RATIO SEMI-DEFINITE CLASSIFIERS

Jonathan Malkin and Jeff Bilmes

Department of Electrical Engineering
University of Washington
Seattle, Washington, USA
{jsm,bilmes}@ee.washington.edu

ABSTRACT

We present a novel classification model that is formulated as a ratio of semi-definite polynomials. We derive an efficient learning algorithm for this classifier, and apply it to two separate phoneme classification corpora. Results show that our disciminatively trained model can achieve accuracies comparable with state-of-the-art techniques such as multi-layer perceptrons, but does not posses the overconfident bias often found in models based on ratios of exponentials.

Index Terms— Pattern recognition, Speech recognition

1. INTRODUCTION

There are many multi-class classifiers, for instance multi-layer perceptrons (MLPs)[3] and Gaussian mixture models. Given sufficient training data, and if we choose the right MLP, it is possible for the model to converge to the true posterior distribution $p(y|x)$. In practice, however, these models, all based on exponentials, tend to produce low-entropy distributions. That is, whether wrong or right, the models are typically quite confident in their decisions.

There are many cases in which we may ask for more from a classifier. There has been recent interest in ranking classifiers, where a classifier not only gives a probability of the correct class but also correctly ranks the classes numerically. Concentrating too much probability mass on the top class will leave little accuracy for alternative classes. This is especially true as the number of classes grows.

Another application, the one motivating our work here, is the Vocal Joystick (VJ) [2]. Combining machine learning, signal processing and an understanding of human-computer interaction, the VJ allows voice-based control for drawing [8], or moving a mouse cursor or robotic arm [9]. Although targeted at individuals with motor impairments, many able-bodied people also enjoy using the VJ.

While the VJ provides users with a high degree of control, motion is typically in one of the cardinal or ordinal directions. The current system maps vocalizations to movement using an approach very similar to the one in [14]. For on-screen motion, vowel quality controls the motion direction and loudness controls the speed. For the robotic arm, vowel quality corresponds to movement across the top of a table with loudness again controlling speed, and pitch controls vertical motion. Here, we focus on vowel quality estimation.

As an assistive device, accuracy and reliability are important, as is providing the flexibility to accomplish new tasks. Currently, the VJ uses output probabilities as mixing weights tor estimate vowel quality. Our experience, however, is that movement is in the direction of one of the pre-defined classes. The reason is simple: the classifiers are over-confident and produce low entropy posteriors.

To overcome a bias towards labeling frames as a single class, earlier work presented a modified Kalman filter which allows a user to move in arbitrary directions [13]. That approach also introduces lag which can make the system more difficult to control. Our goal, therefore, is to find a classifier which will more smoothly transition between classes. A consequence of this is that we expect any such classifiers to produce higher entropy posteriors on average.

We propose a novel model which we call Ratio Semi-Definite Classifiers (RSC). These models are discriminatively trained, with as many parameters per class (in the most general case) as a single Gaussian with a full covariance matrix. Our new model is far from Gaussian, though, as we will describe in the next section. Additionally, RSCs do not rely on any fast-growing functional forms (e.g. exponentials), which we suspect is an important part of why this model produces higher entropy output distributions while retaining reasonable accuracy.

In this paper, we will often refer to higher entropy posteriors as a good thing, in contrast with many other machine learning papers. To be clear, we specifically want higher entropy posteriors in areas where data is sparse or where we have conflicting labels such as around class boundaries. We do not want the classifier to jump steeply from one class to another as we move across boundaries. That is, we want classifiers with good accuracy but that can also produce increased entropy posteriors — higher entropy distributions is a sufficient condition for this. In areas with more densely packed data from a single class, we have no objection to lower entropy distributions although we are tolerant of higher entropy values in those cases as that can be useful for meaningful rankings.

2. PROPOSED MODEL

The basic form of our novel classifier is quite simple:

$$p(y|\mathbf{x}) = \frac{(\mathbf{x}-\mathbf{d}_y)^T A_y (\mathbf{x}-\mathbf{d}_y)}{\sum_k (\mathbf{x}-\mathbf{d}_k)^T A_k (\mathbf{x}-\mathbf{d}_k)} \tag{1}$$

where $\mathbf{x}$ are the input features and the parameters to be learned are $\Theta = \{A_k, \mathbf{d}_k\}_{k=1}^K$ where K is the number of classes. Note that unlike a Gaussian classifier, we have only a polynomial in the numerator and only a polynomial in the denominator. In order to be a valid probability distribution, we must have $A_k \succeq 0 \ \forall k$ and $\forall \mathbf{x} \ \exists i \in \{1,..,K\}$ such that $\mathbf{x}^T A_i \mathbf{x} > 0$. That is, each matrix must be positive semi-definite and at every point at least one matrix must give a strictly positive result. By parameterizing each matrix A as $A = BB^T$, we can guarantee semi-definiteness at the loss of convexity, as seen in [16].

Given training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where $\mathbf{x}_i$ are feature vectors and y_i integral class labels, we can put our problem in a

This material is based on work supported by the National Science Foundation under grant IIS-0326382.

conditional maximum likelihood framework. We find that learning RSCs is solved via an optimization problem:

$$\max_{\Theta} \sum_i \log \frac{(\mathbf{x_i} - \mathbf{d}_{y_i})^T B_{y_i} B_{y_i}^T (\mathbf{x_i} - \mathbf{d}_{y_i})}{\sum_k (\mathbf{x_i} - \mathbf{d}_k)^T B_k B_k^T (\mathbf{x_i} - \mathbf{d}_k)}. \tag{2}$$

Without the parameterization of A, this would be an instance of semi-definite programming [17] and quite computationally expensive. [4] showed that, as long as there are fewer constraints than the rank of the matrices — for any matrix rank — this parameterization will add no additional local extrema. The result is that the optimization will end on either a (nearly) flat constraint face or else will find an optimal solution. Here, we are guaranteed semi-definitenesse so we need not have any constraints. The resulting form can thus be efficiently optimized.

Define $\alpha_{ik} = (\mathbf{x}_i - \mathbf{d}_k)^T B_k B_k^T (\mathbf{x}_i - \mathbf{d}_k)$ and $\beta_i = \sum_k \alpha_{ik}$ so that $\log p(y_i|\mathbf{x}_i) = \log \frac{\alpha_{iy_i}}{\beta_i}$. Differentiating with respect to B_c and $\mathbf{d}_c$, respectively, yields

$$\begin{aligned} \frac{\partial \log p(y_i|\mathbf{x}_i)}{\partial B_c} &= 2\Bigg(\frac{\beta_i - \alpha_{ic}}{\alpha_{ic}\beta_i}(\mathbf{x}_i - \mathbf{d}_c)(\mathbf{x}_i - \mathbf{d}_c)^T B_c \delta(y_i, c) \\ &\quad - \frac{1}{\beta_i}(\mathbf{x}_i - \mathbf{d}_c)(\mathbf{x}_i - \mathbf{d}_c)^T B_c (1 - \delta(y_i, c))\Bigg) \end{aligned} \tag{3}$$

$$\begin{aligned} \frac{\partial \log p(y_i|\mathbf{x}_i)}{\partial \mathbf{d}_c} &= 2\Bigg(\frac{\beta_i - \alpha_{ic}}{\alpha_{ic}\beta_i} B_c B_c^T (\mathbf{x}_i - \mathbf{d}_c)\delta(y_i, c) \\ &\quad - \frac{1}{\beta_i} B_c B_c^T (\mathbf{x}_i - \mathbf{d}_c)(1 - \delta(y_i, c))\Bigg) \end{aligned} \tag{4}$$

There is no analytical solution, but this problem can be solved via stochastic gradient ascent. The optimization can still be quite complex; fast matrix multiplies [1, 19] alleviate this problem.

2.1. RSC Properties

Some properties of our model, and comparisons to other models, follow. First, RSCs do not have any fast growing function, which we believe is important in allowing the production of higher entropy posteriors. Second, a more obvious point, is that a class is more likely if the value of the numerator is large relative to that of other classes. Consequently, the vectors $\mathbf{d}_k$ are *not* class means; we refer to them as shift vectors. If anything, they could perhaps be called anti-means since the shift vector for a given class will in general be pushed away from the mean of that class: if $\mathbf{x} = \mathbf{d}_k$, then class k will have zero probability.

Unfortunately, not all values of the $\mathbf{d}$ vectors yield a well defined model. As Theorem A shows, an RSC is not always continuous.

Theorem A. *Suppose all* $\mathbf{d}_k = \mathbf{d}$. *Then an RSC is not continuous at* $\mathbf{x} = \mathbf{d}$.

Proof. Let a_{ij}^c be the $(ij)^{th}$ element of matrix A and x_n be element n of vector $\mathbf{x}$. Also, let $\mathbf{x}' = \mathbf{x} - \mathbf{d}$, so that the posterior becomes $p(y|\mathbf{x}) = \frac{\mathbf{x}'^T A_y \mathbf{x}'}{\mathbf{x}'^T \sum_k A_k \mathbf{x}'}$. Let $x'_n = 0 \quad \forall n \neq i$, then

$$\lim_{x'_i \to 0} p(y|\mathbf{x}') = \lim_{x'_i \to 0} \frac{x_i'^2 a_{ii}^y}{x_i'^2 \sum_k a_{ii}^k} = \frac{a_{ii}^y}{\sum_k a_{ii}^k}$$

where we apply l'Hôpital's rule twice for the second equality. If we do the same for $x'_n = 0 \quad \forall n \neq j, j \neq i$ we find $\lim_{x'_j \to 0} p(y|\mathbf{x}') = \frac{a_{jj}^y}{\sum_k a_{jj}^k}$. These two limits are not equal in general. □

We add a penalty term while training to avoid areas of the parameter space where the model is not continuous.

The most similar model we have found to ours, and the original motivation for RSCs, was presented in [5]. Their form has $\mathbf{d}_k = \mathbf{0} \; \forall k$, meaning there is a lack of continuity at the origin. Additionally, their model is symmetric about the origin — $p(y|\mathbf{x}) = p(y|-\mathbf{x})$ — and has additional constraints. Specifically, [5] desires each matrix A_k be idempotent and that $\sum_k A_k = \mathbf{I}$, which allows an interpretation that the probabilities are an estimate of the degree to which $\mathbf{x}$ belongs to class k. Requiring idempotent matrices makes the problem NP-hard, so [5] uses a relaxed version of that constraint. As mentioned earlier, we also derived inspiration from the parameterization $A = BB^T$ as seen in [16], although that makes our optimization non-convex.

RSCs thus generalize [5]. The result that the matrices sum to identity as found in [5] is a consequence of other assumptions and allows a model that is truly linear in the class matrices. Our generalized version does not require such a constraint, which also simplifies the optimization.

2.2. Regularization and Penalty

One side-effect of requiring summation to identity as in [5] is implicit regularization. Since regularization has been shown to be important for many machine learning algorithms [18, 3, 12], we have added regularization to our model, along with the aforementioned penalty, both of which we will explain here.

Our full objective function is

$$\max_{\Theta} \sum_i \log p(y_i|\mathbf{x}_i) - \lambda_B \sum_k \|B_k\|_F^2 - \lambda_d \sum_k \|d\|^2 - \lambda_s \frac{1}{|\mathbf{C}|} \tag{5}$$

where $\mathbf{C} = \sum_{k=1}^K (\mathbf{d}_k - \mu)(\mathbf{d}_k - \mu)^T$, with $\mu = \frac{1}{K}\sum_i \mathbf{d}_i$. The first regularization term is the Frobenius norm of the matrices and the second is the L2 norm of the shifts. These terms tend to prefer smaller matrix and shift values. The third, a penalty on the determinant of the covariance matrix of the shift vectors, forces them away from being equal (which causes non-continuity). This ensures that the model will remain continuous everywhere.

We have considered alternative matrix regularizers, for instance $\|B - \alpha\mathbf{I}\|$. Early testing showed no improvement for $\alpha = 1$ and $\alpha = \frac{1}{\sqrt{K}}$, so for now we have focused on the simpler form given in Equation 5.

At first, the penalty term's reliance on a determinant would seem to yield a complex derivative [15]. Differentiating $|\mathbf{C}|^{-1}$ with respect to element n of vector k gives

$$\frac{\partial |\mathbf{C}|^{-1}}{\partial \mathbf{d}_{kn}} = \frac{-1}{|\mathbf{C}|} Tr\left(\mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \mathbf{d}_{kn}}\right)$$

which would require a matrix multiply for every element of every shift vector. By exploiting the derivative of a covariance matrix with respect to one of its constituent vectors and taking advantage of the matrix trace, we see that

$$\frac{\partial |\mathbf{C}|^{-1}}{\partial \mathbf{d}_{kn}} = \frac{-2}{K|\mathbf{C}|} \langle \mathbf{C}_{:,n}^{-1}, (\mathbf{d}_k - \mu) \rangle$$

where $\langle \mathbf{x}, \mathbf{y} \rangle$ is a dot product between vectors and $A_{:,i}$ refers to column i of matrix A. Because of this simple form, we can thus calculate the derivative for the entire vector at once as

$$\frac{\partial |\mathbf{C}|^{-1}}{\partial \mathbf{d}_k} = \frac{-2}{K|\mathbf{C}|} \mathbf{C}^{-1} (\mathbf{d}_k - \mu). \tag{6}$$

Since $\mathbf{C}$ is symmetric, positive semi-definite and the same for all shift vectors, this can be calculated very efficiently.

The gradients of Equation 5, relying on Equations 3, 4 and 6 and the trivial norm derivatives, are thus

$$\frac{\partial}{\partial B_c} = \frac{\partial \log p(y_i|\mathbf{x}_i)}{\partial B_c} - \lambda_B B_c \tag{7}$$

$$\frac{\partial}{\partial \mathbf{d}_c} = \frac{\partial \log p(y_i|\mathbf{x}_i)}{\partial \mathbf{d}_c} - \lambda_d \mathbf{d}_c + \lambda_s \frac{2}{K|\mathbf{C}|}\mathbf{C}^{-1}(\mathbf{d}_c - \mu). \tag{8}$$

For situations where the number of classes is smaller than the number of features, the covariance matrix of the shifts will not be full rank. In these cases, we instead employ a set of random projections [6] and create a set of several matrices with elements distributed as $\mathcal{N}(0,1)$. We then calculate the covariance in that lower-dimensional space. By summing the results over several of these matrices, we achieve the desired effect with high probability while the use of random matrices means the probability of a spurious large penalty should be quite low.

3. EXPERIMENTAL ENVIRONMENT

We have tested our model on two data sets. In both cases, we used MFCCs with first-order deltas giving 26-d feature vectors. Frames were 25ms long with a 10ms shift. We also varied the number of frames in the feature window.

The first data set is the Vocal Joystick Vowel Corpus [10]. This is a set of vowels collected specifically for the VJ project. We created a training set from 21 recording sessions (2 speakers appear twice, although there is only partial overlap in their sounds), a development set of 4 speakers, and a test set of 10 speakers. All speakers come from the earlier data collection efforts described in [10] and capture the wide variability in human vowel production.

Within that corpus, we conducted several sets of experiments. The first used only utterances containing a single vowel. We tested two conditions: for the 4 vowel case, there are approximately 275k training frames (1931 utterances), and 550k frames (3867 utterances) for the 8 vowel case. For both development and testing, we determined accuracy values by splitting the data 6 ways, calculating accuracy over 5 of the 6 sets, and taking the average result.

In addition to the single vowel utterances, we also tested the models on utterances where a speaker shifts from one vowel to another, which we term diphthongs. There are approximately 340k frames in 2140 utterances. These files do not have labels; speakers were supposed to smoothly transition between vowels. The vowel quality at the start and end of the utterances tends to be shifted from the vowel quality seen in single vowels. These files are used only for comparing the entropies of the resulting posteriors.

Our second data set is TIMIT [7], a standard database often used for phone classification. We randomly selected 40 speakers from the training set giving a 400 utterance development set. The test set was unchanged. We used the 39 phone set described in [11].

We have compared our model to a 2-layer MLP and, since the number of parameters are identical, to a Gaussian using single Gaussians with full covariance. For our model, we set $\lambda_B = \lambda_d$ and performed a search to tune the parameter values. We set $\lambda_s = 10^{-10}$ to be small so that it will have a significant effect only if the shifts are nearly equal. For the neural network, we did a complete grid search over a substantial range values to determine the best values for both the number of hidden nodes as well as regularization parameters on both layers in order to compare against the best MLP we could find. Additionally, we compared using feature windows of 1 and 3 frames, and also 7 frames for the VJ Corpus.

4 vowels	Accuracy (%)			Entropy		
Frames	1	3	7	1	3	7
RSC	95.7	97.5	98.1	0.85/0.43	1.13/0.31	0.83/0.37
MLP	97.4	98.2	98.6	0.66/0.40	0.62/0.38	0.31/0.29
Gaussian	97.2	95.2	93.2	0.07/0.19	0.08/0.21	0.06/0.19

8 vowels	Accuracy (%)			Entropy		
Frames	1	3	7	1	3	7
RSC	68.5	71.2	73.4	2.14/0.32	2.51/0.23	2.53/0.22
MLP	71.3	72.2	72.7	1.03/0.58	1.04/0.56	0.90/0.58
Gaussian	69.7	67.6	56.5	0.71/0.57	0.57/0.55	0.29/0.39

Table 1. Development set results for the VJ Corpus. Only the best results for each model are shown. Entropies are given as mean/dev.

4 vowels	Accuracy (%)			Entropy		
Frames	1	3	7	1	3	7
RSC	88.7	90.1	89.9	0.97/0.43	1.21/0.33	0.98/0.40
MLP	90.6	91.1	91.6	0.73/0.42	0.70/0.42	0.41/0.35
Gaussian	89.8	87.9	85.8	0.13/0.26	0.11/0.24	0.10/0.25

8 vowels	Accuracy (%)			Entropy		
Frames	1	3	7	1	3	7
RSC	62.1	63.2	63.4	2.17/0.31	2.52/0.21	2.55/0.21
MLP	67.2	67.8	68.4	1.08/0.58	1.11/0.58	0.97/0.61
Gaussian	61.9	60.7	54.3	0.69/0.57	0.54/0.53	0.31/0.41

Table 2. Test set results for the VJ Corpus. Entropies are given as mean/standard dev.

4. RESULTS AND DISCUSSION

Development set results on the VJ Corpus appear in Table 1. In general, we see the RSC and MLP results improve with increasing window size, whereas the Gaussian shows decreasing accuracy. For the 8-vowel 7-frame case, the one currently in use for VJ mouse control, the RSC even beats the MLP. Despite that, the Gaussian is much more confident than either other classifier, with the MLP still substantially more so than the RSC — admittedly, however, the Gaussian was not trained with any regularization term.

In Table 2 we show results of applying the same models on the test set. The two sets are reasonably different (see [10] for reasons) so accuracies have fallen. In this case, the Gaussian is always rather confident despite being wrong quite often. The MLP shows higher performance than the RSC for accuracy in this case. Testing with other RSC models that had slightly lower development set results improved the accuracy to as high as 64.2%.

Diph.	4-vowel models		
Frames	1	3	7
RSC	1.01/0.46	1.27/0.37	1.04/0.45
MLP	0.82/0.47	0.79/0.47	0.50/0.43
Gaussian	0.16/0.30	0.13/0.28	0.12/0.27
	8-vowel models		
Frames	1	3	7
RSC	2.18/0.36	2.55/0.25	2.56/0.25
MLP	1.03/0.63	1.03/0.63	0.91/0.66
Gaussian	0.63/0.60	0.48/0.55	0.27/0.40

Table 3. Entropy (mean/dev) results for diphthongs.

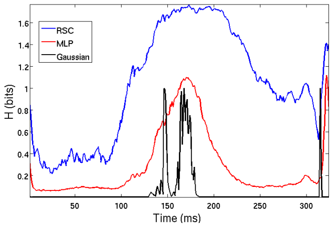

Fig. 1. Entropy plot for a diphthong /ae-i/. Plots use 4 vowel models, 7 frame windows.

Dev.	Accuracy (%)		Entropy	
Frames	1	3	1	3
RSC	46.0	47.7	3.36/0.96	3.30/0.97
MLP	58.3	64.0	2.03/1.13	1.69/1.04
Gaussian	54.9	50.6	1.32/0.93	0.70/0.68

Test	Accuracy (%)		Entropy	
Frames	1	3	1	3
RSC	46.0	47.7	3.36/0.96	3.30/0.98
MLP	57.8	63.4	2.04/1.13	1.70/1.06
Gaussian	54.6	50.6	1.32/0.93	0.70/0.68

Table 4. Development (top) and test (bottom) results for TIMIT with 39 phone classes. Entropies are given as mean/standard dev. Maximum possible entropy ≈ 5.29.

The diphthongs, shown in Table 3 are quite interesting — both the MLP and Gaussian were, on average, even more confident when the utterances crossed vowel boundaries. The RSC, on the other hand, showed very slight upticks in entropy values. This is where we see the most potential for this model. Looking at Figure 1, we see the desired effect: all three models have similar trajectories, but the RSC is never as over-confident, especially in the middle of the utterance, as are the other two models.

As Table 4 shows, the RSC accuracy is worse compared to the other models on TIMIT. This may be due, in part, to the shifts being pushed away from each class with so many more classes. Entropy values, on the other hand, show that the RSC does seem to catch the inherent ambiguity better than do the other models.

5. CONCLUSIONS AND FUTURE WORK

We have introduced a new classification model that is formulated as a ratio of semi-definite polynomials. We have moreover demonstrated that this model can achieve comparable accuracies as state-of-the-art discriminative classifiers, but does not posses the overconfident bias inherent in these models. As with any novel model, there remains much work to be done. First, of course, we would like to find ways to improve the accuracy on data sets such as TIMIT, as well as to investigate ways to help generalization between development and test sets on the VJ Corpus. We plan to explore probabilistic bounds on our projected covariance regularizer in an attempt to show that it will perform as expected with a very high degree of certainty. We would also like to find theoretical bounds on the expected entropy of our classifier versus that of other familiar classifiers.

Beyond purely theoretical work, there are other more practical details waiting to be examined. We have several new variations of RSCs in mind, and several options mentioned in this work. We think allowing adaptation would be useful for the Vocal Joystick and expect to work on that. There is still much to understand about RSCs — we are very excited by the many new possibilities.

Finally, we would like to thank to Amar Subramanya for many useful discussions about this model and that found in [5].

6. REFERENCES

[1] J. Bilmes, K. Asanović, C.-W. Chin, and J. Demmel. Optimizing matrix multiply using PHiPAC: a Portable, High-Performance, ANSI C coding methodology. In *Proceedings of International Conference on Supercomputing*, Vienna, Austria, July 1997.

[2] J. Bilmes et al. The Vocal Joystick: A voice-based human-computer interface for individuals with motor impairments. In *Human Language Technology Conf. and Conf. on Empirical Methods in Natural Language Processing*, 2005.

[3] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[4] S. Burer and R. D. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, Ser. A 103(3):427–444, 2005.

[5] K. Crammer and A. Globerson. Discriminative learning via semidefinite probabilistic models. In *Uncertainty in Artificial Intelligence*, Cambridge, MA, 2006.

[6] S. Dasgupta. Experiments with random projection. In *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence*, San Francisco, CA, 2000.

[7] J. Garofolo et al. *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Linguistic Data Consortium, Philadelphia, PA, 1993.

[8] S. Harada, J. Wobbrock, and J. Landay. VoiceDraw: A hands-free voice-driven drawing application for people with motor impairments. In *ACM SIGACCESS Conf. on Computers and Accessibility*, Tempe, AZ, Oct. 2007.

[9] B. House, J. Malkin, and J. Bilmes. Demo of vj-voicebot: Control of robotic arm with the Vocal Joystick. Presented at ACM SIGACCESS Conf. on Computers and Accessibility, Oct. 2007.

[10] K. Kilanski, J. Malkin, X. Li, R. Wright, and J. Bilmes. The Vocal Joystick data collection effort and vowel corpus. In *Interspeech*, Pittsburgh, PA, Sept. 2006.

[11] K.-F. Lee and H.-W. Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 37(11), Nov. 1989.

[12] X. Li. *Regularized Adaptation: Theory, Algorithms and Applications*. PhD thesis, University of Washington, Seattle, WA, 2007. in preparation.

[13] X. Li, J. Malkin, J. Bilmes, S. Harada, J. Landay, and J. Bilmes. An on-line adaptive filtering algorithm for the Vocal Joystick. In *Interspeech*, Pittsburgh, PA, Sept. 2006.

[14] J. Malkin, X. Li, and J. Bilmes. Energy and loudness for speed control in the Vocal Joystick. In *Automatic Speech Recognition and Understanding*, San Juan, PR, Dec. 2005.

[15] K. B. Peterson and M. S. Pedersen. The matrix cookbook, Feb. 2007.

[16] F. Sha and L. Saul. Large margin hidden Markov models or automatic speech recognition. In *Advances in Neural Information Processing Systems 19*, 2006.

[17] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[18] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, 1995.

[19] R. C. Whaley and J. Dongarra. Automatically tuned linear algebra software. In *Ninth SIAM Conference on Parallel Processing for Scientific Computing*, 1999.