REGULARISING AN ADAPTATION ALGORITHM FOR TONGUE SHAPE MODELS

Mohsen Farhadloo *Miguel Á. Carreira-Perpiñán*

EECS, School of Engineering, University of California, Merced. Merced, CA, USA
Email: {mfarhadloo,mcarreira-perpinan}@ucmerced.edu

ABSTRACT

Realistic data-driven models of the tongue shape can be obtained by learning a nonlinear mapping from tongue landmarks to full contours, trained on a dataset of thousands of contours. Semiautomatic contour extraction from ultrasound takes a lot of time and effort from an expert, so practically it is preferable to adapt a reference model given just a few contours from the new speaker. However, adaptation with very few contours is unreliable and prone to overfitting. We study several forms of regularisation to constrain the adaptation, and determine the optimal amount of regularisation by leave-one-out cross-validation. Our results show that good accuracy models can be found reliably with no user intervention.

Index Terms— tongue model, model adaptation, regularisation, articulatory databases, ultrasound.

1. INTRODUCTION

Models of the shape of the vocal tract, in particular the tongue, are useful in applications such as articulatory synthesis and inversion, animation and visualisation, tracking in biological imaging, and others. Data-driven models, estimated using a collection of thousands of recorded tongue contours (e.g. with ultrasound), allow one to estimate realistic models of the midsagittal tongue contour. In particular, given the location on the tongue contour of 3–4 fleshpoints (landmarks), one can estimate a predictive mapping from the landmarks to the full contour [1, 2, 3]; see fig. 1. Nonlinear mappings [3] achieve errors of 0.2–0.3 mm per point on the tongue, below the ultrasound measurement error (around 0.4 mm). Since collecting contours is costly, it is convenient to adapt automatically a predictive mapping trained on lots of data from one speaker to a new speaker given only a few full contours from the latter. This can be achieved using a feature-transformation approach [4, 5], resulting in errors just slightly larger than training with a large dataset (0.1–0.3 mm more). We are interested in the most useful regime in practice, namely where we have very few adaptation contours (in order that little effort and time is spent in recording and segmenting contours). The adaptation idea of [4, 5] is a feature-transformation approach [6] in a regression setting: we use linear invertible transformations between the reference and target speakers, so that to predict a tongue contour from the target speaker, we first map it linearly to the reference speaker space, then apply its predictive mapping there, and finally map back (with the inverse linear transformation) to the target contour. In [4] it was proposed to use a unique, global transformation for every contour point. This has only 6 free parameters in total and does very robustly with as few as 5 adaptation contours, but its accuracy stagnates quickly with more adaptation contours, leaving an significant accuracy gap compared to retraining a model with those same contours. To correct this, in [5] a purely local approach was proposed, where each contour point has a separate, local linear transformation. This has now quite more parameters and does well when more contours are available, but it is now more prone to overfitting and becomes unreliable with less than 10 contours, requiring regularisation.

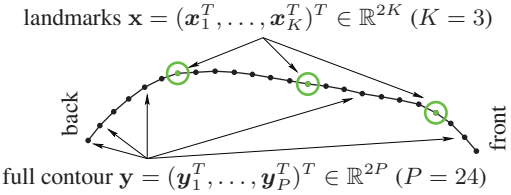

Fig. 1. Prediction problem: given the 2D locations of K landmarks on the tongue midsagittal contour $\mathbf{x}$, reconstruct the entire contour $\mathbf{y}$, represented by P 2D points, by a predictive mapping $\mathbf{y} = \mathbf{f}(\mathbf{x})$.

In this work we consider the local transformation model of [5] and seek two objectives: (1) to regularise its objective function so as to enjoy the benefits of both approaches, by establishing a spectrum between the global approach with few contours and the purely local one with many contours. (2) To determine automatically a point in that spectrum that is optimal for the adaptation dataset provided. These are described in section 4 of the paper, and we evaluate the regularisers (as well as a condition-number regulariser) and leave-one-out cross-validation experimentally in section 5. In addition, section 3 also shows that the adaptation objective function, which appears to suffer from ill-conditioning, can be improved with a suitable rescaling of the variables. We start with a review of the algorithms for learning and adapting the tongue shape predictive model in section 2.

2. LEARNING AND ADAPTING A PREDICTIVE MODEL OF TONGUE SHAPES

2.1. The predictive model of tongue shapes

The goal in this problem is to learn a function which is able to predict the full tongue contour $\mathbf{y} = (\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_P^T)^T \in \mathbb{R}^{2P}$ consisting of P points $\boldsymbol{y}_i \in \mathbb{R}^2$ given only the positions $\mathbf{x} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_K^T)^T \in \mathbb{R}^{2K}$ of K landmarks $\boldsymbol{x}_i \in \mathbb{R}^2$ (fig. 1). The approach proposed in [1] for *linear mappings* and in [3] for *radial basis function (RBF) networks* fits a predictive mapping $\mathbf{f}$ by minimizing the predictive square error $E(\mathbf{f}) = \sum_{n=1}^{N} \|\mathbf{y}_n - \mathbf{f}(\mathbf{x}_n)\|^2$, given a sufficiently large training set, and $\mathbf{f}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{w}$ (linear) or $\mathbf{f}(\mathbf{x}) = \mathbf{W}\Phi(\mathbf{x}) + \mathbf{w}$ (RBF) with M basis functions $\phi_m(\mathbf{x}) = \exp\left(-\frac{1}{2}\|(\mathbf{x} - \boldsymbol{\mu}_m)/\sigma\|^2\right)$. The RBF is trained in an efficient but slightly suboptimal way (as commonly done) by fixing the centers $\boldsymbol{\mu}_m$ by k–means and cross-validating the width σ and the regularisation parameter λ.

2.2. Adaptation with local transformations

Given a small N–contour adaptation dataset $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$, we adapt an existing predictive mapping $\mathbf{f}$ by estimating two invertible linear mappings $\mathbf{g_x}$ and $\mathbf{g_y}$ (with few parameters) that transform the data between the new and old speaker spaces. Each mapping $\mathbf{g}$ is defined as a concatenation of separate, local linear mappings that map a 2D point to another 2D point:

$$\tilde{\mathbf{x}} = \mathbf{g_x}(\mathbf{x}) = \begin{pmatrix} \mathbf{A}_1^{\mathbf{x}}\boldsymbol{x}_1 + \mathbf{b}_1^{\mathbf{x}} \\ \cdots \\ \mathbf{A}_K^{\mathbf{x}}\boldsymbol{x}_K + \mathbf{b}_K^{\mathbf{x}} \end{pmatrix}, \; \tilde{\mathbf{y}} = \mathbf{g_y}(\mathbf{y}) = \begin{pmatrix} \mathbf{A}_1^{\mathbf{y}}\boldsymbol{y}_1 + \mathbf{b}_1^{\mathbf{y}} \\ \cdots \\ \mathbf{A}_P^{\mathbf{y}}\boldsymbol{y}_P + \mathbf{b}_P^{\mathbf{y}} \end{pmatrix}.$$

The adapted predictive mapping is given by $\mathbf{g}_{\mathbf{y}}^{-1} \circ \mathbf{f} \circ \mathbf{g_x}$ and requires estimating $6(K+P)$ parameters that we write collectively as $(\mathbf{A}^{\mathbf{x}}, \mathbf{b}^{\mathbf{x}}, \mathbf{A}^{\mathbf{y}}, \mathbf{b}^{\mathbf{y}})$. The adapted model is linear if $\mathbf{f}$ was linear, and a basis function network where the basis functions are non-radial if $\mathbf{f}$ was a radial basis function network. In the global transformation method of [4], $\mathbf{A}_i^{\mathbf{x}} = \mathbf{A}_j^{\mathbf{y}} = \mathbf{A}$ and $\mathbf{b}_i^{\mathbf{x}} = \mathbf{b}_j^{\mathbf{y}} = \mathbf{b}$, so there were only 6 parameters. To estimate $(\mathbf{A}^{\mathbf{x}}, \mathbf{b}^{\mathbf{x}}, \mathbf{A}^{\mathbf{y}}, \mathbf{b}^{\mathbf{y}})$, we minimize the predictive squared error:

$$\min E(\mathbf{A}^{\mathbf{x}}, \mathbf{b}^{\mathbf{x}}, \mathbf{C}^{\mathbf{y}}, \mathbf{d}^{\mathbf{y}}) = \sum_{n=1}^{N} \left\| \mathbf{y}_n - \mathbf{g}_{\mathbf{y}}^{-1}\mathbf{f}(\mathbf{g_x}(\mathbf{x}_n)) \right\|^2$$

where we introduce new parameters $\mathbf{C}_j^{\mathbf{y}}, \mathbf{d}_j^{\mathbf{y}}$, so we work with

$$\mathbf{y} = \mathbf{g}_{\mathbf{y}}^{-1}(\tilde{\mathbf{y}}) = \begin{pmatrix} \mathbf{C}_1^{\mathbf{y}}\tilde{\mathbf{y}}_1 + \mathbf{d}_1^{\mathbf{y}} \\ \cdots \\ \mathbf{C}_P^{\mathbf{y}}\tilde{\mathbf{y}}_P + \mathbf{d}_P^{\mathbf{y}} \end{pmatrix} \qquad \begin{aligned} \mathbf{C}_j^{\mathbf{y}} &= (\mathbf{A}_j^{\mathbf{y}})^{-1} \\ \mathbf{d}_j^{\mathbf{y}} &= -(\mathbf{A}_j^{\mathbf{y}})^{-1}\mathbf{b}_j^{\mathbf{y}} \end{aligned}$$

instead of $\mathbf{g_y}$, simplifying the optimization (no matrix appears as an inverse).

3. EFFECTS OF RESCALING

During running some preliminary experiments the following observations conducted us towards the fact that our problem is suffering from the bad scaling: slowness of the optimization algorithm (although the BFGS algorithm is expected to be superlinear, the algorithm was so slow), ill-conditioned Hessian, absolutely small step size and tiny deviation from the initial point for portion of the parameters and huge deviation from the initial point for the rest at the end of the optimization. So we decided to normalize the data to have zero mean and approximately unit variance along each dimension (by subtracting the global mean and dividing by the trace of the covariance matrix). The experiments with the normalized data were much faster and less ill-conditioned. For $N = 100$ contours the RBF adaptation algorithm has 160 iterations which takes 5.3 minutes for un–normalized dataset and 2.11 minutes for the normalized one.

4. REGULARISERS

We study three different regularisers in this paper, one that penalises high condition number transformations, and two that encourage all transformation matrices to be identical.

Condition number regulariser According to global feature normalization with partial contours [8], using a regularisation term makes the reconstructed contours more realistic. The motivation behind that particular choice of regularisation was the expectation that the ideal transformation should not be too far from a simple translation, rotation or sheering (which has $\mathrm{cond}\,(\mathbf{A}) = 1$). So choosing a regulariser that penalises the high-condition-number transformations is not a bad choice (Equation(1) which we call it Condition Number Regularizer (Cond#)).

$$\lambda \mathcal{C}(\mathbf{A}^{\mathbf{x}}, \mathbf{C}^{\mathbf{y}}) = \lambda_1 \left(\sum_{i=1}^{K} C(\mathbf{A}_i^{\mathbf{x}}) + \sum_{i=1}^{P} C(\mathbf{C}_i^{\mathbf{y}}) \right) \qquad (1)$$

Directly minimizing $C(\mathbf{A}) = \mathrm{cond}\,(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2$ is difficult, so [8] used instead the much simpler

$$C(\mathbf{A}) = \mathrm{tr}\,(\mathbf{A}^T\mathbf{A}) - D \det(\mathbf{A}^T\mathbf{A})^{1/D} \quad \text{for } \mathbf{A}_{D\times D}, \qquad (2)$$

which satisfies $C(\mathbf{A}) \geq 0$ and $C(\mathbf{A}) = 0$ iff $\mathrm{cond}\,(\mathbf{A}) = 1$ (so it is minimal when $\mathrm{cond}\,(\mathbf{A})$ is minimal), and is piecewise quadratic for $D = 2$. Empirically this is true to some extent, and for good values of λ_1, good solutions can be achieved. However, we observe that not all matrices have so low condition number, so imposing this assumption could be too restrictive.

Chain regulariser In adaptation with local feature transformation [5], we observed that using the same regularisation term as Equation(1) leads to two contradictory effects in two regions. When the number of adaptation contours is small, regulariser causes reduction in the value of point-wise RMSE and its variance, but when the number of adaption contours is large, it causes increase in the value of point-wise RMSE. Also we have observed that in few-contour region the global feature normalization method performs better than its local counterpart. The key aspect of global adaptation method is the application of a similar transformation to all landmark and contour points. Therefore it is a good idea to select a regularisation term that requires all transformations in the local adaptation method, be close to each other when the number of available contours is small. One choice for this purpose is the Chain Regularizer (Chain):

$$R_2(\mathbf{A}^{\mathbf{x}}, \mathbf{C}^{\mathbf{y}}) = \lambda_2(R_2(\mathbf{A}^{\mathbf{x}}) + R_2(\mathbf{C}^{\mathbf{y}})) \qquad (3)$$
$$R_2(\mathbf{A}^{\mathbf{x}}) = \|\mathbf{A}_2^{\mathbf{x}} - \mathbf{A}_1^{\mathbf{x}}\|^2 + \|\mathbf{A}_3^{\mathbf{x}} - \mathbf{A}_2^{\mathbf{x}}\|^2 + \cdots + \|\mathbf{A}_K^{\mathbf{x}} - \mathbf{A}_{K-1}^{\mathbf{x}}\|^2$$

This allows $\mathbf{A}_1$ to drift from $\mathbf{A}_K$ (even if each consecutive pair of matrices are quite close) and so to have somewhat different transformations for the tip and back of the tongue.

Variance regulariser Another regularisation term that is also capable to force all the transformations be close to each other is what we call Variance Regularizer (Var):

$$R_3(\mathbf{A}^{\mathbf{x}}, \mathbf{C}^{\mathbf{y}}) = \lambda_3(R_3(\mathbf{A}^{\mathbf{x}}) + R_3(\mathbf{C}^{\mathbf{y}})) \qquad (4)$$
$$R_3(\mathbf{A}^{\mathbf{x}}) = \mathrm{tr}\,(\mathrm{cov}(\mathrm{vec}\,(\mathbf{A}_1^{\mathbf{x}}), \ldots, \mathrm{vec}\,(\mathbf{A}_K^{\mathbf{x}})))$$

where $\mathrm{vec}\,(\mathbf{A})$ concatenates the columns of a matrix $\mathbf{A}$ into a single column vector. This encourages all matrices to be similar no matters where on the tongue they are.

One could expect that in both Chain and Var regularisers, when the regularisation parameters are very large, they would derive the optimisation algorithm toward transformations which are identical. This allows us to use the $\mathbf{E}$ objective function rather than the proximity objective function $\mathbf{F}$ which was used in the global feature transformation (in the global method, since all the variables are coupled optimising of $\mathbf{E}$ is difficult). We should notice that the chain and variance regularisers are quadratic, so they do not make the optimisation problem, more nonlinear than it already is. However, they couple together all the matrices $\{\mathbf{A}_i\}$ and all the matrices $\{\mathbf{C}_i\}$, which appear decoupled in the Cond# regulariser. However, this makes little difference with BFGS. Also, one may realise that the objective function can be seen as a quadratic-penalty [7] solution for finite penalty of the constrained optimisation problem: $\max_{\mathbf{A}_i, \mathbf{C}_i} E$ s.t. $\mathbf{A}_1 = \cdots = \mathbf{A}_K, \mathbf{C}_1 = \cdots = \mathbf{C}_P$. Thus, for large λ the solution tends to the global method. We compare these three regularisation terms in our experiments.

Leave-One-Out Cross Validation (LOO) In a real situation one may only have access to a few contours that should be used for adaptation and also for finding the best values of the related parameters. A widely used approach in this case would be doing the Leave-One-Out crossvalidation for finding the parameters. As in [9] let $\mathcal{K} : \{1, ..., N\} \mapsto \{1, ..., K\}$ be an indexing function which indicates the partition to which the observation i is allocated by randomisation. Denote by $\hat{\mathbf{f}}^{-k}(\mathbf{x})$ the fitted function, computed with the kth part of the data removed. Then the cross validation estimate of the prediction error is:

$$\mathbf{CV}(\hat{\mathbf{f}}, \lambda) = \frac{1}{N} \sum_{i=1}^{N} L(\mathbf{y}_i, \hat{\mathbf{f}}^{-k(i)}(\mathbf{x}_i, \lambda))$$

In our experiments the tuning parameter λ is one of the λ_1, λ_2 or λ_3 that we introduced in section 4. We find the value for λ which minimises the above equation by looking for it in a particular range of values. We did the LOO for all the three proposed regularisers and the Fig.3 shows the results.

5. EXPERIMENTS

Dataset We use the ultrasound database [3] created at Queen Margaret University and the University of Edinburgh. It contains two speakers (one male, `maaw0`, and one female, `feal0`) with different Scottish accents. Each speaker recorded a set of 20 British TIMIT sentences designed to be phonetically balanced. Recordings for `maaw0` and `feal0` were done in two and one session, respectively. Each tongue contour contains $P = 24$ points for both speakers. We partitioned the data for each speaker to training and testing sets. The male speaker contained 2 236 training frames and 1 491 testing, and the female one contained 4 363 training and 2 909 testing frames.

Results Fig. 2 shows the predictive error per contour point as a function of the number of contours N, for the three types of regularisation that we introduced in section 4: Cond#, Chain and Var. In these experiments we assumed that there were only N contours available for adaptation but the reported errors were based on a separate test dataset to which we had access. With Chain and Var it is possible to find a value for λ which gives small error both in few– and large–contour regions. Therefore we can modulate between purely local ($\lambda = 0$) and purely global ($\lambda \to \infty$) feature transformation methods for Chain and Var but not the Cond# (the error curves in few–contour regions are blown up for Cond#). Both Chain and Var seem to switch continuously from the local case (very low error) to the global one (stagnation error). We can compare these regularisers in terms of final error they achieve and their robustness. In Fig. 2 we define the optimal lower envelope as the minimum achieved error over λ at each value of N . By comparing these lower envelopes we can see for linear adaptation Chain and Var are better than the Cond# and for RBF adaptation all of these three regularisers are comparable (although Var is a bit better than others). In large N region, the Cond# is more robust to the value of regulariser parameter than Chain and Var (all the curves no matter what the λ is, are close together and to the optimal one). However, since the optimal error in large N region is correspondent to $\lambda = 0$, there is no need to regularise in this region. Fig. 3 plots the results of performing the LOO cross validation method when we have only $N = 7$ contours for the adaptation. With small N, the cross-validation curves show significant variability depending on the particular N contours provided (fig. 3 shows two such curves in each plot). However, the location of the optimal λ value does not vary much, which indicates that LOO cross-validation can be used to estimate the optimal amount of regularisation for the adaptation set provided. Also in Fig. 3 the dashed curve is from Fig. 2 and for $N = 7$. The results confirm that even in this practical situation, the value for the λ that LOO cross validation achieves, is close to the best one (for instance for RBF network and Var, the best value for λ is $\lambda_3 = 10^5$ (Fig. 2), which matches with the corresponding results of Fig. 3). It should be noticed that from LOO plots one can eyeball the location of the minimiser (the region in which the minimiser exists in red and blue curves, almost match with the dashed one). The reason that in general the errorbars in Fig. 3 are larger than Fig. 2 is that the results of Fig. 3 are for LOO cross validation which we use only one test vector for computing the test error. However in Fig. 2 as mentioned earlier we had a separate test set. The Var is less sensitive to the exact value of the regulariser parameter than Cond# and Chain (the range that the minimiser exists in Var is wider than Cond# and Chain in Fig. 3). So in general the Var is a better choice to use in a practical problem.

6. CONCLUSION

We have studied several forms of regularisation for adapting a tongue shape model using local feature transformations. While all our regularisers helped to estimate a good accuracy model with very few adaptation contours, we found it most effective to penalise the transformation matrices' variance. This smoothly blends a global adaptation using a single transformation into a local adaptation using a different transformation at each contour point. A near-optimal amount of regularisation for a specific adaptation dataset can be obtained automatically with leave-one-out cross-validation. We have also shown that centering and rescaling the data accelerates the optimisation.
Acknowledgments. Some of the work on section 4 was spurred by comments from Bhiksha Raj and David Noelle. Work funded by NSF award IIS–0711186.

7. REFERENCES

[1] T. Kaburagi and M. Honda, "Determination of sagittal tongue shape from the positions of points on the tongue surface," *J. Acoustic Soc. Amer.*, pp. 1356–1366, Sept. 1994.

[2] P. Badin, E. Baricchin, and A. Vilain, "Determining tongue articulation: From discrete fleshpoints to continuous shadow," in *EUROSPEECH*,1997.

[3] C. Qin, M. Á. Carreira-Perpiñán, K. Richmond, A. Wrench, and S. Renals, "Predicting tongue shapes from a few landmark locations," in *Interspeech*, 2008.

[4] C. Qin and M. Á. Carreira-Perpiñán, "Adaptation of a predictive model of tongue shapes," in *Interspeech*, 2009.

[5] C. Qin, M. Á. Carreira-Perpiñán, and M. Farhadloo, "Adaptation of a tongue shape model by local feature transformations," in *Interspeech*, 2010.

[6] Phil C. Woodland, "Speaker adaptation for continuous density HMMs: A review," in *Adaptation Methods for Speech Recognition, ISCA Tutorial and Research Workshop (ITRW)*,2001.

[7] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer. second edition, 2006.

[8] Chao Qin and M. Á. Carreira-Perpiñán, "Reconstructing the full tongue contour from EMA/X–ray microbeam," in *(ICASSP)*, 2010.

[9] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*, Springer. second edition, 2009.

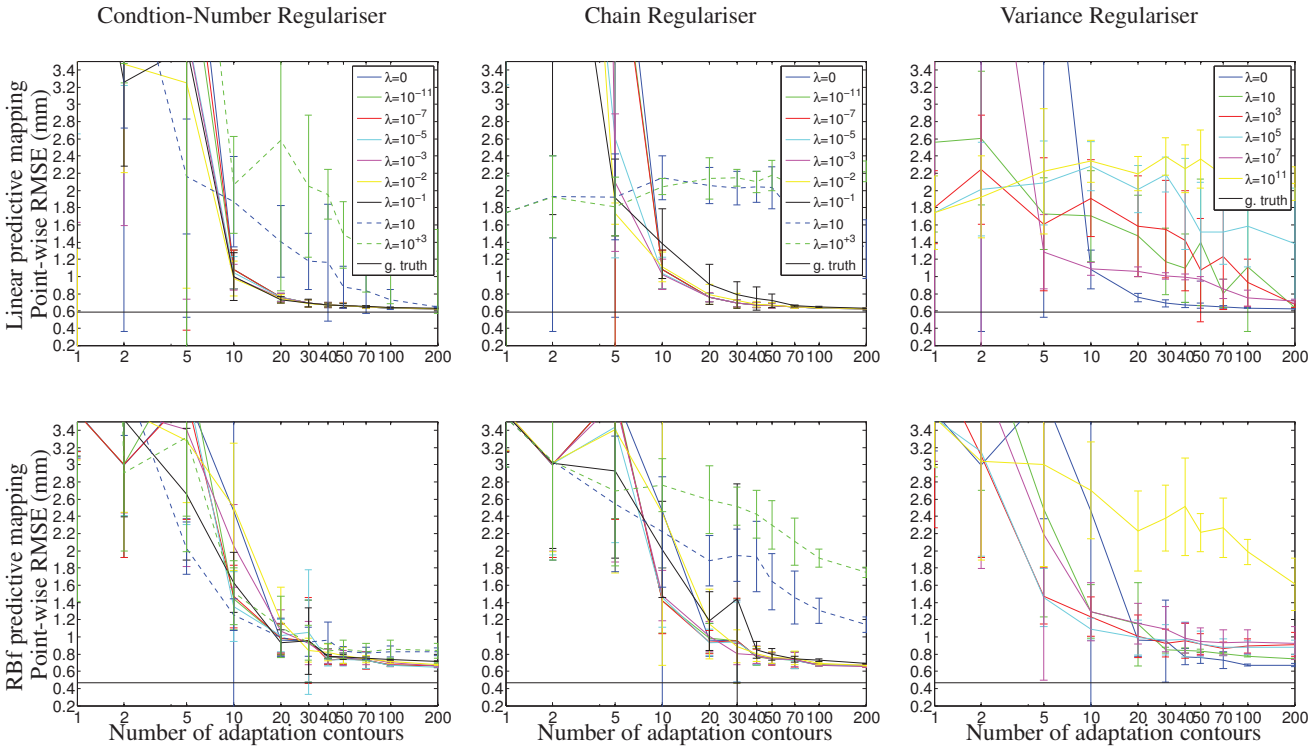

Fig. 2. Predictive error E (as RMSE per contour point in mm) after adaptation as a function of the number of adaptation contours N (for $K = 3$ landmarks). Each curve corresponds to a different value of the λ. Errorbars over 5 random choices of the N.

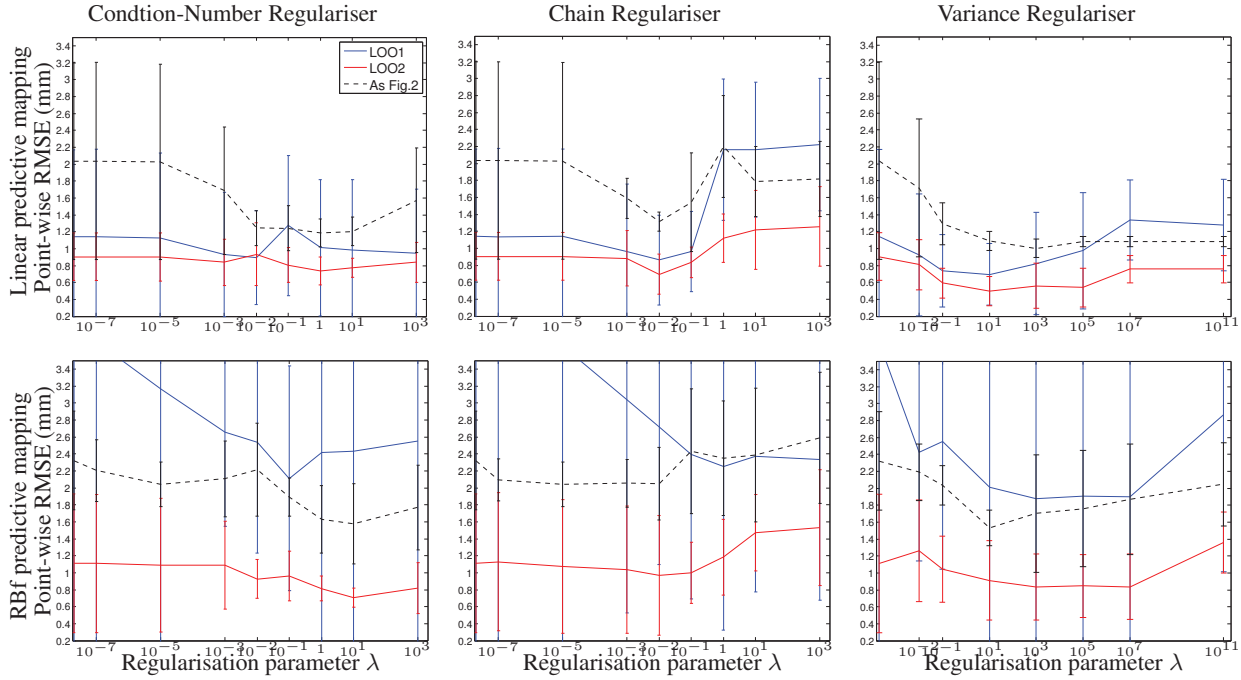

Fig. 3. Leave-one-out cross validation results. Predictive error E (as in fig. 2) after adaptation as a function of the λ (for $N = 7$ adaptation contours) for the normalised dataset. The blue and red curves are each for a different, random set of N contours. The dashed curve is from fig. 2 for $N = 7$.