A Non-parametric Bayesian Network Prior of Human Pose

Andreas M. Lehrmann
MPI for Intelligent Systems
Tuebingen, Germany
alehrmann@tue.mpg.de

Peter V. Gehler
MPI for Intelligent Systems
Tuebingen, Germany
pgehler@tue.mpg.de

Sebastian Nowozin
Microsoft Research
Cambridge, UK
Sebastian.Nowozin@microsoft.com

Abstract

Having a sensible prior of human pose is a vital ingredient for many computer vision applications, including tracking and pose estimation. While the application of global non-parametric approaches and parametric models has led to some success, finding the right balance in terms of flexibility and tractability, as well as estimating model parameters from data has turned out to be challenging. In this work, we introduce a sparse Bayesian network model of human pose that is non-parametric with respect to the estimation of both its graph structure and its local distributions. We describe an efficient sampling scheme for our model and show its tractability for the computation of exact log-likelihoods. We empirically validate our approach on the Human 3.6M dataset and demonstrate superior performance to global models and parametric networks. We further illustrate our model's ability to represent and compose poses not present in the training set (compositionality) and describe a speed-accuracy trade-off that allows real-time scoring of poses.

1. Introduction

Reasoning about human pose is a key ingredient in recent successful applications of computer vision systems [20]. Accurately capturing the variability of human pose is challenging because there is both a variation between different persons as well as a combinatorial number of possible poses a single person can assume.

In this paper we propose a *pose prior*, a generative probabilistic model of static human pose. Such a general purpose prior model is useful in at least two ways; first, it can synthesize realistic poses that can be used for rendering or for generating plausible pose hypotheses, and second, in the context of a larger pose estimation or tracking system it can score any given pose by how a priori likely it is, serving as a more specific regularization term.

A good pose prior must generalize to unseen poses and persons. If it was merely reproducing poses seen in a training dataset it could never span the full variability of human pose. In order to generalize the prior must be *compositional*: it must represent the variations of parts that frequently occur together and produce a pose by combining these parts.

We achieve compositionality by factorizing the pose representation into a Bayesian network [13]. The sparse hierarchical structure of the network enables efficient computation of likelihoods and exact sampling. To apply a Bayesian network on human pose data we need to specify the network structure and conditional probability distributions along the network and it is here that we make two novel technical contributions. First, we enhance the representative power of Bayesian networks by proposing *non-parametric Bayesian networks* in which the conditional distributions are represented by conditional kernel density estimates. Second, we use structure learning to obtain the network structure by finding parts of the pose that strongly depend on each other, leveraging non-parametric mutual information estimators on continuous joint data.

Our data-driven approach is made possible by the recent availability of a large-scale dataset of human pose, the Human 3.6M dataset [12], which captures a large variety of poses and persons. We use this dataset to assess the generalization performance of our approach and demonstrate its good adaption to unseen test poses. Although our learned system is efficient, some applications require direct control of the runtime. For such scenarios we propose an approximation trade-off. With this approximation we demonstrate real-time scoring of Kinect tracker output.

1.1. Related Work

Pose priors are most often used within pose estimation systems and therefore some of the related works we discuss below incorporate a likelihood term that is computed from an observed image. Incorporating such an observation likelihood is possible in our model as well, but in the present work our focus is on a generative model.

A natural idea to build a pose prior is to use the tree structure of the human skeleton as a starting point. Models that follow the skeletal structure are called *kinematic chain mod-*

els [2] and they allow us to incorporate prior beliefs about joint angles. In [17] the authors used a multivariate Normal distribution along the kinematic chain and estimate the parameters from motion capture data. The different choices of possible parametrizations in terms of joint angles or relative world coordinates in a kinematic tree model give rise to qualitatively different behaviours [10]. Despite this flexibility a kinematic tree model has clear limitations, as sharply argued in [15]; it is unable to express the coordination of different limbs and fails to represent global balance and gravity constraints.

We will demonstrate that we can avoid these limitations by using a tree model that does not correspond to the kinematic chain but instead is chosen to optimally approximate the true distribution of poses. The resulting tree no longer corresponds to a skeleton (Figure 1c and 3b) but retains all computational advantages of a tree-structured model.

Previous works have attempted to overcome the limitations of the kinematic tree model in different ways. In [3] the authors have used a global kernel density model on human pose. This model is global and does not reflect the combinatorial nature of human pose hence it is suitable only for modeling specific poses. Another approach proposed in [21] has been to add further interactions to the kinematic tree so that limb-limb coordination and penetration constraints are modelled. This is satisfying as a model but because the model now has cycles, exact inference becomes intractable and the authors have to resort to an expensive approximate particle belief propagation. Likewise in [27] a structure learning heuristic is used to learn a compositional model of pose; exact inference is again intractable and a heuristic based on likely hypotheses is used.

Another popular way to improve over the kinematic tree model is to add latent variables to the model. In [15] the authors augment the kinematic tree model by a few latent variables that are identified by factor analysis. The Gaussian Process latent variable model (GPLVM) [16] has been applied as a pose model [6]. In the GPLVM model a low-dimensional latent space is transformed to pose space by means of a Gaussian Process regression function. The GPLVM model has also been extended to incorporate a temporal model (GPDM) [24]. The Laplacian Eigenmap latent variable model (LELVM) [18] improves on the GPLVM by modeling the manifold of poses using a graph Laplacian and by providing tractable posterior inference in the latent space. An interesting recent model based on a large number of latent binary variables is the implicit mixture of conditional restricted Boltzmann machines (imCRBM) [23]; both estimation and inference are again approximate. While the global latent variable models (GPLVM and LELVM) are flexible they do not provide compositionality. In fact, each training pose is represented as one latent vector and they are not combined in an intelligent way.

2. Non-parametric Bayesian Networks

In this section we introduce our non-parametric Bayesian network model of human pose and show its tractability.

We represent a human body pose by a d-dimensional vector whose components correspond either to angular or xyz coordinates of n joints. Each pose thus decomposes on the joint level, $\mathbf{x} = [x_1, \dots, x_n] \in \mathbb{R}^d$, and we model the angle/position of joint j by a possibly multi-dimensional random variable X_j. The vector of all variables $\mathbf{X} = (X_j)_{j=1,\dots,n}$ defines a high-dimensional pose distribution $q(\mathbf{X})$ whose samples we denote by $\{\mathbf{x}^{(i)}\}_{i=1,\dots,N}$. In principle, we could use a global density estimation technique to learn q. But as discussed in section 1.1, such approaches are either prone to overfitting, lack flexibility or are computationally intractable.

In this work, we therefore take another approach and learn a sparse and non-parametric Bayesian network. A Bayesian network over $\mathbf{X}$ is a pair $(p, \mathcal{G})$ where the distribution p factorizes over the directed acyclic graph $\mathcal{G}$,

$$p(\mathbf{X}) = \prod_{j=1}^{n} p\left(X_j | X_{\mathrm{pa}(j)}\right). \tag{1}$$

The parent operator pa maps an index to the set of parental indices w.r.t. $\mathcal{G}$, [13, 1].

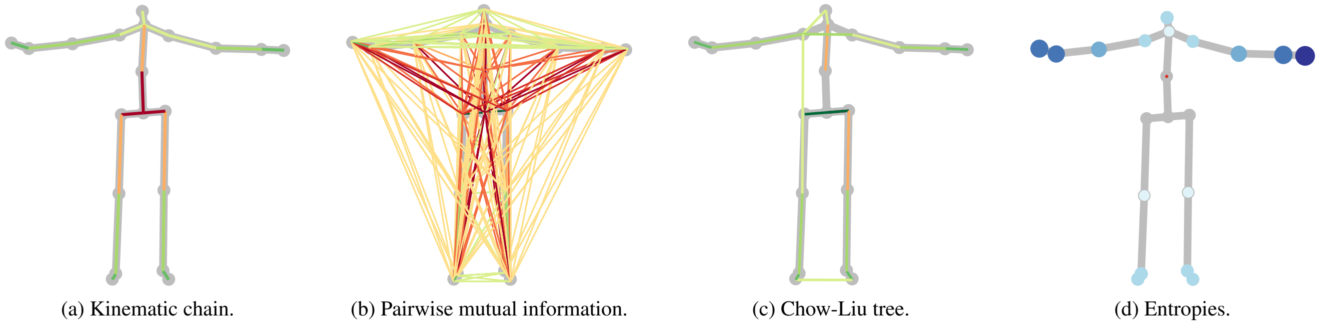
The specification of a Bayesian network hence consists of two parts: The definition of a graph structure $\mathcal{G}$ and the definition of local probabilistic models $p\left(X_j | X_{\mathrm{pa}(j)}\right)$. In the next two sections, we will introduce a fully non-parametric approach for both components.

Our proposed model is different from an earlier proposal, [11]. Their model is based on a mixture distribution over all possible networks and therefore exact likelihood computation is no longer efficient.

2.1. Learning the Graph Structure

The graph structure of a Bayesian network models the local and global (in)dependencies of a distribution. In most cases, the object to be modeled carries some apparent structure and many approaches define $\mathcal{G}$ in a way that reflects the objects intuitive dependencies. In case of the human body, an obvious structure is the kinematic chain, i.e., a tree-structured network with one parent per variable that follows the adjacency of joints in the body (Figure 1a). However, such a canonic representation does not necessarily lead to optimal conditional distributions in an information-theoretic sense.

Therefore, we take another approach and learn $\mathcal{G}$ from data. Since our goal is to learn a sparse structure, we impose the constraint of at most one parent per variable and search for a sparse Bayesian network $(p, \mathcal{G})$ with minimal



(a) Kinematic chain.

(b) Pairwise mutual information.

(c) Chow-Liu tree.

(d) Entropies.

Figure 1: In **(a)**, we show the graph structure of the commonly used kinematic chain together with a non-parametric estimation of mutual information (high MI in green, low MI in red). **(b)** shows the graph of all pairwise mutual informations and **(c)** the corresponding Chow-Liu tree, i.e., the maximum spanning tree of **(b)**. Note how the uninformative edges present in **(a)** are circumvented in **(c)**. The visualization in **(d)** shows a Hinton diagram of the estimated joint entropies.

Kullback-Leibler divergence to $q(\mathbf{X})$, [13],

$$\mathcal{G} := \operatorname*{argmin}_{pa} \mathrm{KL}\left(q(\mathbf{X}) \,\middle\|\, \prod_{j=1}^{n} p\left(X_j | X_{\mathrm{pa}(j)}\right)\right). \tag{2}$$

The network minimizing this distance is known as a Chow-Liu tree and was introduced in [4] for discrete distributions. Given a fully connected graph $\tilde{\mathcal{G}}$ over $\mathbf{X}$ with edge weights w_{jk} set equal to the mutual information $\mathrm{MI}(X_j, X_k)$ between X_j and X_k, the solution to (2) can be shown to be the maximum spanning tree of $\tilde{\mathcal{G}}$ (with edges directed outwards in a consistent way) [1].[1] In contrast to the kinematic chain, a Chow-Liu tree is thus guaranteed to model those pairs of joints that exhibit a high flow of information, independent of their adjacency in the human body.

Here, we use a continuous variant of the Chow-Liu tree, where reliable estimation of mutual information is a hard problem [25]. An ad-hoc resolution is to either discretize the variables or to make simple parametric assumptions. Instead, we employ a fully non-parametric approach based on nearest neighbor distances. We first use the non-parametric entropy estimator [14] in $d_j := \dim(X_j)$ dimensions and calculate

$$\hat{H}(X_j) := \frac{d_j}{N} \sum_{i=1}^{N} \ln \left\| x_j^{(i)} - \eta_j^{(i)} \right\| + c, \tag{3}$$

with the constant $c = \ln(N-1) + \ln V_{d_j} + \gamma$. In the equation above, $\eta_j^{(i)}$ is the nearest neighbor of $x_j^{(i)}$, $V_{d_j} = \pi^{d_j/2} / \Gamma(d_j/2+1)$ is the volume of the d_j-dimensional unit ball, and $\gamma \approx 0.5772$ is the Euler-Mascheroni constant. A more general class of entropy estimators including the one above was shown to be asymptotically unbiased and consistent as $N \to \infty$ in [8]. Using the entropy estimate we can then estimate all pairwise mutual information values by using the relation, [25],

$$\begin{aligned} w_{jk} :=& \hat{\mathrm{MI}}(X_j, X_k) \\ =& \hat{\mathrm{H}}(X_j) + \hat{\mathrm{H}}(X_k) - \hat{\mathrm{H}}(X_j, X_k). \end{aligned} \tag{4}$$

The computed mutual information is visualized in Figure 1b and we can now solve for the Chow-Liu tree [4] by finding the maximum spanning tree [5] to obtain our final result $\mathcal{G}$, shown in Figure 1c.

2.2. Learning the Local Models

Once the network structure is fixed, we need to learn the local conditional distributions $p\left(X_j | \,\mathrm{pa}\left(X_j\right)\right)$ from training data. Since we can estimate them independently, we focus on one of them to keep the notation uncluttered: Let $p(X|Y)$ be one specific local distribution and $\{(x^{(i)}, y^{(i)})\}_{i=1,\ldots,N}$ observations of the corresponding joint distribution $p(X, Y)$. Our approach will be to compute a conditional kernel density estimate (CKDE) in which we can condition on given values $Y = y$ as needed. An unconditional kernel density estimate is given by

$$p\left(x, y\right) := \frac{1}{N|B|} \sum_{i=1}^{N} k\left(B^{-1}\left((x, y) - (x^{(i)}, y^{(i)})\right)\right), \tag{5}$$

where B is the *bandwidth matrix* and k is the *kernel function*. We use Scott's rule [19] to estimate B from the sample covariance. The choice of kernel is only of minor importance, so we use an isotropic Gaussian kernel $k = \mathcal{N}\left(\vec{0}, I\right)$. Equation (5) in this case simplifies to

$$p\left(x, y\right) := \frac{1}{N} \sum_{i=1}^{N} \mathcal{N}\left((x, y) \,\middle|\, (x^{(i)}, y^{(i)}), BB^{\top}\right) \tag{5a}$$

[1] We omit arrows from our network visualizations and implicitly assume the orientations to be directed away from the hip node.

and the conditional distribution for a given value of y is

$$p\left(x|y\right) = \frac{p\left(x,y\right)}{\int_x p\left(x,y\right)\mathrm{d}x}. \tag{6}$$

The evidence term in the denominator requires integration over all non-evident dimensions, which has the analytic solution

$$\int_x p\left(x,y\right)\mathrm{d}x = \frac{1}{N}\sum_{i=1}^{N}\mathcal{N}\left(y \;\middle|\; y^{(i)}, (BB^\top)_{yy}\right), \tag{6a}$$

where $(BB^\top)_{yy}$ denotes the part of $BB^\top$ describing the covariance of y. In summary, we can compute the CKDE density $p(x|y)$ efficiently and at the same asymptotic complexity as the joint KDE density $p(x,y)$.

2.3. Log-likelihoods and Sampling

There are two important operations to perform in applications of our model as a pose prior: computing the *likelihood* of a given pose and *sampling* a pose from the prior. Both operations are efficient as we now show.

Exact log-likelihoods. Given a Chow-Liu/CKDE network with n variables, the log-likelihood $\log p\left(\mathbf{x}\right)$ of a new observation $\mathbf{x} \in \mathbb{R}^d$ is

$$\sum_{j=1}^{n}\left(\log p\left(x_{j,\mathrm{pa}(j)}\right) - \log \int_{x_j} p\left(x_{j,\mathrm{pa}(j)}\right)\mathrm{d}x_j\right). \tag{7}$$

Both parts of the j'th summand have a closed-form solution. Note that the global log-likelihood is composed of many local log-likelihoods, so that we can distinguish likely from unlikely angles/positions on the joint level. This allows a detailed analysis of a pose not possible in global methods.

Sampling. Thanks to the closed-form solution for a conditional Gaussian, we can employ standard ancestral sampling [13], i.e, we find a topological ordering τ for the network structure and draw samples from $p(X_{\tau(j)}|X_{\mathrm{pa}(\tau(j))})$, for $j = 1, \ldots, n$. The only technicality we need to take care of is a conditional reweighting operation of the Gaussian components: A standard kernel density estimate of the form (5a) can be interpreted as a Gaussian mixture with uniform weights and sampling boils down to sampling from a uniformly selected component. In ancestral sampling, on the other hand, we have to deal with conditional distributions. Splitting up the enumerator in (6) shows that we again get a Gaussian mixture model,

$$p\left(x \mid y\right) = \sum_{i=1}^{N} w_i \cdot \mathcal{N}\left(x \;\middle|\; \mu_y^{(i)}, \Sigma_y\right), \tag{8}$$

but this time with non-uniform weights,

$$w_i = \frac{\mathcal{N}\left(y \mid y^{(i)}, (BB^\top)_{yy}\right)}{\sum_{i=1}^{N}\mathcal{N}\left(y \mid y^{(i)}, (BB^\top)_{yy}\right)}. \tag{9}$$

Here, $\mu_y^{(i)}, \Sigma_y$ are the mean and covariance of the i'th Gaussian conditioned on y. Sampling from a local distribution thus consists of two steps: We first select a Gaussian component according to the discrete distribution induced by the weights and then draw a sample from the selected Gaussian conditioned on y.

We see that despite its flexibility the computation in our model is efficient, exact, and simple to implement. We now validate our model experimentally.

3. Experiments

Our experiments are based on two different datasets: the Human 3.6M (H36M) dataset [12] for large-scale experiments and our own Kinect recordings to showcase more specific aspects of our model. For the H36M experiments we use all 7 actors for whom pose data are available and employ a leave-one-person-out scheme: We use all 30 categories of actors $5, 6, 7, 8, 9, 11$ and randomly subsample 15% of all frames without replacement to obtain more independent samples; this constitutes our training set ($\approx 70k$ poses). We use all frames of actor 1 to construct the test set ($\approx 62k$ poses). The H36M skeleton includes some spurious joints that we delete, which results in the same 20 joints present in the Kinect skeleton [20]. All frames are given in relative xyz coordinates centered at the hip node, unless otherwise stated.

3.1. Pose Model

We start by learning a pose model on the H36M training set according to the techniques introduced in section 2. The resulting network structure is displayed in Figure 1c and it is worth noting some of its properties: 1. Three edges connect the left half of the body with the right half, thereby enforcing coherent positions for the feet, hips and shoulders. Note that this does not apply to the kinematic chain. 2. The uninformative pairs of nodes present in the kinematic chain (red edges in Figure 1a) are circumvented in the Chow-Liu tree, thus guaranteeing, from an information-theoretic point of view, optimal conditional distributions under the given constraint of a sparse structure. 3. Subgraphs containing joints with high entropies (Figure 1d), such as the arms and legs, largely follow the kinematic chain. This confirms the intuitive belief that joints with high uncertainty should be conditioned on nearby joints, as they provide the maximum information about a joints position in this case.

One of the advantages of a generative model is that we can immediately check hypotheses, e.g., by drawing samples from the model. Using our Matlab implementation

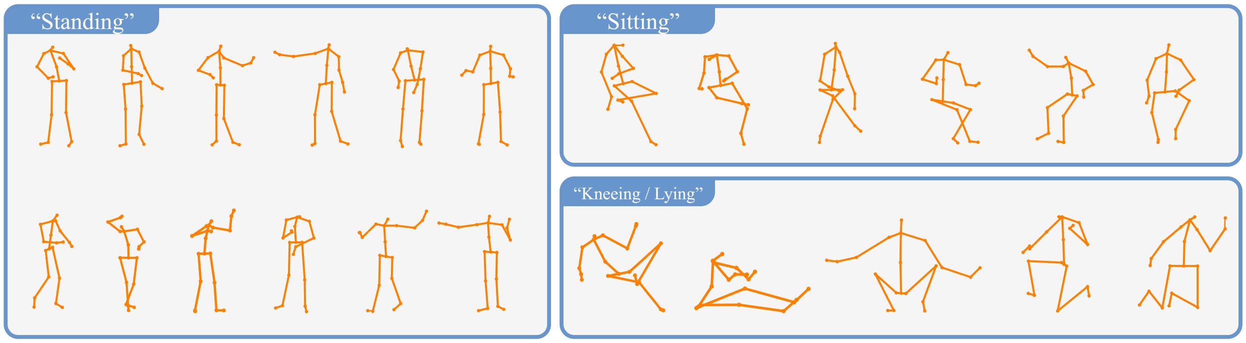


Figure 2: We show samples drawn from our non-parametric pose prior to give an impression of sample quality. All samples are untouched and were generated from a single Chow-Liu/CKDE model.

of the sampling scheme introduced in section 2.3, we are able to generate approximately 10 samples/second. Figure 2 shows a selection of them. Note the variety of poses and their natural appearance, confirming that our model can indeed capture, represent and generate many of the aspects unique to human pose.

3.2. Comparison of Hold-out Log-likelihoods

In this experiment, we evaluate how well our model fits to unseen test poses and how it compares to competing methods. One way to do this in an unsupervised setting is to compute expected hold-out log-likelihoods (ELL) on the H36M test set. As described in section 2, our model consists of two components: estimation of the graph structure (non-parametric Chow-Liu tree) and estimation of the local distributions (conditional kernel density estimation).

We compare this approach to alternatives for these two aspects on training and test data. More specifically, we consider two different ways of estimating the local models and six different graph structures. The options for the conditional distributions are our CKDE approach and a Gaussian linear (GL) network [13]. Cases of badly conditioned covariance matrices are handled by enforcing a lower bound on the eigenvalues.

For the graph structures we consider a global graph with only a single node, a fully independent graph with n nodes but no edges, the kinematic chain, a higher-order kinematic chain, and two variants of the Chow-Liu tree, one with parametric and one with non-parametric estimation of mutual information. In the higher-order kinematic chain each joint is additionally conditioned on its parents' parents. Parametric MI-estimation is based on the entropy of fitted Gaussians. We use parametric MI-estimation for the parametric GL network and our distance-based non-parametric MI-estimation for the non-parametric CKDE network.

The network approaches are complemented by a comparison to the global GPLVM [16], where we employ the popular FITC approximation [22] together with subsampling to achieve tractability. We use a reference implementation[2] and consider embeddings in 1, 3, and 5 latent dimensions, reporting the best ELL.

Table 1: Expected log-likelihoods of GL- and CKDE networks for different graph structures and a comparison to global methods.

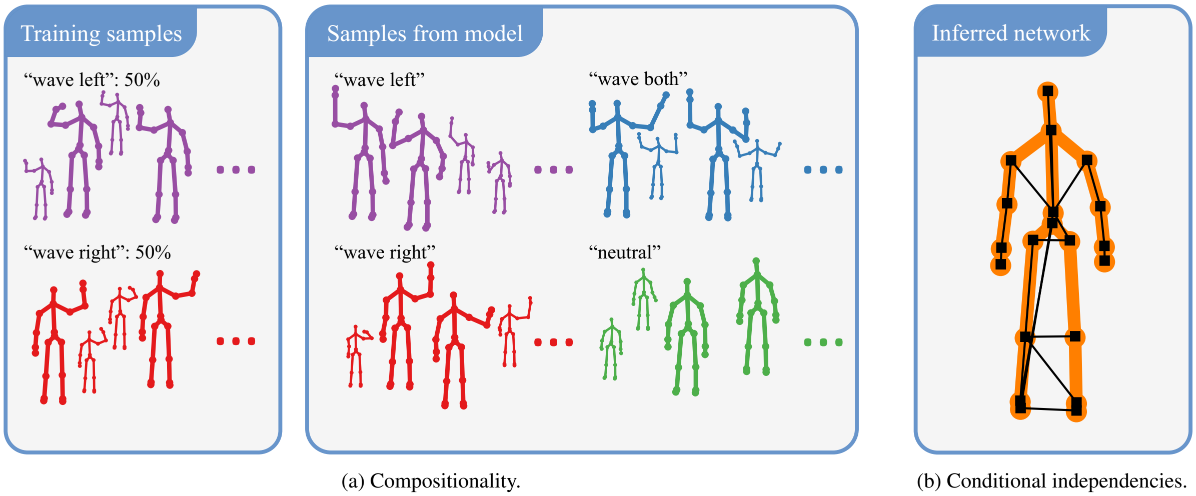
Method	Graph structure	Training	Testing
Gaussian	Global	-266.84	-271.15
KDE	Global	-239.61	-263.77
GPLVM*	Global	-327.85	-341.89
Gaussian linear network	Independent	-352.80	-345.94
	Kinematic chain (order 1)	-311.54	-310.98
	Kinematic chain (order 2)	-305.54	-307.88
	Chow-Liu tree	-283.82	-284.03
CKDE network	Independent	-322.64	-322.25
	Kinematic chain (order 1)	-260.04	-270.52
	Kinematic chain (order 2)	-247.35	-263.83
	Chow-Liu tree **(ours)**	-242.24	$\mathbf{-254.98}$

*25% subsampling; FITC

Results. Our results are shown in Table 1. Among the global methods, the test ELL of a KDE (-264) outperforms both a global Gaussian (-271) and the GPLVM (-342 with 5 latent dim.), despite a spread between training and testing due to overfitting. Although the GPLVM performance could probably be further improved, either by developing better approximations or fine-tuning of the parameters, application of the GPLVM to such a large dataset is an inherently approximate procedure involving a non-convex optimisation problem prone to initialisation and local minima, which is presumably the cause for its poor performance.

Let us now turn to the network approaches and analyze their graph structures. Not surprisingly, a network modeling the joints independently performs worst, with test ELLs of -346 (GL) and -322 (CKDE). Using graph structures based on the kinematic chain increases the test performance

[2] `http://staffwww.dcs.sheffield.ac.uk/people/N.Lawrence/fgplvm/`



(a) Compositionality.

(b) Conditional independencies.

Figure 3: In **(a)**, we show samples from the "wave" training set (left, 2 pose classes) and samples drawn from the learned model (right, 4 pose classes). Our model has learned two poses not available in the training set: "wave both" and "neutral". In **(b)**, we show the inferred network structure. Note that the arms are conditionally independent and can thus be freely combined, whereas the legs are dependent on each other.

to -311 (GL) and -271 (CKDE). Higher-order kinematic chains improve on the results by another $+6.7$ (CKDE) and $+3.1$ (GL) nats. The best graph structure in this comparison are Chow-Liu trees. Their usage results in a big leap in performance, increasing the results again by +8.9 nats in case of the CKDE network and +23.9 nats in case of the GL network. The direct comparison of CKDE- to GL networks is unambiguous: CKDE networks perform consistently better, independent of the graph structure. The combination Chow-Liu/CKDE also performs better than all 3 global methods, making it the best performing approach in this comparison.

3.3. Compositionality

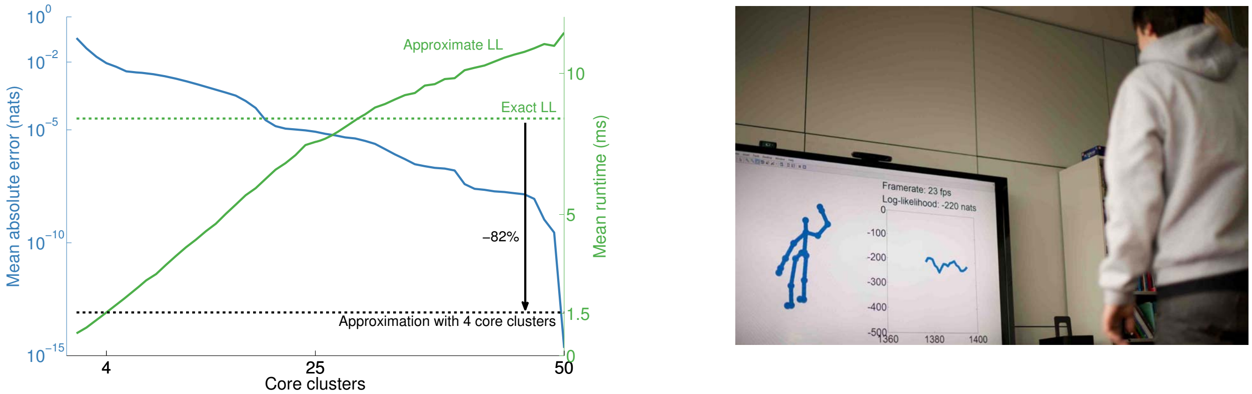
One of the major disadvantages of global non-parametric models is their susceptibility to overfitting; they basically represent and reproduce the training samples. On the other hand, parametric networks based on the kinematic chain are too flexible in the sense that they allow arbitrary combinations of the position of different limbs. This is because different limbs are conditionally independent once their lowest common ancestor w.r.t. the hierarchical tree structure is observed. This is the case, for example, when performing ancestral sampling. At the same time, Gaussian linear networks are not flexible enough in the sense that their local distributions cannot cope with multimodality, which is essential when modeling human pose.

Ideally, we would like to have flexibility and compositionality only where it is adequate and needed. In order to check this property under controlled conditions, we record two different gestures in front of a Kinect: either waving with the left hand only or waving with the right hand only (1000 frames each; Figure 3a (left)). We then learn a pose model according to section 2, draw 5000 samples from it and cluster them into 4 clusters using k-means.

Results. The non-parametrically learned Chow-Liu tree of the model is shown in Figure 3b. Since the arms do not share much information in our example, they are automatically modeled conditionally independent of each other, i.e., we can freely combine their positions. In contrast to that, the position of the right leg tells us a lot about the position of the left leg, since both are parallel to one another throughout the sequence. Consequently, the joint positions of the latter are all modeled conditional on the corresponding joint positions of the former.

The samples generated by this model (Figure 3a (right)) fall into 4 distinct pose classes. Two of the four clusters (coloured in purple and red) correspond to poses also present in the training set. The other two clusters (coloured in blue and green) represent newly learned poses that do not appear in the training data: a neutral pose (both hands lowered) and a pose with both hands raised. The key here is that we do have samples with the left and right hand raised, just never in the same frame. During the sampling process, our model combines the available data to form a new sample that possibly does not resemble any training sample.

In summary, our formulation allows to freely combine substructures, but only if they do not share a lot of information. Joint positions that heavily depend on each other, for instance due to physical constraints, will always be modeled conditionally on each other. Examples include positions of the feet (gravity) and the hips (rigidity), i.e., we get compositionality exactly where we need it.



(a) Accuracy vs. Speed.

(b) Live scoring in front of a Kinect.

Figure 4: Approximation trade-off and real-time inference. In **(a)**, we show the mean accuracy and speed of all approximate local log-likelihoods as a function of the number of core clusters. For comparison we also include the runtime for the computation of exact local log-likelihoods. The black line illustrates the situation for a tolerable ϵ-error of 10^{-2}, corresponding to 4 core clusters. The reduction in computational cost of 82% is big enough to allow real-time applications, shown in **(b)**.

4. Real-time Scoring

Time is a critical factor in applications such as tracking or pose estimation. Our presentation in section 2 has shown that the computation of exact log-likelihoods in our model is tractable, i.e., we do not have to resort to MCMC or similar methods. However, for large datasets the number N of terms in the summations (5a) and (6a) will also become large, resulting in longer runtime. We can speed up inference by considering approximate log-likelihoods. There exist many fast methods for the evaluation of kernel density estimates, but the popular approaches are either not suited for high-dimensional data [7], do not lead to a speed-up for sequential data [26] or are hard to implement due to their complexity [9]. Here, we want to propose a simple alternative to these complex methods that will be sufficient for our purpose. Our experiments will show that the additional decrease in runtime is sufficient to allow the application of our approach in real-time, without having to sacrifice accuracy.

At training time, we cluster all training points into clusters $C_1, \ldots, C_k$ using k-means and build a kd-tree for the cluster centers. At test time, we partition the clusters into a set of core clusters C^e and a set of approximate clusters C^a based on the following scheme: Given a test pose $\mathbf{x} \in \mathbb{R}^d$, we use the kd-tree to determine the clusters whose centers lie closest to $\mathbf{x}$. These make up the core clusters. All remaining clusters are considered approximate clusters. We then evaluate all training points within the core clusters exactly. All the other clusters are evaluated by multiplying the log-likelihood w.r.t. the center with the size of the cluster. The sum in equation (5) thus decomposes into an exact and an approximate sum,

$$p\left(\mathbf{x}\right) = \frac{S_e + S_a}{N|B|}, \tag{10}$$

with

$$S_e = \sum_{C \in C_e} \sum_{j \in C} \kappa\left(B^{-1}\left(\mathbf{x} - \mathbf{x}^{(j)}\right)\right), \tag{11}$$

$$S_a = \sum_{C \in C_a} |C| \kappa\left(B^{-1}\left(\mathbf{x} - \overline{C}\right)\right), \tag{12}$$

where $\overline{C}$ and $|C|$ denote the center and size of cluster C, respectively. In this formulation, those training points contributing most to the log-likelihood are evaluated exactly and those farther away are approximated by their corresponding cluster centers. As the number of core clusters approaches the total number of clusters (or as the number of total clusters approaches the total number of training points), our approximate method converges to the exact log-likelihood.

Since the contribution of a training point to the log-likelihood decreases exponentially with its distance from the test point, a few core clusters should suffice to achieve a high level of accuracy. In order to prove this point, we cluster the entire Human 3.6M training set into 50 clusters, evaluate approximate log-likelihoods for 100 randomly sampled points from the Human 3.6M test set and compare them to their exact counterparts. Figure 4a shows the results in terms of accuracy and speed for a local log-likelihood: If an absolute error of 10^{-2} nats is acceptable, we need as few as 4 core clusters and the runtime is 1.5ms per frame. This compares to 8.4ms for the computation of an exact local log-likelihood. Adding more core clusters further decreases the error, while the runtime increases sublinearly. As the evaluation of a log-likelihood for a Bayesian network in our case requires computation of $2n = 40$ local log-likelihoods (see equation (7)), we achieve a total speed of approx. 61ms per frame (16 fps) on a dataset containing about $70,000$ training points.

This work is accompanied by an open source Matlab suite for Kinect data.[3] Our framework supports recording, training and real-time evaluation of Kinect poses (Figure 4b), making the creation of datasets and integration of the proposed model as part of a larger pipeline very easy.

5. Conclusion

We have introduced a fully non-parametric Bayesian network model of human pose. In order to learn the network structure, we have used a continuous variant of the Chow-Liu tree, in which we have obtained the required estimates of mutual information by means of a non-parametric entropy estimator. We have shown that our model allows for efficient sampling and calculation of exact log-likelihoods.

In our experiments, we have demonstrated that the proposed model achieves a higher expected log-likelihood on the Human 3.6M test set than the 3 global baselines and a Gaussian linear network. The comparison of different graph structures has shown that our non-parametric approach to structure learning outperforms the widely used kinematic chain and also a higher-order variant thereof by a significant margin. We have further illustrated the capabilities of our model to generalize to new poses not present in the training data (compositionality). Finally, we have introduced a fast and accurate method for the computation of approximate log-likelihoods, allowing the application of our approach in real-time.

We expect widespread applicability in domains such as tracking, pose estimation and pose denoising.

Acknowledgements The first author was supported by Microsoft Research through its PhD Scholarship Programme.

References

[1] D. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012. 2, 3

[2] C. Bregler and J. Malik. Tracking people with twists and exponential maps. *CVPR*, 1998. 2

[3] T. Brox, B. Rosenhahn, U. G. Kersting, and D. Cremers. Nonparametric density estimation for human pose tracking. *DAGM*, 2006. 2

[4] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory*, 1968. 3

[5] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2009. 3

[6] C. Ek, P. Torr, and N. Lawrence. Gaussian Process Latent Variable Models for Human Pose Estimation. *MLMI*, 2007. 2

[7] A. Elgammal, R. Duraiswami, and L. Davis. Efficient kernel density estimation using the fast Gauss transform with applications to color modeling and tracking. *PAMI*, 2003. 7

[8] M. Goria, N. Leonenko, V. Mergel, and P. Novi-Inverardi. A new class of random vector entropy estimators and its applications in testing statistical hypotheses. *Journal of Nonparametric Statistics*, 2005. 3

[9] A. Gray and A. Moore. Nonparametric density estimation: Toward computational tractability. *SIAM Data Mining*, 2003. 7

[10] S. Hauberg, S. Sommer, and K. S. Pedersen. Gaussian-like spatial priors for articulated tracking. *ECCV*, 2010. 2

[11] K. Ickstadt, B. Bornkamp, M. Grzegorczyk, J. Wieczorek, M. Sheriff, H. Grecco, and E. Zamir. Nonparametric bayesian networks. *Bayesian Statistics*, 2010. 2

[12] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. Technical report, University of Bonn, 2012. 1, 4

[13] D. Koller and N. Friedman. *Probabilistic Graphical Models*. MIT Press, 2009. 1, 2, 3, 4, 5

[14] L. Kozachenko and N. Leonenko. Sample estimate of the entropy of a random vector. *Problems of Information Transmission*, 1987. 3

[15] X. Lan and D. P. Huttenlocher. Beyond trees: Common-factor models for 2D human pose recovery. *ICCV*, 2005. 2

[16] N. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *NIPS*, 2003. 2, 5

[17] M. W. Lee and I. Cohen. Human upper body pose estimation in static images. *ECCV*, 2004. 2

[18] Z. Lu, M. Á. Carreira-Perpiñán, and C. Sminchisescu. People tracking with the Laplacian eigenmaps latent variable model. *NIPS*, 2007. 2

[19] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 1992. 3

[20] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. F. R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. *CVPR*, 2011. 1, 4

[21] L. Sigal, M. Isard, H. W. Haussecker, and M. J. Black. Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation. *IJCV*, 2012. 2

[22] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. *NIPS*, 2006. 5

[23] G. W. Taylor, L. Sigal, D. J. Fleet, and G. E. Hinton. Dynamical binary latent variable models for 3D human pose tracking. *CVPR*, 2010. 2

[24] R. Urtasun, D. Fleet, and P.Fua. 3D people tracking with Gaussian process dynamical models. *CVPR*, 2006. 2

[25] Q. Wang, S. Kulkarni, and S. Verdú. Universal estimation of information measures for analog sources. *Found. Trends Commun. Inf. Theory*, 2009. 3

[26] C. Yang. Improved fast Gauss transform and efficient kernel density estimation. *ICCV*, 2003. 7

[27] B. Yao and F.-F. Li. Modeling mutual context of object and human pose in human-object interaction activities. *CVPR*, 2010. 2

[3] `http://ps.is.tue.mpg.de/person/lehrmann`