Mixture of Gaussian Processes for Combining Multiple Modalities

Ashish Kapoor, Hyungil Ahn, and Rosalind W. Picard

MIT Media Lab, Cambridge MA 02139, USA,
{kapoor, hiahn, picard}@media.mit.edu,
WWW home page: http://affect.media.mit.edu

Abstract. This paper describes a unified approach, based on Gaussian Processes, for achieving sensor fusion under the problematic conditions of missing channels and noisy labels. Under the proposed approach, Gaussian Processes generate separate class labels corresponding to each individual modality. The final classification is based upon a hidden random variable, which probabilistically combines the sensors. Given both labeled and test data, the inference on unknown variables, parameters and class labels for the test data is performed using the variational bound and Expectation Propagation. We apply this method to the challenge of classifying a student's interest level using observations from the face and postures, together with information from the task the students are performing. Classification with the proposed new approach achieves accuracy of over 83%, significantly outperforming the classification using individual modalities and other common classifier combination schemes.

1 Introduction

There are a growing number of scenarios in pattern recognition where multimodal information is used, and where information from multiple sensors needs to be fused to recover the variable of interest. Multi-sensor classification is a problem that has been addressed previously by using either data-level fusion or classifier combination schemes. In the former, a single classifier is trained on joint features; however, when the data has even one missing channel, a frequent problem, then usually all the data is ignored for that time block, resulting in a significant reduction in the total amount of data for training. One way to address this problem is by training a classifier for each modality that is present, and then combining these for a final decision.

The problem becomes even more challenging when there is labeling noise; that is, some data points have incorrect labels. In many computer vision and HCI applications like emotion recognition, there is always an uncertainty about the true labels of the data; thus, requiring a principled approach to handle any labeling noise in the data.

The highly challenging problem we address in this paper combines the three problems described above: there is multi-sensory data, channels are frequently missing and there might be labeling errors in the data.

We address this challenging problem in a Bayesian framework using a combination of Expectation Propagation [9] and variational approximate inference [1]. The framework utilizes a mixture of Gaussian Processes, where the classification using each channel is learned via Expectation Propagation, a technique for approximate Bayesian inference. The resulting posterior over each classification function is a product of Gaussians and can be updated very quickly. We evaluate the multi-sensor classification scheme on the task of detecting the affective state of interest in children trying to solve a puzzle, combining sensory information from the face, the postures and the state of the puzzle task, to infer the student's state. The proposed unified approach achieves a significantly better recognition accuracy than classification based on individual channels and the standard classifier combination methods. Also, on the affect data set we found that the standard classifier combination rules, which are justified using the probability theory, work better when the individual classifiers are probabilistic (as in the Gaussian Process classification) as opposed to the SVM.

1.1 Previous Work

There are many methods, including Boosting [12] and Bagging [2], which generate an ensemble of classifiers by choosing different samples from the training set. These methods require a common set of training data, which is a set of joint vectors formed by stacking the features extracted from all the modalities into one big vector. As mentioned earlier, often in multi-sensor fusion problems the training data has missing channels and labels; thus most of the data cannot be used to form a common set of training data. Similarly, most of the data remains unused in "feature-level fusion," where a single classifier is trained on joint features.

Kittler et al. [7] have described a common framework for combining classifiers and provided theoretical justification for using simple operators such as majority vote, sum, product, maximum and minimum. Hong and Jain [4] have used a similar framework to fuse multiple modalities for personal identification. Similarly, Han and Bhanu [3] also perform rule-based fusion for gait-based human recognition. One problem with these fixed rules is that, it is difficult to predict which rule would perform best. Then there are methods, such as layered HMMs proposed by Oliver et al. [10], which perform decision fusion and sensor selection depending upon utility functions and stacked classifiers. One main disadvantage of using stacked based classification is that these methods require a large amount of labeled training data. There are other mixture-of-experts [5] and critic-driven approaches [8] where base-level classifiers (experts) are combined using second level classifiers (critics or gating functions) that predict how well an expert is going to perform on the current input. To make a classifier selection, the critic can either look at the current input or base its decision upon some other contextual features as well. For example, Toyama and Horvitz [13] demonstrate a head tracking system based on multiple algorithms, that uses contextual features as reliability indicators for the different tracking algorithms. The framework described by us in this paper is also based on sensor-selection and is most similar

to Tresp [14], where the mixture of Gaussian Processes is described. The key differences include classification based on Gaussian Process rather than regression; also, we use Expectation Propagation for Gaussian Process classification and our classification likelihood is robust to labeling errors and noise. Our framework is also capable of quickly re-learning the classification given updated label associations. Further, we provide a complete Bayesian treatment of the problem rather than using a maximum-likelihood training.

2 Our Approach

Figure 1 shows the model we follow to solve the problem. In the figure, the data $\mathbf{x}^p$ from P different sensors generate soft class labels y. The switching variable λ, determines modalities that finally decide the hard class label $t \in \{1, -1\}$. In section 2.1, we first review classification using Gaussian Process (GP). Section 2.2 then extends the idea to a Mixture of Gaussian Processes and describes how to handle multiple modalities in the same Bayesian framework.

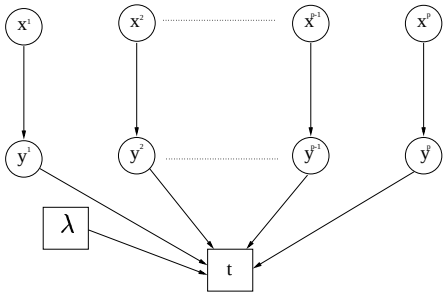

Fig. 1. A mixture of Gaussian Processes for p sensors

2.1 Gaussian Process Classification

Assume we are given a set of labeled data points $\mathbf{X} = \{\mathbf{x}_1, .., \mathbf{x}_n\}$, with class labels $\mathbf{t} = \{t_1, .., t_n\}$. For two-way classification, the labels are, $t \in \{-1, 1\}$. Under the Bayesian framework, given an unlabeled point $\mathbf{x}^*$, we are interested in the distribution $p(t^*|\mathbf{X}, \mathbf{t}, \mathbf{x}^*)$. Here t^* is a random variable denoting the class label for the point $\mathbf{x}^*$. Although, in this paper we only describe how to classify one new point, all the machinery described applies as well to a set of new points without any additional computational overhead.

The idea behind GP classification is that the hard labels $\mathbf{t}$ depend upon hidden soft-labels $\mathbf{y} = \{y_1, ..., y_n\}$. These hidden soft-labels arise due to application of a function f directly on the input data points (i.e. $y_i = f(x_i)\ \forall i \in [1..n]$). Further, we assume a Gaussian Process prior on the function f; thus, the results $\mathbf{y}$ of the evaluation of the function f on any number of input data points $\mathbf{x}$ are jointly Gaussian. Further, the covariance between two outputs y_i and y_j can be specified using a kernel function applied to $\mathbf{x}_i$ and $\mathbf{x}_j$. Formally, $\{y_1, .., y_n\} \sim N(0, K)$ where K is a n-by-n kernel matrix with $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.

The observed labels $\mathbf{t}$ are assumed to be conditionally independent given the soft labels $\mathbf{y}$ and each t_i depends upon y_i through the conditional distribution:

$$p(t_i|y_i) = \epsilon + (1 - 2\epsilon)\Phi(y_i \cdot t_i)$$

Here, ϵ is the labeling error rate and $\Phi(z) = \int_{-\infty}^{z} N(z; 0, 1)$. Very similar likelihoods have been previously used for Gaussian Process classification [11] and

Bayes-point machines [9]. The above described likelihood explicitly models the labeling error rate; thus, the model should be more robust to label noise.

Our task is then to infer $p(t^*|D)$, where $D = \{\mathbf{X}, \mathbf{t}, \mathbf{x}^*\}$. Specifically:

$$p(t^*|D) = p(t^*|\mathbf{X}, \mathbf{t}, \mathbf{x}^*) \propto \int_{\mathbf{y}, y^*} p(t^*|\mathbf{y}, y^*) p(\mathbf{y}, y^*|\mathbf{X}, \mathbf{t}, \mathbf{x}^*) \tag{1}$$

Where the posterior $p(\mathbf{y}, y^*|\mathbf{X}, \mathbf{t}, \mathbf{x}^*)$ can be written as:

$$p(\mathbf{y}, y^*|\mathbf{X}, \mathbf{t}, \mathbf{x}^*) = p(\mathbf{y}, y^*|D) \propto p(\mathbf{y}, y^*|\mathbf{X}, \mathbf{x}^*) p(\mathbf{t}|\mathbf{y})$$

The term $p(\mathbf{y}, y^*|\mathbf{X}, \mathbf{x}^*) \sim N(0, K)$ is the GP prior and it enforces a smoothness constraint. The second term, $p(\mathbf{t}|\mathbf{y})$ incorporates information provided in the labels. In the frameworks described here, $p(\mathbf{y}, y^*|D)$ is approximated as a Gaussian distribution using Expectation Propagation (EP), a technique for approximate Bayesian inference [9]. Assuming conditional independence of labels given the soft-labels, $p(\mathbf{t}|\mathbf{y})$ can be written as:

$$p(\mathbf{t}|\mathbf{y}) = \prod_{i=1}^{n} p(t_i|y_i) = \prod_{i=1}^{n} [\epsilon + (1 - 2\epsilon)\Phi(y_i \cdot t_i)]$$

The idea behind using EP is to approximate $P(\mathbf{y}, y^*|D)$ as a Gaussian. Although the prior $p(\mathbf{y}, y^*|\mathbf{X}, \mathbf{x}^*)$ is a Gaussian distribution, the exact posterior is not a Gaussian due to the form of $p(\mathbf{t}|\mathbf{y})$. Nonetheless, we can use EP to approximate the posterior as a Gaussian. Specifically, the method approximates the terms $p(t_i|y_i)$ as:

$$p(t_i|y_i) \approx \tilde{t}_i = s_i \exp(-\frac{1}{2v_i}(y_i \cdot t_i - m_i)^2) \tag{2}$$

EP starts with the GP prior N(0,K) and incorporates all the approximate terms $\tilde{t}_i$ to approximate the posterior $p(\mathbf{y}, y^*|D) = N(\mathbf{M}, \mathbf{V})$ as a Gaussian. For details readers are encouraged to look at [9]. To classify the test point $\mathbf{x}^*$, the approximate distribution $p(y^*|D) \approx N(M^*, V^*)$ can be obtained by marginalizing $p(\mathbf{y}, y^*|D)$ and then equation 1 can be used:

$$p(t^*|D) \propto \int_{y^*} p(t^*|y^*) N(M^*, V^*) = \epsilon + (1 - 2\epsilon)\Phi(\frac{M^* \cdot t^*}{\sqrt{(1 + V^*)}}) \tag{3}$$

2.2 Mixture of Gaussian Processes for Sensor Fusion

Given n data points $\bar{\mathbf{x}}_1, .., \bar{\mathbf{x}}_n$, obtained from P different sensors, our approach follows a mixture of Gaussian Processes model described in figure 1. Let every i^{th} data point be represented as $\bar{\mathbf{x}}_i = \{\mathbf{x}_i^{(1)}, .., \mathbf{x}_i^{(P)}\}$, and the soft labels as $\bar{\mathbf{y}}_i = \{y_i^{(1)}, .., y_i^{(P)}\}$. Given $\lambda_i \in \{1, .., P\}$, the random variable that determines the combination of the channels for the final classification, the classification likelihood can be written as:

$$P(t_i|\bar{\mathbf{y}}_i, \lambda_i = j) = P(t_i|y_i^{(j)}) = \epsilon + (1 - 2\epsilon)\Phi(t_i \cdot y_i^{(j)})$$

Given $\{\bar{\mathbf{X}}, \mathbf{t}\}$ and $\bar{\mathbf{x}}^*$

Step 1: Initialization

-For all the labeled points $i = 1$ to n do

· Initialize $Q(\lambda_i)$ using uniform distribution

-For all the modalities $p = 1$ to P do

· Incorporate all the labeled data points to obtain a Gaussian posterior for the soft labels:

$$p^0(\mathbf{y}^{(p)}) = N(\mathbf{y}^{(p)}; \mathbf{M}_{\mathbf{y}^{(p)}}, \mathbf{V}_{\mathbf{y}^{(p)}})$$

· Initialize: $Q(\mathbf{y}^{(p)}) = p^0(\mathbf{y}^{(p)})$

Step 2: Variational Updates

-Repeat until change in posteriors is less than some small threshold

· Update $Q(\mathbf{\Lambda})$ using equation 6.

· Update $Q(\bar{\mathbf{Y}})$ using equation 7.

Step 3: Classifying Test Data

-Compute $\hat{\mathbf{\Lambda}} = \arg\max_{\mathbf{\Lambda}} Q(\mathbf{\Lambda})$

-Use P-way classification to get the posterior $Q(\lambda^*)$

-Estimate $p(t^*|\bar{\mathbf{X}}, \mathbf{t})$ using equation 9

Fig. 2. Summary of the algorithm to classify the test data point using a mixture of Gaussian Processes. This algorithm can be readily extended to more than one test points without any computational overhead.

Given a test point $\bar{\mathbf{x}}^*$, let $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}_1, .., \bar{\mathbf{x}}_n, \bar{\mathbf{x}}^*\}$ denote all the training and the test points. Further, let $\bar{\mathbf{Y}} = \{\mathbf{y}^{(1)}, .., \mathbf{y}^{(P)}\}$, denote the hidden soft labels corresponding to each channel of all the data including the test point. Let, $Q(\bar{\mathbf{Y}}) = \prod_{p=1}^{P} Q(\mathbf{y}^{(p)})$ and $Q(\mathbf{\Lambda}) = \prod_{i=1}^{n} Q(\lambda_i)$, denote the approximate posterior over the hidden variables $\bar{\mathbf{Y}}$ and $\mathbf{\Lambda}$, where $\mathbf{\Lambda} = \{\lambda_1, .., \lambda_n\}$ are the switches corresponding only to the n labeled data points. Let $p(\bar{\mathbf{Y}})$ and $p(\mathbf{\Lambda})$ be the priors with $p(\bar{\mathbf{Y}}) = \prod_{p=1}^{P} p(\mathbf{y}^{(p)})$, the product of GP priors and $p(\mathbf{\Lambda})$ uniform. Given $\bar{\mathbf{X}}$ and the labels $\mathbf{t}$, our algorithm iteratively optimizes the variational bound:

$$F = \int_{\bar{\mathbf{Y}}, \mathbf{\Lambda}} Q(\bar{\mathbf{Y}}) Q(\mathbf{\Lambda}) \log\left(\frac{p(\bar{\mathbf{Y}}) p(\mathbf{\Lambda}) p(\mathbf{t}|\bar{\mathbf{X}}, \bar{\mathbf{Y}}, \mathbf{\Lambda})}{Q(\bar{\mathbf{Y}}) Q(\mathbf{\Lambda})}\right) \tag{4}$$

The classification using EP is required only once, irrespective of the number of iterations. In each iteration to optimize the bound given in equation 4, the classification rules are updated using the Gaussian approximations provided by EP. The algorithm is shown in figure 2 and can be divided into 3 steps: initialization, optimization and classification, which are described below.

Step 1: Initialization: In the first step, the approximate posterior $Q(\bar{\mathbf{Y}})Q(\mathbf{\Lambda}) = \prod_{p=1}^{P} Q(\mathbf{y}^{(p)}) \prod_{i=1}^{n} Q(\lambda_i)$ is initialized. Here, $Q(\lambda_i)$ are multinomial distributions and are initialized randomly using a uniform distribution. $Q(\mathbf{y}^{(p)})$ are normal distributions and to initialize them, we first use EP as described in section 2.1, considering all the data points irrespective of the state of the switches. EP results in the approximate Gaussian posteriors $p^0(\mathbf{y}^{(p)}) = N(\mathbf{y}^{(p)}; \mathbf{M}_{\mathbf{y}^{(p)}}, \mathbf{V}_{\mathbf{y}^{(p)}})$ for all $p \in \{1, .., P\}$, which are used to initialize $Q(\mathbf{y}^{(p)})$.A very useful bi-product of EP is the Gaussian approximations of the likelihoods, which would later be used to update our classification during the variational iterations in step 2.

Step 2: Optimization: The bound given in equation 4 is optimized by iteratively updating $Q(\bar{\mathbf{Y}})$ and $Q(\mathbf{\Lambda})$. Given the approximations $Q^k(\mathbf{\Lambda})$ and $Q^k(\bar{\mathbf{Y}})$ from the k^{th} iteration, $Q^{k+1}(\mathbf{\Lambda})$ and $Q^{k+1}(\bar{\mathbf{Y}})$ can be updated using variational updated rules [1]. Specifically, update rules for $Q(\lambda_i)$ and $Q(\mathbf{y}^{(p)})$ are as follows:

$$Q^{k+1}(\lambda_i) \propto \exp\{\int_{\bar{\mathbf{Y}}} Q^k(\bar{\mathbf{Y}}) \log p(t_i|\bar{\mathbf{Y}}, \lambda_i)\}$$

$$Q^{k+1}(\mathbf{y}^{(p)}) \propto \exp\{\int_{\mathbf{\Lambda}} Q^k(\mathbf{\Lambda}) \log p(\mathbf{y}^{(p)}) p(\mathbf{t}|\mathbf{y}^{(p)}, \mathbf{\Lambda})\}$$

The update for $Q(\lambda_i = p)$ can be written as:

$$Q^{k+1}(\lambda_i = p) \propto \exp\{\int_{y_i^{(p)}} Q^k(y_i^{(p)}) \log p(t_i|y_i^{(p)})\} \tag{5}$$

$$= \exp\{\int_{y_i^{(p)}} Q^k(y_i^{(p)}) \log(\epsilon + (1-2\epsilon)\Phi(t_i y_i^{(p)}))\} \tag{6}$$

Equation 6 is intractable but can be computed efficiently by importance sampling using the 1-D Gaussian $Q^k(y_i^p)$ as a proposal distribution. Further, we have the Gaussian approximations from EP for the likelihood term $p(t_i|y_i^{(p)}) \approx s_i^{(p)} \exp(-\frac{1}{2v_i^{(p)}}(y_i^{(p)} \cdot t_i - m_i^{(p)})^2)$. It can be shown that the update rule for $Q(\mathbf{y}^{(p)})$ reduces down to:

$$Q^{k+1}(\mathbf{y}^{(p)}) \propto p(\mathbf{y}^{(p)}) \prod_{i=1}^{n} N(y_i^{(p)}; m_i^{(p)} \cdot t_i, \frac{v_i^{(p)}}{Q^k(\lambda_i)}) \tag{7}$$

This is just a product of Gaussian terms; thus, there is no need to rerun the EP to estimate the new posterior over soft classifications. Further, note that $Q(\lambda_i)$ divides the variance, hence controlling the contribution of each labeled data point for different channels.

Step 3: Classification: In the final step, given the posterior over the switches, $Q(\lambda_i)$ $\forall i \in [1..n]$, we first infer the switches for the test data $\bar{\mathbf{x}}^*$. For this, we do a P-way classification using the GP algorithm described in 2.1 with $\hat{\mathbf{\Lambda}} = \arg\max_{\mathbf{\Lambda}} Q(\mathbf{\Lambda})$ as labels. Specifically, for an unlabeled point $\bar{\mathbf{x}}^*$, P different classifications are done where each classification provides us with q_r^*, where $r \in \{1, .., P\}$, and equals to the probability that channel r was chosen to classify $\bar{\mathbf{x}}^*$. The posterior $Q(\lambda^* = r)$ is then set to $\frac{q_r^*}{\sum_{p=1}^{P} q_p^*}$. In our experiments, for each of these P classifications, we clubbed all the channels together using -1 as observations for the modalities that were missing. Note, that we are not limited to using all the channels clubbed together; but, various combinations of the modalities can be used including other indicator and contextual variables.

Once we have the posterior over the switch for the test data, $Q(\lambda^*)$, we can infer class probability of an unlabeled data point $\bar{\mathbf{x}}^*$ using:

$$p(t^*|\bar{\mathbf{X}}, \mathbf{t}) = \int_{\bar{\mathbf{Y}}, \lambda^*} p(t^*|\bar{\mathbf{Y}}, \lambda^*) Q(\lambda^*) Q(\bar{\mathbf{Y}}) \tag{8}$$

$$= \sum_{p=1}^{P} Q(\lambda^* = p)(\epsilon + (1-2\epsilon)\Phi(\frac{M_{y^{(p)}}^* \cdot t^*}{\sqrt{1 + V_{y^{(p)}}^*}})) \tag{9}$$

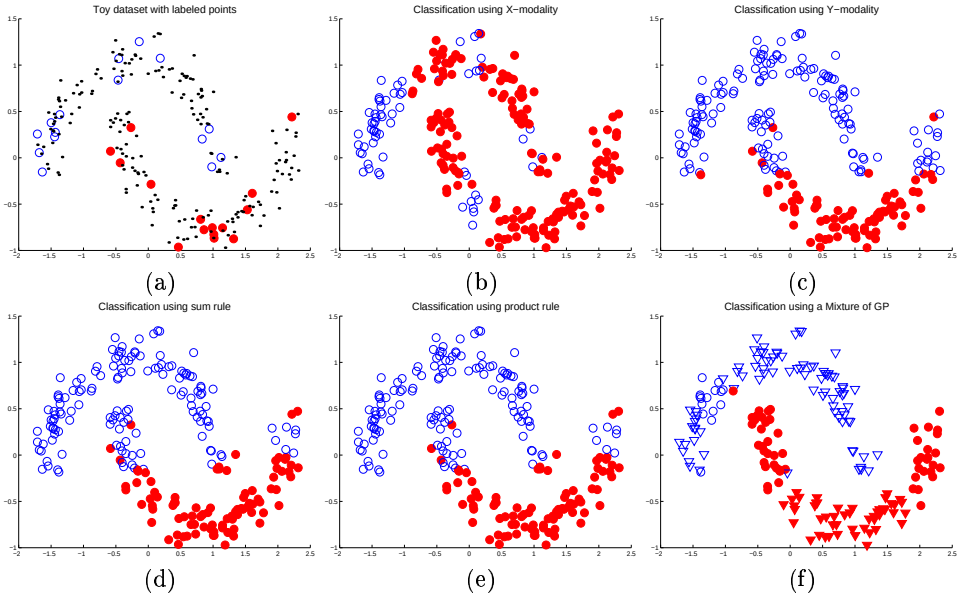

Fig. 3. (a) Toy dataset with the labeled points highlighted, and classification results using (b) X-modality only, (c) Y-modality only, (d) sum rule, (e) product rule and (f) the mixture of GP. The circles in (f) represent points classified with a greater weight on the X-modality and the triangles with a greater weight on the Y-modality.

Here, $M^*_{y^{(p)}}$ and $V^*_{y^{(p)}}$ are the mean and the variance of the marginal Gaussian approximation for p^{th} channel corresponding to the hidden soft label $\bar{\mathbf{y}}^*$.

3 Experiments and Results

We first demonstrate the features of the approach on a toy dataset and then apply it to the task of affect recognition using multiple modalities. We also evaluate the performance of other classifier combination schemes by training SVMs and the GP classifiers on the complete data. These standard classifier combination schemes are shown in Table 1.

Toy Dataset: A toy dataset is shown in figure 3(a), which has been previously introduced by Zhou et al. [15]. The top and the bottom half moon correspond to two different classes. The example shown in the figure has 15 labeled points from each class (30 total) and 100 test points (200 total). First, we perform two GP classifications using the method described in 2.1; one classifies the test points by just using the X-modality (dimension) and the other just using the Y-modality (dimension). Figures 3(b) & (c) show the results of these classifications using each individual modality, which is fairly poor. Figure 3(d) & (e) show classification using the sum and the product rule applied using the result of X and the Y classification. Finally, figure 3(f) shows successful classification

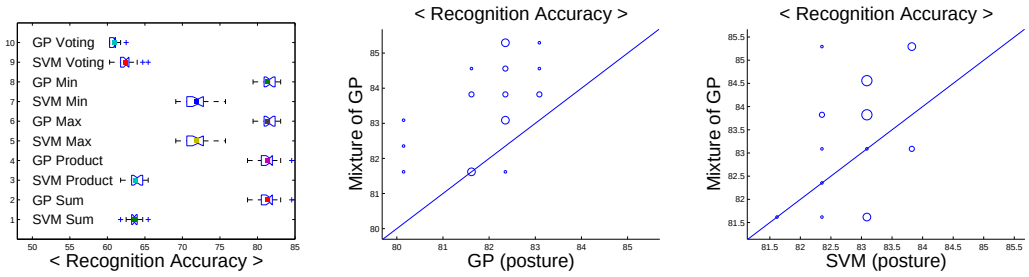

Fig. 4. (a) MATLAB boxplots comparing the standard classifier combination methods for GP and SVM on the affect data. The squares represent the mean, the lines in the middle of the box represents the median, the bounding box represent quartile values and the '+' symbols represent the statistical outliers. (b) Recognition rates of mix of GP vs. GP(posture) and (c) Mix of GP vs. SVM(posture) for the 24 runs. Each point is (accuracy SVM/GP (posture), accuracy mix of GP). Points over the lines correspond to the trials when mix of GP had the better recognition rate. The circle radii represent repeating results; the larger the circle the more the repetition of the points.

using the mixture of GP framework. In figure 3(f) the data points drawn as triangles were classified with a greater weight on the Y modality and the data points drawn as circles with a greater weight on the X-modality. We can see from the figure, that the final classification decision adapts itself according to the input space; thus, demonstrating the capability to perform sensor selection.

Recognizing Affect: We applied the mixture of GP framework to the problem of machine recognition of affect using multiple modalities. We look at the problem of detecting the affective state of interest in a child who is solving a puzzle on the computer. The training and the testing data consists of observations from three different channels: the face, posture and the puzzle activity. Every feature vector corresponding to a datapoint encodes the facial activity, posture activity and game information for a time segment of 8 secs [6]. The database includes 8 children and consists of 61 samples of high-interest, 59 samples of low-interest and 16 samples of refreshing. Only 49 samples had all three channels present. The other 87 samples had the face channel missing. In this paper, we only look at the binary problem of detecting the state of high-interest (61 samples) versus the states of low-interest and refreshing (75 samples).

We trained GP classifiers for each of the three channels using an RBF kernel to compute the similarity matrices for the GP priors with the kernel-width hyper-parameter σ fixed to 0.81, 7.47 and 5.90 for the face, the posture and the puzzle channel respectively. The value of ϵ was fixed at 0.42 for face, 0.0 for posture and 0.37 for the puzzle modality. These parameters were choosen using evidence maximization a standard approach within the Bayesian framework. We randomly selected 87.5% of the points as training data and computed the hyperparamters using evidence maximization. This process was repeated 10 times and the mean values of the hyperparameters were used in our experiments. The *P*-

Table 1. Classifier Combination Methods.

Rule	Criteria
Sum	$p(t=1\|\mathbf{x}^{(1)}..\mathbf{x}^{(P)}) \propto \sum_{p=1}^{P} p(t=1\|\mathbf{x}^{(p)})$
Product	$p(t=1\|\mathbf{x}^{(1)}..\mathbf{x}^{(P)}) \propto \prod_{p=1}^{P} p(t=1\|\mathbf{x}^{(p)})$
Max	$p(t=1\|\mathbf{x}^{(1)}..\mathbf{x}^{(P)}) \propto \max_p p(t=1\|\mathbf{x}^{(p)})$
Min	$p(t=1\|\mathbf{x}^{(1)}..\mathbf{x}^{(P)}) \propto \min_p p(t=1\|\mathbf{x}^{(p)})$
Vote	$p(t=1\|\mathbf{x}^{(1)}..\mathbf{x}^{(P)}) \propto \{ \begin{smallmatrix} 1 & \text{if } \sum_{p=1}^{P} \lceil p(t=1\|\mathbf{x}^{(p)}) \rceil \geq \lceil \frac{P}{2} \rceil \\ 0 & \text{otherwise} \end{smallmatrix} \}$

Table 2. Average recognition rates (standard deviation in parenthesis) for 24 runs on affect data.

	SVM	GP
Face	52.66%(1.4)	52.78%(0.7)
Posture	82.99%(0.6)	82.02%(0.9)
Puzzle	60.82%(1.5)	60.54%(0.9)
Sum	63.63%(0.9)	81.34%(1.2)
Prod	63.76%(0.9)	81.34%(1.2)
Max	71.94%(1.5)	81.37%(1.0)
Min	71.94%(1.5)	81.37%(1.0)
Vote	62.35%(1.2)	60.90%(0.6)
Mix of GP	NA	**83.55%(1.2)**

way classification for estimating the posterior over λ^* was also performed using an RBF kernel with kernel width set to 10.38.

We also evaluate the performance of SVM on this dataset. The SVMs were trained using an RBF kernel and the leave-one-out validation procedure was applied for selecting the penalty parameter C and the kernel width σ. The validation procedure was performed ten times, where each time 87.5% of datapoints were randomly choosen as training data. The mean of the resulting 10 parameters (σ, C) were finally choosen and were equal to (10.48, 1.49), (11.47, 1.33) and (10.66, 2.24) for the face, the posture and the puzzle modality respectively.

We performed 8-fold cross-validation to report the results. In every round the dataset was equally split into 8 parts. The algorithms were tested on every part with the other 7 parts (87.5% of data) used as the training set. Each of these rounds was repeated 24 times to report the results.

First, we compare the performance of standard classifier combination methods for GP based classification and SVMs. The GP classification provides class probabilities for the datapoints, which can directly be used in the standard classifier combination methods (table 1). The sigmoid function can be used to map an SVM output to a value between 0 and 1 and can be used to combine classifiers using the standard rules. There have been many other approaches suggested to convert the SVM output to a probability value and we leave the comparison of those as future work. Figure 4 shows the MATLAB boxplots and compares the performance of the different fixed classifier combination approaches for GP and SVM. The figure plots the mean, the median and quartile values. The figure shows that the GP based classifier combinations outperform the classifier combinations based on the probabilistic interpretation of the SVM output.

Further, table 2 shows the recognition results for each individual modality and many classifier combination rules. Among the individual modalities, the posture channel achieves the highest recognition both with the GP classification and the SVM. Further, it can be easily seen that the classification based on the posture modality outperforms the standard classifier combination rules. Since most of the discriminating information is contained in the posture channel, the standard classifier combination methods don't work well as they assign equal

importance to all the channels. The mixture of GP approach on the other hand is sensitive to this kind of information and thus can adapt to whichever channel works well. The scatter plots shown in the figures 4 (b) and (c) compares the performance of every single trial among the 24 runs of the mixture of GP approach vs SVM/GP classifiers trained on the posture modality. It can be seen clearly that the mixture of GP based approach outperforms the posture modality both when using SVM and GP classification and with table 2 we can see that it outperforms the standard classifier combination methods.

4 Conclusions and Future Work

In this paper, we proposed a unified approach using a mixture of Gaussian Processes for achieving sensor fusion under the challenging conditions of missing channels and noisy labels. We provide a Bayesian algorithm designed with a fast update of classification decisions based on variational and Gaussian approximations. On both a toy example, and on the task of classifying affective state of interest using information from face, postures and task information, the mixture of GP method outperforms several standard classifier combination schemes. Future work includes incorporation of active learning and application of this framework to other challenging problems with limited labeled data.

Acknowledgments

Thanks to Yuan (Alan) Qi for helpful comments. This research was supported by NSF ITR grant 0325428.

References

1. M. J. Beal, Variational Algorithms for Approximate Bayesian Inference, Ph.D. Thesis, University College London, 2003.
2. L. Breiman, Bagging Predictors, *Machine Learning*, Vol. 26(2), 1996.
3. J. Han, B. Bhanu, Statistical Feature Fusion for Gait-based Human Recognition, *CVPR*, 2004.
4. L. Hong, and A. K. Jain, Integrating Faces and Fingerprints for Personal Identification, *PAMI*, Vol. 20(1), 1998.
5. R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G .E. Hinton, Adaptive Mixtures of Local Experts, *Neural Computation*, Vol. 3, pp. 79-87, 1991.
6. A. Kapoor, R. W. Picard and Y. Ivanov, Probabilistic Combination of Classifiers to Detect Interest, *ICPR*, 2004.
7. J. Kittler, M. Hatef, R. P. W. Duin and J. Matas, On Combining Classifiers, *PAMI*, Vol. 20(3), 1998.
8. D. J. Miller and L. Yan, Critic-Driven Ensemble Classification, *Signal Processing*, Vol. 47(10), 1999.
9. T. Minka, Expectation Propagation for Approximate Bayesian Inference, *UAI*, 2001.
10. N. Oliver, A. Garg and E. Horvitz, Layered Representations for Learning and Inferring Office Activity from Multiple Sensory Channels, *ICMI*, 2002.
11. M. Opper and O. Winther, Mean field methods for Classification with Gaussian Processes, *NIPS*, Vol. 11, 1999.
12. R. Schapire, A Brief Introduction to Boosting, *International Conference on Algorithmic Learning Theory*, 1999.
13. K. Toyama and E. Horvitz, Bayesian Modality Fusion: Probabilistic Integration of Multiple Vision Algorithms for Head Tracking, *ACCV*, 2000.
14. V. Tresp, Mixture of Gaussian Processes, *NIPS*, Vol. 13, 2001.
15. D. Zhou, O. Bousquet, T. N. Lal, J. Weston and B. Scholkopf, Learning with Local and Global Consistency, *NIPS*, Vol. 16, 2004.