Interactive Bayesian Optimization for Game Mechanics

Abstract

fill me

Game design often involves a final phase of substantial fine-tuning of game assets. Paradigmatic examples include varying the settings of player-controlled character movement parameters, altering opponent combat statistics, or varying low-level parameters around movement and collision of game objects. Tuning is often a time-consuming and expensive process for several reasons:

1. parameter values must be set to (globally) optimal values, requiring search over a large space
2. evaluation of a setting cannot be done analytically or via simulation but requires costly (in terms of time and money) direct human evaluation
3. quality of a set of parameters may be difficult to specify on a global scale, but instead be relative to other sets of parameters

Interactive Bayesian optimization (closely related to active learning (Settles 2012) and sequential experimental design (Chaloner and Verdinelli 1995)) approaches can address these issues through optimization of design objectives that are expensive to evaluate (Brochu 2010). Employing non-parametric models (here Gaussian Processes) we demonstrate the application of interactive Bayesian optimization to two cases studies of game design tuning in a shoot-em-up game: (1) optimizing player controls to player preferences and (2) adjusting enemy design parameters to enforce a desired level of player behavior.

For control optimization we demonstrate how a preference-learning approach can provide potential control settings to be tested and evaluated against the previous set of controls. Bayesian optimization affords automatic exploration-exploitation trade-offs that enable rapidly (globally) optimizing controls to player preferences via pairwise preference feedback. For enemy design optimization we demonstrate how a designer-specified objective function for player performance statistics can guide building a regression model from enemy parameter settings to desired design features.

First, we discuss related work in game tailoring and adaptation. Second, we motivate and describe our interactive Bayesian optization appraoch, detailing the Gaussian process regression and preference learning models. Third, we describe our shoot-em-up game and describe two empirical human studies demonstrating the efficacy of our approch. We conclude by discussing extensions and the range of applications of this modeling approach.

Related Work

(Yu and Trawick 2011) uses SVM, hard to AL on this

(Hunicke and Chapman 2004) ad hoc

(Yannakakis and Hallam 2009) (Yannakakis, Maragoudakis, and Hallam 2009) GP pref model employed, not found best. however, we use more sophisticated kernel fn in order to better adjust fit parameters across dimensions and get back information on relative importance of dimensions

(Bakkes, Spronck, and van Lankveld 2012)

Interactive Bayesian Optimization

Sequential Bayesian optimization is a modeling approach where a function is optimized through a sequence of points that are tested, each one selected by some algorithm based on previous points. Two functions are involved: (1) the *objective function* that maps inputs to outputs; and (2) the *acquisition function* that maps potential inputs to their value for optimizing the objective function output. In our application we employ Gaussian processes (GPs) for the objective function and a modified expected improvement acquisition function. Our approach falls under the umbrella of interactive Bayesian optimization as each point selected uses feedback provided in interaction with a human (Brochu 2010).

Gaussian processes are a widely used non-parametric modeling technique able to capture complex non-linear relationships within a data set, automatically adjust model complexity to data, and integrate out parameters without user intervention (Rasmussen and Williams 2006). Intuitively, non-parametric models are models that allow for an infinite number of variables to account for the data before selecting only the subset needed to explain a given set of observations. In practice, this leads to models that automatically become more complex to fit a dataset as needed. Bayesian formulations of GP regression and classification automatically trade-off between complexity of a model and fit to a data set, avoiding overfitting and poor generalization problems that occur with optimization approaches. Bayesian model specifications allow parameters of the model to be integrated out, simplifying their use by requiring less user specification. We employ GPs to leverage the benefits of: non-linear mapping from inputs to outputs, automatic complexity adjustment with data collection, and reduced or eliminated parameter specification from users.

Below we describe the formulation of GP regression and GP preference learning and then integrate GP models with active learning methods. Gaussian process regression enables automatic difficulty adjustment by modeling player performance in a game as a non-linear function of game parameters. Gaussian process preference learning enables optimization of game parameters (here controls) to player preferences by modeling player preferences for a set of game parameters as a non-linear function of game parameters than is then forced to pairwise choices between alternatives. Active learning uses a GP objective function to identify desirable next parameter settings to test, guided by a designer-specified acquisition function—here expected improvement—for parameter adjustment. For game performance, designers specify a goal of achieving a given level of in-game performance. For controls, designers specify optimal player preference.

Gaussian Process Regression

Gaussian processes are formally defined as “a collection of random variables, any finite number of which have a joint Gaussian distribution” (Rasmussen and Williams 2006). While allowing an infinite number of variables to be used, any GP model can be computed through a multivariate Gaussian distribution based on the input and output values. Gaussian processes are specified by their mean function ($m(x)$) and covariance function ($k(x, x')$):

$$f(x) \sim GP(m(x), k(x, x'))$$

Intuitively, GP regression learns a model predicting that similar inputs—according to the covariance function—should yield similar outputs. Different choices of the covariance function define different notions of similarity. In our work we employ the automatic relevancy detection (ARD) version of a squared exponential distance:

$$k(x, x') = exp\big(- \frac{1}{2} \Sigma_{l=1}^{d} \kappa^l (x^l - x'^l)^2 \big)$$

where $\kappa^l > 0$ is the ARD parameter for the l-th feature of a d-dimensional data set, serving to control the contribution of this feature to the model. Automatic relevancy detection allows us to optimize model parameters during the fitting process, automatically scaling input dimensions to minimize the impact of irrelevant aspects of the data. Mathematical properties of the GP mean that an initially zero valued mean function will taken on non-zero values after fitting data, allowing the model to be initialized with zero as the mean value (see (Rasmussen and Williams 2006) for additional details on GP regression). In our case we will use such a zero-mean GP.

For our performance regression model we predict player performance (number of times hit) from game parameters controlling enemy attacks (speed and size of bullets along with firing rate). We fit a GP regression model to player data and optimize the covariance function ARD parameters using stochastic gradient descent after each training point received. Since GP regression has a closed-form solution for learning and prediction this task can be done in near-real time with no appreciable time requirements (<1 second for

N training points

).

Gaussian Process Preference Learning

Gaussian process preference learning models choices between two possible alternatives with a GP specifying the underlying value (unobserved) of those alternatives to a user (Chu and Ghahramani 2005).

(Brochu 2010)

Active Learning

(Settles 2012)

Experiment

Game Domain

Methods

Results

Discussion

Acknowledgments

References

Bakkes, S. C.; Spronck, P. H.; and van Lankveld, G. 2012. Player Behavioural Modelling for Video Games. *Entertainment Computing* in press:1–9.

Brochu, E. 2010. *Interactive Bayesian optimization: learning user preferences for graphics and animation*. Ph.D. Dissertation, University of British Columbia.

Chaloner, K., and Verdinelli, I. 1995. Bayesian experimental design: A review. *Statistical Science* 10 (3):273–304.

Chu, W., and Ghahramani, Z. 2005. Preference learning with gaussian processes. In *Proceedings of the 22nd International Conference on Machine learning*, 137–144. ACM.

Hunicke, R., and Chapman, V. 2004. AI for dynamic difficulty adjustment in games. In *Proceedings of the AAAI Workshop on Challenges in Game Artificial Intelligence*.

Rasmussen, C. E., and Williams, C. K. 2006. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, MA.

Settles, B. 2012. *Active learning*, volume 6. Morgan & Claypool Publishers.

Yannakakis, G. N., and Hallam, J. 2009. Real-time game adaptation for optimizing player satisfaction. *IEEE Transactions on Computational Intelligence and AI in Games* 1(2):121–133.

Yannakakis, G. N.; Maragoudakis, M.; and Hallam, J. 2009. Preference learning for cognitive modeling: a case study on entertainment preferences. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 39(6):1165–1175.

Yu, H., and Trawick, T. 2011. Personalized procedural content generation to minimize frustration and boredom based on ranking algorithm. In *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*.