Learning Ambiguities using Bayesian Mixture of Experts

Atul Kanaujia
Department of Computer Science
Rutgers University
Piscataway, NJ 08854
kanaujia@cs.rutgers.edu

Dimitris Metaxas
Department of Computer Science
Rutgers University
Piscataway, NJ 08854
dnm@cs.rutgers.edu

Abstract

Mixture of Experts(ME) is an ensemble of function approximators that fit the clustered data set locally rather than globally. ME provides a useful tool to learn multi-valued mappings(ambiguities) in the data set. Mixture of Experts training involve learning a multi-category classifier for the gates distribution and fitting a regressor within each of the clusters. The learning of ME is based on divide and conquer which is known to increase the error due to variance. In order to avoid overfitting several researchers have proposed using linear experts. However in the absence of any knowledge of non-linearities existing in the data set, it is not clear how many linear experts could accurately model the data.

In this work we propose a bayesian learning framework for learning Mixture of Experts. Bayesian learning intrinsically embodies regularization and model selection using Occam's razor. In the past Bayesian learning methods have been applied to classification and regression in order to avoid scale sensitivity and orthodox model selection procedure of cross validation. Although true Bayesian learning is computationally intractable, approximations do result in sparser and more compact models.

1 Introduction

Mixture of Experts, originally proposed by Jacob et al.[3], provide a modular learning framework that involve multiple function approximators, combined using a multi-category classifier. The original EM algorithm for learning ME maximizes the likelihood by decoupling the learning process into regressor fitting and multi-category classification tasks. Maximum Likelihood(ML) based learning methods typically lead to models with high variance and overfitting. For regression problems ML underestimates the noise level.

The learning in ME is essentially a divide and conquer strategy to train on complex non-linear datasets by breaking it into multiple clusters and fitting surfaces within each of the clusters. The statistical consequence of such a technique is favorable on error due to bias but tends to increase the error due to variance. A simplistic approach to avoid overfitting is to fit linear surfaces. However in the absence of any information about the intrinsic non-linearities existing in the data set, it is difficult to decide on the number of linear experts that can accurately model the data set.

Bayesian model selection offers an elegant solution to avoid overfitting. Waterhouse et. al[12] proposed a bayesian framework to learn mixture of experts with gates obtained using Laplace's approximation. The gate distribution forms an integral component of ME and it is not clear how well the gates are learned on multi-valued mappings. Bishop et al.[1] proposed a generic variational learning framework by optimizing a well defined variational lower bound on the marginal evidence. The algorithm is only suitable for logistic gates and cannot be extended to multi-category gating nodes. They use hierarchical binary tree based structure to learn complex division of input space using binary splits. However no comparison between the original mixture of experts in terms of model complexity and results have been provided.

An important component of the Mixture of Expert implementation is the gate distribution which learns the prior conditional to classify an input $\mathbf{x}$ to the expert cluster. Jordan et. al [3] proposed a double loop EM algorithm for learning the gate distribution as a softmax function using Iterative Reweighted Least Square(IRLS) algorithm. In order to avoid double loop EM, Xu et al.[13] formulated gate distributions as a joint distribution over the constant expert weights and the conditional prior. The alternative method used weighted gaussian distribution for the prior and could be solved analytically.

In this work we propose Mixture of Experts learned using Bayesian theory of model selection and regularization[5]. The experts are learned as Non-linear kernel basis function approximators and the gating network

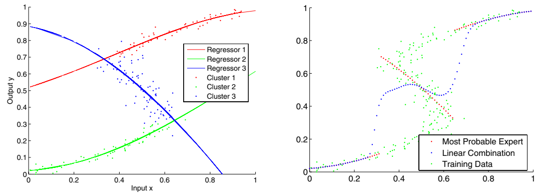

Figure 1. (Left) Clustering of the training data(Right) Prediction using the most probable expert and the linear combination of the experts.

is learned as a sparse bayesian multi-category classifier. The supervised learning framework is formulated as posterior maximization using the regularized Expectation Maximization procedure[6]. The improvement in the prediction accuracy and the model complexity are adequately illustrated using empirical evaluation on the low dimensional synthetic toy data sets. The inverse perspective projection mapping from the 2D image to the 3D human pose is intrinsically multi-valued. Bayesian Mixture of Experts(ME) provides an improved mechanism to learn these multi-valued mappings as 3D state conditionals. The predictions are done based on gate distributions which are learnt as a multi-category classifier. We demonstrate the algorithm on reconstruction of 3D articulated human pose from the 2D image silhouette features.

2 Bayesian Mixture of Experts

Mixture of Experts training involves learning the experts and the gates distribution. The gate distribution g_i is a multi-category classifier to cluster the dataset into several classes. The experts E_i are the regressors that fit each of the clusters locally. Jordan et al. [3] proposed an Expectation Maximization(EM) algorithm based on likelihood maximization for learning the gates g_i and the experts E_i. Given a set of observed variables, $\mathbf{D}$, the EM algorithm tries to estimate the unknown variables $\mathbf{Z}$ by maximizing the expectation of the augmented likelihood $\ell_c(\Omega : \mathbf{D}, \mathbf{Z})$. The unknown variables $\mathbf{Z}$ may denote the class to which observed variables belong. The E-Step consists of estimation of the expected value of the complete likelihood $\ell_c(\Omega : \mathbf{D}, \mathbf{Z})$ using expected value of the hidden variables $\mathbf{Z}$ i.e. $Q(\Omega, \Omega^k) = E\,[\log\,\{\ell_c(\Omega : \mathbf{Z})\}|\,\mathbf{D}, \Omega^k]$. In the M step, the parameters Ω^{k+1} are estimated by maximizing the complete likelihood $\ell_c(\Omega : \mathbf{D}, \mathbf{Z})$ i.e. $\Omega^{(k+1)} = \text{arg } max_\Omega Q(\Omega, \Omega^k)$. Increase in the expected complete likelihood Q implies increase in the incomplete likelihood $\ell(\Omega : D)$. The standard EM algorithm has slow rate of convergence around the maxima and also tends to overfit the data. We propose a learning framework for ME using *regularized EM*[4] algorithm that maximizes the penalized likelihood for estimating the experts and the gate parameters. The standard EM algorithm is an instance of generic class of algorithms called *Proximal Point Algorithms*. Originally these algorithms were introduced by Martinet[6] and Rockafeller[8] for solving objective function with convex constraints. A proximal point algorithm is defined by the iteration in M Step as:

$$\Omega^{(k+1)} = \text{arg } max_\Omega\{\ell_c(\Omega) - \psi_k d(\Omega, \Omega^{(t)})\} \qquad (1)$$

where ℓ_c is the objective function to be maximized and $d(\Omega, \Omega^{(t)})$ is the penalizing function that satisfies the condition $d(\Omega, \Omega^{(t)}) \geq 0$. ψ_k are iteration dependent positive parameters. It can be shown that the objective function $\ell_c(\Omega)$ increases with the iterative M-Step (1) and the EM iterations converges at faster rate compared to the standard EM algorithm [6, 8]. Proximal Point algorithms(PP) constrains the search of parameters Ω in the proximity of previous iteration parameters $\Omega^{(k)}$ thereby yielding a stable iterative algorithm that is robust to divergence due to improper initialization. In our formulation of regularized EM, we maximize posterior distribution instead of the complete data likelihood. This leads to a special case of proximal point algorithm that promotes *sparsity* of weights at every EM iteration of the learning process. In further analysis we assume M experts to be learned on a data set $(\mathbf{t}, \mathbf{x}) = \{(t_1, x_1), (t_2, x_2), \cdots, (t_N, x_N)\}$. The hidden variables $\mathbf{z_i} = \{z_{i1}, \cdots, z_{iM}\}$ are the binary indicator variables that classifies the data point x_i to one of the cluster.

2.1 Experts and Gates Likelihood

We assume the standard gaussian noise model for M expert regressors with the i^{th} expert E_i expressed as kernel basis interpolant $t_n = \{\sum_{j=1}^{N} \phi_i(x, x_j) * \theta_{ij} + \theta_{i0}\} + \mathcal{N}(0, \sigma^2) = \theta_{\mathbf{i}}^{\mathbf{T}} * \mathbf{\Phi}(\mathbf{x})$ where $\mathbf{\Theta} = \{\theta_{\mathbf{1}}, \cdots, \theta_{\mathbf{M}}\}$ are the weight vectors for each of the experts E_i. The augmented likelihood $P(t_n, \mathbf{z_n}|x_n, \Theta, \Lambda) = \prod_{i=1}^{M}\{P(t_n|z_{ni}, x_n, \theta_i, \sigma^2)P(z_{ni}|x_n, \lambda_i)\}^{z_{ni}}$ where $\mathbf{z_n} = \{z_{n1}, \cdots, z_{nM}\}$ are the hidden indicator variables representing the hard clustering of the data point x_n to the M clusters. $P(z_{ni}|x_n, \lambda_i)$ is the gating distribution that assigns a class to each input x_n. The expert likelihoods are gaussian distribution $P(t_n|z_{ni}, x_n, \theta_i, \sigma^2) \propto \exp\{-\frac{\|t_n - \Phi(x_n)^T\theta_i\|^2}{\sigma^2}\}$ with variance as σ^2.

2.2 Bayesian Multi-Category Classification

Gate distribution forms an important component of Mixture of Experts(ME) and is implemented as a

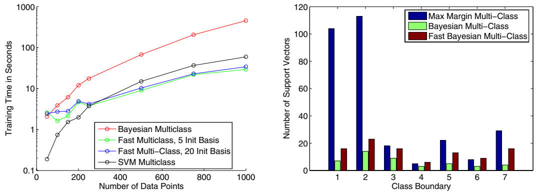

Figure 2. (Left) Training Time on log scale in seconds. (Right) Sparsity of the trained models

multi-category classifier. The multi-category classification learning can be formulated as marginal evidence maximization [10, 7] problem. Likelihood is formulated as a multinomial distribution $P(\mathbf{D}|\mathbf{\Lambda}) = P(z_{ni}|x_n, \lambda_i) = \prod_{k=1}^{M} \prod_{n=1}^{N} \rho_k \{\mathbf{f}(\mathbf{x_n})\}^{z_{nk}}$ for M classes and N observed data pairs $(x_n, \mathbf{z_n})$ with canonical link function as $\rho_j\{\mathbf{f}(\mathbf{x_n})\} = e^{-f_j(x)}/\sum_i^M e^{-f_i(x)}$ where $f_i(x) = \sum_n^N \lambda_{n,i}\Phi(x, x_n) = \lambda_\mathbf{i}^\mathbf{T}\mathbf{\Phi}(\mathbf{x})$, is the kernel basis interpolant at N training points. $\mathbf{\Lambda} = \{\lambda_\mathbf{1}, \lambda_\mathbf{2}, \cdots, \lambda_\mathbf{M}\}$ are the weight parameters for each class and $\mathbf{A} = \{\gamma_\mathbf{1}, \gamma_\mathbf{2}, \cdots, \gamma_\mathbf{M}\}$ are the scale parameters for the weight priors.

Assuming independent weight priors for classes, bayesian learning proceeds by formulating the log posterior distribution $\mathbf{log}\{P(\mathbf{\Lambda}|\mathbf{D}, \mathbf{A})\} = \sum_{k=1}^{M} \sum_{n=1}^{N} z_{nk} \log\{\rho_k\{\mathbf{f}(\mathbf{x_n})\}\} - (\sum_{k=1}^{M} \lambda_\mathbf{k}^\mathbf{T} \gamma_\mathbf{k} \lambda_\mathbf{k})$ where $\gamma_\mathbf{k} = diag(\gamma_{k1}, \gamma_{k2}, \cdots, \gamma_{kN})$ are the individual prior scale parameters for each class k and N training basis vectors. We use Laplace's approximation to estimate the posterior distribution as a gaussian distribution with strong peak at $\mathbf{\Lambda_{MP}} = \{\lambda_\mathbf{1,MP}, \cdots, \lambda_\mathbf{M,MP}\}$

$$P(\mathbf{\Lambda}|\mathbf{D}, \mathbf{A}) \simeq \left\{\prod_{k=1}^{M} P(\lambda_\mathbf{k,MP}|\mathbf{D}, \gamma_\mathbf{k})\right\}$$

$$\exp\left\{\sum_{k=1}^{M} -\frac{1}{2}(\lambda_\mathbf{k} - \lambda_\mathbf{k,MP})^\mathbf{T}\mathbf{C}_\mathbf{k}^{-\mathbf{1}}(\lambda_\mathbf{k} - \lambda_\mathbf{k,MP})\right\} \quad (2)$$

The covariance matrices $\mathbf{C_k}$ are evaluated as hessian of log-posterior of class k. The block diagonal covariance matrix $\mathbf{C}$ for the joint posterior $P(\mathbf{\Lambda}|\mathbf{D}, \mathbf{A})$ is approximated as $\mathbf{diag}\{\mathbf{C_1}, \mathbf{C_2}, \cdots, \mathbf{C_M}\}$ for M classes. Therefore we can factorize the complex multi-variate gaussian (2) into independent gaussians for every class. The posterior distribution is centered around $\mathbf{\Lambda_{MP}}$ which occurs at most probable parameters $\lambda_\mathbf{k,MP}$ for each class k. Maximizing (2) with respect to $\gamma_\mathbf{k}$ gives a closed form update rule for each class k that can be used to estimate weights $\mathbf{\Lambda_{k,MP}}$ and the scale parameters $\gamma_\mathbf{k}$

Fast Online Multi-Category Classification

The computational speed of the Multi-Category classification can be substantially improved by using a bottom up approach (as opposed to pruning based, top-down approach) by adding (or updating) a pool of basis vectors. The algorithm is initiated with randomly selected basis vectors. Multiple passes over the entire training set can be used to add or delete the basis vectors, until the marginal evidence undergoes no change [2]. The contribution of an individual training point towards the marginal likelihood can be computed by decomposing the covariance matrix $\mathbf{C_k}$ of (2) for each class k[2, 1] as

$$\mathbf{C_k} = \mathbf{C_{k,-i}} + \gamma_{i,k}^{-1}\phi_{i,k}\phi_{i,k}^T \quad (3)$$

which expresses the covariance $\mathbf{C_k}$ as the sum of contribution from the individual basis vector $\phi_{i,k}$ and the rest of the model $\mathbf{C_{k,-i}}$. The decomposition yields [2, 1] an augmentation rule for adding new basis vectors to the pool of basis vectors for maximizing the marginal evidence of the hyper-parameters. The algorithm makes a single pass and updates the classifier as it adds, deletes or updates pool of basis vectors. Although suboptimal, the algorithm drastically improves training time without degrading the prediction accuracy. However there is decline in the sparsity of the model. Fig. 2 compares the computational time and the sparsity of the generated models obtained from different algorithms. We compare the results with the multi-category SVM [11]. The classifiers were test on a sample data set containing 7 classes and varying number of points. Training time for multi-category classification algorithm is almost 15 times that of online training using 5 and 20 initial basis vector.

2.3 Regularized Expectation Maximization Algorithm

Learning Mixture of Experts involves estimation of the weight parameters θ_i and λ_i for the experts E_i and the gate distribution g respectively.

Log Posterior: Assuming independent prior distributions with quadratic weight decay, $\log\{P(\Theta)\} = -\sum_{i=1}^{M} \theta_i^T \alpha_i \theta_i$ and $\log\{P(\Lambda)\} = -\sum_{i=1}^{M} \lambda_i^T \gamma_i \lambda_i$ where α_i and γ_i are the hyper-parameters corresponding to the experts and the gate classifier. The log posterior is approximated around the modes of the hyper-parameters:

$$P(\Theta, \Lambda|t_n, x_n, \mathbf{z_n}) \simeq P(\Theta, \Lambda|t_n, x_n, \mathbf{z_n}, \alpha_\mathbf{MP}, \sigma_\mathbf{MP}^\mathbf{2}, \gamma_\mathbf{MP}) \quad (4)$$

The modes of the hyper-parameter distributions are obtained by maximizing the marginal evidence. Assuming independent posterior distributions for hyper-parameters and uniform gamma hyper-priors [5]

$$\{\boldsymbol{\alpha}_\mathbf{MP}, \sigma_\mathbf{MP}^\mathbf{2}, \gamma_\mathbf{MP}\} = \arg max_{\boldsymbol{\alpha}, \sigma^2, \gamma} P(\boldsymbol{\alpha}, \sigma^2, \gamma|\mathbf{t}, \mathbf{x}, \mathbf{z})$$

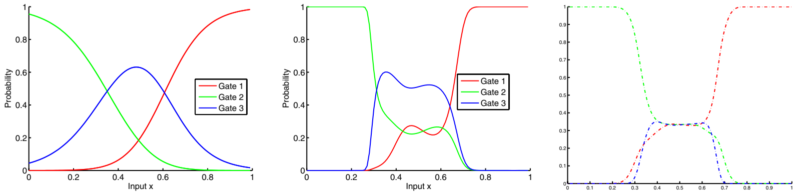

Figure 3. (Left) Gate Distribution learned as softmax function using IRLS.(Middle) Gate Distribution learned as log kernel Bayesian Multi-Category classifier. (Right) The analytical gate distributions

$$\propto \arg max_{\boldsymbol{\alpha},\sigma^2,\gamma} P(\mathbf{t}|\mathbf{x},\mathbf{z},\boldsymbol{\alpha},\sigma^2)P(\mathbf{z}|\mathbf{x},\boldsymbol{\gamma})P(\boldsymbol{\alpha})P(\sigma^2)P(\boldsymbol{\gamma})$$

This yields evidence maximization learning for pruning weights and estimating the full posterior for $\{\Theta, \Lambda\}$. The log posterior is estimated $\mathbf{log}\{P(\Theta, \Lambda|t_n, x_n, \mathbf{z_n})\} \propto P(\Theta|t_n, x_n, \mathbf{z_n})P(\Lambda|\mathbf{z_n}, x_n)$ where $P(\Theta|t_n, x_n, \mathbf{z_n}) \propto$

$$\sum_{i=0}^{M} E[z_{ni}]\text{log}\{P(t_n|z_{ni}, x_n, \theta_i)\} - \sum_{i=1}^{M} \theta_i^T \alpha_{i,MP}\theta_i \quad (5)$$

and $P(\Lambda|\mathbf{z_n}, x_n) \propto$

$$\sum_{i=0}^{M} E[z_{ni}]\text{log}\{P(z_{ni}|x_n, \lambda_i)\} - \sum_{i=1}^{M} \lambda_i^T \gamma_{i,MP}\lambda_i \quad (6)$$

The hyper-parameters $\{\boldsymbol{\alpha}_{MP}, \sigma^2_{MP}, \gamma_{\mathbf{MP}}\}$ are estimated using *type II Maximum Likelihood*. The estimated hyper-paramters are used to prune out weights and generate sparse models at every EM iteration. For the M-Step in the EM iteration we maximize the log posterior instead of the augmented log likelihood. The M Step is equivalent to maximizing the penalized likelihood for a *Proximal Point Iteration* (1) by searching in a region around **0** weights and penalizing non-zero weights. The conditions required for the super-linear convergence are $\alpha_i > 0$ and $\beta_i > 0$ and the quadratic penalties $\theta_i^T\alpha_i\theta_i > 0$, $\lambda_i^T\gamma_i\lambda_i > 0$. These conditions are always satisfied during the EM iteration loop.

Expectation Step: The expectation step involves computing the expected values (denoted as $E[x]$) of the hidden variable.

$$E[z_{ni}] = P(z_{ni} = 1|x_n, t_n, \theta_i, \lambda_i) =$$
$$\frac{P(t_n|x_n, z_{ni} = 1, \theta_i) * P(z_{ni} = 1|x_n, \lambda_i)}{\sum_{m=1}^{M} P(t_n|x_n, z_{mn} = 1, \theta_i) * P(z_{mn} = 1|x_n, \lambda_i)} \quad (7)$$

$E[z_{ni}]$ denotes the soft classification of a point (t_n, x_n) to the i^{th} cluster.

Maximization Step: The maximization step estimates θ_i and λ_i for the expert E_i and the gate distributions g_i respectively, by maximizing the log posterior separately for the experts(5) and the gates(6). The soft clustering computed in the E-step are used to train a multi-category classifier for the gate distribution. A regressor is fitted within each of the cluster obtained from the gate distribution.

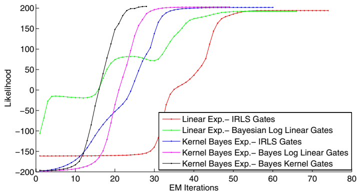

Figure 4. Convergence rates for various implementations of ME

Bayesian Update for Experts: Maximizing (5) is a weighted generalized least square problem and can be solved by reweighting the data terms (t_n, x_n) with $(\sqrt{E[z_{in}]}t_n, \sqrt{E[z_{in}]}x_n)$ and fitting a kernel regressor using *type II Maximum Likelihood*. The weights are pruned by thresholding the scale hyper-parameters α obtained from evidence(marginal likelihood) maximization [5, 10].

Bayesian Update for Gates: Maximizing (6) has the same analytical form as the weight posterior of the multi-category classification problem. The gates are updated by iterative evidence(marginal likelihood) maximization with $E[\mathbf{z_n}]$ as the target class for the input x_n. The learning framework sparsifies the regressors and the gates at each EM iteration using *type II Maximum Likelihood* learning. The Proximal Point step for every EM iteration yields a super-linear convergence rate[6]. We test our formulation on the toy data set where clear multi-valued mappings exist between the predictor and the target variable[1]. The toy data was generated by uniformly sampling 200 values of target variable t in range $\{0, 1\}$ and the predictor variable as $x = t + 0.3 * sin(2\pi t) + \mathcal{N}(0, 0.005)$. Fig. 1 shows the clustering of the training data using conditional posterior distribution obtained in the Expectation step(7). In Fig.4 the convergence rate of our algorithm with various implementations of ME are compared. We used ridge regressor for the Linear Experts and *Kernel Bayes Expert* used RBF kernel regressors trained in bayesian framework. *IRLS gates* denote softmax gates learned using Iterative Reweighted Least Square optimization. *Bayesian Log Linear Gates* denote softmax function learnt in Bayesian framework and *Bayesian Kernel Gates* denote kernel function with softmax link.

Fig.3*(Left,Middle)* compares the gate distributions. The gate distribution due to log kernel basis functions learn the class boundaries better in comparison with the softmax gates learned using IRLS([3]). Fig.3*(Right)* shows the analytically computed gates obtained as likelihood of the fitted regressors to the data set $(g(x_i) = exp\{-\frac{(t_i-(0.3*sin(2*pi*E_i(x))+E_i(x)))^2}{(2*(0.005))}\})$. Clearly the log kernel gates model the distribution more accurately compared to log linear gate distributions. Notice the tails and the sharp change on the cluster boundaries for the gate distribution learned using log kernel bayesian Multi-Category classifier.

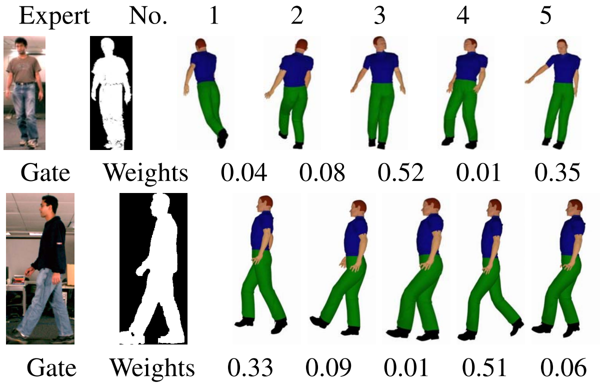
Expert No. 1 2 3 4 5
Gate Weights 0.04 0.08 0.52 0.01 0.35
Gate Weights 0.33 0.09 0.01 0.51 0.06

Figure 5. Predictions from the 5 experts on ambiguous human poses

3 Learning Inverse Prespective Projection

Learning complex inverse mappings require appropriate representations of ambiguities and probabilistic prediction based on spatial and temporal cues. The statistical model should be able to learn all possible ambiguous configurations for an observed input. Learning to reconstruct 3D human pose from 2D image silhouette[9] involves inferencing inverse perspective projection function which is intrinsically ambiguous. Loss of information due to the projection causes forward backward flip ambiguities due to non-observabilities of parts.

We demonstrate the use of Bayesian Mixture of Experts to learn multi-modalities of 3D pose reconstruction using 2D real image silhouette. We train BME on specific poses, using motion capture data, imported to realistically rendered MAYA(Alias Wavefront) model. The 3D pose is represented as 56 joint angles with no global translation. We use the trained model to predict 3D joint angles from real image silhouettes. The silhouettes for the real images were obtained using background subtraction. The shape context features are extracted from the outermost contour. We trained Bayesian ME on the database containing humans in forward and backward poses and sidewalking poses. These poses are difficult to infer from 2D image silhouettes due to forward-backward and legs ambiguities. Fig.5(**Top**) shows the outputs from the 5 experts for the subject facing the camera. 3D configurations predicted from different experts illustrate the ambiguities arising due to the 2D features. The gate weights denote the probabilites associated with each of the expert. Fig.5(**Bottom**) shows the leg ambiguities arising from the walking sequence viewed from the side.

4 Conclusion

In this work we have proposed a framework to learn ambiguities using sparse bayesian mixture of experts. The proposed model can be used with any number of experts and does not require hierarchical structure. We show empirical results to demonstrate the improvement in the sparsity and the representative power of the ME model.

References

[1] C. Bishop and M. Svensen. Bayesian hierarchical mixture of experts. *UAI*, 2003.

[2] A. Faul and M. Tipping. Analysis of sparse bayesian learning. *NIPS*, 2002.

[3] M. Jordan and R. Jacobs. Hierachical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.

[4] H. Li, K. Zhang, and T. Jiang. The regularized em algorithm. *AAAI*, 2005.

[5] D. MacKay. Bayesian interpolation. *Neural Computation*, 1991.

[6] B. Martinet. Regularization d'inequation variationelles par approximations successives. *Rev.Francaise d'Inform. et de Rech Operationnelle*, 1970.

[7] I. Nabney. Efficient training of rbf networks for classification. *IJNS*, 2004.

[8] R. Rockafeller. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 1976.

[9] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. *CVPR*, 2005.

[10] M. Tipping. Sparse bayesian learning and rvm. *JMLR*, 2001.

[11] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector learning for interdependent and structured output spaces. *ICML*, 2004.

[12] S. Waterhouse and D. MacKay. Bayesian methods for mixtures of experts. *NIPS*, 1996.

[13] L. Xu, M. Jordan, and G. Hinton. Alternative model for mixtures of experts. *NIPS*, 1995.