Shared Kernel Information Embedding for Discriminative Inference

Leonid Sigal Roland Memisevic David J. Fleet
Department of Computer Science, University of Toronto
{ls,roland,fleet}@cs.toronto.edu

Abstract

Latent Variable Models (LVM), like the Shared-GPLVM and the Spectral Latent Variable Model, help mitigate over-fitting when learning discriminative methods from small or moderately sized training sets. Nevertheless, existing methods suffer from several problems: 1) complexity; 2) the lack of explicit mappings to and from the latent space; 3) an inability to cope with multi-modality; and 4) the lack of a well-defined density over the latent space. We propose a LVM called the Shared Kernel Information Embedding (sKIE). It defines a coherent density over a latent space and multiple input/output spaces (e.g., image features and poses), and it is easy to condition on a latent state, or on combinations of the input/output states. Learning is quadratic, and it works well on small datasets. With datasets too large to learn a coherent global model, one can use sKIE to learn local online models. sKIE permits missing data during inference, and partially labelled data during learning. We use sKIE for human pose inference.

1. Introduction

Many computer vision problems are amenable to learning some form of mapping from image observations to a 3D object model. Examples include articulated human pose inference [2, 1, 6, 8, 9, 13, 16, 20, 21], articulated pose and shape estimation [18], and hand pose estimation [5]. With such *discriminative methods*, given a set of training samples comprising image features, $\mathbf{x}$, and 3D poses, $\mathbf{y}$, i.e., $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$, the estimation of pose, $\mathbf{y}$, is viewed as a form of 'regression'. This can be formulated in terms of learning the conditional distribution, $p(\mathbf{y} \mid \mathbf{x})$.

In many cases, like human pose inference, both the input (features) and the output (pose) are high-dimensional vectors, i.e., $\mathbf{y} \in \mathbb{R}^{d_\mathbf{y}}$ and $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$ where usually $d_\mathbf{x} > 100$ and $d_\mathbf{y} > 30$. With high-dimensional problems large datasets are usually necessary to learn a conditional distribution that will generalize well. Furthermore since synchronized image and pose data are hard to obtain in practice, one is often forced to work with small or moderately sized labelled data sets, or with unlabelled data using semi-supervised learning [9, 13]. Finally, since pose inference is often ambiguous (i.e., one feature vector is consistent with multiple poses), the conditional distribution, $p(\mathbf{y} \mid \mathbf{x})$, is multi-modal [20].

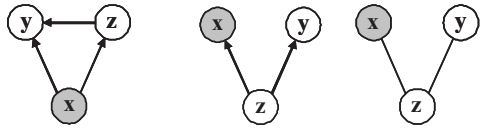


Figure 1. **Graphical models for regression problems.** Gray and white nodes depict observed and hidden variables. MoE and GPLVM are directed models. The Shared KIE is undirected, and can be factored easily in multiple ways.

We introduce a latent variable model called the Shared Kernel Information Embedding (sKIE). It defines a coherent, multi-modal density over one or more *input* feature spaces, the *output* pose space, and a learned low-dimensional latent space. Moreover it is also easy to condition on a latent state, or some combination of input and output states. It can be learned from small datasets, with complexity that is quadratic in the number of training points; the latent model helps to mitigate problems of over-fitting that are common with high-dimensional data. With datasets too large to learn a coherent global model, one can also use sKIE to learn local models in an online fashion. sKIE can deal with missing data during inference, and partially labelled data during learning. We demonstrate the sKIE in the context of discriminative human pose inference.

1.1. Related Work

Approaches to discriminative articulated pose estimation (and tracking) can be loosely classified as either local and global. *Local* methods take the form of kernel regression [16, 21], where one first finds a subset of training exemplars that are similar to the input features (e.g., using K-nearest neighbors). These exemplars are then used to learn an online regressor, like linear locally-weighted regression [16] or Gaussian Process (GP) regression [21]; the latter has the advantage of also producing a confidence measure over the regressed pose. While simple conceptually, the determination of the right local topology and a good distance measure within the neighbourhood can be difficult. Typically these methods require a large set of training exemplars to

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

densely cover the pose space. Furthermore, these methods do not deal with multi-modal conditional distributions (or mappings), so one must first cluster the selected exemplars into local convex sets, for which regression is uni-modal.

Global methods learn a coherent model across the entire training set. Early examples were formulated as Ridge Regression or Relevance Vector Regression [2, 1]. Recent research has focused on multi-valued regression, the most influential of which is the conditional Mixtures of Experts (MoE) model [9, 18, 20]. The MoE model is straightforward and efficient to implement, since training and inference are both $O(N)$. On the other hand, the MoE model typically requires large training sets [13] to properly fit the parameters of the gating and expert functions. This makes it prone to over-fitting with small datasets.

As an alternative, models based on the Gaussian Process Latent Variable Model (GPLVM) [6, 13, 17] and the Spectral Latent Variable Model (SLVM) [10] have been proposed (see Fig. 1). These models exploit an intermediate low-dimensional latent space to effectively regularize the conditional pose distribution (i.e., pose conditioned on features), which helps avoid over-fitting with small training sets. However, as with other Gaussian Process (GP) methods, learning is expensive, $O(N^3)$, and therefore impractical for all but small datasets. Inference with the GP model is also expensive as it involves an $O(N^2)$ optimization (with multiple re-starts) of the likelihood in the latent-space [13]. Sparsification methods [14] reduce learning complexity from $O(N^3)$ to $O(Nd^2)$, where d is the number of *pseudoinputs* (ideally $d << N$), but the use of such techniques is not always straightforward and effective.

sKIE is a generalization of Kernel Information Embedding [12]. It has similar benefits to the GPLVM discussed above, but with lower complexity for learning, $O(N^2)$, and inference, $O(N)$. The other benefits of the sKIE include an explicit density over the latent space, and closed-form expressions for conditional distributions, allowing one to easily condition on a combination of input/output/latent states (see Fig. 1). The sKIE also permits multiple input and output random variables, allowing a dynamic regression from any subset of inputs to any subset of outputs at test time (without learning separate pair-wise models as would be required with MoE, for example). Furthermore, the sKIE can also be used to learn local models in an online fashion (cf. [21]).

2. Kernel Information Embedding

Kernel Information Embeddings (KIE) were introduced in [12] as an unsupervised dimensionality reduction algorithm. Given samples drawn from a distribution $p(\mathbf{x})$, KIE aims to find a low-dimensional latent distribution, $p(\mathbf{z})$, that captures the structure of the data distribution, along with explicit bidirectional probabilistic mappings between the latent space and the data space. In particular, KIE finds the *joint distribution* $p(\mathbf{x}, \mathbf{z})$, that maximizes the *mutual information* (MI) between the latent distribution and the data distribution, i.e.,

$$I(\mathbf{x}, \mathbf{z}) = \int p(\mathbf{x}, \mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})} \, d\mathbf{x} \, d\mathbf{z} \quad (1)$$

$$= H(\mathbf{x}) + H(\mathbf{z}) - H(\mathbf{x}, \mathbf{z}) \quad (2)$$

where $H(\cdot)$ is the usual (differential) Shannon entropy.

Because the data distribution is fixed, maximizing the mutual information (2) reduces to maximizing $H(\mathbf{z}) - H(\mathbf{x}, \mathbf{z})$. This is equivalent to minimizing the conditional entropy $H(\mathbf{x} \,|\, \mathbf{z})$, i.e., the expected negative log likelihood of the data under the joint density $p(\mathbf{x}, \mathbf{z})$. It is interesting to note the similarity to the GPLVM [11] which directly minimizes the negative data log likelihood. Unlike the GPLVM, the KIE provides an explicit density over the latent space.

Given a sample data set $\{\mathbf{x}^{(j)}\}_{j=1}^N$, the KIE objective is optimized to find the corresponding latent positions $\{\mathbf{z}^{(j)}\}_{j=1}^N$. Following [12] we approximate the high-dimensional integrals with kernel density estimates for $p(\mathbf{x})$, $p(\mathbf{z})$, and $p(\mathbf{x}, \mathbf{z})$. If we let $k_{\mathbf{x}}(\cdot, \cdot)$ and $k_{\mathbf{z}}(\cdot, \cdot)$ denote kernels for the data and latent spaces, we can approximate the entropies, $\hat{H}(\cdot)$, and mutual information, $\hat{I}(\cdot)$, as

$$\begin{aligned}\hat{I}(\mathbf{x}, \mathbf{z}) &= \hat{H}(\mathbf{x}) + \hat{H}(\mathbf{z}) - \hat{H}(\mathbf{x}, \mathbf{z}) \\ &= -\frac{1}{N} \sum_i \log \sum_j k_{\mathbf{x}}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ &\quad -\frac{1}{N} \sum_i \log \sum_j k_{\mathbf{z}}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \\ &\quad +\frac{1}{N} \sum_i \log \sum_j k_{\mathbf{z}}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \, k_{\mathbf{x}}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \quad (3)\end{aligned}$$

In what follows we use isotropic, Gaussian kernels for convenience, but one could also use anisotropic kernels.

Inference: Since KIE defines a joint kernel density estimate over latent representatives and observations, inference takes a particularly simple form. It is straightforward to show that the conditional distributions $p(\mathbf{x} \,|\, \mathbf{z})$ and $p(\mathbf{z} \,|\, \mathbf{x})$ are given by weighted kernel density estimates:

$$p(\mathbf{x} \,|\, \mathbf{z}) = \sum_{i=1}^N \frac{k_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^{(i)})}{\sum_{j=1}^N k_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^{(j)})} k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}^{(i)}) \quad (4a)$$

$$p(\mathbf{z} \,|\, \mathbf{x}) = \sum_{i=1}^N \frac{k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}^{(i)})}{\sum_{j=1}^N k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}^{(j)})} k_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^{(i)}) \quad (4b)$$

These conditional distributions are straightforward to compute and they may be multimodal. For Gaussian kernels, they become mixtures of Gaussians.

Learning: Learning the KIE entails the maximization of $\hat{I}(\mathbf{x}, \mathbf{z})$ to find the optimal positions of the latent data representatives. This can be done using any gradient-based optimization method (e.g., conjugate gradient). Towards this end, it follows from (3) that the gradient of $\hat{I}(\mathbf{x}, \mathbf{z})$ with respect to latent position $\mathbf{z}^{(i)}$ is the sum of two terms (since $\hat{H}(\mathbf{x})$ does not depend on $\mathbf{z}^{(i)}$):

$$\frac{\partial \hat{H}(\mathbf{z})}{\partial \mathbf{z}^{(i)}} = -\frac{1}{N} \sum_{j=1}^{N} \left(\kappa_{\mathbf{z}}^{i} + \kappa_{\mathbf{z}}^{j} \right) \frac{\partial k_{\mathbf{z}}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})}{\partial \mathbf{z}^{(i)}} \tag{5}$$

$$\frac{\partial \hat{H}(\mathbf{x}, \mathbf{z})}{\partial \mathbf{z}^{(i)}} = -\frac{1}{N} \sum_{j=1}^{N} \left(\kappa_{\mathbf{xz}}^{i} + \kappa_{\mathbf{xz}}^{j} \right) k_{\mathbf{x}}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \frac{\partial k_{\mathbf{z}}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})}{\partial \mathbf{z}^{(i)}} \tag{6}$$

where $\kappa_{\mathbf{z}}^{i} \equiv (\sum_{l=1}^{N} k_{\mathbf{z}}(\mathbf{z}^{(i)}, \mathbf{z}^{(l)}))^{-1}$ and $\kappa_{\mathbf{xz}}^{i} \equiv (\sum_{l=1}^{N} k_{\mathbf{x}}(\mathbf{x}^{(i)}, \mathbf{x}^{(l)}) k_{\mathbf{z}}(\mathbf{z}^{(i)}, \mathbf{z}^{(l)}))^{-1}$.

Since we are optimizing the positions of latent data representatives (and assuming isotropic kernels) any change in the bandwidth of the latent kernel, $k_{\mathbf{z}}$, is equivalent to rescaling the entire latent space. The choice of the latent space bandwidth $\sigma_{\mathbf{z}}$ is therefore arbitrary, and for the remainder we assume a fixed bandwidth of $\sigma_{\mathbf{z}} = 1$.

The same argument does not hold for the bandwidth of the data space kernel, $\sigma_{\mathbf{x}}$. A common heuristic, which we use below, is to set the bandwidth based on the average distance of nearest neighbors:

$$\sigma_{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} || \mathbf{x}^{(i)} - \mathbf{x}^{(j(i))} || , \tag{7}$$

where $j(i)$ is the index of the nearest neighbor of $\mathbf{x}^{(i)}$. One could also learn the bandwidth using cross-validation.

Regularization: As explained in [12], a trivial way to maximize $\hat{I}(\mathbf{x}, \mathbf{z})$ is to drive all latent positions infinitely far from one another. To avoid this solution, one can include a Gaussian prior over the latent positions, thereby adding $\frac{\lambda}{N} \sum_j ||\mathbf{z}^{(j)}||^2$ to (3) to create the regularized objective function. Here, λ controls the influence of the regularizer. A similar prior is used in the GPLVM [11].

3. Shared Kernel Information Embedding

The Shared KIE (sKIE) is an extension of KIE to multiple (high-dimensional) datasets, with a single hidden cause that is responsible for the variability across all datasets. In what follows we consider the case of two datasets, comprising image feature vectors, $\mathbf{x}$, and poses $\mathbf{y}$. The generalization to more than two datasets is straightforward.

As illustrated in Fig. 1, the sKIE model is undirected and hence it has several natural factorizations. For example, with two datasets one could obtain samples $(\mathbf{x}, \mathbf{y})$ by first sampling from the latent distribution $\mathbf{z}^* \sim p(\mathbf{z})$, and then sampling $\mathbf{x}$ and $\mathbf{y}$ (independently) from their conditionals $p(\mathbf{y} \,|\, \mathbf{z}^*)$ and $p(\mathbf{x} \,|\, \mathbf{z}^*)$. Alternatively, given an observed feature vector $\mathbf{x}^*$ one could draw a sample from the conditional latent distribution $\mathbf{z}^* \sim p(\mathbf{z} \,|\, \mathbf{x}^*)$, and then sample the conditional pose distribution, $p(\mathbf{y} \,|\, \mathbf{z}^*)$.

The sKIE joint embedding for two datasets is obtained by maximizing the mutual information $I((\mathbf{x}, \mathbf{y}), \mathbf{z})$. Assuming conditional independence of $\mathbf{x}$ and $\mathbf{y}$ given $\mathbf{z}$, the MI can be expressed as a sum of two MI terms, i.e.,

$$I((\mathbf{x}, \mathbf{y}), \mathbf{z}) \;=\; I(\mathbf{x}, \mathbf{z}) + I(\mathbf{y}, \mathbf{z}) \,. \tag{8}$$

Like the KIE objective in (3) we maintain a fully non-parametric model, using kernel density estimates for the integrals in (8):

$$\hat{I}((\mathbf{x}, \mathbf{y}), \mathbf{z}) \;=\; \hat{I}(\mathbf{x}, \mathbf{z}) + \hat{I}(\mathbf{y}, \mathbf{z}) \,. \tag{9}$$

In contrast to KIE, here the labelled training data $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$ share a single hidden cause. Thus each pair $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ must be represented by a single, shared latent element $\mathbf{z}^{(i)}$.

We can also incorporate partial data, e.g., image feature vectors $\mathbf{x}^{(j)}$ without corresponding poses, or *vice versa*. In this case there will be latent elements that are only directly constrained by an element in one of the two datasets. More formally, let $\mathcal{I}_{\mathcal{X}}$ and $\mathcal{I}_{\mathcal{Y}}$ denote two sets of indices for the available training data points, with cardinalities $N_{\mathbf{x}}$ and $N_{\mathbf{y}}$. Indices for labelled training samples are included in both sets. Indices for training samples of $\mathbf{x}$ (respectively $\mathbf{y}$) for which there is no corresponding sample from $\mathbf{y}$ ($\mathbf{x}$) exist only in $\mathcal{I}_{\mathcal{X}}$ ($\mathcal{I}_{\mathcal{Y}}$). If we then ignore the terms of the approximate mutual information (9) that are independent of the latent positions, $\mathbf{Z} \equiv \{\mathbf{z}^{(j)}\}_{j=1}^{N}$, where N is the cardinality of $\mathcal{I} = \mathcal{I}_{\mathcal{X}} \cup \mathcal{I}_{\mathcal{Y}}$, the sKIE objective function becomes

$$\begin{aligned} L(\mathbf{Z}) \;=\; & \frac{1}{N_{\mathbf{x}}} \left[\sum_{i \in \mathcal{I}_{\mathcal{X}}} \log \sum_{j \in \mathcal{I}_{\mathcal{X}}} k_{\mathbf{z}}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \; k_{\mathbf{x}}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \right. \\ & \left. - \sum_{i \in \mathcal{I}_{\mathcal{X}}} \log \sum_{j \in \mathcal{I}_{\mathcal{X}}} k_{\mathbf{z}}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \right] \; + \\ & \frac{1}{N_{\mathbf{y}}} \left[\sum_{i \in \mathcal{I}_{\mathcal{Y}}} \log \sum_{j \in \mathcal{I}_{\mathcal{Y}}} k_{\mathbf{z}}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \; k_{\mathbf{y}}(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}) \right. \\ & \left. - \sum_{i \in \mathcal{I}_{\mathcal{Y}}} \log \sum_{j \in \mathcal{I}_{\mathcal{Y}}} k_{\mathbf{z}}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) \right] \end{aligned} \tag{10}$$

3.1. Learning

Learning the sKIE entails the maximization of $L(\mathbf{Z})$ with respect to the unknown latent positions $\mathbf{Z}$. In the experiments below we use a gradient-based approach. The gradients for sKIE optimization have the same form as those for KIE in (5) and (6).

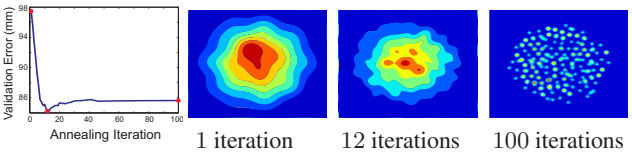


Figure 2. **Annealing with cross-validation.** The left plot shows cross-validation error as a function of the number of annealing iterations. The remaining plots show latent distributions $p(\mathbf{z})$ at annealing levels before, at, and after the minimum of the cross-validation curve. The latent space is initially concentrated, but then spreads out as the annealing progresses. In the limit, the regularizer has no influence and the latent points drift far apart.

Annealing: Like KIE, regularization is necessary to constrain the model. With a mean-zero Gaussian prior over latent positions, the regularized objective function is

$$\hat{L}(\mathbf{Z}) \;=\; L(\mathbf{Z}) \;+\; \frac{\lambda}{N}\sum_{j=1}^{N} ||\mathbf{z}^{(j)}||^2 \;. \qquad (11)$$

When λ is large sKIE tends to keep latent positions tightly clustered and therefore averages over most exemplars. For small λ the latent positions tend to drift apart, producing very little inerpolation between exemplars.

An effective way to learn useful models is to anneal λ during optimization with a stopping criterion determined by cross-validation. We begin with λ large (typically $\lambda = 0.5$), and then gradually reduce it; at each step we typically reduce λ to 90% of its previous value. Cross-validation is used to determine when to stop the annealing. One can use different cross-validation measures. Here we use MSE in discriminative pose inference on a validation set that is disjoint from the training and test set. For example, Fig. 2 shows the effect of typical annealing on the latent sKIE space and illustrates the cross validation error as a function of annealing iterations; the details of the data and model being learned here are given in Fig. 8 in Section 4.2.

Initialization: In the learning experiments below we initialize latent positions using PCA. That is, each labelled training pair $(\mathbf{x}^{(j)}, \mathbf{y}^{(j)})$ become a column of a matrix, the principal directions of which provide a linear subspace embedding with the desired dimension $d_{\mathbf{z}}$. For partially labeled data (i.e., semi-supervised learning), the PCA subspace is first determined from the labelled data. For each unlabelled point, from $\mathbf{x}$ or $\mathbf{y}$, the optimal subspace embedding is found with a pseudoinverse projection.

We have found that the form of the initial guess is not critical. Similar results are obtained with random initialization and a slow annealing schedule. Nevertheless, an initialization strategy that recovers the topology of the latent space, e.g., with LLE or Isomap, is likely to improve the overall performance of sKIE.

3.2. Inference

For discriminative pose inference we want to find likely poses $\mathbf{y}$ conditioned on input image features $\mathbf{x}^*$. We are therefore interested in the conditional pose distribution

$$p(\mathbf{y} \,|\, \mathbf{x}^*) \;\;=\;\; \int_{\mathbf{z}} p(\mathbf{y} \,|\, \mathbf{z})\, p(\mathbf{z} \,|\, \mathbf{x}^*)\, d\mathbf{z} \;. \qquad (12)$$

While we have explicit closed-form expressions for the two conditional factors in the integrand in (12), $p(\mathbf{y} \,|\, \mathbf{x}^*)$ is not straightforward to express or compute in closed-form. It can be approximated with Monte Carlo integration, drawing latent samples and then pose samples as described above, but this can be computationally expensive, especially when $p(\mathbf{y} \,|\, \mathbf{x}^*)$ is multi-modal.

Alternatively, here we focus on identifying the principal modes of $p(\mathbf{y} \,|\, \mathbf{x}^*)$. To this end, we assume that the principal modes of $p(\mathbf{y} \,|\, \mathbf{x}^*)$ coincide with the principal modes of the conditional latent distribution $p(\mathbf{z} \,|\, \mathbf{x}^*)$. That is, we first search for local maxima (MAP estimates) of $p(\mathbf{z} \,|\, \mathbf{x}^*)$, denoted $\{\mathbf{z}_k^*\}_{k=1}^K$ for K modes. From these latent points it is straightforward to perform either MAP inference or take expectations over the conditional pose distributions $p(\mathbf{y} \,|\, \mathbf{z}_k^*)$.

To understand this form of approximate inference, first note that, because features are often ambiguous, we expect $p(\mathbf{y} \,|\, \mathbf{x}^*)$ to be multi-modal. Under sKIE the latent distribution models the joint distribution so, when we condition the latent space on input features, we expect a similarly multi-modal distribution, $p(\mathbf{z} \,|\, \mathbf{x}^*)$. By contrast, conditioned on a highly probable latent point, $\mathbf{z}_k^*$, the pose conditional is usually uni-modal. That is, the multi-modality is manifested mainly in the mapping to the latent space. The latent (joint) space effectively resolves the ambiguity.

Local Modes of $p(\mathbf{z} \,|\, \mathbf{x}^*)$: Given the kernel density estimates in the sKIE model, it is straightforward to show that

$$p(\mathbf{z} \,|\, \mathbf{x}^*) \;=\; \sum_{i=0}^{N} \frac{k_{\mathbf{x}}(\mathbf{x}^*, \mathbf{x}^{(i)})}{\sum_{j=0}^{N} k_{\mathbf{x}}(\mathbf{x}^*, \mathbf{x}^{(j)})} k_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^{(i)}), \qquad (13)$$

To find modes of $p(\mathbf{z} \,|\, \mathbf{x}^*)$ one can choose one or more starting points, and then some form of gradient ascent (e.g., with mean-shift [4]). Starting points could be found by evaluating (13) for each latent representative, and then selecting those with the highest conditional density. One could also draw random samples from $p(\mathbf{z} \,|\, \mathbf{x}^*)$, or, among the training exemplars one could find nearest neighbors to $\mathbf{x}^*$ in feature space, and then begin gradient ascent from their latent representatives (cf., [13]). The most probable latent representatives found in this way are often sufficiently probable in the latent space that subsequent optimization is unnecessary.

Pose Inference: Because the conditional pose distribution is typically uni-modal, it is reasonable to use conditional expectation to find the mean and covariance in the

pose space. From the form of the conditional distributions (4a), the mean of $p(\mathbf{y} \mid \mathbf{z}^*)$ is a convex combination of training exemplars. That is,

$$E[\mathbf{y}|\mathbf{z}^*] \;=\; \sum_{i=1}^{N} \frac{k_{\mathbf{z}}(\mathbf{z}^*, \mathbf{z}^{(i)})}{\sum_{j=1}^{N} k_{\mathbf{z}}(\mathbf{z}^*, \mathbf{z}^{(j)})}\, \mathbf{y}^{(i)} \qquad (14)$$

One can show that the conditional covariance is given by

$$C_{\mathbf{y}\,|\,\mathbf{z}^*} \;=\; C_{k_\mathbf{y}} + \sum_i \frac{k_{\mathbf{z}}(\mathbf{z}^*, \mathbf{z}^{(i)})}{\sum_l k_{\mathbf{z}}(\mathbf{z}^*, \mathbf{z}^{(l)})}\, \mathbf{y}^{(i)}\mathbf{y}^{(i)\,\mathrm{T}} - \sum_{i,j} \frac{k_{\mathbf{z}}(\mathbf{z}^*, \mathbf{z}^{(i)})\; k_{\mathbf{z}}(\mathbf{z}^*, \mathbf{z}^{(j)})}{\left(\sum_l k_{\mathbf{z}}(\mathbf{z}^*, \mathbf{z}^{(l)})\right)^2}\, \mathbf{y}^{(i)}\mathbf{y}^{(j)\,\mathrm{T}} \qquad (15)$$

where $C_{k_\mathbf{y}}$ is the kernel covariance matrix; here $C_{k_\mathbf{y}} = \sigma_\mathbf{y}^2 \mathbf{I}$.

3.3. Complexity

Learning sKIE has complexity $O(N^2)$, while computing conditional distributions (4) is $O(N)$. This compares favorably with the GPLVM, for which learning is $O(N^3)$ due to the inversion of the $N \times N$ kernel matrix, and inference is $O(N^2)$. For both the GPLVM and KIE one can also achieve greater efficiencies with sparsification methods (e.g., [14]), and numerical approximations such as fast multipole methods (e.g., [15]). One can also employ fast mean-shift algorithms for more efficient mode finding on the conditional distributions (e.g., [7]).

4. Experiments

4.1. S-Curve Data

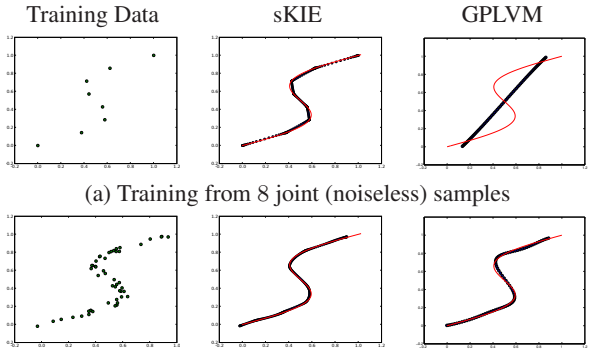
Following [13, 20] we first describe experiments with a synthetic 'S'-curve. Points were sampled from a $2D$ density $p(\mathbf{x}, \mathbf{y})$ defined by

$$\mathbf{x} \;=\; t + sin(2\pi t) + \eta_\mathbf{x}\;, \qquad \mathbf{y} \;=\; t + \eta_\mathbf{y}\;,$$

where $\eta_\mathbf{x}$ and $\eta_\mathbf{y}$ are IID mean-zero Gaussian noise with variance σ^2, and $t = \mathcal{U}(0, 1)$ is uniform. The conditional density $p(\mathbf{y} \mid \mathbf{x})$ has up to 3 modes (see Fig. 3).

Figure 3 shows sKIE models and GPLVMs learned from noisy and noiseless data (the GPLVM was learned from the joint data samples). Both models learn a 1D latent space, encoding the shared structure of the 2D joint distribution. The GPLVM does not capture the S-curve with only 8 samples, consistent with [13]. sKIE does a better job with 8 points. In presence of the noise, sKIE recovers the structure of the curve well, while the GPLVM exhibits a small bias in the upper lobe. Finally, we note that the Mixture of Experts (MoE) model cannot be trained with 8 points. With 50 points MoE can be trained but typically overfits and overestimates the variance (not shown due to space limitations).

Figure 4 shows sKIE learned from partially labeled data. While sKIE tolerates significant amounts of unlabeled data, the model often generalizes well without it, so the unlabeled data mainly helps to reduce the variance of the output estimates. With only 8 fully labeled samples the reconstruction of the joint distribution is smooth (see Fig. 3), but the variance is high because the data are sparse.



(a) Training from 8 joint (noiseless) samples

(b) Training from 50 noisy ($\sigma^2 = 0.1$) joint samples

Figure 3. **Comparison of sKIE and GPLVM on S-curve data.** The red curves depict the true mean of the joint density. For the sKIE and GPLVM the blue points are produced by uniformly sampling latent positions $\mathbf{z}$, and then taking the mean of $p(\mathbf{x}, \mathbf{y}|\mathbf{z})$.

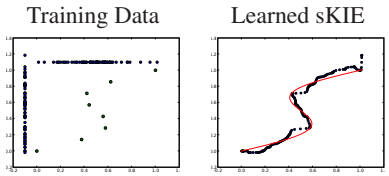


Figure 4. **Semi-supervised Learning.** The sKIE is learned from 8 joint and 142 marginal noisy samples. The curve is not as smooth as in the fully supervised case, but the model does tolerate nearly 20 times as many unlabbeled as labeled samples.

4.2. Human Pose Estimation

We next consider human pose inference. We use two datasets, one synthetic, called POSER [1], and the HUMANEVA dataset with synchronized video and mocap data.

Poser Dataset: POSER contains 1927 training and 418 test images, synthetically generated from mocap data (54 joint angles per frame). The image features and error metric are provided with the dataset [1]. The 100D feature vectors encode the image silhouette using vector-quantized shape contexts. The mean RMS error is given by

$$E_{ang}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{M} \sum_{m=1}^{M} |\, (\hat{\mathbf{y}}_i - \mathbf{y}_i) \mod 360^\circ \,|\,. \qquad (16)$$

Here, $M = 54$, and $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ correspond to the i-th joint angles for the true and estimated poses.

HumanEva Dataset: HUMANEVA-I [19] contains synchronized multi-view video and mocap data. It comprises 3

subjects performing multiple activites. Here we use walking and jogging sequences[1] with observations from 3 color cameras[2]. This includes $5,985$ training samples (image-pose pairs) and $6,291$ test samples (for global models, we only use subject S1, for which there are $2,190$ training and $2,625$ test samples). Where smaller training sets are used, data are randomly sampled from the entire training set.

Features: Following [3] our features are based on shape context descriptors. From each image we extract a silhouette using background subtraction, to which we fit a bounding box. The shape context representation is constructed by randomly sampling 400 points on internal edges and outer contours of the silhouette, from which histograms are constructed[3]. We then cluster $40,000$ randomly sampled histograms to learn a codebook of size 300. Shape context histograms are subsequently vector-quantized using that codebook. The final 300D feature vectors are normalized to unit length. This choice of feature vector was motivated by simplicity and ease of implementation; better features have been shown to perform favorably on HUMANEVA-I (e.g., hierarchical features [9] - HMAX, Spatial Pyramid, Hyperfeatures, and Vocabulary Trees have all been explored).

Errors: Pose is encoded by 15 3D joint centers defined relative to the pelvis in camera-centric coordinates, so $\mathbf{y} \in \mathbb{R}^{45}$. Estimation errors (in mm) are measured as average Euclidian distance to the $M = 15$ markers [19],

$$E_{pos}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{M} \sum_{m=1}^{M} ||(\hat{\mathbf{y}}_i - \mathbf{y}_i)|| \,. \qquad (17)$$

Monocular Pose Inference: Fig. 5 shows how the performance of sKIE depends on the number of training examples. For each of POSER and HUMANEVA we learn 10 sKIE models with dimensions 2 and 5 respectively, using random subsets of training data for each model. The HUMANEVA-I dataset is highly variable and noisy, so learning a higher dimensional embedding improves performance (see Fig. 6).

For this experiment and others below, unless stated otherwise sKIE inference proceeds as follows: Given a feature vector $\mathbf{x}^*$, we find the most probable training examplar according to the conditional latent distribution $p(\mathbf{z}|\mathbf{x}^*)$, from which we use mean-shift to find a local maxima. Conditioned on this point, $\mathbf{z}^*$, we compute the mean pose from $p(\mathbf{y}|\mathbf{z}^*)$. We also implemented Nearest Neighbor (NN) regression and kernel regression (with Gaussian kernels). Performance for all estimators is simply the average error (i.e., (16) and (17)) over the respective test sets. Standard error bars are also shown in all cases.

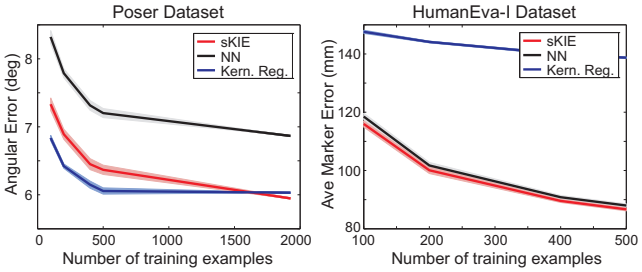


Figure 5. **Single Hypothesis Inference.** The plots show the average error (and standard error bars - using shading) for sKIE, NN Regression, and Kernel Regression on HUMANEVA-I (right) and POSER (left) as a function of the training set size.

sKIE consistently outperforms NN Regression. Kernel Regression performs favorably compared to sKIE on POSER data (with few training examples). We postulate that this is due in part to the relatively clean data that does not exhibit as much multi-modality. Kernel Regression is sensitive to the kernel bandwidth and performance can quickly degrade as the kernel bandwidth increases[4]. sKIE seems to be relatively insensitive to kernel bandwidths in both the input and output spaces.

We also conducted a set of experiments in which sKIE produces multiple predictions (called k-sKIE, for k predictions). In doing so we explore several alternate forms of inference: (1) sampling K samples from $p(\mathbf{z}|\mathbf{x}^*)$, followed by mode finding, then estimation of the mean pose from the conditional pose distribution; (2) starting with latent representatives associated with the K nearest neighbors to $\mathbf{x}^*$ in the input feature space, from which we compute the mean pose conditioned on each corresponding latent point; and (3) like (2) but with intermediate latent mode finding. In all cases the annealing schedule was $\lambda_{i+1} = 0.9\lambda_i$ starting with $\lambda_0 = 0.5$ and was run for a maximum of 20 annealing iterations.

Fig. 6 shows the results. For comparison, we also include the performance of k-nearest neighbors, kernel regression, and standard sKIE inference used in Fig. 5. Notice that utilizing nearest neighbors as a way of finding good latent states performs better than random sampling with hill climbing, though for the POSER dataset even the latter produces more accurate results than k-NN for $k \leq 4$. The best performance at $k = 1$ occurs with mode finding from the conditional nearest neighbor. Nevertheless, it is somewhat poorer then conditional expectation based on the nearest neighbors without mode finding in most other cases. In any case it is clear that the latent model carries significant value in regularizing the inference; i.e., with sKIE we al-

[1] We ignore walking from subject 3 due to corrupt motion capture data

[2] Like [3, 21] we do not use the 4 greyscale views.

[3] Histograms are computed at 12 angular and 5 radial bin log-polar resolution, with minimal and maximal radial extant set to 1/8 and 3 of the mean distance between all 400 sampled points. The histograms are thereby invariant to the overall scale of the sillhouette.

[4] Performance of kernel regression on HUMANEVA-I in Fig. 7 can be improved by manually tuning kernel bandwidths, but for consistency with sKIE we used the same heuristic, Eq. (7), for all models in all experiments.

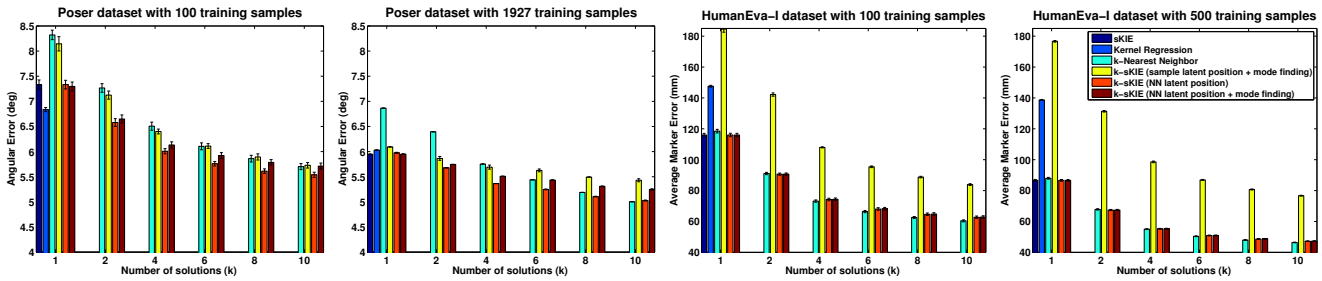


Figure 6. **Multi-Hypothesis Inference.** Graphs show the performance of sKIE as a function of the number of hypotheses, k (for 100 and 1927 training examples for POSER data and 100/500 training examples for HUMANEVA-I data). We compare performance of sKIE with k-Nearest Neighbor Regression and Kernel Regression; different inference methods are explored as described in the text.

Algorithm / Dataset	HUMANEVA-I	POSER
Linear Regression [6]	-	7.70 (deg)
Nearest Neighbor	70.15 (mm)	6.87 (deg)
GPLVM [6]	-	6.50 (deg)
Kernel Regression	138.13 (mm)	6.03 (deg)
Gaussian RVM [1]	-	6.00 (deg)
Global sKIE	-	5.95 (deg)
Local sKIE (25 neigh)	64.63 (mm)	5.77 (deg)

Figure 7. **Performance of the local sKIE model.**

ways perform better then k-NN for any k in POSER experiments, and better then k-NN for $k < 4$ for HUMANEVA-I.

Monocular Local Pose Estimation: The entire HUMANEVA-I dataset is so large that learning a coherent global model is prohibitively expensive. In such cases one can learn local sKIE models online for inference.

For each test case we first find the 25 training samples whose features vectors are closest to the test feature vector. From those 25 samples we learn an sKIE with a 2D latent space. To speed up the online learning sKIE was trained by running a fixed (5) number of annealing iterations starting at $\lambda = 0.5$ and taking 100 gradient steps for every annealing iteration. Once trained, the mode of the conditional latent distribution was found using optimization, from which the conditional mean was used as the estimated pose. The full datasets were used for both testing and training. As shown in Fig. 7, the online sKIE performs favorably with respect to all methods considered, including the GPLVM, whose performance was reported with POSER data in [6].

Multi-Variable Shared-KIE Models: An advantage of sKIE over direct regression models (e.g., MoE) is its ability to learn joint models over many variables, allowing conditioning on any subset of them at test time. With other regression models a separate regression function (or conditional distribution) would have to be learned between all combinations of test inputs. To illustrate these benefits of sKIE we trained two models with more that one input and one output, one for multi-view pose inference, and one for tracking.

Multi-view pose inference: Using the HUMANEVA-I dataset we learned an sKIE with input features from three synchronized cameras $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ and pose $\mathbf{y}$. The model can be conditioned on any subset of inputs. The results, explained in Fig. 8, clearly show the ambiguities that arise in pose inference from only one or two views.

Monocular Tracking: For Bayesian pose tracking we want to condition on both the current image features, $\mathbf{x}_t$, and the pose at the previous time, $\mathbf{y}_p$. Fig. 9 shows such a model. It can be used to initialize a tracking discriminatively, or generatively as a motion prior. Other methods (e.g., [2, 20]) would need several models to perform the same task. Again, in Fig. 9 it is interesting to see that the conditional latent distributions, conditioned on features, is sometimes multi-modal (frame 430), and sometimes uni-modal (frame 586).

5. Conclusions

This paper describes a new shared latent variable model called the shared Kernel Information Embedding (sKIE). This model has several appealing properties, namely, that (1) it utilizes a latent space as an intermediary in the inference process and hence allows learning from small datasets (i.e., can generalize well); (2) it provides closed-form multi-modal conditional distributions, conditioning on the input space (or spaces) of features, the shared latent space, and output pose space; (3) it defines coherent densities over these spaces; (4) it has favorable complexity of $O(N^2)$ for training and $O(N)$ for inference. Furthermore, sKIE can deal with missing data during inference, and partially labeled data during learning.

Acknowledgements: This work was supported in part by NSERC Canada, the Canadian Institute for Advanced Research (CIFAR), and a grant from Bell University Labs.

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. *CVPR*, 2:882–888, 2004.

[2] A. Agarwal and B. Triggs. Learning to track 3d human motion from silhouettes. *ICML*, 9–16, 2004.

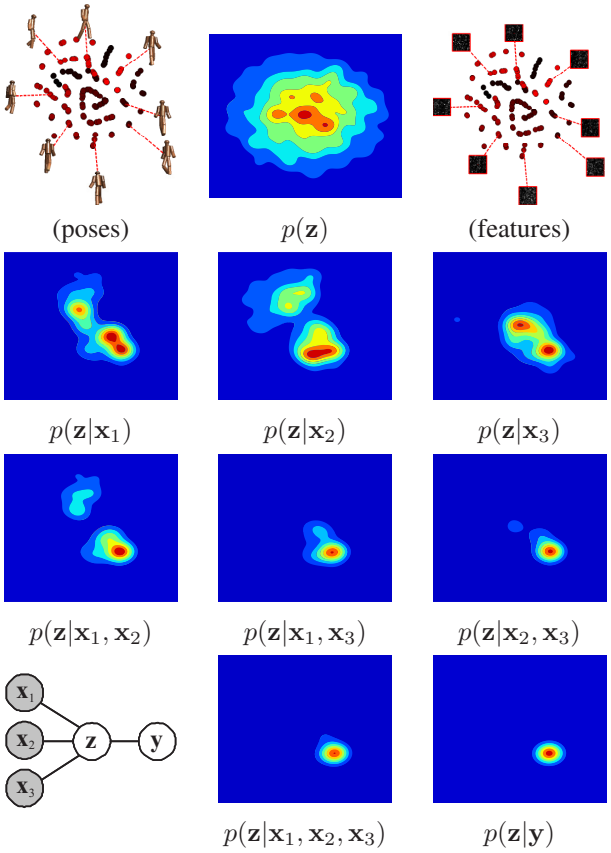


Figure 8. **Shared KIE for multiview pose inference.** The sKIE is learned from 200 random samples of walking from Subject 1 in HUMANEVA-I. Inputs are 300D shape context features from each of three synchronized cameras, $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, plus a 45D pose vector (i.e., a joint model over 947 dimensional data). Top row depicts the learned 2D shared latent space. Remaining rows depict conditional latent space distributions obtained by conditioning on subsets of the views and on the true pose (*bottom-right*). Top-left figure illustrates the individual training sample positions in the latent space colored by their proximity within the sequence. Nearby poses do end up close to one another in the latent space. Also note that the conditional distributions obtained by conditioning on observations and on the pose give consistent densities in the latent space.

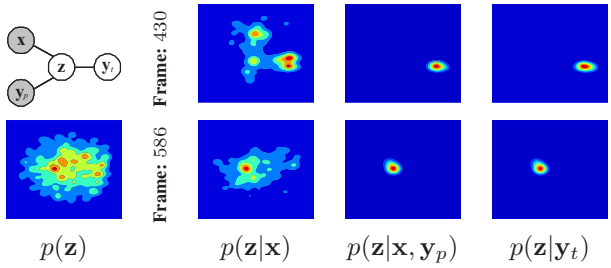


Figure 9. **Shared KIE for tracking.** Illustrated is the latent-space, the latent-space conditioned on one current observation, and latent-space conditioned on the current observation and previous pose. We utilize the 200 samples from HUMANEVA-I dataset (camera 1) used to train the model in Fig. 8.

[3] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas. Fast algorithms for large scale conditional 3d prediction. *CVPR*, 2008.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI*, 24(5):603–619, 2002.

[5] T. de Campos and D. Murray. Regression-based hand pose estimation from multiple cameras. *CVPR*, 1:782–789, 2006.

[6] C. Ek, P. Torr, and N. Lawrence. Gaussian process latent variable models for human pose estimation. *Work. Mach. Learn. Multimodal Interact.*, LNCS 4892:132–143, 2007.

[7] B. Han, D. Comaniciu, Y. Zhu, and L. Davis. Sequential kernel density approximation: Application to real-time visual tracking. *IEEE Trans. PAMI*, 30(7):1186–1197, 2008.

[8] T. Jaeggli, E. Koller-Meier, and L. V. Gool. Monocular tracking with a mixture of view-dependent learned models. *AMDO*, LNCS 4069:494–503, 2006.

[9] A. Kanaujia, C. Sminchisescu, and D. Metaxas. Semi-supervised hierarchical models for 3d human pose reconstruction. *CVPR*, 2007.

[10] A. Kanaujia, C. Sminchisescu, and D. Metaxas. Spectral latent variable models for perceptual inference. *ICCV*, 2007.

[11] N. D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *J. Machine Learning Res.*, 6:1783–1816, Nov. 2005.

[12] R. Memisevic. Kernel information embeddings. *ICML*, 633–640, 2006.

[13] R. Navaratnam, A. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. *ICCV*, 2007.

[14] J. Quiñonero-Candela and C. Rasmussen. A unifying view of sparse approximate gaussian process regression. *J. Machine Learning Res.*, 6:1939–1959, 2006.

[15] V. Raykar and R. Duraiswami. *The Improved Fast Gauss Transform with applications to machine learning*. MIT Press, 2006.

[16] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. *ICCV*, 2:750–759, 2003.

[17] A. Shon, K. Grochow, A. Hertzmann, and R. Rao. Learning latent structure for image synthesis and robotic imitation. *NIPS*, 1233–1240, 2006.

[18] L. Sigal, A. Balan, and M. J. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. *NIPS*, 2007.

[19] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *TR CS-06-08, Brown University*, 2006.

[20] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. *CVPR*, 1:390–397, 2005.

[21] R. Urtasun and T. Darrell. Local probabilistic regression for activity-independent human pose inference. *CVPR*, 2008.