Learning discontinuities for switching between local models [*]

Marc Toussaint and **Sethu Vijayakumar**
Institute of Perception, Action and Behavior
School of Informatics, University of Edinburgh
The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK.
mtoussai@inf.ed.ac.uk, sethu.vijayakumar@ed.ac.uk

1 Introduction

Locally weighted learning techniques, in particular LWPR [Vijayakumar *et al.*, 2002], have successfully been used for high-dimensional regression problems. Their robustness and efficient online versions are crucial in robotic domains where, for instance, an inverse model of an articulated dynamic robot has to be learned in real-time. Such models map a high-dimensional state (e.g., joint angles and velocities) and a desired change of state to the required motor signals (torques).

While typically such mappings are assumed to be smooth, in real world scenarios, there are many interesting cases where the functions of interest are truly discontinuous. Some examples include contacts with other objects (in particular the ground), with other parts of the body, or with "joint limits". In fact, many interesting interactions with the environment manifest themselves through discontinuities in the sensorimotor data.

In this paper, we show how discontinuous switching between local regression models can be learned. The general topic of switching models has been discussed before, e.g., in the context of state space models [Ghahramani and Hinton, 1998; Pavlovic *et al.*, 2000] or multiple inverse models [Wolpert and Kawato, 1998]. Generally, the question of which particular model receives responsibilities for a given input can be modeled as a hidden variable i in a generative mixture model. In our case, we assume that the responsibility index i can be predicted from the input (the robot state). Thus, inferring a model for i corresponds to classifying the input domain into regions for each sub-model.

Since in robotic domains, local learning is crucial to prevent interference and allow for online adaptation techniques, we propose a model of the responsibility index i which is itself a composition of local classifiers. Multiple pairwise classifiers are concatenated to construct a complete model in the form of a product-of-sigmoids, which is capable of learning complex, sharply bounded domains for each local model in lieu of the typical Gaussian kernels.

2 Learning a family of models

Given training data $\{(\boldsymbol{x}_k, y_k)\}_{k=1}^M$ with inputs $\boldsymbol{x}_k$ and outputs y_k, the first level goal of our algorithm is to learn a *family* of models $\{\phi_1, .., \phi_N\}$ such that every datum can be explained by at least one model. The problem of predicting which particular model ϕ_i is responsible for a given input $\boldsymbol{x}$ is solved on a higher level as explained in the next section. In formal notations, we assume a mixture model

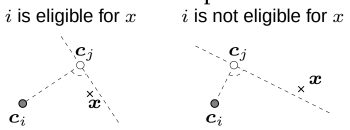
$$P(y|\boldsymbol{x}) = (1-\epsilon) \sum_{i=1}^N P_{\text{loc}}(i|\boldsymbol{x})\; P(y|i,\boldsymbol{x}) + \epsilon\, \mathcal{U}(y) \;,$$

$$P(y|i,\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left\{-\frac{|y-\phi_i(\boldsymbol{x})|^2}{2\,\sigma^2}\right\} \;,$$

where $\mathcal{U}(y)$ is a uniform distribution accounting for background noise (e.g. outliers) and i is the hidden variable specifying the particular model that generates a datum. Since we aim for localized models, we impose a locality constraint already at this level as follows: Let $\boldsymbol{c}_i$ denote the mean input (*center*) on which model ϕ_i has been trained on. For a given input $\boldsymbol{x}$, the ith model is eligible if and only if there does not exist a jth model which has its center "between" $\boldsymbol{c}_i$ and $\boldsymbol{x}$. More precisely,

$$P_{\text{loc}}(i|\boldsymbol{x}) = 0 \iff \exists j : \langle \boldsymbol{x}-\boldsymbol{c}_j, \boldsymbol{c}_i-\boldsymbol{c}_j\rangle < 0 \;,$$

where $\langle\cdot,\cdot\rangle$ is the scalar product in input space:



Further, $P_{\text{loc}}(i|\boldsymbol{x})$ is uniform over all eligible i's. Given a current family of models, we can infer a posterior on the responsibility index i for a given datum $(\boldsymbol{x}, y)$, using Bayes rule:

$$P(i|y,\boldsymbol{x}) = \frac{1}{Z}\, P(y|i,\boldsymbol{x})\; P_{\text{loc}}(i|\boldsymbol{x}).$$

Calculating the MAP assignment $\hat{i}$ allows us to associate every training datum with the most likely model. Using this, the sufficient statistics of each local model ϕ_i is updated. Further, the data that is labeled as "yet unmodeled" (which is inferred to have been generated by $\mathcal{U}(y)$) is used to generate a new family member by the following heuristic (compare RANSAC): A random datum $(\boldsymbol{x}, y)$ is selected from the unmodeled data; the K closest neighbors of $(\boldsymbol{x}, y)$ (w.r.t., Euclidean input distance) are chosen as initial training data for the new model, where K is a random Poisson number with mean $3\,d$ (here, d is the input dimensionality). Finally, models that receive too few MAP responsibilities (less than $10\,d$ in our experiments) are discarded. This iterative process can be repeated until no new models are generated.

[*] The first author acknowledges support by the German Research Foundation (DFG), Emmy Noether fellowship TO 409/1-1.

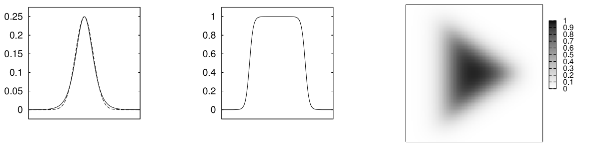


Figure 1: Kernels that can be represented as a product of sigmoids in 1D and 2D.

This general scheme of family learning can be realized with any type of models ϕ_i. In the experiments, we will choose ϕ_i to be linear functions, learned with Partial Least Squares (PLS) regression. PLS, involving an intermediate lower-dimensional projection, has been proven efficient for high-dimensional problems [Vijayakumar *et al.*, 2002].

3 Products of sigmoids for switching

On the second level of our algorithm, the goal is to learn a predictive model $P(i|\boldsymbol{x})$ of the latent responsibility index i that is more precise than the uninformed prior $P_{\text{loc}}(i|\boldsymbol{x})$. Given some data, it is easy to decide whether two models are "potentially neighbored"—namely whether there exists data for which both models are eligible—based on their centers. For each pair (ij) of neighbored models, we learn a sigmoidal function $\psi_{ij}(\boldsymbol{x})$, where $\psi_{ij} \equiv 1 - \psi_{ji}$. The product of such sigmoids around a submodel i defines a coefficient $\beta_i(\boldsymbol{x})$ that we associate with the submodel for a given input $\boldsymbol{x}$,

$$\beta_i(\boldsymbol{x}) = \frac{1}{Z'} \prod_j \psi_{ij}(\boldsymbol{x}) \ , \quad \psi_{ij}(\boldsymbol{x}) = \frac{1}{1 + \exp[-\phi_{ij}(\boldsymbol{x})]} \ .$$

Here, Z' normalizes β_i over i. As indicated, we represent sigmoids ψ_{ij} with a scalar function ϕ_{ij}. Fig. 1 illustrates the kind of kernels can be represented as products of sigmoids.

The sigmoids $\psi_{ij}(\boldsymbol{x})$ are meant to represent the likelihood that a model i rather than j is responsible for an input $\boldsymbol{x}$, conditioned on that either i or j is responsible. The product combination is comparable to an AND voting. The MAP labeling $\hat{\imath}$ we introduced in the previous section is now used to train these sigmoids. In the experiments, we consider ϕ_{ij} to be linear functions, again learned with PLS.

4 Experiments

We tested the algorithm on piecewise linear, discontinuous test functions. A test function has 3 parameters: the input dimension d, the number L of linear pieces it is composed of and the output noise σ. The localities, slopes and boundaries of the linear pieces are sampled randomly. Fig. 2(a,b) display learning results from a 1D example in comparison to LWPR. Fig. 2(c,d) display two error curves on 10-dimensional test functions over the rather large input domain $[-1, 1]^{10}$. The *family error* is the MSE of the best fitting eligible model $\phi_{\hat{\imath}}$ (averaged over an independent test data set); the *classification error* counts how often the product of sigmoids correctly predicts $\phi_{\hat{\imath}}$ to be the best fitting model for a given input (i.e., $\text{argmax}_i \beta_i = \hat{\imath}$). In the experiments we find that the algorithm reliably generates a family with optimal family error at the noise level ($\sigma^2 = 0.01$). In 5 dimensions (not displayed here) the classification error rapidly converges to zero while in 10 dimensions, the classification error converges to around 4%. For more results see homepages.inf.ed.ac.uk/mtoussai/projects/05-ijcai.

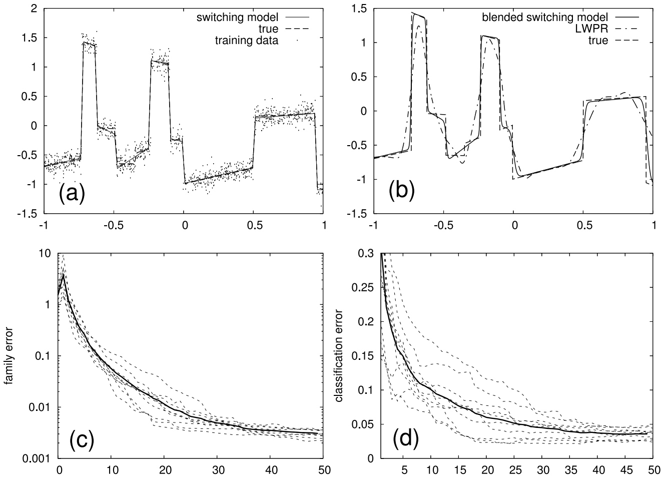


Figure 2: (a) A 1D test function with d=1, L=10, σ=0.1. Learned switching model after 20 iterations on M=1000 training data points. (b) The blended switching model: $y(\boldsymbol{x}) = \sum_i \beta_i(\boldsymbol{x}) \, \phi_i(\boldsymbol{x})$ compared to LWPR. (c) Family error (cf. Sec. 4) and (d) classification error for 10 runs on random test functions with d=10, L=10, σ=0.1, and M=10 000. The bold line is the average over all curves.

5 Discussion

The presented model addresses the problem of handling the discontinuities that naturally arise, e.g., in sensorimotor data during interaction with a structured environment. Our model extends earlier local learning approaches in several ways: The responsibility region associated with each local model (learned with the product of sigmoids) has a much more versatile boundary shape compared to typical Gaussian kernels. Problems associated with initialization of kernel shapes or widths and the heuristic choice of an ad hoc number of submodels are circumvented by the robust incremental allocation of new models. Although we consistently used PLS as the underlying regression machinery, the general model allows to utilize any efficient single model learner to represent the local models ϕ_i as well as the classifier functions ϕ_{ij}. Future work will in particular investigate non-linear learners for the local models as well as the boundary classifiers.

References

[Ghahramani and Hinton, 1998] Z. Ghahramani and G.E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12:963–996, 1998.

[Pavlovic *et al.*, 2000] Vladimir Pavlovic, James M. Rehg, and John MacCormick. Learning switching linear models of human motion. In *NIPS*, pages 981–987, 2000.

[Vijayakumar *et al.*, 2002] Sethu Vijayakumar, Aaron D'Souza, Tomohiro Shibata, Jorg Conradt, and Stefan Schaal. Statistical learning for humanoid robots. *Autonomous Robot*, 12:55–69, 2002.

[Wolpert and Kawato, 1998] D.M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1998.