Bayesian Feature Selection for Hearing Aid Personalization

Citation for published version (APA):

Ypma, A., & Vries, de, B. (2008). Bayesian Feature Selection for Hearing Aid Personalization. In *Bayesian Feature Selection for Hearing Aid Personalization, MLSP-07, Proceedings, Thessaloniki, Greece, 2007* (pp. 425-430) <https://doi.org/10.1109/MLSP.2007.4414344>

DOI:

[10.1109/MLSP.2007.4414344](https://doi.org/10.1109/MLSP.2007.4414344)

Document status and date:

Published: 01/01/2008

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

BAYESIAN FEATURE SELECTION FOR HEARING AID PERSONALIZATION

Alexander Ypma¹, Serkan Özer², Erik van der Werf¹ and Bert de Vries^{1,2}

¹GN ReSound Research, GN ReSound A/S, Horsten 1, 5612 AX Eindhoven

²Signal Processing Systems group, Dept. Electrical Engineering, TU Eindhoven
s.ozero@student.tue.nl, {aypma, evdwerf, bdevries}@gnresound.com

ABSTRACT

We formulate hearing aid personalization as a linear regression. Since sample sizes may be low and the number of features may be high we resort to a Bayesian approach for sparse linear regression that can deal with many features, in order to find efficient representations for on-line usage. We compare to a heuristic feature selection approach that we optimized for speed. Results on synthetic data with irrelevant and redundant features indicate that Bayesian backfitting has labelling accuracy comparable to the heuristic approach (for moderate sample sizes), but takes much larger training times. We then determine features for hearing aid personalization by applying the method to hearing aid preference data.

1. HEARING AID PERSONALIZATION

Modern digital hearing aids contain advanced signal processing algorithms with many parameters. These are set to values that ideally match the needs and preferences of the user. Because of the large dimensionality of the parameter space and unknown determinants of user satisfaction, the fitting procedure becomes a complex task. Some of the user parameters are personalized by the hearing aid dispenser based on the nature of the hearing loss. Other parameters may be tuned on the basis of models for e.g. loudness perception [1]. But not every individual user preference can be put as a preset into the hearing aid: some particularities of the user may be hard to represent into the algorithm, and the user's typical acoustic environments and preference patterns may be changing. Therefore we should personalize a hearing aid during usage to actual user preferences.

The algorithms introduced in [2] are able to learn preferred parameters from control operations of a user. We formulated the personalization problem as a linear regression from acoustic features to preferred hearing aid parameters. It is however not known which features are necessary and sufficient for explaining the user preference. Both the type of the feature and the appropriate time scale at which the feature is computed are unknown. Taking 'just every interesting feature' into account may lead to high-dimensional

input vectors, containing irrelevant and redundant features¹ that make on-line computations expensive and hamper generalization of the model. This is even more of an issue since the number of user adjustments may be small. We therefore choose a Bayesian feature selection scheme that can deal with many features and still obtain a sparse and well-generalizing model for observed preference data (in order to make the on-line sound processing as efficient as possible). We study the behaviour of such a feature selection scheme with synthetic data, and we compare with two benchmark methods for feature selection and linear regression. We then analyse preference data from a listening test, for efficient personalization of a hearing aid algorithm.

Fig. 1. Hearing aid personalization via user control

The personalization problem is illustrated in figure 1. A hearing aid performs a sound processing function $f(i_t, y_t)$ where the functional form depends on the hearing aid parameter y_t . The Automatic Control (AC) unit takes input x_t , which holds a vector of acoustic features from the input signal i_t and maps it to an output v_t . The user will be feeding back corrections to the hearing aid via a control wheel to adjust for an AC-driven parameter value v_t different from the desired value. This user feedback signal e_t is absorbed by a user-specific parameter vector b in order to achieve smaller future corrections from the user (hence: higher user satisfaction). The parameter vector y_t will reflect the user-

¹Irrelevant features do not contribute to the output as such, whereas redundancy refers to features that are correlated with other features (which do not contribute to the output when the correlated features are also present).

preferred parameter value, and is given by $y_t = \mathbf{x}_t^T \mathbf{b} + e_t$. Finding the optimal $\mathbf{b}$ coefficients boils down to a linear regression problem, which can be seen when analyzing the personalizing problem [2].

We now turn to the problem of finding a sparse set of acoustic features for efficient processing on a hearing aid. Since user preferences are expected to change mainly over long-term usage, the coefficients $\mathbf{b}$ are considered stationary for a certain data collection experiment. We therefore propose off-line feature selection based on a set of pre-collected preference data. To emphasize the off-line nature, we will index samples with i rather than t in the rest of the paper.

2. BAYESIAN FEATURE SELECTION

Backfitting [3] is a method for estimating additive linear models of the form $y(\mathbf{x}) = \sum_{m=1}^d g_m(\mathbf{x}; \theta_m)$. The non-linear basis functions g_m absorb parameters b_m as $g_m = b_m f_m$, and they contain auxiliary parameters θ_m that determine the shape of the basis function. Since we assume a linear map, the individual functions $f_m(\mathbf{x}; \theta_m)$ are also linear, and there are no auxiliary parameters θ_m . Backfitting decomposes the statistical estimation problem into d individual estimation problems by creating "fake targets" for each g_m function. It effectively decouples the inference in each dimension, leading to a highly efficient algorithm.

2.1. Probabilistic backfitting

A probabilistic version of backfitting is derived in [4], where variables z_{im} are introduced (figure 2) that act as unknown (fake) targets for each branch of regression input. One assumes that z_{im} and y_i are conditionally Gaussian:

$$y|\mathbf{z}_i \sim \mathcal{N}(\mathbf{1}^T \mathbf{z}_i, \psi_y) = \mathcal{N}\left(\sum_{m=1}^d z_{im}, \psi_y\right) \quad (1)$$

$$z_{im}|\mathbf{x}_i \sim \mathcal{N}(g_m(\mathbf{x}_i), \psi_{zm}) = \mathcal{N}(b_m f_m(\mathbf{x}_i), \psi_{zm})$$

Parameters $\phi = \{\{b_m, \psi_{zm}\}_{m=1}^d, \psi_y\}$ can be optimized using the EM framework for maximum likelihood fitting [5]. There one distinguishes between observed variables $\mathbf{x}_{\mathcal{D}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ and hidden variables $\mathbf{x}_{\mathcal{H}} = \{\mathbf{z}_i\}_{i=1}^N$ and one maximizes the expected *complete data* loglikelihood $\langle \ln p(\mathbf{x}_{\mathcal{D}}, \mathbf{x}_{\mathcal{H}}; \phi) \rangle = \langle \ln p(\mathbf{y}, \mathbf{Z}|\mathbf{X}; \mathbf{b}, \Psi_{\mathbf{z}}, \psi_y) \rangle$. Here, $\mathbf{Z}$ denotes the N by d matrix of all z_{im} and design matrix $\mathbf{X}$ contains the $f_m(\mathbf{x}_i)$ of all data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. The EM update equations for b_m , the noise variances ψ_y and ψ_{zm} and the moments of $\mathbf{Z}$ have complexity $O(d)$. For brevity, we only give the equation for the weights: $b_m^{(n+1)} =$

$$\omega'_m b_m^{(n)} + \omega_m \frac{\sum_{i=1}^N \left(y_i - \sum_{k \neq m}^d b_k^{(n)} f_k(\mathbf{x}_i) \right) f_m(\mathbf{x}_i)}{\sum_{i=1}^N f_m(\mathbf{x}_i)^2} \quad (2)$$

Fig. 2. Graphical model for Bayesian Backfitting, from [4]

where $\omega'_m = 1 - \omega_m$, $\omega_m = \frac{\psi_{zm}}{s}$ and $s = \psi_y + \sum_{m=1}^d \psi_{zm}$. This is 'probabilistic backfitting': when the factor ψ_{zm}/s is put to 1, the standard backfitting update remains.

2.2. Automatic relevance determination

Probabilistic backfitting makes it possible to use a Bayesian framework to regularize its least-squares solution against overfitting. Furthermore, for certain choice of the prior over the coefficients $\mathbf{b}$ a sparse model is favored. E.g. one can place individual precision variables α_m over each of the regression parameters b_m (figure 2) and choose the priors:

$$\begin{aligned} \mathbf{b}|\boldsymbol{\alpha} &\sim \prod_{m=1}^d \text{Normal}(b_m; 0, 1/\alpha_m) \\ \boldsymbol{\alpha} &\sim \prod_{m=1}^d \text{Gamma}(\alpha_m; \lambda_\alpha, \nu_\alpha^{(m)}) \end{aligned} \quad (3)$$

It can be shown [6] that the marginal prior over the coefficients is a multidimensional Student t-distribution, which places most of its probability mass along the axial ridges of the space. At these ridges the magnitude of only one of the parameters is large, hence favoring a sparse solution (this procedure is called *automatic relevance determination* or ARD). Extracting marginal posteriors such as $\ln p(\mathbf{b}|\mathbf{x}_{\mathcal{D}}; \phi)$ is intractable, but can be done approximately using *variational Bayes*. In this approach, one assumes that variables and parameters factorize in the posterior, so the approximate joint posterior $Q(\mathbf{Z}, \mathbf{b}, \boldsymbol{\alpha}) = Q(\mathbf{Z})Q(\mathbf{b}, \boldsymbol{\alpha})$. This leads to approximate marginal posteriors $Q(\mathbf{b}), Q(\boldsymbol{\alpha})$ of the form

$$\begin{aligned} Q(\boldsymbol{\alpha}) &= \prod_{m=1}^d \text{Gamma}(\alpha_m; \hat{\lambda}_\alpha, \hat{\nu}_\alpha^{(m)}) \\ Q(\mathbf{b}) &= \prod_{m=1}^d t_\nu(b_m; \mu_{b_m}, \sigma_{b_m}^2) \end{aligned} \quad (4)$$

and $Q(\mathbf{Z})$ resembles the EM update expression, with parameters b_m replaced by their expectations $\langle b_m \rangle$. Note that in this approximation the posterior correlations between $\mathbf{b}$ and α are retained [4]. Inference is done using the variational Bayesian EM algorithm, where one alternates variational E and M steps (inferring above posteriors) with a maximization step in the hyperparameters. Substituting the expressions for $\langle z_{im} \rangle$ in the update equations for the distribution of $Q(\mathbf{b})$ gives a 'backfitting like' update for the regression coefficients [4]. The complexity of the full variational EM algorithm remains linear in the dimensionality d .

2.3. Practical issues

When the hyperparameters $\lambda_\alpha, \nu_\alpha^{(m)}$ are initialized to small values, we have effectively an uninformative prior on precision variable α . Because of the ARD mechanism, irrelevant components will have $\langle \alpha_m \rangle \rightarrow \infty$, so the posterior distribution over the corresponding coefficient b_m will be narrow around zero. The significance of a posterior mean value unequal zero can be tested directly via a t -test on the posterior means $\langle b_m \rangle$ (see [7]), since the marginal posterior over coefficients b_m is a factorial t -distribution (4). This leaves a fully automatic regression and feature selection method, where the only remaining hyperparameters are the initial value for the noise variances ψ_{zm}, ψ_y , the level of the t -test and the convergence criteria for the variational EM loop.

3. FAST HEURISTIC FEATURE SELECTION

This section presents two fast greedy heuristic feature selection algorithms specifically tailored for the task of linear prediction. The algorithms apply (1) Forward Selection (FW) and (2) Backward Elimination (BW), which are known to be computationally attractive strategies that are robust against overfitting [8]. In our implementation both algorithms apply the following general procedure.

1. *Preprocessing* For all features (input and output) subtract the mean and scale to unit variance. Features without variance are removed. For efficiency, second order statistics are pre-calculated on the full data.
2. *Crossvalidation* Repeat the following steps 10 times
 - (a) Split dataset: randomly take out 10% of the samples for validation. The statistics of the remaining 90% are used to generate the ranking.
 - (b) Heuristically rank the features (see below).
 - (c) Evaluate the ranking to find the number of features k that minimizes the validation error.
3. *Wrap up* From all values k (found at 2c) select the median k_m (round down in case of ties). Then, for

all rankings, count the occurrences of a feature in the top- k_m to select the k_m most popular features and finally optimize their weights on the full dataset.

The difference between the two algorithms lies in the ranking strategy used at step 2b. However, before discussing this difference we first focus on the evaluation.

3.1. Efficient evaluation

The standard least squares error of a linear predictor, using weight vector $\mathbf{b}$, ignoring a constant term for the output variance, is calculated by

$$J = \mathbf{b}^T \mathbf{R} \mathbf{b} - 2 \mathbf{r}^T \mathbf{b} \quad (5)$$

where $\mathbf{R}$ is the auto-correlation matrix defined as

$$\mathbf{R} = \sum_i \mathbf{x}_i \mathbf{x}_i^T \quad (6)$$

and $\mathbf{r}$ is the cross-correlation vector defined as

$$\mathbf{r} = \sum_i y_i \mathbf{x}_i \quad (7)$$

Finding the optimal weights for $\mathbf{b}$, using standard least-squares fitting, requires a well-conditioned invertible matrix $\mathbf{R}$, which we ensure using the standard regularization technique of adding a small fraction to the diagonal elements of the correlation matrix. Since the regularized matrix $\mathbf{R}$ is a non-singular symmetrical positive definite matrix, we can use a Choleski factorization, providing an upper triangular matrix $\mathbf{C}$ satisfying the relation $\mathbf{C}^T \mathbf{C} = \mathbf{R}$, to efficiently compute the least-squares solution

$$\mathbf{b} = \mathbf{R}^{-1} \mathbf{r} = \mathbf{C}^{-1} (\mathbf{C}^{-1})^T \mathbf{r} \quad (8)$$

Moreover, since intermediate solutions of actual weight values are often unnecessary, because it suffices to have an error measure for a particular subset s (with auto- and cross-correlations $\mathbf{R}_s$ and $\mathbf{r}_s$ obtained by selecting corresponding rows and columns of $\mathbf{R}$ and $\mathbf{r}$, and $\mathbf{C}_s$ being the corresponding Choleski factorization), we can directly insert (8) into (5) to efficiently obtain the error on the training set using

$$J_s = -(\mathbf{C}_s^{-1} \mathbf{r}_s)^T (\mathbf{C}_s^{-1} \mathbf{r}_s) \quad (9)$$

Obtaining a Choleski factorization from scratch, to test a selection of k features, requires a computational complexity of $O(k^3)$, the subsequent matrix division then only requires $O(k^2)$. It is possible to update the factorization incrementally, reducing that complexity also to $O(k^2)$. Matlab's implementation of the complete factorization, however, proved to be rather efficient, so we did not use this option.

3.2. Forward selection and backward elimination

Forward selection repetitively expands a set of features by always adding the most promising unused feature. Starting from an empty set, features are added one at a time. Once selected features are never removed. To identify the most promising feature FW investigates each (unused) feature, directly calculating errors using (9). In principle the procedure can provide a complete ordering of all features. The complexity, however, is dominated by the largest sets, so needlessly generating them is rather inefficient. FW therefore stops the search early once the minimal validation error has not decreased for at least 10 runs.

Backward elimination employs the reverse strategy of FW. Starting from a complete set of features it generates an ordering by each time taking out the least promising feature. To identify the least promising feature our algorithm investigates each feature still part of the set, and removes the one that provides the largest reduction (or smallest increase) of criterion (9). Since BW spends most time at the start, when the feature set is still large, not much can be gained using an early stopping criterion. Hence, in contrast to FW, BW always generates a complete ordering of all features.

4. EXPERIMENTS

4.1. Evaluation of Bayesian backfitting

We generated artificial regression data as follows:

1. Choose total number of features d and number of irrelevant features d_{ir} . The number of relevant features $d_{rel} = d - d_{ir}$.
2. Generate N samples from a normal distribution of dimension $d - \frac{d_{ir}}{2}$. Pad the input vector with $\frac{d_{ir}}{2}$ zero dimensions.
3. Regression coefficients $b_m, m = 1, \dots, d$ were drawn from a normal distribution, and coefficients with value $|b_m| < 0.5$ were clipped to $|0.5|$. The first $\frac{d_{ir}}{2}$ coefficients were put to zero.
4. **Optional:** choose number of redundant features $d_{red} = (d - d_{ir})/2$. The number of relevant features is now $d_{rel} = d_{red}$. Take the relevant features $[\frac{d_{ir}}{2} + 1, \dots, \frac{d_{ir}}{2} + d_{rel}]$, rotate them with a random rotation matrix and add them as redundant features by substituting features $[\frac{d_{ir}}{2} + d_{rel} + 1, \dots, \frac{d_{ir}}{2} + d_{rel} + d_{red}]$.
5. Outputs were generated according to the model, and Gaussian noise was added at an SNR of 10.
6. An independent test set was generated in the same manner, but the output noise was zero in this case (i.e. an infinite output SNR).

7. In all experiments, inputs and outputs were scaled to zero mean and unit variance after the data generation procedure was performed. Unnormalized weights were found by inverse transforming the weights found by the algorithms. The noise variance parameters ψ_{zm} and ψ_y were initialized to $0.5/(d+1)$, thus assuming a total output noise variance that is 0.5 initially.²

4.1.1. Detecting irrelevant features

In the first experiment, $d_{ir} = 10$, so the first and the last five input features were irrelevant for predicting the output, and all other features were relevant. We varied the number of samples N as $[50, 100, 500, 1000, 10000]$ and studied two dimensionalities $d = [15, 50]$. We repeated 10 runs of each feature selection experiment (with each time a new draw of the data), and trained both Bayesian and conventional feature selection methods on the data. The Bayesian method was trained for 200.000 cycles maximum or when the likelihood was improving less than $1e-4$. We compared (i) *labelling error*, i.e. we counted the total number of mislabelings of a feature and normalized by the total number of features present in 10 runs; (ii) *mean normalized prediction error on the test set*; (iii) *computational complexity* (Matlab *tic, toc*). The mean labelling results over 10 repetitions³ are shown in figure 3. We see that for both 15 and 50 features

Fig. 3. Mean labeling error versus log sample size. Upper graph is for $d = 50$, lower graph for $d = 15$.

and moderate to high sample sizes⁴ *variational Bayesian backfitting (VB) outperforms FW, and performs similar to BW*. For small sample sizes, FW and BW outperform VB.

²We noticed that initializing the noise variances to large values led to slow convergence with large sample sizes. Initializing to $0.5/(d+1)$ alleviated this problem.

³The result for $(d, N) = (50, 10000)$ is based on 5 runs.

⁴We define moderate sample size as $N = [100, \dots, 1000]$ for $d = 15$ and $N = [1000, \dots, 10000]$ for $d = 50$.

As for prediction quality of the learned model using the selected features only, performance of all methods was similar (not shown). Apparently, including irrelevant features with small weights contributes little to the prediction error (but features may be expensive to compute!). *Finally, the occasional improved accuracy of VB comes at the expense of larger training times* (see figure 5).

4.1.2. Detecting redundant features

We then added redundant features to the data, i.e. we included optional step 4. in the data generation procedure. In a new experiment, d was varied and the output SNR was set at 10. Since the relevant and the redundant features may be interchanged⁵, we determined the size of the redundant subset in each run (which should equal $d_{\text{red}} = [5, 10, 20]$ for $d = [20, 30, 50]$, respectively). In figure 4 we plot the mean

Fig. 4. Estimated d_{red} vs. log sample size. Upper, middle, lower graphs are for $d = 50, 30, 20$ and $d_{\text{red}} = 20, 10, 5$.

size of the redundant subset estimated in 10 runs for different d, d_{red} , including one standard deviation error bars. *For moderate sample sizes, both VB and the benchmark methods detect the redundant subset* (though they are biased to somewhat larger values), but accuracy of VB estimate drops with large sample sizes. When inspecting the likelihood curves for these cases, it turned out that we did not reach convergence after 200,000 iterations, likely causing the worsening. Further, in figure 5 it is shown that VB scales much less favourably with sample size than the benchmark methods, but on the other hand scales more favourably with input dimensionality. The superlinear scaling behaviour of the benchmark methods with number of relevant features may eventually result in higher computation times for FW and BW. We conclude that VB is able to detect both irrelevant

⁵A rotated set of relevant features may be considered by a feature selection method as more relevant than the original ones, in which case the originals become the redundant ones.

Fig. 5. Running time vs. log sample size. Upper, middle and lower graphs for each method are for $d = 50, 30, 20$.

and redundant features in a relatively reliable manner for dimensionalities up to 50 (which was the maximum dimensionality studied) and moderate sample sizes. The benchmark methods seem more robust to small sample problems.

4.2. Feature selection in preference data

We implemented a hearing aid algorithm on a real-time platform and made one of the processing parameters of the virtual hearing aid on-line modifiable by the user. Six normal hearing subjects were exposed in a lab trial to an acoustic stimulus that consisted of several speech and noise snapshots picked from a database (each snapshot typically in the order of 10 seconds) which were combined in several ratios and appended. This led to one long stream of signal/noise episodes with different types of signals and noise in different ratios. The subjects were asked to listen to this stream several times in a row, and adjust the processing parameter as desired. Each time an adjustment was made, the acoustic input vector and the desired hearing aid processing parameter were stored. At the end of an experiment a set of input-output pairs was obtained from which a regression model could be inferred using off-line training. We postulated that two types of features bear information about the user preference. Feature 1 was expected to be correlated with feature 3 and 4 (features referred to as *red1*, *red2*, *red3*), whereas feature 2 was expected to be very different (referred to as *indep*). Further, features 2, 3 and 4 were computed at 6 different time scales, whereas the first feature was the same at all time scales, leading to $3 \times 6 + 1 = 19$ features. The number of adjustments for each of the subjects 1 to 6 was [43, 275, 703, 262, 99, 1020]. *This means that we are in the realm of moderate sample size and moderate dimensionality, for which VB is accurate* (see section 4). We then trained VB on the six datasets. In figure 6 we show for subjects 3 and 6 a Hinton diagram of the posterior mean values for the

variance (i.e. $\frac{1}{\langle \alpha_m \rangle}$). Subjects 3 and 6 adjust the hearing aid

Fig. 6. ARD-based selection of hearing aid features. Shown is a Hinton diagram of $\frac{1}{\langle \alpha_m \rangle}$, computed from preference data. Left: subject no. 3. Right: subject no. 6. Horizontal (left to right): time scale at which a feature is computed [1, 2, 3.5, 5, 7.5, 10]. Vertical (top to bottom): features [red1, indep, red2, red3]. Box size denotes relevance.

parameter primarily based on features *indep* and *red2*. Two other subjects only used feature *indep*, whereas one subject used all features *indep*, *red1*, *red2*, *red3* (to some extent). One last subject's data could not be fit reliably (noise variances ψ_{zm} were high for all components).

5. DISCUSSION

From our *synthetic data* experiments, we conclude that VB is a useful method for doing accurate regression and feature selection at the same time, provided sample sizes are moderate to high and computation time is not an issue. When redundant features are present and sample sizes are high, one has to train for many iterations in order to reach convergence. However, labelling and prediction accuracies are comparable to the results obtained with our speed-optimized benchmark methods, which take much less computing time and scale more favourably with sample size. Only when dimensionalities are very high, we expect that the linear scaling behaviour of VB with dimensionality will pay off. One further advantage of the VB approach over the benchmark methods is that one can compute a predictive distribution using the inferred parameter posteriors, allowing for a prediction with confidence levels. = From our *preference data* experiment, we noted that 4 out of 6 users personalized the hearing aid parameter based on (some of the) features *indep* and *red2*, which seem a good choice for inclusion in an on-line learning algorithm. For one of the users, either the sample size was too low, his preference was too noisy or the linearity assumption of the model might not hold. In the future, an experimental setup with a different acoustic stimulus might be considered to possibly diminish the noise in the preference data.

6. CONCLUSION

We compared Bayesian backfitting feature selection to a speed-optimized heuristic approach. Results on synthetic data indicate that the method has labelling accuracy comparable to a fast heuristic approach (for moderate sample sizes), but takes much larger training times. VB is therefore a useful alternative to benchmark methods FW and BW mainly when the sample size is moderate (accuracy is good), the number of features is high (linear scaling with dimensionality), and predictions with confidence levels are needed. In a hearing aid personalization problem, 4 out of 6 subjects showed preferences based on two types of features, giving valuable clues for feature choice in on-line algorithms.

7. ACKNOWLEDGMENTS

The authors like to thank Job Geurts for help with the listening experiments and Tjeerd Dijkstra for useful discussions.

8. REFERENCES

- [1] S. Launer and B. Moore, "Use of a loudness model for hearing aid fitting: On-line gain control in a digital hearing aid," *Int. J. of Audiology*, pp. 262 – 273, 2003.
- [2] A. Ypma, B. de Vries, and J. Geurts, "Robust volume control personalization from on-line preference feedback," in *Proc. 2006 IEEE International Workshop on Machine Learning for Signal Processing, MLSP'06*, Maynooth, Ireland, September 6-8 2006, pp. 441–446.
- [3] T. Hastie and R. Tibshirani, *Generalized Additive Models*, Chapman & Hall, 1990.
- [4] A A D'Souza, *Towards tractable parameter-free statistical learning*, Ph.D. thesis, University of Southern California, 2004.
- [5] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. of Royal Statistical Soc. B*, vol. 39, pp. 1–38, 1977.
- [6] M. Tipping, "Bayesian inference: An introduction to principles & practice in machine learning," in *Advanced Lectures on Machine Learning*, 2003, pp. 41–62.
- [7] J. Ting et al, "Predicting EMG data from m1 neurons with variational bayesian least squares," in *Advances in Neural Information Processing Systems 18 (NIPS 2005)*, Cambridge, MA: MIT Press., 2005.
- [8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.