Cross-Modal Clustering

Michael H. Coen

MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street
Cambridge, MA 02139
mhcoen@csail.mit.edu

Abstract

This paper presents a self-supervised algorithm for learning perceptual structures based upon correlations in different sensory modalities. The brain and cognitive sciences have gathered an enormous body of neurological and phenomenological evidence in the past half century that demonstrates the extraordinary degree of interaction between sensory modalities during the course of ordinary perception. This paper presents a new framework for creating artificial perceptual systems inspired by these findings, where the primary architectural motif is the cross-modal transmission of perceptual information to enhance each sensory channel individually. The basic hypothesis underlying this approach is that the world has regularities – natural laws tend to correlate physical properties – and biological perceptory systems have evolved to take advantage of this. They share information continually and opportunistically across seemingly disparate perceptual channels, not epiphenomenologically, but rather as a fundamental component of normal perception. It is therefore essential that their artificial counterparts be able to share information synergistically within their perceptual channels, if they are to approach degrees of biological sophistication. This paper is a preliminary step in that direction.

Introduction

Most of the enormous variability in the world around us is unimportant. Variations in our sensory perceptions are not only tolerated, they generally pass unnoticed. Of course, some distinctions are of paramount importance and learning which are meaningful as opposed to which can be safely ignored is a fundamental problem of cognitive development. This process is a component of perceptual grounding, where a perceiver learns how to make sense of its sensory inputs. The perspective in this paper is that this is a clustering problem, in that each sense must learn to organize its perceptions into meaningful categories. That animals do this so readily belies its complexity. For example, people learn phonetic structures for languages simply by listening to them; the phonemes are somehow extracted and clustered from auditory input even though the listener does not know in advance how many unique phonemes are present in the signal. Contrast this with a standard mathematical approach to clustering, where some knowledge of the clusters, e.g., how many there are or their distributions, must be known a priori in order to derive them. Without knowing these parameters in advance, algorithmic clustering techniques may not be robust (Still and Bialek 2004). Assuming that in many circumstances animals cannot know the parameters underlying their perceptual inputs, how then do they learn to organize their sensory perceptions reliably?

This paper presents an approach to clustering based on observed correlations between different sensory modalities. These cross-modal correlations exist because perceptions are created through physical processes governed by natural laws (Thompson 1917, Richards 1980, Mumford 2004). An event in the world is simultaneously perceived through multiple sensory channels in a single observer, and the hypothesis of this paper is that these correspondences play a primary role in organizing the sensory channels individually. The approach described here uses this hypothesis to propose a new framework for grounding artificial perceptual systems. We note this technique is not specific to a single modality.

To develop this framework, we introduce a mathematical model of *slices*, which are topological manifolds that encode dynamic perceptual states and are inspired by surface models of cortical tissue (Citti and Sarti 2003, Ratnanather et al. 2003). Slices are able to represent both symbolic and numeric data and provide a natural foundation for aggregating and correlating information. We then present a cross-modal clustering algorithm that organizes slices based on their temporal correlations with other slices. The algorithm does not presume that either the number of clusters in the data or their distributions is known beforehand. We then examine the outputs and behavior of this algorithm on simulated datasets.

To motivate this work, we first consider a concrete example that raises the possibility that not only can senses influence one another during perception, which has been well established (e.g., Stein and Meredith 1993, Calvert et al. 2004), they can also influence the perceptual structures used internally by other senses. In essence, we examine the possibility that senses can be perceptually grounded by bootstrapping off each other.

In *Proceedings of the Twentieth National Conference on Artificial Intelligence*
(AAAI'05) pp. 932-937. Pittsburg, PA. 2005.

Speechreading

Our example begins with the 1939 World's Fair in New York, where Gordon Peterson and Harold Barney (1952) collected samples of 76 speakers saying sustained American English vowels. They measured the fundamental frequency and first three formants for each sample and noticed that when plotted in various ways (Figure 1), different vowels fell into different regions of the formant space. This regularity gave hope that spoken language – at least vowels – could be understood through accurate estimation of formant frequencies. This early hope was dashed in part because co-articulation effects lead to considerable movement of the formants during speech (Holbrook and Fairbanks 1962). Although formant-based classifications were largely abandoned in favor of the dynamic pattern matching techniques commonly used today (Jelinek 1997), the belief persists that formants are potentially useful in speech recognition, particularly for dimensional reduction of data.

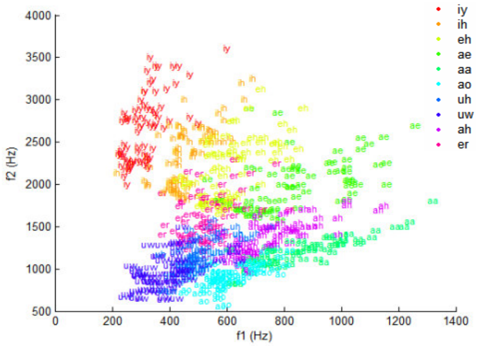

Figure 1 – Labeled scatterplot of the first two formants for English vowels (Peterson and Barney 1952).

It has long been known that watching the movement of a speaker's lips helps people understand what is being said. (viz. Bender 1981, p41). The sight of someone's moving lips in an environment with significant background noise makes it easier to understand what the speaker is saying; visual cues – e.g., the sight of lips – can alter the signal-to-noise ratio of an auditory stimulus by 15-20 decibels (Sumby and Pollack 1954). The task of lip-reading has by far been the most studied problem in the computational multimodal literature (e.g., Mase and Pentland 1990, Huang et al. 2003, Potamianos et al. 2004), due to the historic prominence of automatic speech recognition in computational perception. Although significant progress has been made in automatic speech recognition, state of the art performance lags human speech perception by up to an order of magnitude, even in highly controlled environments (Lippmann 1997). In response to this, there has been increasing interest in non-acoustic sources of speech information, of which vision has received the most attention. Information about articulator position is of particular interest, because in human speech, acoustically ambiguous sounds tend to have visually unambiguous features (Massaro and Stork 1998). For example, visual observation of tongue position and lip contours can help disambiguate unvoiced velar consonants /p/ and /k/, voiced consonants /b/ and /d/, and nasals /m/ and /n/, all of which can be difficult to distinguish on the basis of acoustic data alone.

Articulation data can also help to disambiguate vowels. Figure 2 contains images of a speaker voicing different sustained vowels, corresponding to those in Figure 1. These images are the unmodified output of a mouth tracking system written by the author, where the estimated lip contour is displayed as an ellipse and overlaid on top of the speaker's mouth. The scatterplot in Figure 2 shows how a speaker's mouth is represented in this way, with contour data normalized such that a resting mouth configuration (referred to as *null* in the figure) corresponds with the origin, and other mouth positions are viewed as offsets from this position. For example, when the subject makes an /iy/ sound, the ellipse is elongated along its major axis, as reflected in the scatterplot.

Given this example, it may be surprising that our interest in this paper is not how to do speechreading. Rather, we are concerned with a more basic problem: how do sensory systems learn to segment their inputs? In the plots above, the data are conveniently labeled, which makes the correspondences and the clusters obvious, even when they are overlapping. However, perceptual events in the world are generally not accompanied with explicit labels. Instead, animals are faced with data like those in Figure 3 and must somehow learn to make sense of them. This process is not confined to development, as perceptual correspondences are plastic and can change over time.

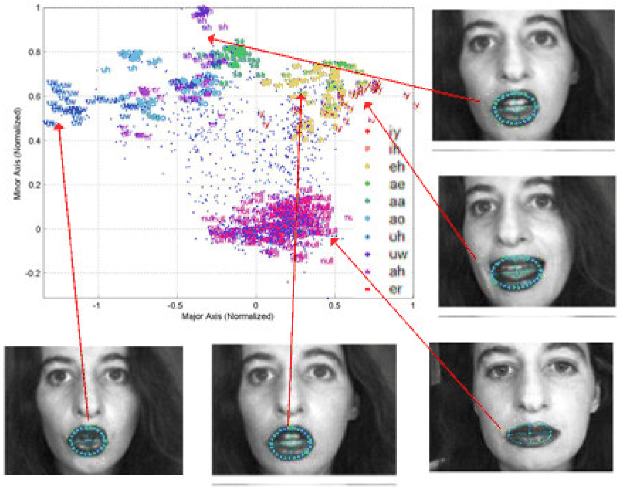

Figure 2 – Modeling lip contours with an ellipse. The scatterplot shows normalized major (x) and minor (y) axes for ellipses corresponding to the same vowels as those in Figure 1.

We would therefore like to have a general purpose way of taking data (such as shown in Figure 3) and deriving the kinds of correspondences and segmentations (as shown between Figures 1 & 2) without external supervision. Instead, we will use the cross modal correspondences among the inputs to derive a training signal.

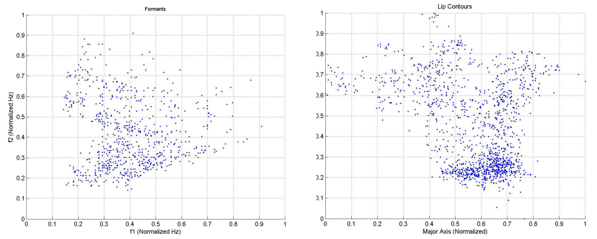

Figure 3 – Unlabeled sensory inputs. Formants F1 and F2 are displayed on the left and lip contour positions corresponding to major and minor axes of fit ellipses are shown on the right. All data are normalized.

Algorithmic Framework

Intuitively, we are going to proceed by hyperclustering each perceptual space into a codebook. This means we are going to partition it into far more regions than necessary for categorizing its perceptual data. The partitioning is created with a variant of k-means clustering discussed below. We call this codebook a *slice*, through analogy to a small slice of cortical tissue.

Our goal is then to combine the codebook clusters into larger regions that reflect the actual perceptual categories represented in sensory data. To do this, we need some meaningful way to compare the similarity of different clusters in the same codebook. We introduce the notion of a *reverse Hebbian projection* of a codebook region, which provides an intuitive way to view co-occurrence relations between different slices. We then compare the reverse Hebbian projections of different regions using the Kantorovich-Wasserstein distance (Gibbs and Su 2002). This provides a non-Euclidean measure of how similar two codebook clusters in the same slice appear from the perspective of the other modalities that co-occur with them. Using this metric, we define a cross-modal clustering algorithm to combine codebook regions into larger regions corresponding with the actual perceptual categories of interest. This process continues iteratively in parallel across all the slices in a system, allowing them to perceptually bootstrap on one another, until a well-defined termination criterion is met. The reliance on co-occurring slices for defining this codebook distance metric is why this is called *cross-modal* clustering.

We define a *slice* S_i as an n-dimensional topological manifold that represents the perceptual organization of an individual sensory system. Each slice consists of a set of clusters c_j that partition it. These partitions are created by first using k-means clustering for some high value of *k*, e.g., 100. This initial clustering is then iteratively refined by recursively reclustering individual regions to normalize the overall cluster density. What value of *k* should be used? Because we are creating a codebook first rather than directly separating the data into perceptual categories, this approach is less sensitive to the selection of *k* than would be the case otherwise. Codebook creation tends towards being a heuristic process and a rule of thumb in this domain is using 5-10 times the maximum number of possible clusters provides satisfactory results. Using more than is necessary simply means more training data may be required. Using fewer runs the risk of missing categories, but recent results have indicated this condition may be both detectable and correctable.

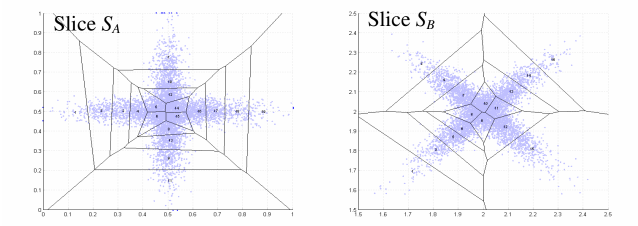

Figure 4 – Codebooks for two slices S_A and S_B corresponding to hypothetical modalities, each containing a mixture of two Gaussian inputs.

Without loss of generality, we assume that all perceptual dimensions are normalized to [0, 1], so that each cluster $c_j \subset [0,1]^n$. We say that a perceptual input *activates* whichever cluster contains it. By way of example, Figure 4 contains the codebooks of two slices S_A and S_B, corresponding to hypothetical modalities each containing a mixture of two Gaussian inputs.

A *data stream* $d_1,..,d_n \in I$ is a series of sensory inputs. With respect to any data stream D, we are interested in the conditional probabilities of clusters in different slices being active simultaneously.

Consider two slices S_A and S_B, partitioned into regions such that $S_A = \{p_1, p_2, ..., p_m\}$, $S_B = \{q_1, q_2, ..., q_n\}$. For event *x*, let *h(x) = # of times event x occurs*, so *Pr(q|p) = h(p,q)/h(p)*. We define a *region* $r \subseteq S_A$ as a set of codebook clusters.

We then define the *Hebbian projection* of region $r \subseteq S_A$ onto S_B as:

$$H(r) = \left[\Pr(q_1 \mid r), \Pr(q_2 \mid r), ..., \Pr(q_n \mid r)\right]$$

The Hebbian projection is a spatial probability distribution that provides a picture of how slice S_B appears whenever region *r* in S_A is active.

We now define weighted versions of these as follows:

$$h_\omega(r) = \sum_{p \in r} \omega_p h(p), \text{ where } \omega = \left[\omega_{p_1}, ..., \omega_{p_m}\right], \ \sum \omega_i = 1$$

$$\Pr_\omega(q \mid r) = h_\omega(r,q) / h_\omega(r) = \sum_{p \in r} \omega_p h(p,q) \Big/ \sum_{p \in r} \omega_p h(p)$$

$$H_\omega(r) = \left[\Pr_\omega(q_1 \mid r), \Pr_\omega(q_2 \mid r), ..., \Pr_\omega(q_n \mid r)\right]$$

Finally, we define the *reverse Hebbian Projection* of region $r \subseteq S_A$ onto S_B as:

$$\widehat{H}(r) = H_{H(r)}(S_B) = \left[\Pr_{H(r)}(p_1 \mid S_B), ..., \Pr_{H(r)}(p_m \mid S_B)\right]$$

The reverse Hebbian projection of a region *r* is a spatial probability distribution that answers the question: *what other regions does slice S_B think region r is similar to in slice S_A?* It is equal to the Hebbian projection of all of S_B onto S_A, weighted by the Hebbian projection of the region *r* onto S_B.

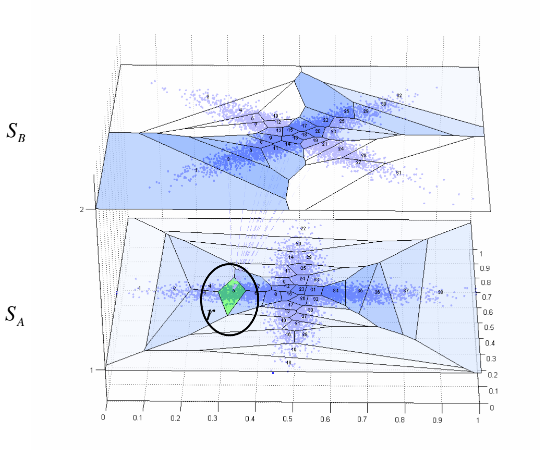

Figure 5 – A visualization of the *reverse Hebbian projection* of the circled cluster *r* in slice S_A (on the bottom) onto slice S_B (on the top). It shows (on the bottom) which other regions in S_A are thought to be similar to *r* from the perspective of S_B. On top, slice S_B is shaded by the Hebbian projection of *r*.

Recall that our goal is to combine codebook clusters in a slice into larger regions corresponding to actual perceptual categories. Doing this requires a similarity metric between different slice regions, and we formulate one by comparing the probability distributions corresponding to the regions' reverse Hebbian projections. It is important to note that these distributions are on n-dimensional data. It is therefore not possible to use one dimensional measures, e.g., Kolmogorov-Smirnov distance, to compare them. Instead, we measure the difference in the spatial distributions of region projections using a higher dimensional metric, the Kantorovich-Wasserstein distance d_W:

Consider the reverse projections for two regions in S_A:

$$\widehat{H}(r_1) = H_{H(r_1)}(S_B) \text{ and } \widehat{H}(r_2) = H_{H(r_2)}(S_B)$$

The Wasserstein distance between them is defined to be:

$$d_W(\widehat{H}(r_1), \widehat{H}(r_2)) = \inf_J \left\{ E\left[d(x,y)\right] : L(x) = \widehat{H}(r_1), L(y) = \widehat{H}(r_2) \right\}$$

where the infimum is taken over all joint distributions J on x and y with marginals $\widehat{H}(r_1)$ and $\widehat{H}(r_2)$, and *d()* is the distance function Δ defined below.

Determining the value of d_W is a well-studied optimization problem. In our framework, it is equal to:

$$d_W\left(\widehat{H}(r_1), \widehat{H}(r_2)\right) = \tfrac{1}{m} \min_{j_1,\ldots,j_n} \sum_{i=1}^{m} \left[d\left(\Pr_{H(r_1)}(p_i \mid S_B), \Pr_{H(r_2)}(p_{j_i} \mid S_B) \right)^2 \right]^{1/2}$$

where the minimum is taken over all possible permutations of $\{1,\ldots,m\}$, where *m* is the number of clusters in S_A. In practice, the Hungarian algorithm (Kuhn 1955) solves this optimization problem effectively. Also, notice that as *k* increases in our domain, the Hebbian conditional probabilities for most cluster pairs are expected to approach or are equal to zero. The algorithm capitalizes on this and therefore, remains practical even for very large values of k.

We are now in a position to define a *cross-modal distance metric* Δ_B between two regions r_1 and r_2 in slice S_A, with respect to another slice S_B:

$$\Delta_B(r_1, r_2) = \left[(1-\lambda) \left[d_E(r_1, r_2) \right]^2 + \lambda \left[d_W\left(\widehat{H}(r_1), \widehat{H}(r_2) \right) \right]^2 \right]^{1/2}$$

The function d_E is the Euclidean distance between cluster centroids. The parameter λ defines the relative importance of Euclidean with respect to cross-modal distances. In most applications, we set $\lambda = 1$, so Euclidean distance is ignored. When spatial locality is important, e.g., in handwriting recognition, smaller values may be useful.

This distance function lets us compare clusters based on both their spatial distance (the d_E term) and their apparent distance as seen from another slice (the d_W term). *Thus, even if two clusters are spatially far apart, their effective distance may be quite small if other co-occurring slices cannot tell them apart.* Conversely, two clusters that are near each other in space may be deemed effectively far apart if they seem different from a cross-modal perspective. We can therefore use this aggregate measure to determine which clusters in a slice actually represent the same percepts and which do not. In practice, we repeatedly recompute Δ by defining it through an iterated function system; details are beyond the scope of this short paper.

The *self distance* of a region *r* in slice S_A measures its own distance to its reverse Hebbian projection, with respect to another slice:

$$s(r) = \Delta\left(H_A(r), \widehat{H}_B(r) \right)$$

Note that the term $H_A(r)$ has a very simple interpretation. It is simply the relative probabilities of the codebook clusters in S_A given that region *r* is active. Self distance is a measure of how far away region *r* is from the other regions to which it is similar. We use this notion to define the *neighborhood* of a region *r*:

$$N(r) = \left\{ c_i : \Delta(r, c_i) < s(r) \right\}$$

The neighborhood of a region is the set of codebook clusters that it is closest to, namely, those within its own self distance. Note that this may include codebooks that are not members of the given region.

Lastly, we define the threshold *t(r)* of a region *r* in slice S_A with respect to slice S_B:

$$t(r) = \sum_{p_i \in N(r)} \Delta_B(r, p_i)\, \widehat{H}(r, p_i) \Big/ \sum_{p_i \in N(r)} \widehat{H}(r, p_i)$$

The threshold is a measure of the distance between a region *r* and the other regions from which it cannot be distinguished, according to the perspective of a co-occurring slice.

Cross-Modal Clustering

We now present an algorithm for combining a slice's codebook clusters into regions that actually reflect the perceptual structure of its input. The algorithm proceeds in a two step fashion. First, it looks for two regions in each slice that are closest together according to the Δ metric defined above. If these regions are within each other's thresholds, they are joined. If not, it considers the next two closest clusters, and so on.

Next, the algorithm greedily moves clusters to their closest regions, if they are not already there.

The algorithm then repeats, until nothing can be found to merge in any of the system's slices, upon which it stops. It is guaranteed to terminate because each step reduces the total number of regions in the system by at least 1. In the worst case, therefore, the number of loops would be $k(n-1)$, where k corresponds to the maximum number of codebook regions and n is the total number of slices.

Algorithm:

Given: k, the number of codebook clusters, and λ, the parameter for weighting Euclidean and cross-modal distances. In most applications, we set $\lambda = 1$ and ignore Euclidean distances.

Initialization: Collect perceptual data to generate the initial codebook for each slice. Standard practice in codebook generation is to collect some constant multiple of the number of codebook regions. The examples here use $30k'$, where k' is equal to k plus however many extra regions are generated in the density normalization procedure described above.

Learning: At this point, all of the slices have been clustered into codebooks. As each perceptual input is received, update the conditional probability matrices between slices. The actual inputs themselves need not be stored. Learning can continue throughout the life of the system, in particular, during the region building step.

Region Building:

```
for each slice $S_i$ and for each cluster $c_a \in S_i$ :
   create a region $r_a$ in $S_i$ containing only $c_a$
end for
while (true)
    for each pair of co-occurring slice $S_i$, $S_j$ :
       % Step 1: Find a pair of regions to merge
         for each pair $r_a$, $r_b \in S_i$, sorted by $\Delta_j(r_a, r_b)$
            if $\Delta_j(r_a, r_b) < \min\left(t_j(r_a), t_j(r_b)\right)$
               merge$(r_a, r_b)$ ; break;  % exit inner for loop
            end if
         end for
       % Step 2: Make sure clusters are in closest regions
       for each cluster $c \in S_i$ :
            let $r = \min_{r \subset S_i} \arg\left[\Delta_j(c, r)\right]$
            make codebook cluster $c$ a member of region $r$
       end for
    end for
    if nothing was merged in any slice on the step
       either wait for new data or exit
    end if
end while
```

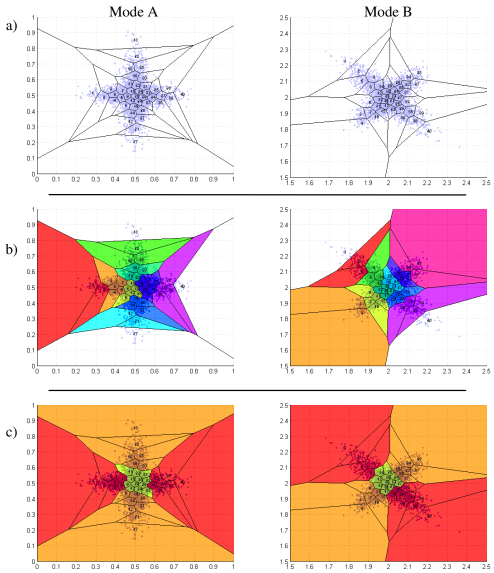

Figure 6 – Three stages in cross-modal clustering. The top shows the initial codebook creation in each slice. The middle show intermediate region formation. The bottom shows the correctly clustered outputs. The confusion regions between the categories is indicated in the center.

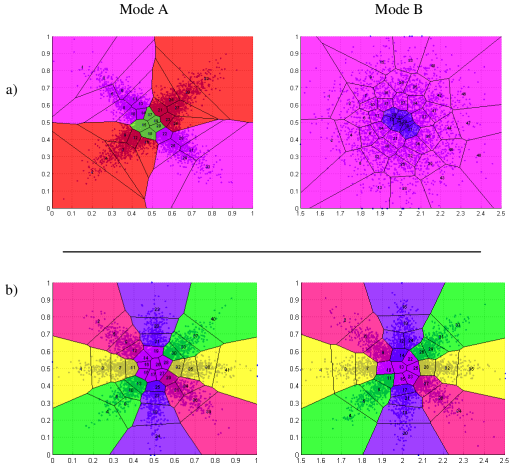

Figure 7 – Outputs of the cross-modal clustering for other sets of inputs. On the top, the algorithm is able to extract an embedded perceptual category in Mode B through reference to Mode A. On the bottom, it correctly separates four overlapping Gaussian distributions in each modality.

Figure 6 shows the output of this algorithm running on two datasets corresponding to Gaussian mixture models. This example is interesting because it is the simplest complex input, namely, two overlapping distributions. The algorithm is able to determine the correct number of regions, even though they overlap and it does not know at the start how many regions to expect in the data. It is also able to create individual regions from non-adjacent components, because it is ignoring Euclidean distances.

The algorithm also identifies the confusion area between the regions, which can be quite useful in perceptual disambiguation and is the subject of our current work. In Figure 7(a), the algorithm is able to detect a region embedded in another one in slice B. Notice here that sizes of regions do not matter, because they are compared only on the basis of cross-modal rather than Euclidean distances.

Discussion

We have presented a cross-modal clustering algorithm that is able to correctly cluster multiple unknown data distributions. It does this by taking advantage of the high degree of correlation among the distributions, which in perceptual systems is due to the natural laws that govern the generation of perceivable events.

The algorithm is also able to function exclusively in low dimensional spaces. Although it can also handle higher dimensional data, it is important to note that cortical architectures make extensive use of low dimensional spaces, e.g., throughout visual, auditory, and somatosensory processing (Amari 1980, Bednar et al. 2004), and this was a primary motivating factor in the development of Self Organizing Maps (Kohonen 1984). Surprisingly, low dimensional spaces can be difficult to work with because they are so crowded. Percepts tend to overlap in complex ways and clustering them generally proves challenging. It is reasonable to assume that animals make extensive use of cross-modal information to guide this process, and we have very tentatively examined the possibility here that artificial perceptual systems may benefit from using conceptually similar mechanisms.

Acknowledgements

The author is grateful to Whitman Richards and Howard Shrobe for encouragement and feedback and also thanks the anonymous reviewers for their helpful comments. This research is sponsored by AFOSR under contract #F49610-03-1-0213.

A color version of this paper is available at:
http://www.csail.mit.edu/~mhcoen/aaai05.pdf

References

Amari, S. Topographic organization of nerve fields. *Bulletin of Mathematical Biology*, 42:339-364. 1980.

Bednar, J.A., Choe, Y., De Paula, J., Miikkulainen, R., Provost, J., and Tversky, T. Modeling Cortical Maps with Topographica, *Neurocomputing*, in press. 2004.

Bender, R. E. The Consquest of Deafness (3rd Ed.). Danville, Il: Interstate Publishers. 1981.

Calvert, A.G., Spence, C., and Stein, B.E. The Handbook of Multisensory Processes. Bradford Books. 2004

Citti, G. and Sarti, A. A cortical based model of perceptual completion in the roto-translation space. In *Proceeding of the Workshop on Second Order Subelliptic Equations and Applications.* Cortona. 2003.

Gibbs, A.L and Su, F.E. On choosing and bounding probability metrics. *International Statistical Review*, vol. 70, number 3, 419-435. 2002

Holbrook, A., and Fairbanks, G. Diphthong Formants and their Movements. *J. Speech Hear. Res.* 5, 38-58. 1962

Jelinek, F. Statistical Methods for Speech Recognition. MIT Press. Cambridge, MA. 1997.

Kohonen, T. Self-Organization and Associative Memory. Springer-Verlag, Berlin. 1984.

Kuhn, H. The Hungarian method for the assignment problem. *Naval Res. Logist. Q.* 2:83—97. 1955.

Lippmann, R.P. Speech recognition by machines and humans. *Speech Communication* 22, 1-15. 1987.

Mase, K., and Pentland, A. Automatic Lipreading by Computer. *Trans. IEEE.*, vol. J73-D-II, No. 6, pp. 796-803, June 1990

Massaro, D.W. & Stork, D. G. Sensory integration and speechreading by humans and machines. *American Scientist*, 86, 236-244. 1988.

Mumford, S. Laws in Nature. London, England: Routledge. 2004.

Peterson, G.E. and Barney, H.L. Control methods used in a study of the vowels. *J.Acoust.Soc.Am.* 24, 175-184. 1952.

G. Potamianos, C. Neti, J. Luettin, and I. Matthews, Audio-Visual Automatic Speech Recognition: An Overview. In: *Issues in Visual and Audio-Visual Speech Processing*, G. Bailly, E. Vatikiotis-Bateson, and P. Perrier (Eds.), MIT Press (In Press), 2004

Ratnanather, J. T., Barta, P. E., Honeycutt, N. A., Lee, N. G., Morris, H. M., Dziorny, A. C., Hurdal, M. K., Pearlson, G. D., and Miller, M. I. Dynamic programming generation of boundaries of local coordinatized submanifolds in the neocortex: application to the Planum Temporale, *NeuroImage*, vol. 20, pp. 359-377, 2003

Richards, W. (ed.) Natural Computation. Cambridge, MA. The MIT Press. 1988.

Stein, B.E., and Meredith, M. A. 1994. The Merging of the Senses. Cambridge, MA. MIT Press.

Still, S., and Bialek, W. How many clusters? An information theoretic perspective, *Neural Computation*. 16:2483-2506. 2004.

Sumby, W.H., and Pollack, I. Visual contribution to speech intelligibility in noise. *J. Acoust. Soc. Am.* 26:212-215. 1954.

Summerfield, Q. Some preliminaries to a comprehensive account of audio-visual speech perception, in Dodd, B. and Campbell, R., editors, Hearing by Eye: The psychology of lip-reading. Lawrence Erlbaum Associates, Hillsdale NJ. pgs 3-52. 1987.

Thompson, DW. On Growth and Form. New York: Dover Publications. 1917 revised 1942.