Local Gaussian Processes for Pose Recognition from Noisy Inputs

Martin Fergie
mfergie@cs.man.ac.uk
Aphrodite Galata
a.galata@cs.man.ac.uk

The University of Manchester,
Oxford Road,
Manchester,
M13 9PL, UK

Abstract

Gaussian processes have been widely used as a method for inferring the pose of articulated bodies directly from image data. While able to model complex non-linear functions, they are limited due to their inability to model multi-modality caused by ambiguities and varying noise in the data set. For this reason techniques employing mixtures of local Gaussian processes have been proposed to allow multi-modal functions to be predicted accurately [10]. These techniques rely on the calculation of nearest neighbours in the input space to make accurate predictions. However, this becomes a limiting factor when image features are noisy due to changing backgrounds. In this paper we propose a novel method that overcomes this limitation by learning a logistic regression model over the input space to select between the local Gaussian processes. Our proposed method is more robust to a noisy input space than a nearest neighbour approach and provides a better prior over each Gaussian process prediction. Results are demonstrated using synthetic and real data from a sign language data set and HumanEva [9].

1 Introduction

Inferring the poses of articulated bodies such as hands and people directly from images requires models which are able to handle complex high dimensional and non-linear data. Recently, discriminative models that learn a mapping directly from simple image features to the pose space have grown in popularity due to their fast inference [1, 4, 6, 10]. This mapping is non-linear, multi-modal and highly noisy due to image ambiguities and subject variations.

Gaussian processes provide a flexible framework for modelling non-linear noisy data and have been used for human pose estimation [11, 14]. They model a set of training inputs with a joint Gaussian distribution with zero mean and a covariance given by a covariance function $k(\mathbf{x},\mathbf{x}')$ where $\mathbf{x}$ are training inputs. Despite their power and flexibility, they suffer from a number of limitations. Gaussian processes give a predictive Gaussian distribution over the output space with a single mean and variance. This means that they are unable to handle multi-modalilty in the mapping from image features to the pose space. Similarly, they are commonly used with a stationary covariance function which is independent of translations in the input space. The leads to a model where the learnt signal noise is independent of the training inputs. As a result, prediction accuracy suffers when a data set contains regions

BMVC 2010 doi:10.5244/C.24.98

with varying amounts of noise, as the Gaussian process model will attempt to average these. Finally, learning complexity is $O(N^3)$ time and $O(N^2)$ storage which makes learning from large data sets impractical without employing sparsification techniques which affect prediction accuracy [7].

Urtasun and Darrell have recently proposed a model that addresses some of these limitations by employing multiple Gaussian processes, each modelling a local region of the data set [11]. In their method each local Gaussian process has it's own set of hyper parameters allowing multi-modality to be modelled. Learning is also possible on large data sets as time complexity is now $O(TS^3)$ for T local experts each of size S where $S << N$.

However, their method of choosing which local Gaussian process to use when inferring the predicted output given a test input requires calculating the nearest neighbours in the input space to find the Gaussian process trained on that region. When using such a model for pose estimation, the input space can contain large amounts of noise due to changing backgrounds and illumination. This means that the nearest neighbour calculation can be dominated by background information rather than the subject itself. Figure 1 shows a test image and it's nearest neighbours in the training set calculated using the Euclidean distance between a grid of SIFT descriptors [10] calculated for each image. This shows that images that are close in feature space are not necessarily close in pose space. As a result, predictions are made using Gaussian processes which are trained on other regions of the data set, leading to poor prediction accuracy.

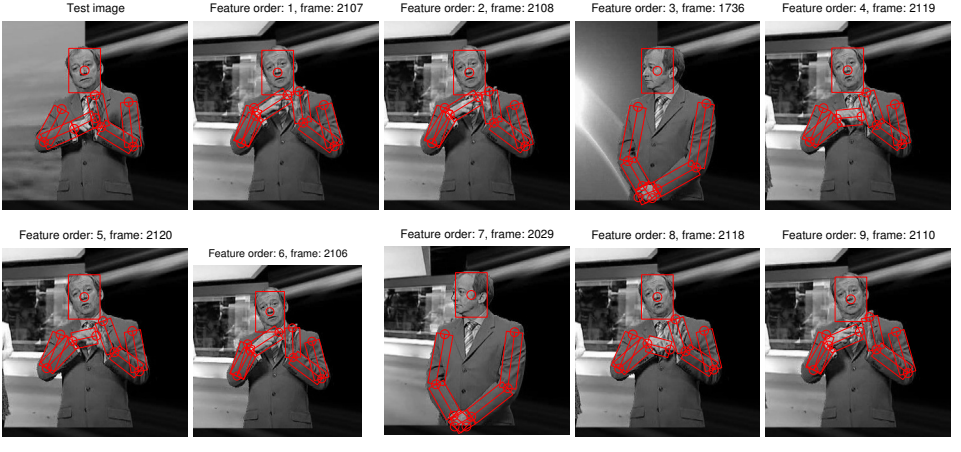

Figure 1: Visualisation of nearest neighbours in feature space with ground truth pose data shown in red. Top left image is a test image, and the remaining images are its nearest neighbours in the feature space calculated from an independent training set. The calculated nearest neighbours have very similar backgrounds but significant pose variation.

2 Local Gaussian Process Regression

A Gaussian process is a collection of random variables with a joint Gaussian distribution defined over them [13]. For a training set $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ and $\mathbf{Y} = \{\mathbf{y}_n\}_{n=1}^N$ consisting of inputs

$\mathbf{x}_i$ and noisy outputs $\mathbf{y}_i$ Gaussian processes model the function f such that

$$\mathbf{y}_i = f(\mathbf{x}_i) + \varepsilon_i \tag{1}$$

where $\varepsilon_i = \mathcal{N}(0, \sigma^2_{noise})$ and σ^2_{noise} is the noise variance.

Gaussian process regression is formulated as a Bayesian approach that assumes a Gaussian process as a prior distribution over the function values,

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(0, \mathbf{K}), \tag{2}$$

where $\mathbf{f} = [f_1, \ldots, f_N]^T$ is a vector of function output values $f_i = f(\mathbf{x}_i)$ and $\mathbf{K}$ is the covariance matrix given by covariance function $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. The covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$ can take many forms. In this paper we use a combination of a squared exponential function, a bias and a white noise term,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2_{signal} \exp\Big(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{P}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}{2} \Big) + \theta + \sigma^2_{noise} \delta_{ij}, \tag{3}$$

where the model hyper-parameters are $\Theta = \{\mathbf{P}, \sigma^2_{signal}, \theta, \sigma^2_{noise}\}$, $\mathbf{P}$ is a diagonal matrix of length scales $p_1 \ldots p_D$ and δ_{ij} is Kronecker's delta function. These hyper-parameters can be learnt automatically from the training set [13].

For an unseen point $\mathbf{x}_*$ and its associated unknown noisy output $\mathbf{y}_*$ the Gaussian prior placed over the function f with zero mean and covariance $k(\mathbf{x}_i, \mathbf{x}_j)$ leads to the distribution

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & k(\mathbf{x}_*, \mathbf{X}) \\ \mathbf{k}(\mathbf{x}_*, \mathbf{X})^T & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right) \tag{4}$$

where $k(\mathbf{x}_*, \mathbf{X})$ is the covariance between the test data and the training data. By conditioning this on the observed data, the predictive equations for the Gaussian process can be obtained

$$\mu(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{X}) \mathbf{K}^{-1} \mathbf{Y} \tag{5}$$

$$\sigma(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{X})^T \mathbf{K}^{-1} k(\mathbf{x}_*, \mathbf{X}), \tag{6}$$

where $\mu(\mathbf{x}_*)$ is the predicted output mean, $\sigma(\mathbf{x}_*)$ is the prediction variance [13].

Since the distribution over the test output $\mathbf{y}_*$ is Gaussian, any multi-modal regions of the data set are averaged and modelled as noise through the prediction variance σ^2_*. Similarly, since the noise hyper-parameters are independent of translations for the inputs, $\mathbf{x}$, all regions of the data set are modelled with the same amount of signal noise (σ^2_{noise} in eq. 3). Finally, the inversion of the covariance matrix $\mathbf{K}$ in equation 5 makes inference on large data sets unfeasible.

To overcome this, Urtasun and Darrell [11] proposed a model consisting of multiple Gaussian process models each acting as an expert for a region of the data set. Each expert is trained on a neighbourhood of training inputs which are local in pose space. This means that multi-modal regions of the data set will be modelled by placing a Gaussian process expert on each mode. A model with T experts each of size S is trained by selecting T training points as expert centres and learning the hyper-parameters to model its S nearest neighbours in the pose space. For an expert $i \in \{1, \ldots, T\}$ a training point n is selected as the expert centre and is trained using the S points nearest to $\mathbf{y}_n$. The expert centres can be chosen randomly or by using a clustering algorithm. The left plot of figure 2 illustrates how the training points of each expert are selected on a toy data set.

Prediction for a test point $\mathbf{x}_*$ is made by selecting the M nearest training points, $\eta_{\mathbf{x}_*} = \{\eta_i\}_{i=1}^{M}$, to the test point $\mathbf{x}_*$ in the input space. The value M is manually set to control the number of neighbouring training points that predictions are made from and is typically set to 10. A prediction for each η_i is made by selecting the nearest expert to η_i in the pose space, and using S nearest pose neighbours as the training set for that expert. These predictions are then combined as a Gaussian mixture distribution where the distribution over pose is given by

$$p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{Y},\Theta) = \sum_{i=1}^{M} \pi_i p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X}_{\zeta_i},\mathbf{Y}_{\zeta_i},\Theta) = \sum_{i=1}^{M} \pi_i \mathcal{N}(\mu_i(\mathbf{x}_*),\sigma_i(\mathbf{x}_*)), \tag{7}$$

where ζ_i are the nearest neighbours in pose space for η_i and π_i is a function of the inverse prediction variance. The prediction for each expert is now given by

$$\mu_i(\mathbf{x}_*) = k(\mathbf{x}_*,\mathbf{X}_{\zeta_i})\mathbf{K}_{\zeta_i}^{-1}\mathbf{Y}_{\zeta_i}, \tag{8}$$

$$\sigma_i(\mathbf{x}_*) = k(\mathbf{x}_*,\mathbf{x}_*) - k(\mathbf{x}_*,\mathbf{X}_{\zeta_i})^T\mathbf{K}_{\zeta_i}^{-1}k(\mathbf{x}_*,\mathbf{X}_{\zeta_i}). \tag{9}$$

Figure 2 shows the training points selected for each expert on the left, and the predictions made for a set of unique test points on the right. The model is able to learn a mapping over the multi-modal region of the data set that a single Gaussian process model would just average.

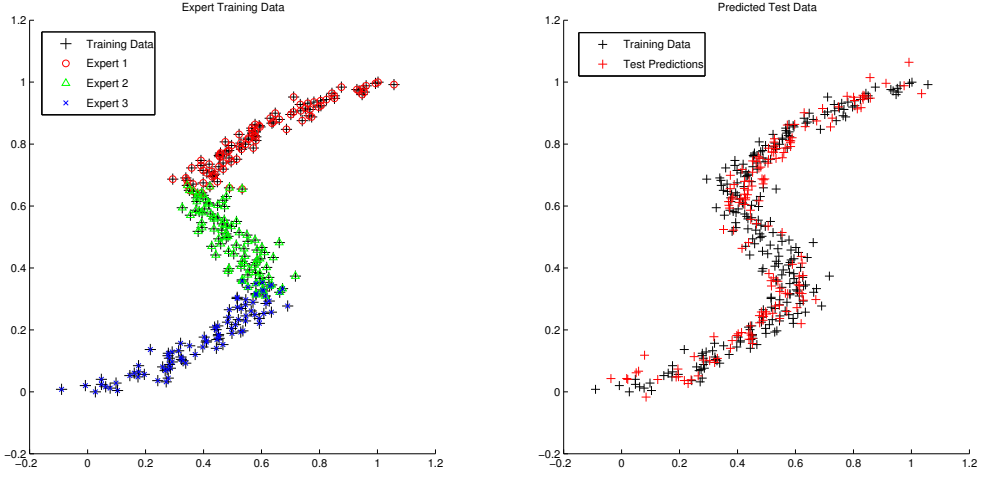

Figure 2: Local Gaussian process experts using method presented in [11]. The model attempts to learn the inverse of $x = y + 0.3sin(2\pi y) + \varepsilon$ where $\varepsilon = \mathcal{N}(0, 0.05^2)$ leading to a noisy multi-modal regression problem. The model is trained with 3 experts, each comprising of 90 training points. Expert centres are selected using k-means. The left plot shows the points selected for training each Gaussian process expert and the right plot shows the predicted values y_* for an independent test set.

The main limitation of Urtasun and Darrell's method is that it relies heavily on the calculation of the nearest neighbours $\eta_{\mathbf{x}_*}$ of each test point $\mathbf{x}_*$ in the input space. When estimating pose directly from images there is often significant noise in the input space caused by changing backgrounds and subject variations. This violates the assumption that points near in the

input space are near in the pose space and the calculation of $\eta_{\mathbf{x}_i}$ can be dominated by noise. Figure 3 shows the toy problem from figure 2 with extra dimensions of noise added to the input space. This causes Urtasun and Darrell's method to select the wrong experts for prediction resulting in inaccurate predictions, even in regions of the data set where there is only one mode.

Further, the reliance on using the inverse prediction variance as a prior over the predictions π_i from each expert is problematic since the prediction variance of a Gaussian process measures both signal noise and prediction uncertainty. Thus, an expert which is trained on a region of the data set which contains more noise will receive a lower prior over an expert trained on a region containing little noise. This gives an unfair bias to regions of the dataset which contain less noise in the prediction. This effect is seen in figure 2 where the middle region of the data modelled by expert 2 receives very few predictions. The expert modelling this region has a higher signal noise than the other experts and therefore its prior, π_i, is lower.

The following section explains how these effects can be overcome by using a logistic regression model to calculate π_i as opposed to the inverse variance.

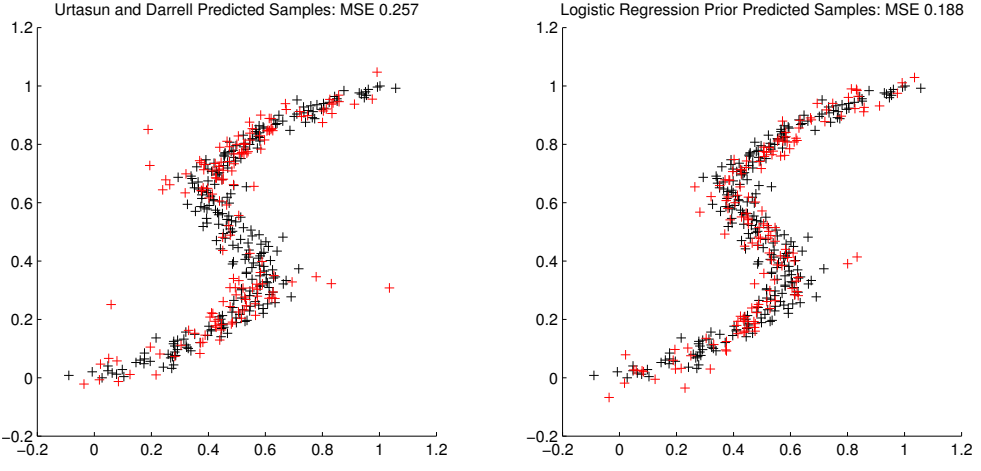

Figure 3: Toy problem as in figure 2 with noise added to the input space. Inputs X are appended with a vector sampled from a distribution $\mathcal{N}(0,10)$. The left panel shows the method presented in [11]. The noise has caused the predictions to be chosen from incorrect experts resulting in lower prediction accuracy. The right panel shows the same problem but prediction is performed using the method outlined in section 3. Mean squared error in the left panel is 0.254, and the right panel 0.188.

3 Expert Selection Using Logistic Regression

To alleviate the problems of relying on calculating nearest neighbours in noisy data, a more powerful model can be learnt to from select which experts predictions should be made. Taking inspiration from the mixtures of experts literature [3, 12], this paper employs a logistic regression model to provide a prior over expert predictions.

In a mixtures of experts model, a logistic regression model is used to select between a set of linear models playing a similar role to π_i in equation 7. Logistic regression models the probability of an input $\mathbf{x}$ belonging to a class c_i. The probability of the class conditioned on the input, $p(c_i|\mathbf{x})$, is given by

$$p(c_i|\mathbf{x}) = \frac{e^{(\mathbf{w}_i^T \mathbf{x})}}{\sum_{j=1}^{T} e^{(\mathbf{w}_j^T \mathbf{x})}}. \tag{10}$$

This is a linear combination of the inputs $\mathbf{x}$ and a vector of weights $\mathbf{w}_i$ which is then evaluated with the softmax function such that $p(c_i|\mathbf{x}) \in (0,1)$.

The weights $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^{T}$ are learnt using standard gradient optimisation techniques to find $\hat{\mathbf{W}}$ such that

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}}[L(\mathbf{W}) + \log p(\mathbf{W})], \tag{11}$$

where $L(\mathbf{W})$ is the log likelihood given by

$$L(\mathbf{W}) = \sum_{n=1}^{N} \sum_{i=1}^{T} \log p(c_{in}|\mathbf{x}_n, \mathbf{w}_i) \tag{12}$$

and $p(\mathbf{W})$ is the prior on the weights. The prior on the weights is chosen as a Laplacian prior such that $p(\mathbf{W}) \propto \exp(-\lambda ||\mathbf{W}||_1)$ where $||\mathbf{W}||_1 = \sum_{i=1}^{T} |\mathbf{w}_i|$ is the l_1 norm. Using a Laplacian prior promotes sparsity in the weights making the model robust to noise in the input space [5]. The influence of the prior is controlled by λ which is set using cross validation on the training set.

Training the local Gaussian process regression model is similar to [11] except the extra step of training the logistic regression model. The logistic regression model is trained by assigning a class label c_i to each training point (x_n, y_n) which indicates which expert that training point is closest to in pose space. Predictions for a test point $\mathbf{x}_*$ are now made by making a prediction from all individual experts and using $p(c_i|\mathbf{x}_*)$ as a prior on the predictions, replacing π_i in equation 7. The predictive distribution now becomes

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}, \Theta) = \sum_{i=1}^{T} p(c_i|\mathbf{x}_*) p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}_{\vartheta_i}, \mathbf{Y}_{\vartheta_i}, \Theta) = \sum_{i=1}^{T} p(c_i|\mathbf{x}_*) \mathcal{N}(\mu_i(\mathbf{x}_*), \sigma_i(\mathbf{x}_*)), \tag{13}$$

where $p(c_i|\mathbf{x}_*)$ is given by the logistic regression model, and T is the number of experts. Equations 8 and 9 for the predicted mean and variance for each expert no longer make predictions using the local neighbour hood ζ_i as no nearest neighbours are calculated. Instead, the training set that was used to train each expert ϑ_i is used. Thus, the predictive mean and variance for each expert now becomes

$$\mu_i(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{X}_{\vartheta_i}) \mathbf{K}_{\vartheta_i}^{-1} \mathbf{Y}_{\vartheta_i}, \tag{14}$$

$$\sigma_i(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{X}_{\vartheta_i})^T \mathbf{K}_{\vartheta_i}^{-1} k(\mathbf{x}_*, \mathbf{X}_{\vartheta_i}). \tag{15}$$

It should be noted that this method predicts from all T experts as opposed to making the M predictions in the method presented by Urtasun and Darrell [11]. In practice, this is in fact considerably faster as the kernel matrices $\mathbf{K}_{\vartheta_i}$ can be stored and reused as opposed to having to recalculate $\mathbf{K}_{\zeta_i}$ for each of the M nearest neighbours to the test point $\mathbf{x}_*$. It would be

simple to reduce the number of expert predictions made by simply choosing the M experts that maximise $p(c_i|\mathbf{x}_*)$.

Figure 3 shows the toy problem from figure 2 with extra dimensions of noise added to the input. The left panel shows predictions made on a test set using the method outlined in [11] and the right panel shows predictions made using the method presented in this paper. Urtasun and Darrell's method is no longer able to model the data accurately showing significant errors even in areas where there is one mode. Our method is able to cope better with the noisy input data showing less errors in the unimodal sections of the data set and giving a lower error overall. The middle section of the dataset is also represented better using our method as the prior given by $p(c_i|\mathbf{x}_*)$ is not dependent on the signal noise of the Gaussian process expert. Recall that π_i from equation 7 is a function of the inverse variance. This has the effect of giving a lower weighting to experts that model regions of the dataset where there is more noise in the outputs. The expert modelling the middle section of the dataset in figure 2 has higher noise hyperparameters (σ^2_{signal} and σ^2_{noise} in eq. 3) than the other two experts. As a result, fewer samples are drawn for the middle expert from the predictive distribution given in 7. This is shown in the left panel of figure 2. In comparison, the right panel of figure 3 using a logisitc regression prior does a much better job of representing the middle region.

4 Experimental Evaluation

4.1 Sign Language Data Set

We have used the sign language data set from [2] to compare the two methods outlined in this paper. The data set consists of 5910 frames of sign language captured from BBC television with manually annotated joint positions for the head and arms. A variety of image features are calculated for each frame and then the models outlined in this paper are used to learn a mapping directly from these features to the joint positions. The background is split into two halves, a static back drop and the images from the television program itself. As these images are constantly changing, the image features and regression model must be robust to this noise. The image features used are hierarchical features (HMAX) [8] and grids of SIFT descriptors [4]. The sequence has been split into 400 frame chunks, 10 chunks are randomly selected for training and the remaining 4 are used for testing. Dividing the data in this way allows behaviours from different regions of the sequence tested, but minimises the number of adjacent training and test frames.

Table 1 shows a comparison between the choice of prior for weighting the contribution from each expert while performing prediction. A prediction is made from all T experts and then different priors are used to evaluate $\mathbb{E}[p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}, \Theta)]$ for each test point. This tests the suitability of the different priors without relying on calculation of nearest neighbours in the input space. It shows that the inverse variance does not provide a suitable prior over predictions and gives higher errors than using a logistic regression prior. This is likely because the prediction variance of a Gaussian process models both signal noise and prediction uncertainty as stated in section 2. Therefore any noisy regions of the dataset will incorrectly given a lower weight in the final prediction.

Table 2 shows a direct comparison between Urtasun and Darrell's method [11] and the method outlined in this paper. These results show that our method is able to predict the pose with comparable or better accuracy than Urtasun and Darrell's method. However, it has the advantage of enabling prediction to be made considerably faster as the covariance matrices

Image Feature	Inverse Variance	Logistic Regression
HMAX	13.64	8.73
SIFT Grid	27.8623	8.90

Table 1: Comparison of prior calculation on sign language data set. Prediction is made from all T experts as in equation 13 and then different priors are used to weight each prediction. Inverse variance is π_i from equation 7 and logistic regression uses $p(c_i|\mathbf{x}_*)$ from equation 13. Errors are mean Euclidean distance per pixel calculated between the ground truth and $\mathbb{E}[p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{Y},\Theta)]$.

Image Feature	Urtasun and Darrell [11]	Logistic Regression
HMAX	9.09	8.73
SIFT Grid	10.67	8.90

Table 2: Comparison of Urtasun and Darrell's method [11] predicted using equation 7 and the method outlined in this paper predicted using equation 13 on sign language data set. Errors are mean Euclidean distance per pixel calculated between the ground truth and $\mathbb{E}[p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{Y},\Theta)]$.

used in prediction do not need to be recalculated for each test point. To compute predictions for the 1600 frames in the sign language dataset, our method took 5 minutes and [11] took 38 minutes. A sequence of frames showing pose estimation are included in figure 4.

4.2 HumanEva Data Set

We have also evaluated our method using the HumanEva data set [9]. We have taken 4800 frames of Subject 1 performing Walking, Boxing and Jogging actions taken from 3 cameras. The data has been divided into 200 frame chunks, 19 chunks are randomly selected for training, the remaining 5 are used for testing. Using images from multiple cameras introduces changes in the image background which would otherwise be static. We make monocular predictions of 3D joint positions from a single camera angle by rotating the joint positions into each camera's coordinate frame, then subtracting the root of the skeleton. This results in a joint representation that is independent of camera angle and global position. Hierarchical features (HMAX) from [8] are used as they do not require an accurate bounding box around the subject.

The results are shown in table 3. Our method offers an improvement in accuracy over Urtasun and Darrell's method.

Urtasun and Darrell [11]	Logistic Regression
58.37	53.65

Table 3: Comparison between Urtasun and Darrell's method and our method on the HumanEva data set as described in section 4.2. Errors are mean Euclidean distance per pixel calculated between the ground truth and $\mathbb{E}[p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{Y},\Theta)]$

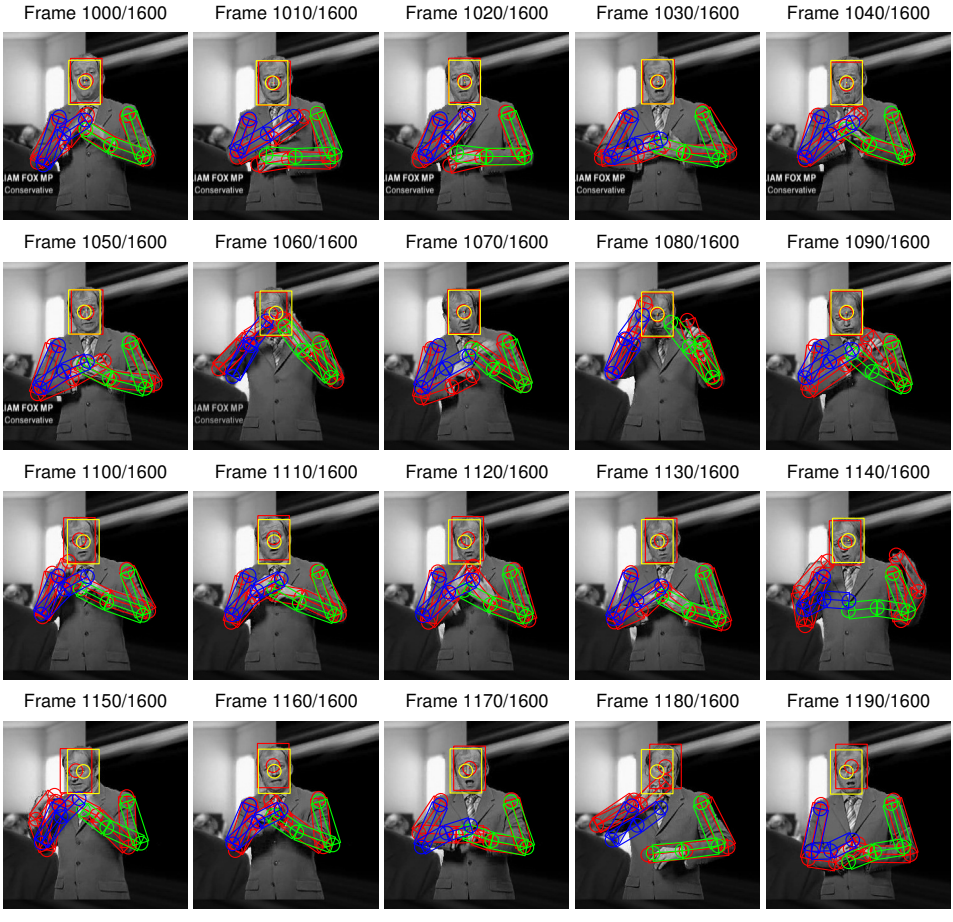

Figure 4: Results from our proposed method on the sign language data set. Ground truth information is in red, tracking results are in blue green and yellow.

5 Discussion

In this paper we have proposed a method for performing accurate regression for noisy, multi-modal and non-linear data. Our approach uses a mixture of Gaussian processes which are selected using a logistic regression model over the input space. We have shown that our method can achieve accurate regression on both real and synthetic data sets and overcomes some of the limitations of previous methods.

Specifically, by replacing the prior over predictions in [] with a logisitic regression model we have shown that more accurate regression can be achieved. By removing the reliance on calculating nearest neighbours in the input space at test time, our method is also more robust to dealing with noisy features caused by changing backgrounds and subject variations.

References

[1] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):44–58, Jan. 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.21.

[2] P. Buehler, MR Everingham, DP Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language TV broadcasts. In *Proceedings of the 19th British Machine Vision Conference*, pages 1105–1114. BMVA Press, 2008.

[3] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2):181–214, 1994.

[4] A. Kanaujia, C. Sminchisescu, and D. Metaxas. Semi-supervised hierarchical models for 3d human pose reconstruction. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383341.

[5] B. Krishnapuram, L. Carin, MAT Figueiredo, and AJ Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):957–968, 2005.

[6] Huazhong Ning, Wei Xu, Yihong Gong, and T. Huang. Discriminative learning of visual words for 3d human pose estimation. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. ISSN 1063-6919. doi: 10.1109/CVPR.2008.4587534.

[7] J. Quiñonero-Candela and C.E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

[8] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2:994–1000 vol. 2, June 2005. ISSN 1063-6919. doi: 10.1109/CVPR.2005.254.

[9] L. Sigal and M.J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown Univertsity TR*, 2006.

[10] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Learning joint top-down and bottom-up processes for 3d visual inference. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2:1743–1752, 2006. ISSN 1063-6919. doi: 10.1109/CVPR.2006.169.

[11] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. ISSN 1063-6919. doi: 10.1109/CVPR.2008.4587360.

[12] S. Waterhouse, D. MacKay, and T. Robinson. Bayesian methods for mixtures of experts. In *Advances in Neural Information Processing Systems 8: Proceedings of the 1995 Conference*, page 351. MIT Press, 1996.

[13] CKI Williams and CE Rasmussen. Gaussian Processes for Regression. *Advances in Neural Information Processing Systems*, 8:514–520, 1996.

[14] Xu Zhao, Huazhong Ning, Yuncai Liu, and T. Huang. Discriminative estimation of 3d human pose using gaussian processes. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec. 2008. doi: 10.1109/ICPR.2008.4761707.