The Bayes Point Machine for Computer-User Frustration Detection via PressureMouse

Yuan Qi
Media Lab, MIT
Cambridge, MA, 02139
yuanqi@media.mit.edu

Carson Reynolds
Media Lab, MIT
Cambridge, MA, 02139
carsonr@media.mit.edu

Rosalind W. Picard
Media Lab, MIT
Cambridge, MA, 02139
picard@media.mit.edu

ABSTRACT

We mount eight pressure sensors on a computer mouse and collect mouse pressure signals from subjects who fill out web forms containing usability bugs. This approach is based on a hypothesis that subjects tend to apply excess pressure to the mouse after encountering frustrating events. We then train a Bayes Point Machine in an attempt to classify two regions of each user's behavior: mouse pressure where the form-filling process is proceeding smoothly, and mouse pressure following a usability bug. Different from current popular classifiers such as the Support Vector Machine, the Bayes Point Machine is a new classification technique rooted in the Bayesian theory. Trained with a new efficient Bayesian approximation algorithm, Expectation Propagation, the Bayes Point Machine achieves a person-dependent classification accuracy rate of 88%, which outperforms the Support Vector Machine in our experiments. The resulting system can be used for many applications in human-computer interaction including adaptive interface design.

1. INTRODUCTION

The Bayes Point Machine is a Bayesian linear classifier that can be converted to a nonlinear classifier by using feature expansions or kernel methods as the Support Vector Machine (SVM). By approximating the Bayesian average, it achieves good generalization performance [2, 4].

In this paper, we apply the Bayes Point Machine in an attempt to detect computer-users' frustration, by classifying pressure signals collected from a computer mouse specifically designed for this purpose, namely, PressureMouse [5]. The PressureMouse has eight pressure sensors so that it can collect a pattern of information related to how the user handles the mouse. In our experiment, every subject is asked to fill out a multiple-page web form using a normal keyboard and a PressureMouse. From the standpoint of the user, the PressureMouse works just like a normal mouse, requiring no training or special effort to use. Subjects are not told anything about the purpose of the mouse until after the experiment is concluded; thus, we expect that subjects used the PressureMouse as an ordinary mouse. We hypothesize that events that are frustrating will show up in patterns of use of the mouse.

To intentionally invoke a subject's frustration, the information entered on a web form was erased following a fictitious problem. Subjects were forced to re-enter all of the data on the web page before being allowed to continue. This data loss event, coupled with time pressure placed on the subjects, is likely to induce mild frustration.

Post-test questionnaires and interviews suggest that the data-loss event invokes the subject's frustration. Furthermore, the data sets suggest that the frustration can affect the way the subject handles the mouse. By applying the Bayes Point Machine, we were able to reliably distinguish between mouse pressure signals gathered during the first time filling out the page (before the error notice) and the second time filling out the same page (after the error notice).

2. A NEW APPROACH FOR CLASSIFICATION: THE BAYES POINT MACHINE

2.1 Bayes Point

A linear classifier classifies a point $\mathbf{x}$ according to

$$t = \mathrm{sign}(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})) \quad (1)$$

for some parameter vector $\mathbf{w}$ (the two classes are $t = \pm 1$). The basis function $\boldsymbol{\phi}(\mathbf{x}_i)$ allows the classification boundary to be nonlinear in the original features. This is the same likelihood used in logistic regression and in Gaussian process classifiers.

Given a training set $D = \{(\mathbf{x}_1, t_1), ..., (\mathbf{x}_N, t_N)\}$, the likelihood for $\mathbf{w}$ can be written as

$$p(\mathbf{t}|\mathbf{w}, X) = \prod_i p(t_i|\mathbf{x}_i, \mathbf{w}) = \prod_i \Psi(t_i \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_i)) \quad (2)$$

where $\mathbf{t} = \{t_i\}_{i=1}^N$, $X = \{\mathbf{x}_i\}_{i=1}^N$, $\Psi(\cdot)$ is the cumulative distribution function for a Gaussian. We can also use the step function or logistic function as $\Psi(\cdot)$.

What distinguishes the Bayes Point Machine is how we train the weights. Given a new input $\mathbf{x_{N+1}}$, we approximate the predictive distribution:

$$p(t_{N+1}|\mathbf{x}_{N+1}, \mathbf{t}) = \int p(t_{N+1}|\mathbf{x}_{N+1}, \mathbf{w})p(\mathbf{w}|\mathbf{t})d\mathbf{w} \quad (3)$$

$$\approx P(t_{N+1}|\mathbf{x}_{N+1}, \langle\mathbf{w}\rangle) \quad (4)$$

PUI 2001 Orlando, FL USA

where $\langle \mathbf{w} \rangle$ denotes the posterior mean of the weights, called the Bayes Point. This approach does not require voting a large number of classifiers nor does it assume that the posterior is adequately captured by its mode, as in logistic regression.

2.2 Expectation Propagation

Expectation Propagation(EP) [4] exploits the fact that the likelihood is a product of simple terms. If we approximate each of these terms well, we can get a good approximation to the posterior. Expectation Propagation chooses each approximation such that the posterior using the term exactly and the posterior using the term approximately are close in KL-divergence. This gives a system of coupled equations for the approximations which are iterated to reach a fixed point.

Expectation Propagation can also be viewed as a powerful extension of assumed-density filtering (ADF) [1, 3]. The ADF method is a sequential technique for approximating a posterior distribution that can be used in stochastic process modeling and online learning. Expectation Propagation extends ADF by using iterative batch-version refinements; this enables EP to utilize the information from the whole data sequence and to greatly improve the approximation quality.

First, a Gaussian prior distribution is assigned for $\mathbf{w}$

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_i \mathcal{N}(w_i|0, \alpha_i^{-1}) \tag{5}$$

where $\boldsymbol{\alpha} = \{\alpha_i\}$ is a hyperparameter vector. Later, we assign $\alpha_i = 1$ for all i.

Denote the exact terms by $g_i(\mathbf{w})$ and the approximate terms by $\tilde{g}_i(\mathbf{w})$:

$$p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}) \propto p(\mathbf{w}|\boldsymbol{\alpha}) \prod_i p(t_i|\mathbf{w}) = p(\mathbf{w}|\boldsymbol{\alpha}) \prod_i g_i(\mathbf{w}) \tag{6}$$

$$\approx p(\mathbf{w}|\boldsymbol{\alpha}) \prod_i \tilde{g}_i(\mathbf{w}) \tag{7}$$

For the Bayes Point Machine, the approximate terms are chosen to be Gaussian, parameterized by (m_i, v_i, s_i):

$$\tilde{g}_i = s_i \exp(-\frac{1}{2v_i}(t_i \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} - m_i)^2). \tag{8}$$

This makes the approximate posterior distribution also Gaussian:

$$p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}) \approx q(\mathbf{w}) = \mathcal{N}(\mathbf{m}_w, \mathbf{V}_w). \tag{9}$$

To find the best term approximations we proceed as follows: (to save notation, $t_i \boldsymbol{\phi}(\mathbf{x}_i)$ is written as $\boldsymbol{\phi}_i$)

1. Initialization Step:

   Set $\tilde{g}_i = 1$: $v_i = \infty$, $m_i = 0$, and $s_i = 1$.

   Also, set the prior: $\mathbf{m}_w = \mathbf{0}$, $\mathbf{V}_w = \mathrm{diag}(\boldsymbol{\alpha})$, $\alpha_i = 1$ for all i.

2. Loop until all (m_i, v_i, s_i) converge:

   Loop $i = 1, \ldots, N$:

   (a) Remove the approximation $\tilde{g}_i$ from $q(\mathbf{w})$ to get the 'leave-one-out' posterior $q^{\setminus i}(\mathbf{w})$, which is also Gaussian: $\mathcal{N}(\mathbf{m}_w^{\setminus i}, \mathbf{V}_w^{\setminus i})$. From $q^{\setminus i}(\mathbf{w}) \propto q(\mathbf{w})/\tilde{g}_i$, this implies

   $$\mathbf{V}_w^{\setminus i} = \mathbf{V}_w + \frac{(\mathbf{V}_w \boldsymbol{\phi}_i)(\mathbf{V}_w \boldsymbol{\phi}_i)^T}{v_i - \boldsymbol{\phi}_i^T \mathbf{V}_w \boldsymbol{\phi}_i} \tag{10}$$

   $$\mathbf{m}_w^{\setminus i} = \mathbf{m}_w + (\mathbf{V}_w^{\setminus i} \boldsymbol{\phi}_i) v_i^{-1} (\boldsymbol{\phi}_i^T \mathbf{m}_w - m_i) \tag{11}$$

   (b) Putting the posterior without i together with term i gives $\hat{p}(\mathbf{w}) \propto g_i(\mathbf{w}) q^{\setminus i}(\mathbf{w})$. Choose $q(\mathbf{w})$ to minimize $KL(\hat{p}(\mathbf{w}) \mid\mid q(\mathbf{w}))$. Let Z_i be the normalizing factor.

   $$\mathbf{m}_w = \mathbf{m}_w^{\setminus i} + \mathbf{V}_w^{\setminus i} \rho_i \boldsymbol{\phi}_i \tag{12}$$

   $$\mathbf{V}_w = \mathbf{V}_w^{\setminus i} - (\mathbf{V}_w^{\setminus i} \boldsymbol{\phi}_i)(\frac{\rho_i \boldsymbol{\phi}_i^T \mathbf{m}_w}{\boldsymbol{\phi}_i^T \mathbf{V}_w^{\setminus i} \boldsymbol{\phi}_i})(\mathbf{V}_w^{\setminus i} \boldsymbol{\phi}_i)^T \tag{13}$$

   $$Z_i = \int_{\mathbf{w}} g_i(\mathbf{w}) q^{\setminus i}(\mathbf{w}) \mathrm{d}\mathbf{w} = \Psi(z_i) \tag{14}$$

   where

   $$z_i = \frac{(\mathbf{m}_w^{\setminus i})^T \boldsymbol{\phi}_i}{\sqrt{\boldsymbol{\phi}_i^T \mathbf{V}_w^{\setminus i} \boldsymbol{\phi}_i + 1}} \tag{15}$$

   $$\rho_i = \frac{1}{\sqrt{\boldsymbol{\phi}_i^T \mathbf{V}_w^{\setminus i} \boldsymbol{\phi}_i + 1}} \frac{\mathcal{N}(z_i; 0, 1)}{\Psi(z_i)} \tag{16}$$

   (c) From $\tilde{g}_i = Z_i \frac{q(\mathbf{w})}{q^{\setminus i}(\mathbf{w})}$, update the term approximation:

   $$v_i = \boldsymbol{\phi}_i^T \mathbf{V}_w^{\setminus i} \boldsymbol{\phi}_i \Big(\frac{1}{\rho_i \boldsymbol{\phi}_i^T \mathbf{m}_w} - 1 \Big) \tag{17}$$

   $$m_i = \boldsymbol{\phi}_i^T \mathbf{m}_w^{\setminus i} + (v_i + \boldsymbol{\phi}_i^T \mathbf{V}_w^{\setminus i} \boldsymbol{\phi}_i) \rho_i \tag{18}$$

   $$s_i = Z_i \sqrt{1 + v_i^{-1} \boldsymbol{\phi}_i^T \mathbf{V}_w^{\setminus i} \boldsymbol{\phi}_i} \exp(\frac{\rho_i}{2} \frac{\boldsymbol{\phi}_i^T \mathbf{V}_w^{\setminus i} \boldsymbol{\phi}_i}{\boldsymbol{\phi}_i^T \mathbf{m}_w}) \tag{19}$$

3. Finally, compute the normalizing constant and the evidence:

   $$B = (\mathbf{m}_w)^T \mathbf{V}_w (\mathbf{m}_w) - \sum_i \frac{m_i^2}{v_i} \tag{20}$$

   $$p(D|\boldsymbol{\alpha}) \approx \int \prod_i \tilde{g}_i(\mathbf{w}) \mathrm{d}\mathbf{w} \tag{21}$$

   $$= |\mathbf{V}_w|^{1/2} \exp(B/2) \prod_i s_i \tag{22}$$

Denote the dimensionality of a training data point after feature expansion as d. The computational time of this algorithm is $O(d^2)$ for processing each data point, and therefore $O(Nd^2)$ per iteration. Only 5–6 iterations are needed to make the algorithm converge in our experiments.

In sum, we train a Bayes Point Machine through step 1 to step 3 of EP as described in the above. After the convergence of the algorithm, we obtain the posterior mean $\mathbf{m}_w$ of the weigths $\mathbf{w}$ as the Bayes Point. Note that $\mathbf{m}_w$ is the same as $\langle \mathbf{w} \rangle$ defined in equation (4). After obtaining the Bayes Point $\mathbf{m}_w$, we can classify a new test data point by letting $\mathbf{w} = \mathbf{m}_w$ in equation (1).

3. A NEW TOOL FOR HUMAN-COMPUTER INTERACTION: PRESSUREMOUSE

In this section, we present a new unobtrusive tool for human-computer interaction, the PressureMouse [5]. The PressureMouse (figure 1) is a special mouse that is equipped with eight pressure sensors so that it can collect pressure signals from computer users. Since a computer user may hold the mouse tighter or looser under different situations, the PressureMouse provides an opportunity to passively sense some aspects of muscle tension from the user. To design the PressureMouse, we briefly considered the use of Indium-Tin Oxide and other materials as the sensing material, but shied away from their use because of toxic properties or the high temperatures required to cure them.

Figure 1: PressureMouse: four foam sensors at the back of the mouse and two on each side.

After a literature review, we learned that force sensitive resistors are composed of a conductive elastomer of some sort and electrodes. As force is applied to the elastomer or foam, it becomes more dense and thus more conductive. As a result, to make more sensitive force-sensitive material, what is needed is a conductive foam that compresses under light loads.

The anti-static foam that is used to package electronic components works well as a conductive elastomer for light loads. The new pressure sensors for the PressureMouse are constructed based this kind of foam (figure 2).

Figure 2: PressureMouse tactile sensors.

These sensors have a greater dynamic range because the foam compresses under light loads. Unfortunately, they are not very elastic, meaning that after being loaded, they take a small amount of time to decompress. The unprocessed mouse data is 8 dimensions of 8-bit analog data captured at 60 Hz.

These sensors, applied to several points of a mouse, allow us to determine if the user is touching the mouse or not, and how hard the user is touching the mouse. This construction was designed to test the hypothesis that users would apply different patterns of pressure to the mouse under episodes of frustration or stress vs. under normal use. If this hypothesis is true, then the mouse may potentially provide a form of passive frustration sensing, without requiring conscious manipulation from the user. Here we assume a user will change the way he or she holds the mouse when encountering a frustrating situation, which is confirmed in our experiments as shown in Section 5.

4. EXPERIMENT DESIGN: FRUSTRATING COMPUTER-USERS

In our experiments, a subject is asked to fill out a multiple-page web form to put their resume online at a job site. They are told that we will be asking them some questions about usability of the site after the task. A page of the web form is shown in figure 3.

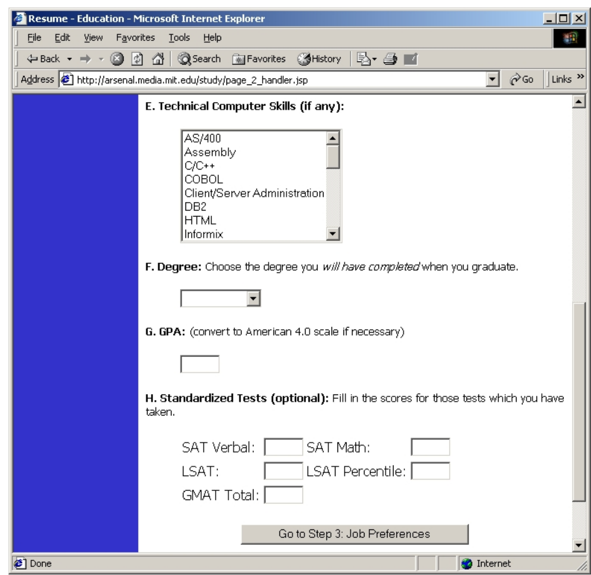

Figure 3: One page of the web form used in our experiments

In laboratory experiments or usability tests, subjects may not care enough about the task for it to matter to them if there is a problem with it. In order to increase the likelihood that usability bugs would induce frustration, time was made salient to the subjects by emphasizing that their time is very important, and that "most people completed the form in 15 minutes, and most MIT students completed it in 10 minutes." A timer set to 15 minutes was placed in view. Thus, anything that caused a delay might be more likely to frustrate the user. After filling out the second page of the web forms, which asks for the date of the user's degree, school where it was obtained, technical computer skills of the user, GPA, and (optional) standardized test scores, the user clicks to the next page. At that point an error message alerts the user that the date format is wrong. (Whatever format the user entered we can tell them they should have used another format). When the user goes back to the page to re-enter the date correctly, he or she sees that the page lost all of the

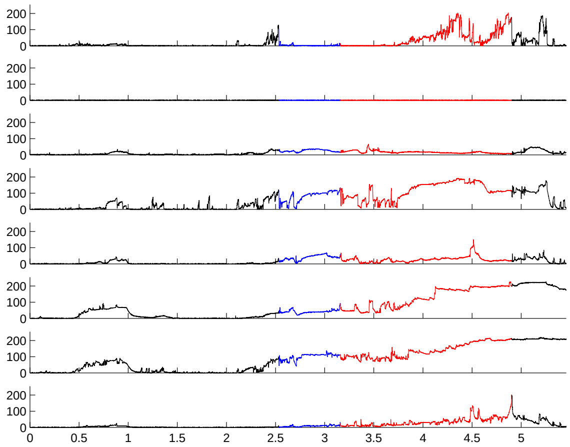

Figure 4: Mouse pressure signals of a subject. The unit of the horizontal axis is one minute. Blue denotes the first time filling out the web page with the date; red denotes the second time filling out the same page after the data loss. Black denotes other time periods, which are not used in this paper's analysis.

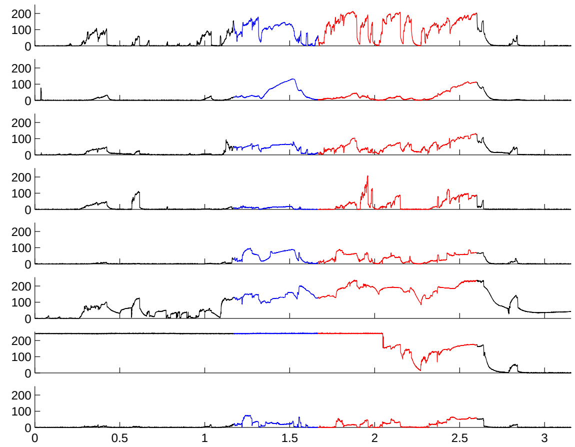

Figure 5: Mouse pressure signals of another subject, with red/blue/black coding as above.

data just entered. We expect that this event should mildly increase the user's frustration level, and we hypothesize that this would affect how they handle the mouse.

5. DATA ANALYSIS

We apply the Bayes Point Machine trained with Expectation Propagation to the classification of the pressure signals. We use two features, the mean and the variance of the pressure signals over a half-second window (i.e., 30 data points because the signals are sampled at 60 Hz as mentioned before).

As shown in figures 4 and 5, refilling the web form is strongly correlated with a pattern change in the pressure signals. Additionally, when we compare figures 4 and 5 we can see that different subjects exhibit different patterns of mouse use. For instance, the second channel shows no activity for the subject in figure 4, but the subject in figure 5 exerts pressure on this channel. Unfortunately, our prototype sensors do not perform well all the time. If we examine figure 5, we see the 7^{th} sensor does not work correctly; it 'flatlines' for the first part initial part of the data collection.

For the Bayes Point Machine, we choose the nonlinear basis expansion $\boldsymbol{\phi}(\mathbf{x})$ in equation (2) over the data points as follows:

$$\boldsymbol{\phi}(\mathbf{x}) = [K(\mathbf{x},\mathbf{x}_1), \cdots, K(\mathbf{x},\mathbf{x}_i), \cdots, K(\mathbf{x},\mathbf{x}_N)]^T$$

where $K(\mathbf{x},\mathbf{x}_i)$ is a basis function, which is chosen to be a Gaussian with variance being 0.1. Using the above feature expansion, we actually have $O(N^3)$ as the training time of Expectation Propagation, because here the feature dimensionality d equals the size of a training set N. The data is labeled as -1 during the first time filling the forms, and 1 during the second time filling the forms. The labeling is reasonably correlated with the subjects' frustration level, though it is not completely accurate. The data set for each subject is randomly split into a training set and a test set, in the ratio 55% : 45%. We test the Bayes Point Machine on five subjects. For comparison, we also test the Support Vector Machine on these data sets [6]. The Gaussian kernel with variance of 0.1 is used for SVM. The classification results are summarized in table 1. Running on a Pentium 3 computer, the average time for classifying a test data point in our matlab implementation is around 8 ms, which is fast enough to be used for real-time applications.

6. CONCLUSION AND FUTURE WORK

Trained with Expectation Propagation, the Bayes Point Machine achieves an average 11.87% person-dependent classification error rate, which outperforms the Support Vector Machine in our experiments. By combining the Bayes Point Machine with a PressureMouse, this frustration-detection system can be used for many real-time applications in human-computer interaction, for example:

1. Adaptive Interface Design: This PressureMouse sensor and detection algorithm can be used as a design tool for guiding the development of user interfaces, helping find and eliminate events that invoke users' frustration.

2. Building Reinforcement Learner: The device can be used to provide the reward function in a reinforcement learning process, associating punishment with excessive pressure applied to the mouse.

As to the future work, we aim to build person-independent classifiers for the mouse pressure signals. Also, we are working on additional algorithms and sensors that integrate multiple modes of data in an effort to provide increasingly confident and robust estimates of users' expressions.

7. REFERENCES

[1] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Uncertainty in AI*, volume 11, 1998.

Table 1: Test performance of EP classifiers on five computer-users' frustration detection. For comparison, we also list the test performance of SVMs here. In the table, EP-BPM denotes the Bayes Point Machine trained by Expectation Propagation.

Test error rate (%)	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Average Error
SVM	4.72	38.89	8.85	12.00	9.21	14.73
EP-BPM	3.94	35.56	6.19	9.71	3.95	11.87

[2] R. Herbrich, T. Graepel, and C. Campbell. Bayes point machine: Estimating the Bayes point in kernel space. In *IJCAI Workshop SVMs*, pages 23–27, 1999.

[3] H. Kushner and A. Budhiraja. A nonlinear filtering algorithm based on an approximation of the conditional distribution. *IEEE Transaction Automatic Control*, 45:580–585, 2000.

[4] T. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT, Jan. 2001. www.media.mit.edu/~tpminka/papers/learning.html.

[5] C. Reynolds. the sensing and measurement of frustration with computers. Master's thesis, MIT, 2001.

[6] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.