems but it is hard for simple, elegant models to represent messy human behaviour. Data driven approaches have the advantage that they are precisely tailored to specific real interactions, but struggle to generalise robustly, so we consider a construction which divides up the problem of modelling interaction into two streams, **forward** and **inverse**.[2]

[2] **Forward**, first-principles models of interaction derived from physics, physiology and psychology, where parameters are not fully known, with *epistemic* uncertainty and unique solutions, e.g. implementing an executable simulation of what sensor vector we would observe for a given finger pose.
Inverse, data-driven models of interaction, learned from observed interaction with machine learning, with *aleatoric* uncertainty and non-unique solutions, e.g. training a regression model to predict pose from a touch sensor vector.

This splits the modelling task into two parts with complementary strengths, bringing generalisable, testable simulation models into the forward construction and powerful machine learned predictive models into the inverse construction. We show that this is both an intellectually appealing way of partitioning the interaction problem, and concretely demonstrate that probabilistic fusion of forward and inverse models leads to performance which exceeds either approach alone.

BACKGROUND

Forward/inverse models in HCI

How does this relate to the study of Human–Computer Interaction? Forward models of human or sensor behaviour tend to be more straightforward to specify theoretically and to acquire empirical data from their associated behaviour, and as described above, they more frequently have a unique mapping. Interface designers, however, if they want the computer to be able to respond to the human behaviour, need inverse models which can let them determine the intention behind the sensed observations.

Forward and inverse models can address different subproblems in interaction; some problems are easy to model and simulate underpinned with strong scientific knowledge, and some are much more easily formulated as a black-box learning problems that can be fed to machine learning algorithms. Combining these approaches allows us to fall back when one approach fails. For example a sensor might exceeds theoretically expected bound and not be well predicted by a forward model; or an action may be sensed in a context that lies outside the space of training data used to implement a classifier and be poorly inverted by a black box inverse model.

These inverse problems often have non-unique solutions, and are based on incomplete and noisy observations. E.g., we can specify models of human physiology relating to movements of the arm and

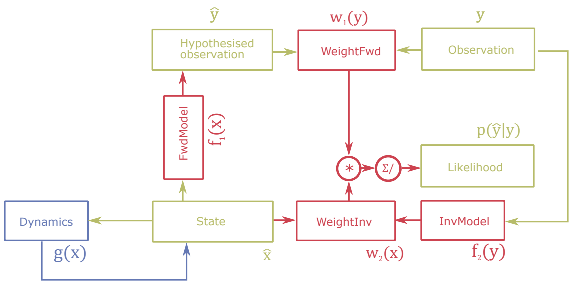

Figure 3: A particle filter architecture which uses a forward model to propose hypothesized hidden states, and reweights using a machine-learned inverse model.

(1) Bayesian analysis, integrating over models of conditional distributions.
(2) The creation of a complex **nonlinear, multivariable regression mapping**. This is an *inverse model* from a possibly high-dimensional sensor-space X to the original $(x, y, z, \theta, \phi, \psi)$ vector.
(3) The creation of a **causal forward model** from (x, y, z, θ, ψ) to image space X, which can then be differentiated to find the values of (x, y, z, θ, ψ) which minimise the difference between the observed sensor readings X and the inferred readings $\hat{X}$.

hand, and gradually improve their fidelity via theoretical insight, a range of experimental techniques and high-fidelity sensors in carefully controlled lab settings. Advanced sensors like those used in Brain Computer Interfaces explicitly include forward models of the skull to infer the brain state from EEG sensors, but the same principle can be applied in all interactive systems. The challenge for the interface designer is, however, to infer from some possibly cheap and low fidelity sensor information which human intentions were associated with movements of their body, which led to the series of sensor readings.

Similarly, on the engineering side, for a user-interface sensor input system we can often predict the behaviour of the forward model of the system from knowledge of the technical specifications of the series of components used. However, it is difficult to predict whether these will be sensitive enough to be good enough for the interaction requirements, without either building the system and testing it, or by finding inverse models which let us predict the expected accuracy of the resulting system, for typical inputs. The ability to create such a model computationally, in advance of construction would be a useful tool in the design stage of a project.

Inverse problems in human behaviour. Collaborative efforts to generate executable computational models of human behaviour include [7, 10, 13]. Applications of such models in HCI include [2, 3, 15]. [11] introduced the use of computationally efficient Approximate Bayesian Computation (ABC) approaches to solve inverse problems in HCI. Their approach did not specify an explicit forward model *a priori*, but rather learned a forward model for human cognitive processes which would optimise a plausible cost function.

EXAMPLE: 3D TOUCH INFERENCE TASK

As an example, imagine designing an interaction solution for a novel capacitive screen sensor. Progress in design of capacitive screen technology has led to the ability to sense the user's fingers up to several centimetres above the screen. The ability to sense finger position and pose accurately a distance from the device screen allows designers to create novel interaction styles, and researchers to better track, analyse and understand human touch behaviour. However, the inference of position and pose is a classic example of an ill-posed inverse problem, given only the readings from the two-dimensional capacitive sensor pads, making the solutions inherently uncertain.

To infer finger pose and position away from the touch surface we need a) a sensing technology which can detect the human hand at a distance from the screen and b) an inference mechanism which can estimate the pose and position given the raw sensor readings. The sensor technologies involved will rarely provide a simple reading which will return the position (x, y, z) and pose (pitch, roll and yaw – θ, ϕ, ψ). Inferring these values can be done with three general approaches.

FORWARD & BACKWARD COMBINATIONS

One approach to combine the forward ('causal') and inverse (or 'regression') modelling approaches is via the use of the regression model to create an initial condition for the optimisation algorithm or a particle filter approaches to refine. Particle filters use forward and inverse models, and have been used in a range of computer vision [5] and HCI applications [4, 16]. A particle filter can perform *stateful* tracking; i.e. track the inferred finger pose over time. The particle filter (or sequential Monte Carlo filter) is a probabilistic predictor-corrector, which maintains an estimate of distribution over possible finger poses $P(\mathbf{y_t}), \quad \mathbf{y_t} = [x_t, y_t, z_t, \theta_t, \phi_t]$ as a set of discrete samples. It updates this distribution by weighting samples according to how close their expected sensor state would be to true observations of the sensor $\mathbf{X}$. Internal dynamics which model the potential evolution of inferred finger poses allow the filter to predict likely future states when observations are noisy or incomplete. We use both our forward model which predicts $\hat{\mathbf{X}}$ given $\mathbf{y}$ (to predict the sensor states we would expect from an estimated internal state) **and** our inverse model which predicts $\hat{\mathbf{y}}$ from $\mathbf{X}$ in the filter. Figure 3 illustrates a particle filter architecture which naturally combines forward and inverse models.

NEXT STEPS IN INTEGRATING FORWARD AND INVERSE MODELS

Practical computational models

Often, scientific knowledge and models will be locked into legacy code or commercially confidential software, such that the forward models are essentially black-box models which can be executed to generate data, but cannot be conveniently interrogated internally. The availability of flexible algorithmically differentiable models which can learn from data, such as those used in deep learning environments such as Tensorflow, means that if we can stimulate the forward model sufficiently to be able to machine learn a 'clone' of it, we also have an analytically differentiable model.[3]

[3] *Black-box models of complex simulations:* Flexible statistical models have been used to create more efficient representations of computationally complex simulators [6, 12, 17]. This requires initial simulation to generate training data for a machine learning solution, which can then run more rapidly than the original data – it can be viewed as a 'glorified lookup table' which performs inference between the observations to avoid exponential explosions of required storage. We can represent the simulator in the form of a function $y = f(x)$. Each run of the simulator is defined to be the process of producing one set of outputs y for one particular input configuration x. We assume that the simulator is deterministic, that is, running the simulator for the same x twice will yield the same y.

Transfer learning: In many machine learning tasks it is possible to get a lot of data for a closely related task. The forward simulation approach has the advantage that we can simulate the capacitance for an ideal sensor without disturbances due to sensor noise. This is potentially useful as part of the representation learning process, as it can help the forward model generate the appropriate structures. Furthermore, we can simulate a wide range of finger types, poses and positions, but the simulated model will always be an approximation of the real sensors.

Accelerated, differentiable and invertible simulation

Machine learning models can be used to create a computationally efficient, accelerated implementation of the forward, or causal model implemented by e.g. an electrostatic simulation (or from physical experiments) *which is also algorithmically differentiable.* This can then be used in real-time to infer the most likely inputs, by performing gradient-based optimisation to minimise the distance between the current sensed values and those of the model output. It also means that, as shown in Figure 4, we can concatenate a series of simulation tools, 'clone' them with machine learning tools and then efficiently differentiate the models for optimisation or control purposes This simulation pipeline for the sensor hardware can be augmented with biophysical models of human movement, e.g. [3, 15].

Further recent activity in the machine learning community has explored analytically invertible deep neural networks [1, 8, 9], which have the potential to learn both forward and inverse models simultaneously, in a differentiable structure. [4]

[4] An advantage of the conversion of general simulation code to a deep network running on a Tensorflow graph is the ease with which we can analytically calculate the model gradients – a simple call to `tf.gradients(loss, model.input)` suffices. This is useful, e.g. when inferring which inputs might have caused an observed sensor vector. Starting from a random initial guess, x_0, and adapting $x \propto \frac{\partial L}{\partial x}$, for our loss function L, we can converge on a plausible input vector x.

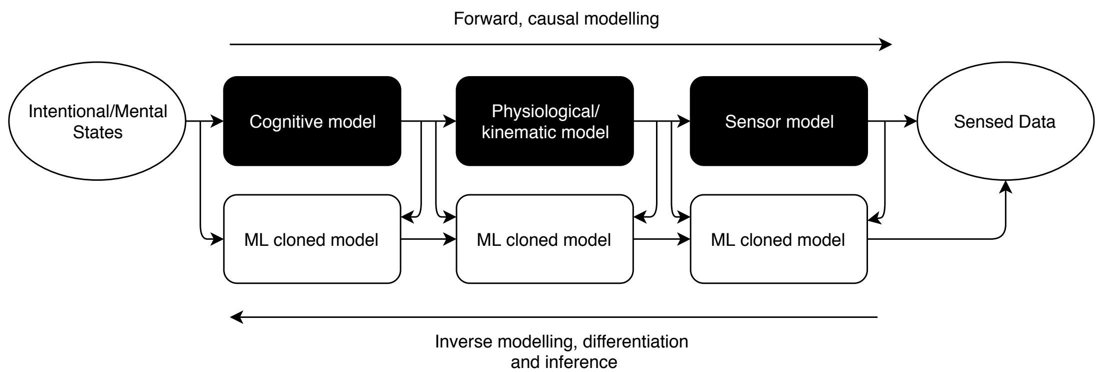

Figure 4: Concatenated differentiable models to solve forward and inverse problems in HCI contexts.

CONCLUSIONS

If the inverse problem for a particular task and interface combination is difficult to solve, it means the interface is likely to become difficult to use because the system cannot reliably infer intention from sensed data. We suggest to computationally design interfaces to make them easy to invert; such that smooth, natural, predictable interactions on the part of a user correspond to efficient tours of the information space, augmented with rich, human-centric feedback. We can design systems such that tools that humans are good at controlling can generate intention signals that are easy to reliably identify, and we can do so in simulation.

When designing novel systems it would be advantageous to be able to create a simulation pipeline from which we can predict the performance of different physical sensor systems and the associated algorithms for inference of intention from human movements. Furthermore, theoretical models and physical observations can be combined with modern machine learning models to create accurate differentiable and invertible models, simplifying the inference of intent from sensed signals.

Systematic application of forward and inverse models has the potential to turn the analysis and design of input systems for human–computer interaction from an art into a science.

ACKNOWLEDGEMENTS

We thank EPSRC for support from grant EP/R018634/1, *Closed-Loop Data Science for Complex, Computationally- and Data-Intensive Analytics*

REFERENCES

[1] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe. 2018. Analyzing Inverse Problems with Invertible Neural Networks. *ArXiv e-prints* (Aug. 2018).

[2] Myroslav Bachynskyi. 2016. *Biomechanical models for human-computer interaction*. Ph.D. Dissertation. Universität des Saarlandes Saarbrücken.

[3] Myroslav Bachynskyi, Gregorio Palmas, Antti Oulasvirta, Jürgen Steimle, and Tino Weinkauf. 2015. Performance and Ergonomics of Touch Surfaces: A Comparative Study using Biomechanical Simulation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1817–1826.

[4] Michael J Black and Allan D Jepson. 1998. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *European conference on computer vision*. Springer, 909–924.

[5] Andrew Blake and Michael Isard. 1997. The condensation algorithm-conditional density propagation and applications to visual tracking. In *Advances in Neural Information Processing Systems*. 361–367.

[6] Stefano Conti, John Paul Gosling, Jeremy E. Oakley, and Anthony O'Hagan. 2009. Gaussian process emulation of dynamic computer codes. *Biometrika* (2009), asp028.

[7] Scott L Delp, Frank C Anderson, Allison S Arnold, Peter Loan, Ayman Habib, Chand T John, Eran Guendelman, and Darryl G Thelen. 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering* 54, 11 (2007), 1940–1950.

[8] L. Dinh, J. Sohl-Dickstein, and S. Bengio. 2016. Density estimation using Real NVP. *ArXiv e-prints* (May 2016).

[9] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. 2018. FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models. *ArXiv e-prints* (Oct. 2018).

[10] Robert Hester, Alison Brown, Leland Husband, Radu Iliescu, William Andrew Pruett, Richard L Summers, and Thomas Coleman. 2011. HumMod: a modeling environment for the simulation of integrative human physiology. *Frontiers in physiology* 2 (2011), 12.

[11] Antti Kangasrääsiö, Kumaripaba Athukorala, Andrew Howes, Jukka Corander, Samuel Kaski, and Antti Oulasvirta. 2017. Inferring Cognitive Models from Data Using Approximate Bayesian Computation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1295–1306.

[12] Marc C. Kennedy and Anthony O'Hagan. 2001. Bayesian calibration of computer models. *Journal of the Royal Statistical Society. Series B, Statistical Methodology* (2001), 425–464.

[13] Misagh Mansouri and Jeffrey A Reinbolt. 2012. A platform for dynamic simulation and control of movement based on OpenSim and MATLAB. *Journal of biomechanics* 45, 8 (2012), 1517–1521.

[14] Tony O'Hagan. 2004. Dicing with the unknown. *Significance* 1, 3 (2004), 132–133.

[15] Antti Oulasvirta, Sunjun Kim, and Byungjoo Lee. 2018. Neuromechanics of a Button Press. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 508.

[16] Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. 2011. AnglePose: robust, precise capacitive touch tracking via 3D orientation estimation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2575–2584.

[17] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. 1989. [Design and Analysis of Computer Experiments]: Rejoinder. *Statist. Sci.* (1989), 433–435.

[18] Albert Tarantola. 2005. *Inverse problem theory and methods for model parameter estimation*. Vol. 89. SIAM.