Robot Speech Learning via Entropy Guided LVQ and Memory Association

Qiong Liu
liu@pal.xerox.com
FX Palo Alto Laboratory
3400 Hillview Ave. Bldg. 4
Palo Alto, CA 94304, USA

Stephen Levinson, Ying Wu, Thomas Huang
{sel, yingwu, huang}@ifp.uiuc.edu
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

Abstract

The goal of this project is to teach a computer-robot system to understand human speech through natural human-computer interaction. To achieve this goal, we develop an interactive and incremental learning algorithm based on entropy-guided LVQ and memory association. Supported by this algorithm, the robot has the potential to learn unlimited sounds progressively. Experimental results of a multilingual short-speech learning task are given after the presentation of the learning system. Further investigation of this learning system will include human-computer interactions that involve more modalities, and applications that use the proposed idea to train home appliances.

1 Introduction

As computer programming gets more and more labor intensive, programming computers through multi-modality human-computer interaction becomes an attractive idea for further exploration. The idea of programming a computer through human-computer interaction is originally proposed by Alan Turing in 1950 [16]. Since then, very few researchers try this idea in practice because of technical limitations. In the past decade, researchers started several projects that are related to this idea. These projects include the "Cog" project in MIT AI lab [1], the "Robocat" project in Bell Labs [8], the "Sail" project in Michigan State University [18], the "Toco" project in MIT Media Lab [15], and the "Illy" project in University of Illinois at Urbana-Champaign [11]. In these projects, the "Cog" project mainly focuses on the robot hardware construction; the "Sail" project mainly focuses on computer vision research; the other three projects mainly focuses on spoken language acquisition research.

In our project, we choose spoken language acquisition research as a starting point. The terminology of "spoken language acquisition" denotes the process of a computer to learn the signal pattern and the meaning of human speech. This terminology is also used in [6] with a similar meaning. The choice of this research starting point has two advantages over the whole project. First, the robot speech ability can help us to retrieve what the robot has learned in a convenient way. This is important for debugging our system. Second, speech ability can give the robot more flexibility to express itself than the robot action ability can.

In early research in Bell Labs, Henis and Levinson built a mobile robot for spoken language acquisition [8]. Their system directly translates the robot speech input into text with a pre-trained speech recognizer. The direct translation limits their system for natural language acquisition (e.g. acquire English and Chinese at the same time). The language acquisition approach proposed by Roy and Pentland in the late 1990s tackled the text representation problem encountered in early research by using phonemes to represent speech directly [15]. However, the phoneme recognition system in their design still cannot provide sufficient flexibility to the language acquisition task.

In these early projects, features from various modalities, such as speech and vision, are associated via a predefined discrete set (i.e. word or phoneme). This type of association strategy is illustrated in Fig. 1. Due to this type of system construction, the definition of the discrete set becomes crucial for the system performance. In other words, the human defined discrete set may become a bottleneck for the system performance.

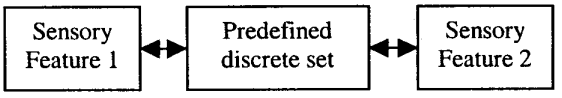

Figure 1. Sensory data are associated through predefined discrete set.

To overcome problems encountered in early research, we propose building relations between sensory features directly through interactive and incremental learning. In this paper, we will present an algorithm that allows a multisensory mobile robot to learn and understand human speech through direct association between speech and other sensory inputs. With this novel approach, the speech feature representations are used directly for communication without translating this media (speech) into another media (text) for manipulation. Thus, the robot can learn speech in various (potentially any) languages. Since our system avoids the predefined discrete set, it is more flexible for language acquisition than early systems.

The rest of this paper is organized as follows. In Section 2, we discuss various classifiers and their suitability

0-7803-7044-9/01/$10.00 ©2001 IEEE

to the robot-learning project. Section 3 focuses on an information transmission model for human-computer interaction. This model is useful to guide the computer training process and our algorithm design. Section 4 describes our learning algorithm in details. Section 5 presents some learning experiments with synthetic data. Section 6 describes our speech acquisition experiments. The conclusion and future directions are given in Section 7.

2 Various Pattern Classifiers and the Robot Project

The sensory data are generally manipulated in groups by a robot *brain* (i.e. a computer). Therefore, the learning process of a robot should start from the learning of a pattern classifier that forms data groups. Since the robot is expected to learn in any environment, the classifier designed for the robot should have the ability to learn any pattern classification task online. Besides the ability to learn any task, the robot classifier should also have the ability to get increasingly complicated as it is exposed to more data.

For discussion proposes, we separate existing pattern classifiers into four categories:

1) Classifiers based on parametric density estimation.
2) Classifiers based on parametric boundary approximation.
3) Classifiers based on non-parametric density estimation.
4) Classifiers based on non-parametric boundary approximation.

In these classifiers, the first type of classifiers can work very well when the joint probability density $p(x,y)$ is known. However, $p(x,y)$ is unknown for most pattern classification cases. For the robot-learning task, it is impossible to know the probability density forms of the sensory inputs before the robot start to learn. Therefore, the classifiers based on parametric density estimation are not going to be used for the robot-learning task.

When the pdf $p(x,y)$ is unavailable, the second type of classifier is generally used. This type of classifier directly adjusts a parametric decision boundary $f(x)=0$ to approximate the optimal decision boundary. These approaches include the Widrow-Hoff approach [19], the Ho-Kashyap approach [9], the Fisher linear discriminant approach [3], and the newly emerged Support Vector Machine (SVM) [17] classifier. All classifier variations in this category can be viewed as searching an optimal decision boundary in a predefined function space. It is often tricky to find a function space that can describe all unknown class boundaries well. Since the robot and the designer do not know what kind of data the robot will experience in its lifetime, this type of classifier is not appropriate for the robot project.

When the prior knowledge of a learning task is unavailable, the third type of classifier is often used. The idea of this type of classifier is to estimate the probability density value of each class at every testing point, and use the Bayes decision theory to perform classification at the testing point. Typical approaches in this category are the Parzen-window approach [14] and the k_n-nearest-neighbor approach [2]. Since classical Pazen-window approach and k_n-nearest-neighbor approach need to memorize a large number of training data for accurate density estimation, these classifiers are not appropriate to learn a large amount of contents when the memory space and computational power are limited. Therefore, we will not select pattern classifiers in this category for our robot-learning task.

Classifiers in the fourth category can asymptotically achieve a low pattern classification error rate without *a priori* knowledge. The basic idea of this type of pattern classifier is to break the boundary optimization problem into a collection of sub-problems, and approximates the optimal decision boundaries piece by piece. The classification boundary can take the form of piecewise hyper-plane, piecewise hyper-sphere, or piecewise spline surface etc. A typical classifier in this category is based on Vector Quantization (VQ) technique [7]. As the VQ technique relaxes the requirement on memory space and computational power, classifiers in this category seem more suitable for the robot-learning task than pattern classifiers in other categories.

There are various approaches to train the vector representations of a VQ based classifier. These approaches include the LBG algorithm [12], self-organizing map (SOM) [10], growing cell structure (GCS) [4], neural gas (NG) [13], and growing neural gas (GNG) [5], etc. Among these approaches, the LBG algorithm has the problem of keeping "dead units" (the units that are not used by any input). The result of this algorithm is also greatly affected by initialization. The SOM approach and the GCS approach use fixed network dimensionality. Fixing the dimensionality makes the growth of the SOM network difficult. It also restricts the natural representation-vector update of both GCS and SOM. Even though the NG approach eliminates the fixed dimensionality constraint for more learning flexibility, it cannot dynamically add new representation vectors for increasingly complicated learning tasks. The GNG approach goes further to allow the number of vectors change during its learning process. However, its cost for maintaining the neighborhood connections of each representation vector is too expensive for the robot online learning task. Because of these limitations of existing VQ training approaches, we propose an entropy-guided LVQ algorithm for the robot-learning task.

3 Information Transmission Model for Human-computer Interaction

The robot training process can be viewed as an information transmission procedure. The information distribution in a sensory-data feature space is vital for the robot training and the algorithm design. Assuming that we denote $\boldsymbol{x}$ as a point in the feature space, and $\boldsymbol{y}$ as a possible label of $\boldsymbol{x}$, the

conditional probability of y at point x can be expressed by $p(y|x)$. With these notations, the information related to point x can be measured by conditional entropy

$$H(Y \mid X = x) = -\sum_{y} p(y \mid x)\log(p(y \mid x)) \cdot \quad (1)$$

Since $p(y|x)$ varies from point to point, the information related to every feature point varies in the feature space. Suppose we want to separate two classes y_1 and y_2 based on the Bayes decision theory, we know that $p(y_1|x)=p(y_2|x)$ at a boundary point x of class y_1 and class y_2. Based on information theory, this equation indicates that the boundary points have more information than points that are not on the boundary. Generally speaking, a point near to a classification boundary has more information contents than a point that is distant from classification boundaries, provided that the density functions vary smoothly in the feature space. These facts suggest a robot instructor spending most teaching effort on data that are close to decision boundaries. On the other hand, the robot brain should assign sufficient resources to learn the data that are close to decision boundaries.

Based on the analysis of the information transmission model, we propose the following strategy to assist the robot training process. When a robot responds correctly to its input, a teacher will consider that the robot can "understand" the testing sample and therefore do nothing to the robot. However, when the robot responds incorrectly to its input, the teacher will consider that the robot cannot "understand" the testing sample and therefore should repeatedly teach the robot about the correct response to the testing sample. Through repeatedly training, the robot will perform correct actions on selected testing samples. This teaching strategy is different from some existing training schemes. However, it aligns well with our daily experiences.

4 An Interactive and Incremental Learning Algorithm

4.1 Entropy guided LVQ

Learning vector quatization (LVQ) is a popular algorithm in pattern-classification. It is generally initialized by VQ algorithms we described in section 2. The problems of those algorithms limit their application in the robot-learning task. Beside those problems we mentioned in section 2, many representation vectors generated by those VQ algorithms are often assigned far away from decision boundaries. For pattern recognition purpose, the representation vectors that are distant from decision boundaries are generally not very useful. These representation vectors can be reduced if we use conditional entropy to control the new representation-vector insertion process. Our algorithm assigns new representation vectors through Voronoi cell split. Let x be an input vector, y be a label, R be the region occupied by a Voronoi cell, n be the number of training data that fall in R, p be the probability estimation $p(y|R)$, we can define an information (entropy) related value I_R with Eq. (2). If the value I_R is greater than a threshold, the cell will split into two cells to increase the local cell resolution.

$$I_R = -n \cdot \sum_{y} p(y \mid R)\log(\, p(y \mid R)) \quad (2)$$

To avoid the *dead unit* (i.e. the Voronoi cell that is not a winner for a long time) problem commonly encountered by hard competitive learning algorithms, such as the LBG algorithm, our algorithm uses an age record to keep track of the usage of every representation vector, and periodically removes dead-units for efficient resource allocation.

With the described mechanism, Voronoi cells near classification boundaries can get more chances than cells distant from classification boundaries to split into fine cells. This cell information value estimation is useful to control the Voronoi cell resolution according to the external stimulation frequency and the information distribution in the feature space. Because the Voronoi cell resolution can be easily controlled through the external stimulation, it is easy to train a computer to distinguish minor feature differences through repeatedly training with similar examples. This teaching process can simulate the teaching process of a human.

The LVQ mechanism used in our algorithm can be described as follows. Let m_i and m_j be the two nearest representation vectors to the input x, α *and* β are learning constants. For input x with label y, the boundary adjustment mechanism can be described with Eq. (3).

$$\begin{aligned} &If\,(p(y \mid R_j) > p(y \mid R_i))and(x \in R_i) \\ &m_j(t+1) = m_j(t) + \alpha(x - m_j(t)) \\ &Else \\ &m_i(t+1) = m_i(t) + \beta(x - m_i(t)) \end{aligned} \quad (3)$$

Eq. (3) may be calculated incrementally. That is the feature we want in our learning system. The local posterior probability estimation is useful for the memory association among different modalities. The probability estimation is also useful to suppress the noise disturbance. As an algorithm property, it is not difficult to prove that the error rate of our classifier can approach the Bayes error rate asymptotically.

4.2 Building relations among different modalities

Building relations among different modalities in a robot learning system is very important for a robot to perform interesting behaviors, such as performing an action according to a speech command and naming an object with speech etc. The algorithm we presented in previous sections can build a large number of representations for inputs from each sensor. The relations of different inputs can be built through associations based on statistics of synchronized events.

Fig. 2 demonstrates a representation-association model between two modalities. In this model, the association between two different representations is tracked by a

counter for these two representations. If a representation X_p happens with a representation Y_q at nearly the same time, the $Counter_{p,q}$ will be increased by 1. When the robot detects an input X_p, the computer can always find its associated representations in another modality based on $p(Y|X_p)$. On the other hand, when the robot detects Y_q, the computer can always find associated representations based on $p(X|Y_q)$. In our current system, the associated representations are chosen based on the maximum value of $p(Y|X_p)$ or the maximum value of $p(X|Y_q)$.

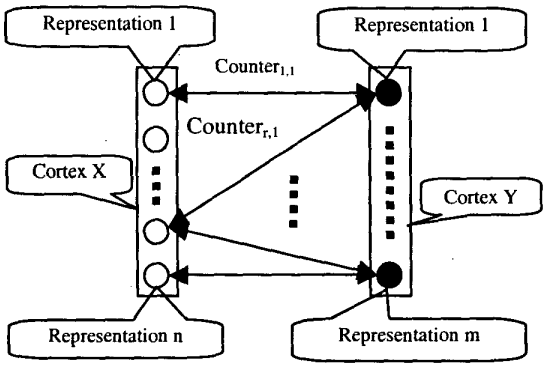

Figure 2. Representation association model between two modalities

5 Experiments with Synthetic Data

For the robot-learning task, our algorithm has many advantages over existing competitive learning algorithms. It does not have the fixed network structure to constrain the growth and training of the network; it can add new representation vectors incrementally according to task requirement; it can also save large memory space and computation for tracking the neighborhood connections of every representation vector. Beside these advantages, using the information related value to guide the cell-split is more meaningful than using the quantization error to guide the representation vector insertion for classification error-rate control. Using the information related value to control the cell-split also makes it easy for the teacher to master the robot teaching strategy, and makes the algorithm robust to data outliers. To illustrate how our algorithm works, we conduct experiments using synthetic data before we try the algorithm with a speech-learning task.

5.1 Synthetic Data

In the synthetic data experiments, the distribution of the data is illustrated in Fig. 3. In Fig. 3, the data inside the square box belong to class 1, and the data outside the box belong to class 2. The range of all data in each dimension is within [-3,3]. The data are uniformly distributed in each class, and the data frequency of each class is proportional to the area that each class occupies.

For this pattern classification task, an ideal VQ based pattern classifier can separate the data perfectly with 5 representation vectors. It is the best solution we can get when we know the underlying data distribution. In the following illustration, it is assumed that we don't know the underlying data distribution. The task in our experiments is to let a classifier learn the pattern classification task from the data randomly generated according to the underlying data distribution.

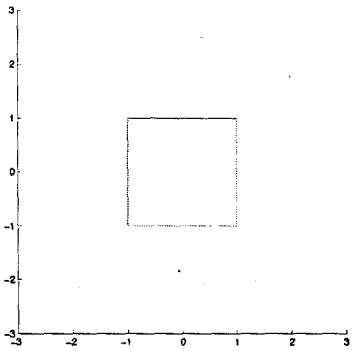

Figure 3. A synthetic pattern classification task with class-1 inside the box and class-2 outside the box.

5.2 Experiments with Synthetic Data

In our learning experiment, we assume that we have 100,000 labeled training samples generated randomly from the above distributions. We skip the cell deletion process for easy analysis. Under this condition, the number of Voronoi cells used by our algorithm is determined by the threshold value for cell split.

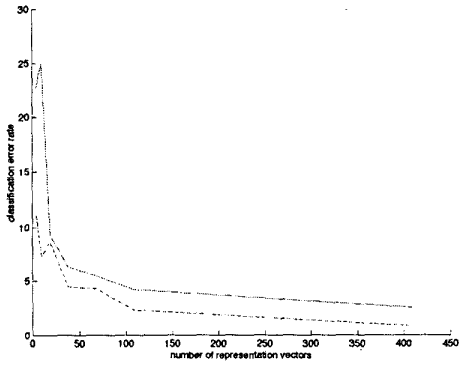

Figure 4. Classification error rate comparison. The solid line reflects the error rate change of randomly generated VQ classifiers. The dashed line reflects the error rate change of entropy-guided VQ classifiers.

We first test the entropy-guided VQ without using Eq. (3) for supervised boundary refinement. The training process starts from a single point on (0,0). It uses all data falling in a Voronoi cell to train the corresponding representation vector in the same way. The result of this

experiment is compared with a simple VQ classifier, which uses the same number of vector representations randomly generated according to the data distribution. The comparison result is shown in Fig. 4.

From the experimental results shown in Fig. 4, we observe that the entropy-guided VQ algorithm generally performs better than the algorithm that randomly assigns representation vectors according to data distribution. Since both algorithms are based on VQ principles to perform the pattern classification task, and every approach has the potential to generate a classifier that can solve this problem perfectly, we cannot claim that our algorithm can always perform better than the other approach just based on this experiment. However, since all classifiers in the experiment are generated independently, it is reasonable to conclude that the entropy-guided VQ algorithm can allocate representation vectors more efficiently than the other algorithm in general.

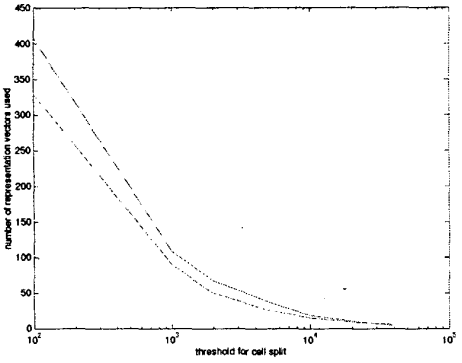

Figure 5. Number of representation-vectors used by entropy-guided VQ classifiers and entropy-guided LVQ classifiers. The solid line corresponds to entropy-guided VQ classifiers. The dashed line corresponds to entropy-guided LVQ classifiers.

We also tested the synthetic data on the entropy-guided LVQ algorithm. This algorithm is different from the algorithm used in previous comparison because it uses labeled data to refine the decision boundary construction instead of using all data in the same way. The classifiers constructed with this algorithm are compared with the classifiers generated by entropy-guided VQ algorithm. The comparison result is shown in Fig. 5 and Fig. 6.

From Fig 5, we notice that the number of representation vectors used by a classifier decreases as the threshold for cell split increases. We also notice that the decision boundary refinement procedure can help a classifier to use less resource to perform the pattern classification task in general.

In Fig. 6, we notice that the dashed line is above the solid line when the cell split threshold is very high. Because the number of representation vectors generated for a classifier is generally low when the cell split threshold is high, we consider the data in that range to be too noisy to reflect the general trend of the algorithm performance. In Fig. 6, the dashed line is under the solid line for most values as the number of representation vectors is reasonably large. Since all our classifiers are constructed independently with randomly generated training samples, it is reasonable to conclude that the entropy-guided LVQ algorithm can work more efficiently than the entropy-guided algorithm.

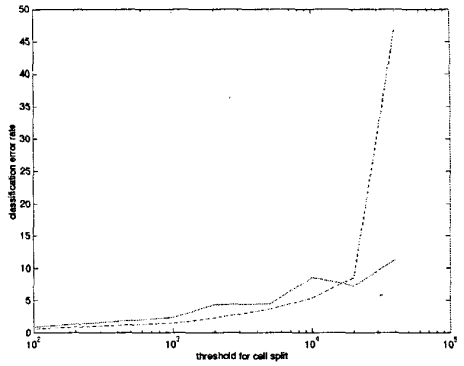

Figure 6. Classification error rate comparison. The solid line corresponds to entropy-guided VQ classifiers. The dashed line corresponds to entropy guided LVQ classifiers.

6 Speech Acquisition Experiments

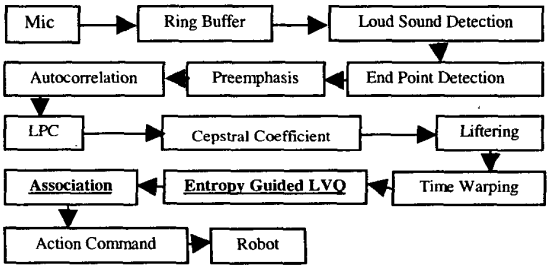

Figure 7. Construction block diagram of the audio learning system

We are presently using our algorithm to enable a mobile robot to learn spoken language through its interactions with humans. The system structure is described in Fig. 7. In this system configuration, the "Mic → ... → Liftering" parts try to simulate some major functions of the human hearing system. The same structure is used in state-of-the-art speech recognizers for extracting useful audio features. The "Time Warping" component is used to align inputs with representation vectors. The approaches we present in previous sections are used to represent and classify the input signals.

Currently, we are using two modalities to test our language acquisition system. These two modalities are speech and tactile sense. The robot can automatically build associations between these two modalities according to the association model we presented in the last section. For

example, if we frequently say "forward" in any language to the robot when we push the robot forward, the robot will build associations between the "forward" speech and the push action. After the association is constructed, the "forward" speech command will be able to activate a "push" feeling on a corresponding tactile sensor.

Our experiment is to teach the robot to act according to our short speech commands in different languages. The robot knows nothing about any speech utterances when it starts to learn. As we teach the robot, we say speech commands to the robot while we force the robot to perform some actions through pushing its corresponding sensors. This teaching process will follow the procedure we proposed in section 3. In other words, we may issue a speech command to the robot and wait for its reaction. If the reaction is correct, we know that the robot can understand the testing command, and therefore try to test it on other commands. If the robot cannot react correctly to our speech, we know that the robot cannot understand the specific command when it experiences the test. Therefore, we should pronounce this command more to the robot, and tell the robot the meaning of the command through pushing corresponding tactile sensors. Following this teaching process, we try control commands in English, Chinese and Japanese with different speakers. The control commands are "forward", "back", "left", "right", "straight", and "stop" in English and their corresponding Chinese and Japanese versions. The robot can distinguish these commands after we teach it for about twenty minutes. Following the interactive training, when we say a command to the robot in English, Chinese, or Japanese, it will move according to our commands. The accuracy of the robot responses is about 100%. Demonstrations of this project can be found at http://www.ifp.uiuc.edu/speech/.

7 Conclusion and Future Directions

In this project, a novel learning framework is developed to facilitate human-computer interaction. In the long run, we want to expand our learning framework to more modalities, and enable the robot to communicate with us through natural human-computer interactions. At present, we are working on more efficient temporal sequence processing method, and speech synthesis for the robot. After the completion of the synthesis program, we expect that the robot can explain its action in speech. In the long run, we also want to expand our learning framework to many other modalities, and enable the robot to express its sensory detection and internal state in speech and body motions. The demonstrations for these expectations may include naming an object in speech or expressing its starvation (low battery) in speech etc. Other important and interesting topics in this research project are speech sentence learning and music sequence learning. To assemble more complicated body parts for the robot will also be encouraging.

8 Acknowledgement

We want to thank Prof. David Kriegman for his suggestions to our research.

Reference

[1] R.A. Brooks, COG THE HUMANOID ROBOT. http://www-caes.mit.edu/mvp/ html/cog.html, 1995.

[2] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, 2000.

[3] R.A. Fisher, The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7 Part II: 179-188, 1936.

[4] B. Fritzke, Growing Cell Structure – A Self-Organizing Network for Unsupervised and Supervised Learning. Neural Networks, vol. 7, no. 9, pp. 1441-1460, 1994.

[5] B. Fritzke, Some Competitive Learning Methods. http://www.neuroinformatik.ruhr-uni-bochum.de/ini/ VDM/, (1997).

[6] A. L. Gorin, S. E. Levinson, and A. Sanker, "An Experiment in Spoken Language Acquisition," *IEEE Transaction on Speech and Audio Processing*, vol. 2, no. 1, part II, January, 1994.

[7] R.M. Gray, Quantization. IEEE Transaction on Information Theory, vol. 44, no. 6, 1998.

[8] E. A. Henis, and S. E. Levinson, "Language as part of sensorymotor behavior," *Proc. AAAI Symposium*, Cambridge, MA, November 1995.

[9] Y. C. Ho, and R. L. Kashyap, A class of iterative procedures for linear inequalities. J. SIAM Control, vol. 4, no. 1, pp. 112-115, 1966.

[10] T. Kohonen, *Self-Organizing Maps*. Springer, 1995.

[11] S.E. Levinson, Q. Liu, Ruei Sung Lin, Weiyu Zhu, Chris Dodsworth, Matt Kleffner, Kevin Squire, Danfeng Li, and Jun Huang, Automatic Language Acquisition, http://www.ifp.uiuc.edu/speech/

[12] S.P. Lloyd, Least squares quantization in pcm. Technical notes, Bell Laboratories, 1957. Published in 1982 in IEEE Transactions on Information Theory.

[13] T.M. Martinetz, and K.J. Schulten, A "neural gas" network learns topologies. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, Artificial Neural Networks, pp. 397-402, North-Holland, Amsterdam, 1991.

[14] E. Parzen, On estimation of a probability density function and mode. Annals of Mathematical Statistics, 33: 1065-1076, 1962.

[15] D. Roy, and A. Pentland, "Learning Words from Natural Audio-Visual Input," *Proc. Int. Conf. Spoken Language Processing*, vol. 4, pp. 1279, Sydney, Australia, December 1998.

[16] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433-460, 1950.

[17] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[18] J. Weng, Learning in Computer Vision and Beyond: Development, 1999.

[19] B. Widrow, and M. E. Hoff, Adaptive switching circuits. Electronic Labs, Stanford University, Stanford, Calif., Tech. Rept. 1553-1, 1960.