INTERSPEECH 2009 BRIGHTON

The Semi-Supervised Switchboard Transcription Project

Amarnag Subramanya and Jeff Bilmes

Department of Electrical Engineering, University of Washington, Seattle, USA.

asubram@u.washington.edu, bilmes@ee.washington.edu

Abstract

In previous work, we proposed a new graph-based semi-supervised learning (SSL) algorithm and showed that it outperforms other state-of-the-art SSL approaches for classifying documents and web-pages. Here we use a multi-threaded implementation in order to scale the algorithm to very large data sets. We treat the phonetically annotated portion of the Switchboard transcription project (STP) as labeled data and automatically annotate (at the phonetic level) the Switchboard I (SWB) training set and show that our proposed approach outperforms state-of-the-art SSL algorithms as well as a state-of-the-art strictly supervised classifier. As a result, we have STP-style annotations of the entire SWB-I training set which we refer to as *semi-supervised STP* (S3TP).

Index Terms: semi-supervised learning, phone classification, graph-based learning.

10.21437/Interspeech.2009-554

1. Introduction

A large number of standard learning algorithms, such as multi-layer perceptrons (MLP), Gaussian mixture models (GMM), support vector machines (SVM), and hidden Markov models (HMM), assume the existence of *labeled* data where each input is annotated with the desired output(s). In many practical problems, such as speech recognition, annotating training data is time-consuming, cumbersome and error-prone [1]. As a result, in the recent past there has been much interest in *semi-supervised* learning (SSL) [2]. Here, one makes use of small amounts of labeled data with relatively large amounts of unlabeled data to train classifiers. As unlabeled data is obtained very easily, SSL does not suffer from the same drawbacks as supervised learning. For a survey of SSL algorithms, see [3] (and the references therein).

In *transductive learning*, one is only interested in classifying a closed set of inputs [4]. In other words, in transductive learning, the training set is treated as labeled data, the test set as unlabeled data and goal is to infer the labels of the unlabeled data – this is very similar to the goal of SSL, the difference being that in transductive learning we are only interested in obtaining a classification for the samples in the test set [2]. In practice though, the above difference is often blurred as a transductive learner can be utilized to classify any given input [3]. For the purposes of completing a partially transcribed corpus, however, transductive learning is precisely the problem we wish to solve.

There are a number of instances of the successful application of SSL to speech recognition. For example, training an HMM-based speech recognizer using the Baum-Welch algorithm when intra-unit segmentations are not given is an instance of SSL. *Self-training* [5] is another SSL algorithm that has been successfully applied to a number of problems in speech and language. There one trains a system using the labeled data which is then used to annotate the unlabeled data. Subsets of the unlabeled data that have been classified above some confidence are then added to the training set (i.e., treated as being labeled) and used to train a new model. The above is repeated until convergence. Examples include [6, 7], where acoustic and language models for broadcast news were trained in a semi-supervised manner. In [6] the confidence measures were generated by comparing the recognizer output against the closed captions. In problems related to conversational and/or spontaneous speech, however, one does not usually have access to closed captions or other sources of information from where accurate confidence measures may be derived. In this paper, we explore an alternative set of models for semi-supervised learning where a graph is used as a regularizer on both the labeled and unlabeled data to train models.

In graph-based SSL, one assumes that the data (both labeled and unlabeled) is embedded within a low-dimensional manifold expressed by a graph. In other words, each data sample is represented by a vertex within a weighted graph with the weights providing a measure of similarity between vertices. For a survey of current state-of-the-art graph-based SSL algorithms, see section 9 in [3]. Graph-based SSL algorithms are discriminative, transductive and non-parametric. There are a number of reasons why graph-based algorithms are well-suited for our purposes here – (a) they have been shown to outperform other SSL approaches (see chapter 19 in [2]), and more importantly (b) as human speech is produced by the movement of a small (finite) number of articulators, it has often been argued that speech can be embedded within a low-dimensional manifold thereby making it particularly suitable for graph-based SSL [8].

A majority of the current graph-based SSL algorithms have a number of drawbacks: (a) they are based on minimizing squared error, which is not suitable for classification problems [9, 10, 11]; and (b) they assume binary classification requiring the use of computationally expensive extensions such as one vs. rest for multi-class problems. To overcome the above issues, in [12] we proposed a graph-based SSL algorithm based on minimizing the Kullback-Leibler (KL) divergence between distributions that encode class membership probabilities. Unlike squared loss, which is based on absolute error, KL-Divergence is based on relative error. In addition KL-Divergence is asymptotically consistent w.r.t. the underlying distributions [11]. In [12], we showed that the above algorithm generalizes in a straightforward manner to multi-class problems and outperforms the state-of-the-art in the case of the semi-supervised document and web-page classification tasks.

Yet another common criticism of graph-based SSL algorithms (and SSL in general) is the lack of algorithms that scale to very large data sets. Most of the results in SSL thus far have been on relatively small-sized data sets. For example, the largest graph-based SSL application to date had about 900,000 samples [13]. To give the reader an intuition of the scale of the above data set, if we had a phone classification problem with

about *2.5 hours* of training data and assumed a frame rate of 100 Hz, we would have about 900,000 frames. Clearly 2.5 hours of training data is very small compared to the thousands of hours of training data used to train current state-of-the-art speech recognizers [14]. Thus, as the basic premise in SSL is that adding large amounts of unlabeled data leads to improved performance, we require algorithms that scale easily to large amounts of data.

In this paper we extend the algorithm proposed in [12] in two important ways: **1)** We show how the algorithm may be scaled to very large data sets by making use of a multi-threaded implementation. **2)** We use the above parallel implementation to annotate the SWB-I training set at the phonetic level. We use the phonetically annotated portion of STP as labeled data and treat the rest of SWB-I as unlabeled and transduce their labels. This phonetically annotated version of SWB-I, which we refer to as *semi-supervised STP* (S3TP). S3TP data could be useful for the training of large vocabulary systems and for speech research in general.

2. Graph-based Semi-Supervised Learning

Given a set of labeled samples, denoted by $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, and a set of unlabeled samples, denoted by $\mathcal{D}_u = \{\mathbf{x}_i\}_{i=l+1}^{l+u}$, our goal is to infer the labels of the unlabeled samples (this is the transductive learning problem). We define X and Y to be the input and output spaces of the classifier respectively, with $\mathbf{x}_i \in \mathrm{X}$ and $y_i \in \mathrm{Y}$ (here, Y is the set of phones to be recognized). The first step in any graph-based SSL algorithm is the construction of a weighted undirected graph, $\mathcal{G} = (V, E)$ where $V = \{1, \cdots, l+u\}$ is the set of vertices representing both the labeled and unlabeled samples and $E \subseteq \{V \times V\}$ is the set of edges. We use $w_{ij} = [\mathbf{W}]_{ij}$ to denote the weight of the edge between vertices i and j which represent samples $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively. $\mathbf{W}$ is referred to as the weight (or affinity) matrix of $\mathcal{G}$. As will be seen shortly, the input features $\mathbf{x}_i$ affect the final classification results via the graph, $\mathbf{W}$. Thus graph construction is crucial to the success of any graph-based SSL algorithm and currently "is more of an art, than science" [15]. While there are a number of ways of constructing the graph (see section 6.2 in [3]), in our case, we use symmetrized k-nearest neighbor (k-NN) graphs. We set $w_{ij} = \mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j)$ if vertex j is one of vertex i's k nearest neighbors or vice versa, and otherwise $w_{ij} = 0$. In the above $\mathrm{sim}(\mathbf{x}_a, \mathbf{x}_b)$ is a measure of similarity between samples $\mathbf{x}_a$ and $\mathbf{x}_b$. It is assumed that $\mathrm{sim}(\mathbf{x}_a, \mathbf{x}_b) = \mathrm{sim}(\mathbf{x}_b, \mathbf{x}_a)$. We discuss more about the choice of the similarity measure in section 4.

For every $i \in V$, we use two multinomial distributions $p_i(y)$ and $q_i(y)$, $y \in \mathrm{Y}$. Here, the p_i's and q_i's are the distributions to be learned. In addition, for all the labeled vertices $j = 1 \ldots l$, we define $r_j(y)$, $y \in \mathrm{Y}$. The r_j's are derived from the labels as follows – if $y_j = \bar{y} \in \mathrm{Y}$, then $r_j(y) = \delta(y = \bar{y})$. If there is uncertainty associated with the labels, it can be encoded using r_j. We define $\mathrm{p} = (p_1, \ldots, p_{l+u})$ (in practice, p is a $(l+u) \times |\mathrm{Y}|$ matrix) and $\mathrm{q} = (q_1, \ldots, q_{l+u})$. In [12], we proposed optimizing the following objective

$$C(\mathrm{p}, \mathrm{q}) = \sum_{i=1}^{l} D_{KL}\big(r_i || q_i\big) + \mu \sum_{i=1}^{l+u} \sum_{j \in \mathcal{N}'(i)} w'_{ij} D_{KL}\big(p_i || q_j\big)$$

$$- \nu \sum_{i=1}^{l+u} H(p_i) \text{ s.t. } \sum_{y} p_i(y) = 1, \ \sum_{y} q_i(y) = 1,$$

and $p_i(y), \ q_i(y) \geq 0$. In the above, $H(p_i) = -\sum_{\mathrm{y}} p_i(\mathrm{y}) \log p_i(\mathrm{y})$ is the Shannon entropy function of p_i, $D_{KL}(p_i||q_j)$ is the KL-divergence between p_i and q_j, μ and ν are hyper-parameters whose selection we discuss in section 4. Also $w'_{ij} = \left[\mathbf{W}'\right]_{ij}$, $\mathbf{W}' = \mathbf{W} + \alpha \mathbf{I}_n$, $\mathcal{N}'(i) = \{i\} \cup \mathcal{N}(i)$ and $\alpha \geq 0$. While it may seem that we have two ways of computing the most likely class for vertex i (i.e., using $p_i(y)$ or $q_i(y)$), α, which is a hyper-parameter in C ensures that $p_i(y) = q_i(y) \ \forall \ i, y$. For all experiments in this paper we set $\alpha = 1.0$.

The first term in C penalizes the solution $q_i, \ i \in \{1, \ldots, l\}$, when it is far away from r_i, but does not insist that $q_i = r_i$, as allowing for deviations from r_i can help especially with noisy labels [16]. The second term in C, the so-called *graph-regularizer*, penalizes a lack of consistency with the geometry of the data induced by the graph. If w_{ij} is large, we prefer a solution in which p_i and q_j are close in the KL-divergence sense. While KL-divergence is asymmetric, given that $\mathcal{G}$ is undirected implies $\mathbf{W}$ is symmetric ($w_{ij} = w_{ji}$) and as a result the second term is inherently symmetric. In addition the second term acts as a 'glue' between the p_i's and q_i's. The last term maximizes entropy and thus encourages each p_i to be close to the uniform distribution. This acts as a guard against degenerate solutions commonly encountered in SSL [17, 18]. It can be seen that the above approach generalizes easily to multi-class problems and also has the ability to incorporate label uncertainty [12].

It can be shown that $C(\mathrm{p}, \mathrm{q})$ is convex in (p, q) and can be solved using alternating-minimization (it does not admit a closed form solution). Each iterative update has a closed form solution and can be shown to converge to the correct minimum. It has been shown (see [12]) that C may be solved using the following update equations –

$$p_i^{(n)}(y) = \frac{1}{\mathcal{Z}_i} \exp^{\frac{\beta_i^{(n-1)}(y)}{\gamma_i}},$$

$$q_i^{(n)}(y) = \frac{r_i(y)\delta(i \leq l) + \mu \sum_j w'_{ji} p_j^{(n)}(y)}{\delta(i \leq l) + \mu \sum_j w'_{ji}},$$

where n represents the iteration index, $\gamma_i = \nu + \mu \sum_j w'_{ij}$, $\beta_i^{(n-1)}(y) = -\nu + \mu \sum_j w'_{ij} (\log q_j^{(n-1)}(y) - 1)$, and $\mathcal{Z}_i$ is a normalizing constant to ensure p_i is a valid probability distribution. Note that each iteration of the proposed framework is relatively simple to implement. Henceforth we refer to the proposed objective optimized using alternating minimization as *measure propagation* (MP). For more details on this approach and the use of alternating minimization for optimization, see [12].

We compare the performance of MP against that of the *label propagation* (LP) algorithm. We choose LP instead of other alternatives such as spectral graph transduction [18] and manifold regularization [10] because (a) LP scales easily to large data sets, and (b) it has been shown to give state-of-the-art performance on a number of benchmark SSL tasks [2] and phone classification [19]. The LP objective is given in [16] and the update equations are given by

$$p_i^{(n)}(y) = \frac{r_i(y)\delta(i \leq l) + \nu u(y) + \mu \sum_j w_{ij} p_j^{(n-1)}(y)}{\delta(i \leq l) + \nu + \mu \sum_j w_{ij}}$$

where $u(y) = 1/|\mathrm{Y}|$, μ and ν are hyper-parameters.

3. Scalability to Large Datasets

We are now in an era of multi-core processing where individual processors are not getting faster but computers come with mul-

tiple processors. As a result, algorithms need to be amenable to parallel computing across multiple cores within the same shared-memory machine and/or across different machines. This is all the more important in the case of SSL as the basic premise here is that unlabeled data is obtained very cheaply and adding more unlabeled data leads to better performance. As stated above, the largest dataset used within a graph-based SSL framework is on the order of 900,000 samples [13]. Clearly, this is a fraction of the amount of unlabeled data at our disposal. For example, on the Internet we create 1.6 billion blog posts, 60 billion emails, 2 million photos and 200,000 videos every day [20].

The updates for both p and q are easily parallelized. For each vertex in the graph, during each iteration, we have one distribution (p or q) that is held fixed, while the other (q or p) is updated. Consider a multi-threaded application in which each thread operates on a subset of the graph and updates the appropriate distribution while the other is held fixed. We implemented such a multi-threaded application using POSIX threads in C++. This was used to generate the results presented in the next section and enabled us to operate on a data set as large as STP.

4. Experimental Setup & Results

In our experiments we use the Switchboard (SWB) I training set [14, 1]. It consists of 2,400 two-sided telephone conversations among 543 speakers [21]. SWB is used almost ubiquitously for the training of large vocabulary conversational speech recognition systems [14, 1]. The SWB-I corpus comes with word level annotations. In addition, phone level annotations generated in a semi-automatic manner by using a large vocabulary speech recognizer are also available [22]. However, as the speech recognizer has a non-zero error rate, these phone-level transcriptions are considered less reliable.

The *Switchboard Transcription Project* (STP) [23] was undertaken to accurately annotate SWB at the phonetic and syllable levels. One of the goals was that such data could then be used to improve the performance of conversational speech recognition systems. As the task of phonetically annotating speech is time-consuming and error-prone, only 75 minutes of speech segments selected from different SWB conversations were annotated at the phone level and about 150 minutes annotated at the syllable level. Clearly, having access to such phonetic-level annotations for all of SWB could be useful for the speech community. Our goal here is to treat the phonetically annotated portion of STP as labeled data and use it to annotate all of SWB in STP style, i.e., at the phonetic speech-frame level. The result of the above is what we call the S3TP corpus and concomitantly we are able to show that our approach scales to very large data sets.

In the following, we refer to SWB-I that is not a part of the STP as *SWB-STP*. We randomly split the phonetically annotated part of STP data into training, development, and test sets containing 70%, 10% and 20% of the data respectively. The above was repeated 10 times (i.e., we had 10 each of training, development and test sets). In each case, the training set was treated as labeled data, the hyper-parameters were turned on the corresponding development set and the performance was evaluated on the test sets. In order to observe the effects of adding unlabeled data to phone classification performance, we added the unlabeled SWB-STP data in stages. The percentage, s, included, 0%, 2%, 5%, 10%, 40%, 60%, and 100% of SWB-STP. We ran both the LP algorithm and our proposed approach MP. In the case when s =100%, there were about 120 million samples. Both LP and MP are graph based approaches and we used the same graph in each case (only the hyper-parameters were tuned separately).

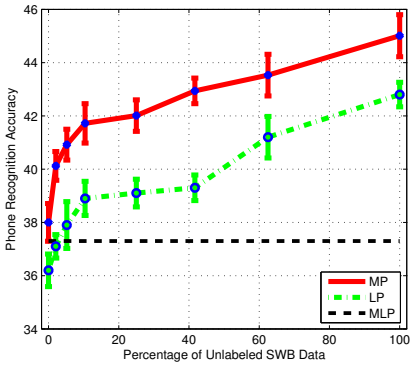

Figure 1: Phone Accuracy vs. Percentage of SWB training data. Phone Accuracy was measured on the STP data. Note that when all the SWB data was added, the resulting graph had **120 million** vertices. The dashed black line shows the performance of a MLP measured using the $s = 0\%$ case over the same training, development and test sets as MP and LP.

Next, we describe graph construction. We constructed a new graph for each value of s over all the STP data and $s\%$ of SWB-STP data. The features $\mathbf{x}_i$ were extracted in the following manner – the wave files were first segmented and then windowed using a Hamming window of size 25ms at 100Hz. We then extracted 13 PLP coefficients from these windowed features and appended both deltas and double-deltas resulting in a 39 dimensional feature vector. As phone classification performance is improved by context, we used a 7 frame context window (3 frames in the past and 3 in the future) yielding a 273 dimensional $\mathbf{x}_i$. We used

$$\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j) = \exp\{-(\mathbf{x}_i - \mathbf{x}_j)^T \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j)\}$$

as the similarity measure to generate the weights w_{ij}. In the above, Σ is the covariance matrix computed using all of SWB. As a result of the scale of the dataset, it is not possible to generate the k-NN graph using *brute-force* search which is $O((l+u)^2)$. However fast k-NN search is a well researched problem with many approximate solutions. A majority of these solutions are based on the classic *kd-tree* [24] data structure. Here we make use of the Approximate Nearest Neighbor (ANN) library [1] [25]. It constructs a modified version of the kd-tree data structure which is then used to query the nearest neighbors (see [25]). We constructed different symmetrized k-NN graphs using ANN for each value of s. For all the experiments, we used $k = 10$. The labeled and unlabeled points in the graph changed based on training, development and test sets used.

For each case, we ran a search over $\mu \in \{$1e–8, 1e–4, 0.01, 0.1$\}$ and $\nu \in \{$1e–8, 1e–6, 1e–4, 0.01, 0.1$\}$ for both LP and MP. The hyper-parameters were chosen based on the performance on the development set and the same value was used to measure the accuracy on the test set. The mean phone accuracy on the different test sets (and the standard deviations) is shown in figure 1 for the different values of s. The result obtained using $s = 0\%$ corresponds to the case when only STP data was used. We would like to point out that our results at $s = 0\%$

[1] Available at `http://www.cs.umd.edu/~mount/ANN/`

outperform the state-of-the-art. As a reference, at $s = 0\%$, a L2 regularized MLP with a 9 frame context window gave 37.3% mean accuracy on the exact same training/dev/test sets while MP gave 38.0% (this is a significant improvement at the 0.0001 level according to a difference of proportions significance test). In figure 1, the performance of the MLP is shown by the dashed black line. The MLP was evaluated using only the STP data, i.e., no unlabeled data was used. Phone classification on conversational speech is considered a harder problem in comparison to phone classification using read speech (e.g., using the TIMIT corpus) and our baseline MLP result is considered state-of-the-art for that model [26]. The results also show that MP significantly improves over the performance of LP for all values of s. Another very encouraging trend is that the phone classification performance improves with the addition of unlabeled data.

5. S3TP Corpus

The S3TP corpus is available at `http://ssli.ee.washington.edu/s3tp`. In addition to the frame-level classification, the above web-page also contains frame-level posteriors. One useful application of the posteriors would be the generation of n-best lists which can be used for the training or re-scoring of speech recognition systems. Further, it could be insightful to look at a comparison of the human annotations of STP and the automatically generated machine annotations. Are there classes of phones where the two agree and other classes where they disagree? We believe that this can lead to new insights into automatic processing of speech signals using machines. In general, we believe that semi-supervised phonetic annotation could be a valuable tool in the hands of speech scientists.

6. Discussion

We have shown how our previously proposed graph-based SSL algorithm can scale to very large data sets by making use of a multi-threaded implementation. We have also shown that it outperforms state-of-the-art SSL algorithms for the phone classification task. When running timing tests on the multi-threaded implementation, we found that in some cases we were able to only achieve a sub-linear speed-up. We conjecture that this is due to poor microprocessor cache performance and have recently developed a node-ordering algorithm that can pre-process the graph to mitigate such problems. Another area for future work is the extension of this approach to speech recognition, i.e., SSL over sequences. We also wish to apply this approach to articulatory labellings of speech corpora, which are even more difficult to obtain than phonetic labellings.

7. References

[1] A. Subramanya, C. Bartels, J. Bilmes, and P. Nguyen, "Uncertainty in training large vocabulary speech recognizers," in *Proc. of the IEEE Workshop on Speech Recognition and Understanding*, 2007.

[2] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-Supervised Learning*. MIT Press, 2007.

[3] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.

[4] V. Vapnik, *Statistical Learning Theory*. Wiley Series, 1998.

[5] H. J. Scudder, "Probability of Error of some Adaptive Pattern-Recognition Machines," *IEEE Transactions on Information Theory*, vol. 11, 1965.

[6] T. Kemp and A. Waibel, "Unsupervised training of a speech recognizer: Recent experiments," in *Proceedings of Eurospeech*, 1999.

[7] L. Lamel, J. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer, Speech and Language*, 2002.

[8] A. Jansen and P. Niyogi, "Semi-supervised learning of speech sounds," in *Interspeech*, 2007.

[9] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. of the International Conference on Machine Learning (ICML)*, 1999.

[10] M. Belkin, P. Niyogi, and V. Sindhwani, "On manifold regularization," in *Proc. of the Conference on Artificial Intelligence and Statistics (AISTATS)*, 2005.

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

[12] A. Subramanya and J. Bilmes, "Soft-supervised text classification," in *EMNLP*, 2008.

[13] I. W. Tsang and J. T. Kwok, "Large-scale sparsified manifold regularization," in *Advances in Neural Information Processing Systems (NIPS) 19*, 2006.

[14] G. Evermann, H. Y. Chan, M. J. F. Gales, B. Jia, D. Mrva, P. C. Woodland, and K. Yu, "Training LVCSR systems on thousands of hours of data," in *Proc. of ICASSP*, 2005.

[15] X. Zhu, "Semi-Supervised Learning with Graphs," Ph.D. dissertation, Carnegie Mellon University, 2005.

[16] Y. Bengio, O. Delalleau, and N. L. Roux, *Semi-Supervised Learning*. MIT Press, 2007, ch. Label Propogation and Quadratic Criterion.

[17] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 19–26.

[18] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proc. of the International Conference on Machine Learning (ICML)*, 2003.

[19] A. Alexandrescu and K. Kirchhoff, "Graph-based learning for phonetic classification," in *ASRU*, 2007.

[20] A. Tomkins, "Keynote speech," CIKM Workshop on Search and Social Media, 2008.

[21] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proceedings of the ICASSP*, vol. 1, San Francisco, California, March 1992, pp. 517–520.

[22] N. Deshmukh, A. Ganapathiraju, A. Gleeson, J. Hamaker, and J. Picone, "Resegmentation of switchboard," in *Proceedings of the ICSLP*, Sydney, Australia, November 1998, pp. 1543–1546.

[23] S. Greenberg, "The Switchboard transcription project," The Johns Hopkins University (CLSP) Summer Research Workshop, Tech. Rep., 1995.

[24] J. Friedman, J. Bentley, and R. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transaction on Mathematical Software*, vol. 3, 1977.

[25] S. Arya and D. M. Mount, "Approximate nearest neighbor queries in fixed dimensions," in *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 1993.

[26] N. Morgan, "Personal communication," 2009.