LAPLACIAN EIGENMAPS AND SPECTRAL TECHNIQUES FOR EMBEDDING AND CLUSTERING

Mikhail Belkin and Partha Niyogi, NIPS' 01

Presented by Talha Cihad Gulcu

October 11, 2011

Organization

- Introduction
- The Algorithm
- Justification
  - Laplacian Operator and Optimal Embedding
  - Heat Kernels and Weight Matrix
- Examples

Introduction (1/2)

- The problem of representing low dimensional data lying in a very high dimensional space is encountered in a variety of areas, such as artificial intelligence, information retrieval, and data mining.
- Although there are numerous works on dimensionality reduction, the majority of the works do not make use of the manifold on which the data reside.
- In this paper, a new algorithm which has an explicit connection with the geometric structure of the manifold enclosing data points is presented.

Introduction (2/2)

- The justification of the algorithm comes from the optimal embedding of manifolds, which is related to Laplacian operator.
- The Laplacian matrix of the graph can be regarded as an approximation of the Laplacian operator on the manifold.
- Therefore, the embedding map proposed here can be viewed as an approximation to the natural map associated with the Laplacian operator.

The Algorithm (1/3)

- The objective can be described as embedding the data points $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^l$ to a lower dimensional space $\mathbb{R}^m$.
- The algorithm consists of three steps. The first step is constructing the graph.
- The nodes i and j are connected by an edge if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close enough". As a measure of proximity, one can consider the following options:
  - ϵ-neighborhoods: Nodes i and j are connected if $||\mathbf{x}_i - \mathbf{x}_j||^2 < \epsilon$.
  - n nearest neighbors: Nodes i and j are connected if i is among n nearest neighbors of j or j is among n nearest neighbors of i.

The Algorithm (2/3)

- The second step is weighting the edges. In this step, there are two variations as well:
  - Heat kernel: If the vertices i and j are connected by an edge, choose W_{ij} as

    $$W_{ij} = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{t}}$$

    The reason why W_{ij} is chosen in this way is explained later.
  - Simple-minded: Choose $W_{ij} = 1$, if the nodes i and j are connected by an edge. We do not have any parameter t to be determined in this case.

The Algorithm (3/3)

- The third step is the eigenmaps step.
- For the each connected part of the graph, we find the eigenvalues and eigenvectors for the generalized problem

$$L\mathbf{y} = \lambda D\mathbf{y} \tag{1}$$

where D is the diagonal matrix having the diagonal entries $D_{ii} = \sum_j W_{ji}$, and $L = D - W$ is the Laplacian matrix.

- Equation (1) always have the trivial eigenvector $\mathbf{1} = (1, \ldots, 1)^T$ associated with the eigenvalue 0. Ignoring the trivial eigenvector, let $\mathbf{y}_1, \ldots, \mathbf{y}_{k-1}$ be the solutions of (1) having eigenvalues $\lambda_1 \leq \cdots \leq \lambda_{k-1}$.
- Then, the algorithm maps each $\mathbf{x}_i$ to $(\mathbf{y}_1(i), \ldots, \mathbf{y}_m(i))$.

Justification (1/6)

- For simplicity, let's first consider the case $m = 1$, i.e., the case when data points are mapped to the real line.
- Denoting the map from the graph to the real line as

$$\mathbf{y} = (y_1, y_2, \ldots, y_k)^T$$

the cost to be minimized by the mapping can be written as $\sum_{i,j}(y_i - y_j)^2 W_{ij}$, or equivalently,

$$\begin{aligned} \sum_{i,j}(y_i^2 + y_j^2 - 2y_i\, y_j)W_{ij} &= \sum_i y_i^2 D_{ii} + \sum_j y_j^2 D_{jj} - 2\sum_{i,j} y_i y_j W_{ij} \\ &= 2\mathbf{y}^T(D - W)\mathbf{y} \\ &= 2\mathbf{y}^T L\mathbf{y} \qquad (2) \end{aligned}$$

Justification (2/6)

- Then, the solution of this minimization problem can be expressed as

$$\text{argmin}_{\mathbf{y}^T D\mathbf{y}=1} \mathbf{y}^T L\mathbf{y} \tag{3}$$

- In (3), the constraint $\mathbf{y}^T D\mathbf{y} = 1$ removes the arbitrary scaling factor in the embedding.
- But, as (2) implies, L is a positive semidefinite matrix, and $\mathbf{y}^T L\mathbf{y} = 0$ if and only if $\mathbf{y} = \mathbf{1}$. To eliminite the trivial solution which maps all the vertices to the same value, we reformulate the solution as

$$\text{argmin}_{\substack{\mathbf{y}^T D\mathbf{y}=1 \\ \mathbf{y}^T D\mathbf{1}=0}} \mathbf{y}^T L\mathbf{y} \tag{4}$$

Justification (3/6)

- Then, the following theorem establishes the justification of the algorithm proposed.

  Theorem 1. *The solution* $\mathbf{y}_{opt}$ *for* (4) *is the eigenvector of* (1) *having the smallest non-zero eigenvalue.*

  Proof. First observe that if $\mathbf{y}_1$ and $\mathbf{y}_2$ are two solutions of (1) having different eigenvalues, we have $\mathbf{y}_1^T D\mathbf{y}_2 = 0$ and $\mathbf{y}_1^T L\mathbf{y}_2 = 0$, since L is a symmetric matrix.

  Then, we define the inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$ as $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T D\mathbf{y}$, and orthonormalize the eigenvectors of (1) having the same eigenvalue by using Gram-Schmidt process. In this way, we get an orthonormal basis $\mathbf{y}_0, \ldots, \mathbf{y}_{k-1}$ consisting of the solutions of (1).

Justification (4/6)

Now, any $\mathbf{y} \in \mathbb{R}^k$ can be written as $\mathbf{y} = \sum_{i=0}^{k-1} \alpha_i \mathbf{y}_i$, and the requirements $\mathbf{y}^T D\mathbf{y} = 1$ and $\mathbf{y}^T D\mathbf{1} = 0$ can be expressed as $\sum_{i=0}^{k-1} \alpha_i^2 = 1$, and $\alpha_0 = 0$, respectively.

Thus, we have $\sum_{i=1}^{k-1} \alpha_i^2 = 1$, and from the inequality $\mathbf{y}^T L\mathbf{y} = \sum_{i=1}^{k-1} \lambda_i \alpha_i^2 \geq \lambda_{min} \sum_{i=1}^{k-1} \alpha_i^2 = \lambda_{min}$, we conclude that the eigenvector of (1) having the smallest nonzero eigenvalue gives $\mathbf{y}_{opt}$. □

- When the data points are mapped to $\mathbb{R}^m$, $m > 1$, the mapping can be fully described by the $k \times m$ matrix $Y = [\mathbf{y}_1 \mathbf{y}_2 \ldots \mathbf{y}_m]$ where i^{th} row Y_i^T refers to the embedding of i^{th} vertex.

Justification (5/6)

- The cost function $\sum_{i,j}(y_i - y_j)^2 W_{ij}$ is extended to the case $m > 1$ as $\sum_{i,j} ||Y_i - Y_j||^2 W_{ij} = \text{tr}(Y^T LY)$.
- Thus, similar to the case $m = 1$, we formulate the optimal embedding as

$$\operatorname{argmin}_{\substack{Y^T DY = I_m \\ Y^T D\mathbf{1} = \mathbf{0}}} \text{tr}(Y^T LY) \tag{5}$$

- In (5), the constraint $Y^T DY = I_m$ prevents the m-dimensional embedding collapse onto a subspace having dimension less than m.

Justification (6/6)

- Then, the following theorem establishes the validity of the algorithm for the case $m > 1$.

  Theorem 2. *The solution* Y_{opt} *for* (5) *is given by*

  $$Y_{opt} = [\mathbf{y}_1 \mathbf{y}_2 \ldots \mathbf{y}_m] \tag{6}$$

  where $\mathbf{y}_i$ *is the solution of* (1) *having the* i^{th} *smallest nonzero eigenvalue.*

- Therefore, the algorithm proposed minimizes a cost function that a good clustering algorithm should minimize. Hence, the algorithm presented in this paper makes sense.

- Moreover, the algorithm is very simple, and basically consists of one sparse eigenvalue problem.

Laplacian Operator and Optimal Embedding

- Let $\mathcal{M}$ be a smooth manifold lying in $\mathbb{R}^k$, and let f be a mapping from $\mathcal{M}$ to $\mathbb{R}$.
- For any $x \in \mathcal{M}$, and a small perturbation δx, we have

$$|f(x+\delta x) - f(x)| \approx |\langle \nabla f(x), \delta x \rangle| \leq ||\nabla f|| \, ||\delta x||$$

  where ∇f refers to the gradient of f.
- Hence, if $||\nabla f||$ is small, points close to x is mapped to points close to $f(x)$. Then, a map which preserves the locality best can be described as

$$\text{argmin}_{||f||_{L^2(\mathcal{M})}=1} \int_{\mathcal{M}} ||\nabla f(x)||^2 \, dx \tag{7}$$

Laplacian Operator and Optimal Embedding

- From the Stokes theorem and the identity $\mathcal{L}f = \operatorname{div}\nabla(f)$, where div refers to divergence and $\mathcal{L}$ refers to Laplacian operator, we get

$$\int_{\mathcal{M}} ||\nabla f(x)||^2 \, dx = \int_{\mathcal{M}} \mathcal{L}f(x)f(x)\, dx$$

- Therefore, Laplacian is a positive semidefinite operator, and the solution of (7) should be the eigenfunction of $\mathcal{L}$ having the smallest eigenvalue.
- This result is analogous to Theorem 1, namely the result that the mapping which minimizes the cost considered is the eigenvector of L having the smallest nonzero eigenvalue.

Heat Kernels and Weight Matrix (1/4)

- Let $f : \mathcal{M} \rightarrow \mathbb{R}$ be a function, and $\mathbf{x}_1, \ldots, \mathbf{x}_k$ be data points on $\mathcal{M}$.
- Consider the heat equation $\frac{\partial u}{\partial t} = \mathcal{L}u$ with the initial condition $u(x,t)\big|_{t=0} = u(x,0) = f(x)$.
- The solution of the heat equation is given by

  $$u(x,t) = \int_{\mathcal{M}} H_t(x,y)\, f(y)\, dy$$

  where $H_t(x,y)$ is the Green's function corresponding to the heat equation. Thus,

  $$\mathcal{L}f(x) = \mathcal{L}u(x,0) = \frac{\partial}{\partial t}\left[\int_{\mathcal{M}} H_t(x,y)\, f(y)\, dy\right]\Bigg|_{t=0} \tag{8}$$

Heat Kernels and Weight Matrix (2/4)

- If $||x-y||$ and t are sufficiently small, the heat kernel can be approximated as $H_t(x,y) \approx (4\pi t)^{-\frac{n}{2}} e^{-\frac{||x-y||^2}{4t}}$, where $n = dim\mathcal{M}$.
- Moreover, $H_t(x,y)$ tends to impulse function $\delta(t)$ as t approaches 0. Thus, we have

$$\lim_{t\to 0}\int_{\mathcal{M}} H_t(x,y)\, f(y)\, dy = f(x)$$

- Then, from (8), we obtain

$$\begin{aligned}\mathcal{L}f(x) &\approx \frac{u(x,t)-f(x)}{t}\\ &\approx -\frac{1}{t}\left[f(x) - (4\pi t)^{-\frac{n}{2}}\int_{\mathcal{M}} e^{-\frac{||x-y||^2}{4t}} f(y)\, dy\right] \qquad (9)\end{aligned}$$

for t sufficiently small.

Heat Kernels and Weight Matrix (3/4)

- For the data points $\mathbf{x}_1, \ldots, \mathbf{x}_k$, (9) can be approximated as

$$\mathcal{L}f(\mathbf{x}_i) = -\frac{1}{t}\left[f(\mathbf{x}_i) - \frac{1}{k}(4\pi t)^{-\frac{n}{2}} \sum_{\substack{\mathbf{x}_j \\ 0<||\mathbf{x}_j - \mathbf{x}_i||<\epsilon}} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{4t}} f(\mathbf{x}_j) \right] \tag{10}$$

- (10) defines the discrete Laplacian operator. The coefficient $1/t$ has no effect on eigenvectors.

- Choosing f as a constant function, from (10), we derive

$$\frac{1}{k}(4\pi t)^{-\frac{n}{2}} \sum_{\substack{\mathbf{x}_j \\ 0<||\mathbf{x}_j - \mathbf{x}_i||<\epsilon}} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{4t}} = 1$$

which justifies the choice $D_{ii} \triangleq \sum_j W_{ji}$ in the algorithm.

Heat Kernels and Weight Matrix (4/4)

- Finally, (10) implies that we get a discrete counterpart of the Laplacian operator if we choose W_{ij} as

$$W_{ij} = \begin{cases} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{4t}}, \text{ if } ||\mathbf{x}_i - \mathbf{x}_j|| < \epsilon \\ 0, \text{ otherwise} \end{cases}$$

- To sum up, we have shown that
  - The eigenfunction of the Laplacian operator having the smallest nonzero eigenvalue gives the best locality preserving map for the manifolds.
  - The eigenvectors of the L used in the algorithm are the eigenvectors of the discrete Laplacian, since the L and discrete Laplacian are constant multiples of each other.

Therefore, the algorithm given is justified by the role of Laplacian operator in optimal embedding.

Examples (1/8)

- Example 1- A Toy Vision Example: In this example, there is a 40×40 visual field, and 1000 images having either a vertical or a horizontal bar are chosen at random.
- Each image is to mapped to $\mathbb{R}^2$. Thus, the data lying in $\mathbb{R}^{1600}$ is to be embedded to $\mathbb{R}^2$.
- Fig. 1 shows the result of the algorithm compared to principal component analysis.

Examples (2/8)

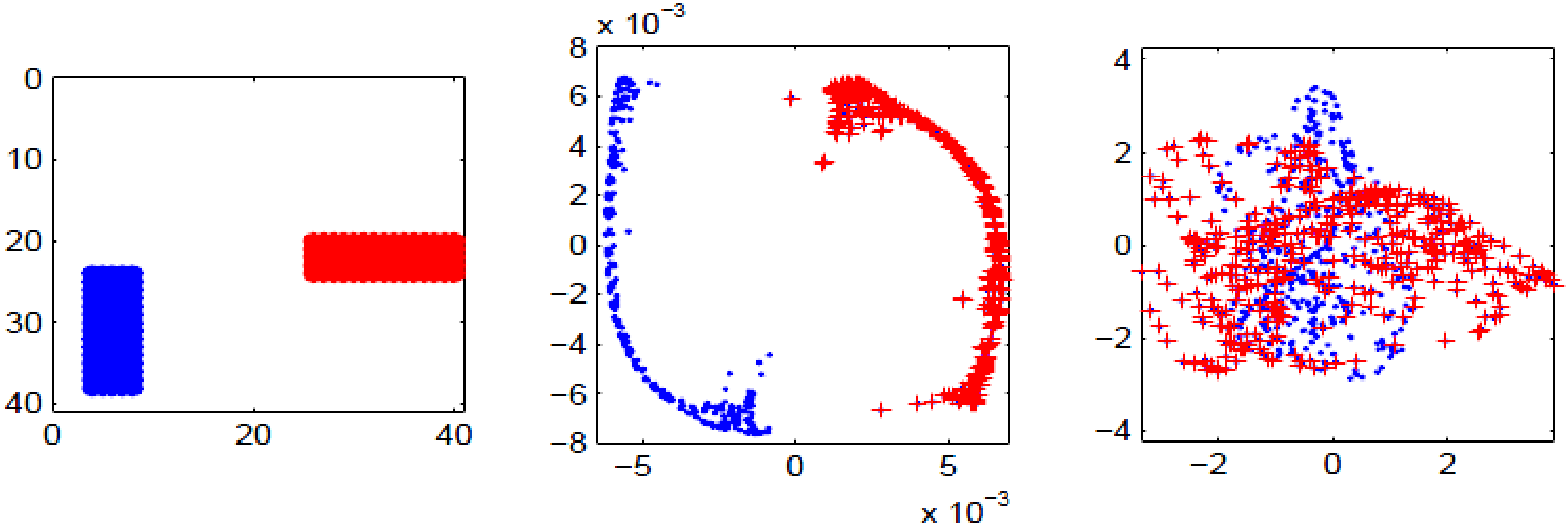

Figure 1: The left panel shows a horizontal and a vertical bar. The middle panel is a two dimensional representation of the set of all images using the Laplacian eigenmaps. The right panel shows the result of a principal components analysis using the first two principal directions to represent the data. Dots correspond to vertical bars and '+' signs correspond to horizontal bars.

Examples (3/8)

- Example 2- Words in the Brown Corpus: Brown Corpus is a collection of texts compiled from works published in the United States in 1961. It contains a total of roughly one million words.
- The algorithm presented here is applied to the 300 most frequent words in the Brown corpus.
- Each word is represented as a vector in $\mathbb{R}^{600}$ based on its left and right neighbors.
- Then, this 300 word is embedded to $\mathbb{R}^2$. Fig. 2 and Fig. 3 show the results.

Examples (4/8)

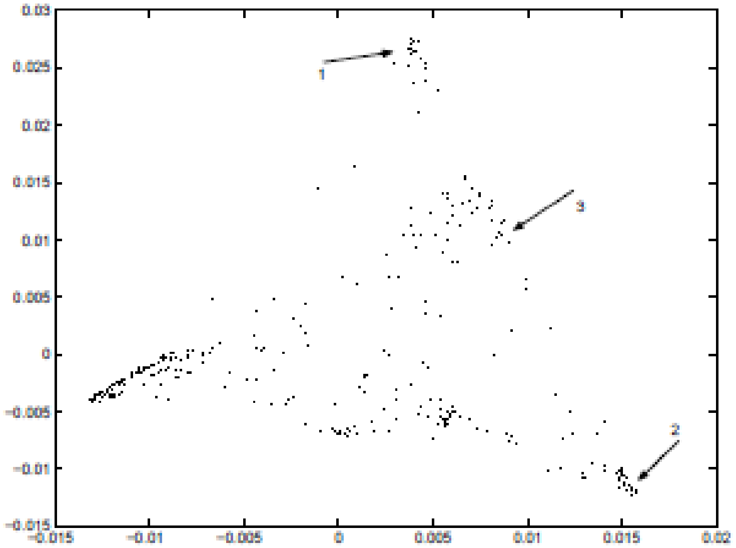

Figure 2: 300 most frequent words of the Brown corpus represented in the spectral domain.

Examples (5/8)

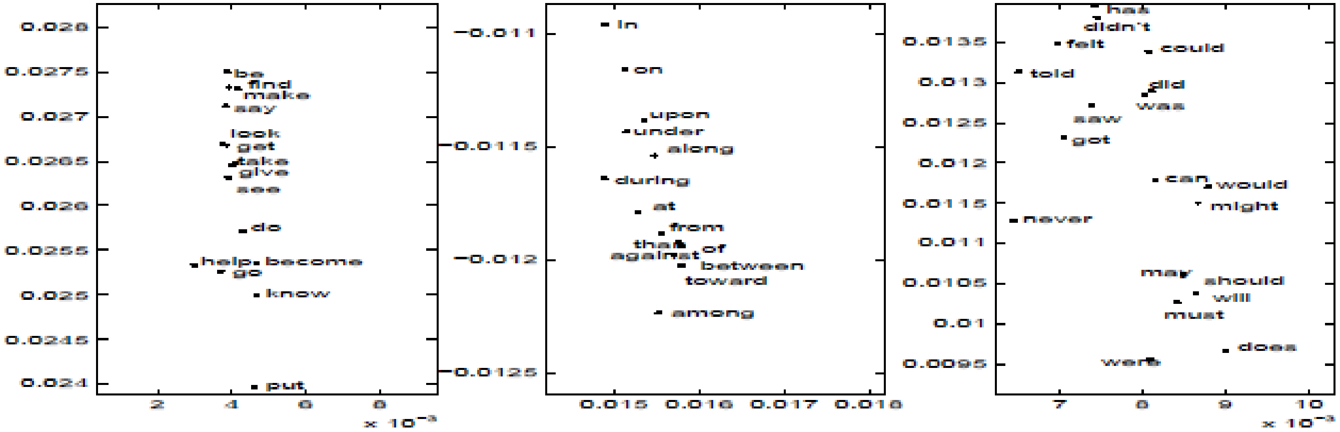

Figure 3: Fragments labeled by arrows in figure 2, from left to right. The first contains infinitives of verbs, the second contains prepositions and the third mostly modal and auxiliary verbs. We see that syntactic structure is well-preserved.

Examples (6/8)

- Example 3- Speech: Lastly, a sentence of speech sampled at 1 kHz is considered.
- Short time Fourier spectra were computed at 5 ms intervals yielding 685 vectors of 256 Fourier coefficients for every 30 ms chunk of the speech signal.
- Then, each vector is mapped to $\mathbb{R}^2$. From Fig. 4, we see that the vectors corresponding to fricatives, periodic sounds, and closures form 3 main clusters.
- Fig. 5 illustrates three selected region of the representation space in detail.

Examples (7/8)

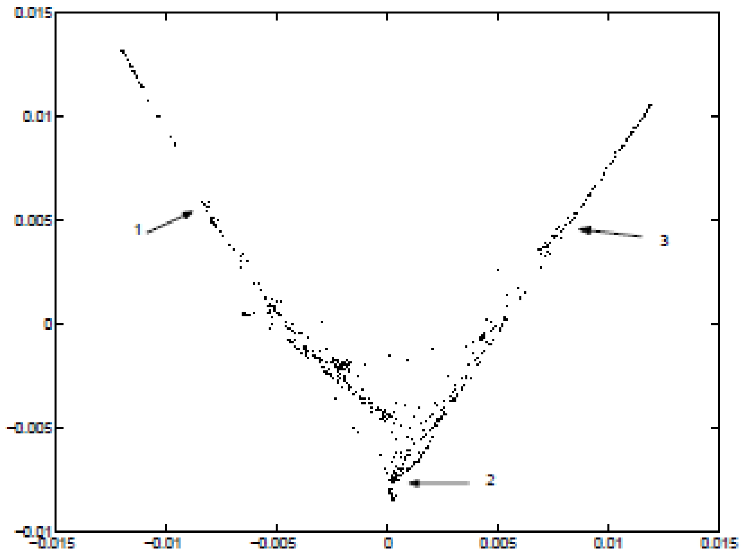

Figure 4: 685 speech datapoints plotted in the two dimensional Laplacian spectral representation.

Examples (8/8)

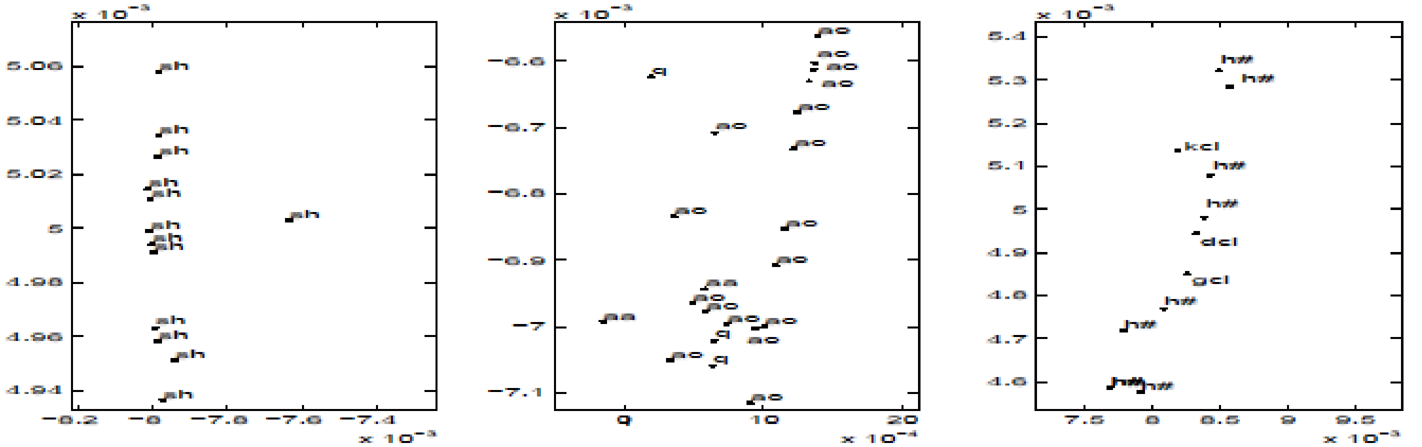

Figure 5: A blowup of the three selected regions in figure 4, from left to right. Notice the phonetic homogeneity of the chosen regions. Note that points marked with the same symbol may arise from occurrences of the same phoneme at different points in the utterance. The symbol “sh” stands for the fricative in the word *she*; “aa”,”ao” stand for vowels in the words *dark* and *all* respectively; “kcl”,”dcl”,”gcl” stand for closures preceding the stop consonants “k”,”d”,”g” respectively. “h#” stands for silence.