XPod: A Human Activity Aware Learning Mobile Music Player

Sandor Dornbush, Jesse English, Tim Oates, Zary Segall, Anumpam Joshi
University of Maryland, Baltimore County
{sandor1, english1, oates, zary, joshi} @ umbc.edu

Abstract

The XPod system, presented in this paper, aims to integrate awareness of human activity and musical preferences to produce an adaptive system that plays the contextually correct music. The XPod project introduces a "smart" music player that learns its user's preferences and activity, and tailors its music selections accordingly. We are using a BodyMedia device that has been shown to accurately measure a user's physiological state. The device is able to monitor a number of variables to determine its user's levels of activity, motion and physical state so that it may predict what music is appropriate at that point. The XPod user trains the player to understand what music is preferred and under what conditions. After training, the XPod, using various machine-learning techniques, is able to predict the desirability of a song, given the user's physical state.

Figure 1: Proposed XPod Form Factor

1 Introduction

In this paper we study the problem of creating a music player that plays contextually correct music by using a system of physiological sensors to monitor a user's state. We propose using machine learning algorithms to estimate a user's preference of songs in various situations. Several machine learning algorithms were used to model the complex relationships between an individual's musical preferences and his or her activities. We studied several machine learning systems on a modified version of the existing mobile MP3 player, XPod. This player will attempt to select the song best suited to the current activity of the user. XPod was previously reported on in [Dornbush *et al.*, 2005], wherein the system was designed to use a database and a neural network to suggest music to users. In this paper we expand on that work, and approach many different machine-learning algorithms, with varied results.

This paper is layed out in the following format: in Section 2, we will review similar work in this field. In Section 3, we will discuss the motivation for creating such a device. In Section 4, we will speak about the data collected and how it was used to produce learning algorithms. We will then discuss the results of five experiments in Section 5, and end with a discussion on future research in this area in Section 6.

2 Related Work

Context aware systems can greatly improve a users experience with computer systems. For example several systems relate user activity to mobile phones [Bylund and Segall, 2004; Siewiorek *et al.*, 2003]. This paper proposes an extension to the mobile MP3 player, XPod, which is able to automate the process of selecting the song best suited to the current activity of the user. That system used a collection of machine learning techniques to play contextually correct music. The array of user state information was converted to one of three states (active, passive and resting) using a decision tree. The state was used as a key into the database of state dependant ratings. That was combined with the results from a neural network that estimated the users preference of a song. While this showed the potential of an adaptive context aware music player it had significant limitations.

The concept of a music player that is aware of the user's activity has made it into the mainstream market with industry leaders Nike and Apple teaming up to deliver an iPod

that wirelessly communicates with a sensor in a Nike running shoe[Nike and Apple, 2006]. That system has had limited market success so far, but has shown that there is consumer interest. The primary use case for this system is to record and analyze a runners athletic performance. That system is built to facilitate the user's athletic training, not to allow the user to train the system. We uniquely address the problem of a context aware music player that learns a users preferences.

Other researchers have studied the relationship between a users activity and the music selection played for them. In [Park *et al.*, 2006] the authors explore the idea of using Bayesian networks to predict the proper music for the current situation. The authors used a fairly small set of attributes to describe the context. The authors only used attributes from the large area context such as weather, user gender, time and season.

Sonic City[Gaye *et al.*, 2003; Gaye and Holmquist, 2004] developed a wearable jacket that created music based on the sensed light, noise and movement. In this experiment the user and the environment together create music.

[Elliott and Tomlinson, 2006] developed a system that correlates the song played to a users pace. This system used machine learning to determine which song to play, matching the song's beats per minute(BPM) to the users pace. While this naive approach provides a level of context aware music it is not sufficient. For one this does not address the question of what music to play when a user is not active or when participating in an activity such as bicycling where the concept of pace is not well defined. Secondly in our experiments BPM is not sufficient to characterize a piece of music. We found some pieces of music with high BPM that our user found suitable to low activity level, and pieces of music with low BPM that were suitable for high activity level.

3 Motivation

The XPod concept is based on the idea of automating much of the interaction between the music player and its user. The XPod project introduces a "smart" music player that learns its user's musical preferences for different activities, and tailors its music selections accordingly. The device is able to monitor a number of variables to determine its user's levels of activity, motion and physical states at the current moment and predict what music would be appropriate. The XPod user trains the player to predict the preference of music under various conditions. After an initial training period, XPod is able to use its internal algorithms to make an educated selection of the song that would best fit its user's emotion and activity.

Before playing a song the internal algorithm is used to predict the users rating of that song in their current state. That prediction is used to weigh the chance that the current song will be played. A song with a low expected rating may be skipped in the current state. Every song has a chance of being played at any time. This is done so that the XPod explores the feature space and does not get stuck playing a few songs. For example if a song is rated zero once, that should not be interpreted rating that song as zero in all contexts. In a different state that song might be rated a four. Our goal would be to model the level of knowledge for every song; then use that

Figure 2: An author collecting training data.

	Name	Type
Input	Galvanic Skin Response	Real value
	Mean Acceleration Longitudinal	Real value
	Std. Dev. Acceleration Longitudinal	Real value
	Mean Acceleration Transversal	Real value
	Std. Dev. Acceleration Transversal	Real value
	Skin Temperature	Real value
	Heat Flow	Real value
	Heat Flow Cover	Real value
	Transversal Cadence	Integer
	Longitudinal Cadence	Integer
	Time of Day	Integer
	Day of Week	Integer
	Song Genre	Symbolic
	Song Artist	Symbolic
	Song Album	Symbolic
	Song Title	Symbolic
	Beats Per Minute	Integer
Output	User's Action	Integer{0-4}

Table 1: Input and output fields for XPod classifiers.

Genre	Artists	Tracks
Alternative Rock	Red Hot Chili Peppers	17
Blues	Louie Armstrong	20
Electronic	M.I.A.	12
Funk	Mofofunka	15
Hip-Hop	Black Eyed Peas	14
	Digable Planets	11
	Kanye West	20
Jazz	Art Blakey	8
	John Coltrane	13
	Miles Davis	10
Reggae	Bob Marley	10
Rock	Beatles	11
	Phish	16
	Rolling Stones	10
	Grateful Dead	8
	Jimi Hendrix	6
	Santana	9
Ska	Tokyo Ska Paradise Orchestra	27

Table 2: Music library.

model to weight the trust we have in the rating. At this point we used a fixed discount of the predicted rating.

We propose a form factor illustrated in Figure 1 where the device is mounted on an armband matching the current widespread usage patterns of MP3 players. In addition a device on the arm can capture an accurate view of how a user is moving his or her body.

4 XPod Dataset

The XPod system is comprised of a standard MP3 playing device and a human body sensor. The device tracks and stores a record of song meta-data as a song is played, including artist, album, genre, title, and beats-per-minute. In addition, the system records the time of day, a user's rating (from 0 to 4 stars), and a full range of physical responses from the user's body. The full set of attributes recorded are detailed in Table 1. These measurements include skin temperature, heat flow, two dimensions of acceleration, cadence, and galvanic skin response, which is a measure of how much sweat is on the user's skin. Each of the symbolic attributes, artist, album, title, and genre are all expanded into a large number of binary attributes. This was done so that the symbolic attributes could be accurately handled by the numerical algorithms, such as SVMs and neural networks. For this reason the total number of attributes in our experiments, including the user state, is 289. Typically each instance is a sparse array with most attributes set to 0 or false.

To gather the information about the user's physical state, a BodyMedia [BodyMedia, 2006] device was used. This device straps on to the arm, and broadcasts its readings wirelessly to a nearby system, which records the data for use by the XPod. The BodyMedia device is capable of monitoring a user's physiological and emotional state[Nasoz *et al.*, 2003]. In this paper we focus on the physiological state; however this system should be able to adapt musical preference to the emotional state.

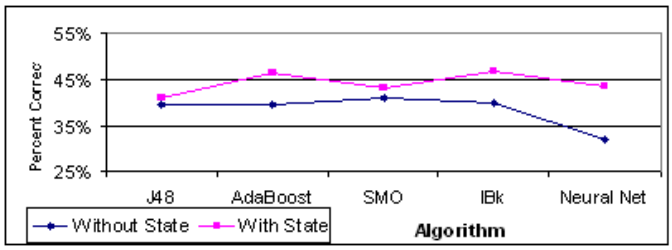
5 Machine Learning Algorithms

To test the XPod, we trained several independent learning algorithms on our test data. To construct our dataset, we gathered 239 different mp3 song files. Each song was analyzed to find the beats per minute. A researcher collected training information using a prototype system. The prototype shown in Figure 2 involved a tablet computer and a BodyMedia device.

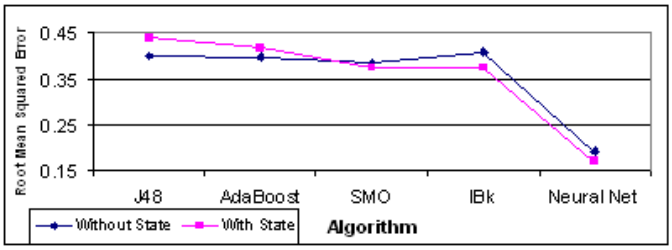
A researcher on the XPod team proceeded to record training instances in a variety of physical situations (exercise, mild activity, rest, etc.). A training instance, or data point, includes a value for each field in Table 1. 565 training instances were recorded. For each instance the XPod player would rate a song and play that song. If the rating matched the researcher's preference he took no action. If the rating did not match the preference the researcher gave the song a rating from 0 to 4, reflecting how appropriate the researcher felt the song was at that time. A rating of 0 would result in the music player skipping the remainder of the song. Each classification algorithm was trained on some or all of the training instances. That training was used to predict how a user would rate a song in the future.

It is our goal to show that a music player can choose the contextually correct music if it uses information about a user's physiological state. To prove this theory we created two sets of machine learning systems, those trained with user state information, and those without user state information. "State" refers to the array of information gained from the BodyMedia device, as well as any other outside information, such as date and time. Due to our small sample size, activity can be inferred from the date and time. For this reason those variables where included in state information. We will consider our experiment successful if the system is more accurate when it has access to the state information than when it does not have access to state information.

We used 10 fold cross-validation to measure the accu-



(a) % Accuracy of various learning algorithms.



(b) RMSE of various learning algorithms.

Figure 3: Performance of learning algorithms

racy of the machine learning algorithms. We also experimented with Leave-One-Out-Cross-Validation(LOOCV). We found very similar results between the two methods. Since LOOCV is much more expensive we have reported the results of 10 fold cross-validation. We used classifiers from the open source Weka library[Witten and Frank, 2005] and neural networks from the open source Joone library[Marrone and Team, 2006].

In the following experiments each training instance could be classified into one of five classes. The expected performance of a random algorithm would be 20%. All of the algorithms performed significantly better than random.

5.1 Decision Trees

The first classifier used was the decision tree algorithm (J48)[Quinlan, 1993]. When learning without state, the decision tree was able to properly classify the training data 39.47% of the time. However, when using state, the decision tree was able to properly classify the training data 41.06% of the time. The accuracy of decision trees was not the best in the survey, however they do show a slight (2%) advantage of using state information in the learning algorithm. The poor generalization of J48 can be seen in the fact that the system not using state information had less error than the system that did use state information.

5.2 AdaBoost

The J48 classifier improved significantly when it was boosted with AdaBoost (AdaBoostM1)[Freund and Schapire, 1996]. This classifier was correct against the training data 39.47% without state, and 46.55% with state. AdaBoost suffered from the same generalization performance inversion as the original J48 classifier. This showed that AdaBoost is somewhat effective at increasing the performance of the J48 algorithm.

5.3 Support Vector Machine (SVM)

The third classifier we experimented with was support vector machines (SMO)[Platt, 1998; Keerthi *et al.*, 2001]. SVM's generalized well and had a little improvement when using state (43.19%) over not using state (40.89%). In this case the SVM was almost able to divide the dataset into the researcher's preference based solely on the musical data. When adding in state, the dimension space changed minimally, adjusting enough to shuffle a few incorrectly classified instances to the proper area.

The small difference between the SVM trained with state information and without state information, (2%) is likely a result of the relatively large feature space and small training set. The expressiveness of the second order kernel allows the SVM to identify the user's preference without the state information.

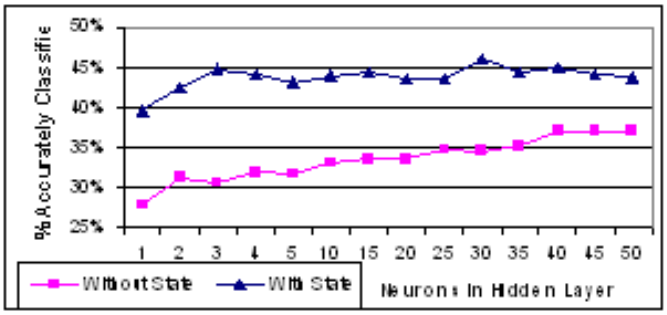
5.4 K-Nearest Neighbors(KNN)

We had surprisingly positive results from the lazy classifier: k-nearest neighbors (IBK)[Aha and Kibler, 1991]. We allowed Weka to choose the optimal number of neighbors. The best number of neighbors was found to be 9. Results showed a 7% increase in accuracy when using state (46.72%) over not using state (39.82%). More importantly KNN had a low root mean squared error (RMSE) (0.3753).

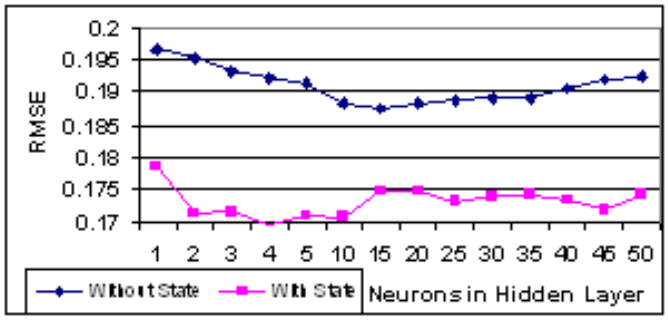
5.5 Neural Networks

We had very promising results from a neural network trained on this data-set. We created a three layer network with 288, or 276 inputs depending on whether state information was used. A small hidden layer and a single neuron output was used. We experimented with a variety of different size hidden layers from 1 to 50. The results of these experiments are shown in 4(a) and 4(b) We found very similar results with a small number of hidden nodes, 3, as when we used a large number of hidden nodes, 50. As the size of the hidden layers grows, the accuracy of the network using state information does not increase much. However the accuracy of the network not using state information does increase. We believe that the more complex networks are better at memorizing erroneous information to accurately rate the songs.

We had difficulties with over training. The network would find the best validation error in the first 100 training epochs. We used early stopping to keep the best network on the validation data. We are investigating ways to avoid this problem. We were able to achieve respectable performance with a network given state information. That network correctly classified instances 43.54% much better than the 31.87% accuracy without state information. This is not the best percent accuracy, however it did get the best results in terms of RMSE (0.17). The neural network had a fraction of the RMSE of the other methods.



(a) % Accurately identified using different size hidden layers.



(b) RMSE using different size hidden layers.

Figure 4: Performance of different size networks.

6 Conclusion and Future Work

Our goal is to show that a music player trained with a user's physical activity and preference can choose the contextually correct music. All of the systems evaluated performe significantly better than random(20% accuracy). As shown in Figure 3(a), given state information every system chooses the exactly correct label more often than the same algorithm without state information. For each testing instance there are five possible categories. Figure 3(b) shows that the tree based algorithms tend to generalize poorly, evidenced by the fact that systems using state information had greater RMSE than the systems not using state information. The other algorithms were able to generalize well and achieved high accuracy and low RMSE.

We believe that if we collected still more training instances the difference between the performance of the statefull and the stateless system would grow. Presumably if we collected enough training instances we would find a pair of instances that are identical on all non-state attributes but have different ratings. Then any classifier without state information would have to give both instances the same label. However only one could possibly be correct. A classifier given the same instance including the state information has a chance to classify both instances correctly. Therefore if we collected more training instances the difference between stateful and stateless systems should increase.

Although the lazy classifiers tended to perform well, in practice this may not be the case. Specifically, a portable music device is not likely to have high processing power. Given an active user of such a device, listening for multiple hours a day over the course of one or two years, the device would search an instance space of over tens of thousands of data points. Performing a calculation like this may be more than inefficient: it could be wholly impractical.

Support vector machines may be well suited to the task as they can begin to classify new instances having very little training data to build on. From the end-user's perspective, this is a desirable feature, as the user would need to spend very little time setting up the system, and more time enjoying the benefits. Further, SVMs are capable of classifying in a very high dimensional space while only performing calculations in a much smaller number of dimensions. However it is not clear if SVMs could be created on a constrained device. Perhaps the SVMs would be created on a unconstrained device such as a PC. Then the trained SVM would then be transferred to the portable device. Even constrained devices can evaluate a SVM.

Decision trees would likely be the most computational feasible classifier, as they can be converted into a rule set, which can be evaluated very rapidly. As we've shown in this application, decision trees perform better with boosting.

Our view is that the neural network is the most promising result. Although it did not get the highest exact accuracy it tended to get very close to the right answer, reflected in the small RMSE. Since the result was used to influence pseudo-random choice of music it is actually more important to be close than to be exactly accurate. For example if the correct rating should be 4 but the system responds with 3, that will result in a low RMSE, but will not count towards the percent correct. A rating of 0 would result in the same percent accurate, however a much higher RMSE. Since a song rated 0 would be skipped, but a song rated three or four would likely be played, a close answer is almost as good as the correct answer. For that reason we feel that the low RMSE of the neural network indicates that it would be the most useful algorithm. Many embedded devices such as mobile phones already employ neural networks, therefore it should be possible to use neural networks in mobile music playing devices.

In future work we will investigate other meta-data that could be associated with the music. We have used relatively simple music analyzing software to find the beats per minute, however it is possible to find much more by analyzing the music[Logan and Salomon, 2001]. It would also be interesting to investigate human generated meta data in community systems such as the Pandora Project[Westergren, 2006] or Audoscrobbler[Audioscrobbler, 2006]. Any new meta-data regarding songs could be included as additional inputs into the machine learning algorithms. We will investigate augmenting the training instances already collected with additional meta-data. Our goal will be to see if there is a significant increase in performance given new information.

We will investigate prototyping this system in a physical device. While the BodyMedia device provides many different attributes a satisfactory system could likely be built with a

selection of those attributes. An inspection of the decision tree built by J48 shows 20 decisions based on acceleration, almost four times more than the sum of all decisions based on other state variables.

Nokia research has generously granted our research group two 5500 Sport phones[Nokia, 2006]. These phones have accelerometers to measure a users activity level and the capability to play MP3's. We expect to implement the XPod system on this device in the near future.

We have shown the relative advantages of different machine learning systems at choosing the contextually correct music. People have shown an interest in this type of system. However more works need to be done to further refine this system.

7 Acknowledge

We would like to thank Chad Eby for the renditions of the proposed XPod form factor and Nokia for the donation of the 5500 Sport phones.

References

[Aha and Kibler, 1991] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[Audioscrobbler, 2006] Audioscrobbler. http://www.audioscrobbler.net/, 2006.

[BodyMedia, 2006] BodyMedia. http://www.bodymedia.com/, 2006.

[Bylund and Segall, 2004] M. Bylund and Z. Segall. Towards seamless mobility with personal servers. *INFO Journal*, May 2004.

[Dornbush *et al.*, 2005] Sandor Dornbush, Kevin Fisher, Kyle McKay, Alex Prikhodko, and Zary Segall. XPod a human activity and emotion aware mobile music player. In *Proceedings of the International Conference on Mobile Technology, Applications and Systems*, November 2005.

[Elliott and Tomlinson, 2006] Greg T. Elliott and Bill Tomlinson. Personalsoundtrack: context-aware playlists that adapt to user pace. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 736–741, New York, NY, USA, 2006. ACM Press.

[Freund and Schapire, 1996] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. pages 148–156, San Francisco, 1996. Morgan Kaufmann.

[Gaye and Holmquist, 2004] Lalya Gaye and Lars Erik Holmquist. In duet with everyday urban settings: a user study of sonic city. In *NIME '04: Proceedings of the 2004 conference on New interfaces for musical expression*, pages 161–164, Singapore, Singapore, 2004. National University of Singapore.

[Gaye *et al.*, 2003] Lalya Gaye, Ramia Mazé, and Lars Erik Holmquist. Sonic city: the urban environment as a musical interface. In *NIME '03: Proceedings of the 2003 conference on New interfaces for musical expression*, pages 109–115, Singapore, Singapore, 2003. National University of Singapore.

[Keerthi *et al.*, 2001] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Computation*, pages 637–649, 2001.

[Logan and Salomon, 2001] B. Logan and A. Salomon. A music similarity function based on signal analysis. In *IEEE International Conference on Multimedia and Expo, ICME*, 2001.

[Marrone and Team, 2006] Paolo Marrone and Joone Team. Joone, 2006. http://www.jooneworld.com/.

[Nasoz *et al.*, 2003] Fatma Nasoz, Kaye Alvarez, Christine L. Lisetti, and Neal Finkelstein. Emotion recognition from physiological signals for presence technologies. *International Journal of Cognition, Technology and Work - Special Issue on Presence*, 6, 2003.

[Nike and Apple, 2006] Nike and Apple. http://www.apple.com/ipod/nike/, 2006.

[Nokia, 2006] Nokia. 5500 sport phone, 2006.

[Park *et al.*, 2006] H. S. Park, J.O. Yoo, and S. B. Cho. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *International Conference on Fuzzy Sytems and Knowledge Discovery (FSKD'06)*, pages 970–979, 2006.

[Platt, 1998] J. Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization. Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[Quinlan, 1993] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[Siewiorek *et al.*, 2003] Daniel Siewiorek, Asim Smailagic, Junichi Furukawa, Neema Moraveji, Kathryn Reiger, and Jeremy Shaffer. Sensay: A context-aware mobile phone. In *ISWC*, 2003.

[Westergren, 2006] Tim Westergren. Music genome project, 2006. http://www.pandora.com/.

[Witten and Frank, 2005] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*, volume 2nd Edition. Morgan Kaufmann, San Francisco, 2005.