# Visualizing and Classifying Odors Using a Similarity Matrix

**Liran Carmel** **Yehuda Koren** **David Harel**
Department of Computer Science and Applied Mathematics
The Weizmann Institute of Science, Rehovot 76100, Israel
{liran,yehuda,harel}@wisdom.weizmann.ac.il

## Abstract

*The Lorentzian model is an analytic expression that describes the time response of electronic nose sensors. We show how this model can be utilized to calculate a normalized similarity index between any two measurements. The set of similarity indices is then used for two purposes: visualization of the data, and classification of new samples. The visualization is carried out using graph drawing tools, and the results are shown to bear some desired properties. The classification is done using a majority-decision type algorithm, and is demonstrated to have very low error rate.*

## Keywords

electronic noses, similarity index, feature extraction, Lorentzian model, graph drawing, visualization, classification

## INTRODUCTION

Visualization and classification are among the most addressed issues in the data analysis of electronic noses (eNoses). Classification, which is the task of determining the identity of incoming samples, is by far the most popular form of analysis. It is realized by a variety of supervised learning algorithms, details of which can be found in, e.g., [2] or [3]. Visualization is the task of representing the eNose measurements as points in two-dimensional (and in some occasions three-dimensional) space, in a way that faithfully captures their inter-relationships. Undoubtedly, principal component analysis (PCA) is the most widely used visualization algorithm.

Whatever algorithm is used, would it be for visualization or for classifications, it requires ***feature vectors*** as input. A ***feature set*** is a small set of parameters that somehow describe the entire time response of a certain sensor in a certain measurement. The ***feature vector*** is comprised of the collection of the feature sets of all the sensors in a particular measurement. Typically, the height of the signal is taken as a single feature per sensor (see, e.g., [3]), resulting in feature vectors, to be hereinafter called the ***height feature vectors***, whose length is the number of sensors in the eNose.

The feature vectors serve well for many applications, but might introduce unavoidable difficulties for others. One problem is that the eNose sensors might be scaled differently, even if they are made of the same technology. Consequently, the various features must be preprocessed, so as to bring them onto common grounds. This is typically done by standardizing the features or by normalizing the feature vectors. Another problem is that many of the algorithms assume some metric for the feature space. For example, the popular K-nearest-neighbors (KNN) algorithm calculates the Euclidean distance between pairs of feature vectors, and PCA can be interpreted as a projection in an Euclidean space. Yet, there is no *a priori* reason to associate any specific metric with a particular feature space.

We suggest a different approach, which enables visualization and classification without having to use directly the feature vectors. Instead, we propose a simple method for measuring the amount of similarity between any two measurements, in a way that allows for natural comparison between differently scaled signals. We further support this approach by introducing two algorithms, one for visualization and one for classification, which take these similarities as their input. We show that the resulting visualization is of high quality and often contains information not present in standard approaches, and that the probability of classification errors is very low.

## Experimental

We have tested our algorithms against a large dataset that we have collected using the MOSESII eNose [6] with two sensor modules: an eight-sensor quartz-microbalance (QMB) module, and an eight-sensor metal-oxide (MOX) module. (Reviews on these technologies can be found in, e.g., [2] or [7].) The samples were put in 20-ml vials in HP7694 headspace sampler, which heated them to 40ºC and injected the headspace content into the eNose. There, the analyte was first introduced into the QMB chamber, whence it followed to the 300ºC heated MOS chamber. The injection lasts for 30 seconds, and is followed by a 15 minute purging stage using synthetic air.

The dataset includes 30 volatile odorous pure chemicals listed alphabetically in Table 1. These chemicals were intentionally chosen from many different families, so that they would represent a broad range of possible stimuli. Each chemical was measured in batches, with a single batch containing at least seven successive measurements. Different batches of the same chemical were usually taken in totally different dates. In total, we have performed 300 measurements, with an average of ten per chemical.

## The Similarity Matrix

For $n$ measurements, the similarity matrix $S$ is an $n \times n$ matrix, with $S_{ij}$ measuring the similarity between measurements $i$ and $j$. Preferably, the similarity values should be normalized, i.e., scaled between 0 and 1, with a 1 denoting identical measurements.

**Table 1: The 30 pure chemicals in our dataset**

| | |
|---|---|
| 1. 1s-(-)-α-pinene | 16. ethyl-2-methylbutyrate |
| 2. 1s-(-)-β-pinene | 17. ethyl-3-methylthiopropionate |
| 3.1-phenyl-1,2-propanedione | 18. ethyl-n-valerate |
| 4. 2-acetylpyridine | 19. ethyl acetoacetate |
| 5. 2,3-heptanedione | 20. ethyl caproate |
| 6. 4-methylanisole | 21. ethylpyrazine |
| 7. alpha-angelica lactone | 22. phenylacetaldehyde dimethyl acetal |
| 8. amyl butyrate | 23. propylidene phthalide |
| 9. butyl butyrate | 24. R-(-)-limonene |
| 10. butyl butyryl lactate | 25. S-(-)-limonene |
| 11. butylidene phthalide | 26. terpinotene |
| 12. cis-3-hexenyl acetate | 27. trans-2-hexenal |
| 13. cis-6-nonenal | 28. trans-2-hexenol |
| 14. citral | 29. trans-2-methyl-2-pentenoic |
| 15. ethyl-2-methyl-4-pentenoate | 30. trans-2-octenal |

As we shall see next, we are able to obtain such normalized similarity values, exploiting a feature extraction technique developed by our group [1]. The idea that underlies this technique is to model the time-dependency of the response by an analytic expression, which is completely characterized by a small set of parameters; these parameters are then taken as the feature set. For every measurement we find the corresponding values of the features by carrying a fast and robust curve-fitting procedure. The analytic expression, called the ***Lorentzian model***, is derived from a very simple physical description of the measurement process. The Lorentzian model is explicitly written as

$$r(t)=\begin{cases} 0 & t<t_0 \\ \beta\tau\tan^{-1}\left(\frac{t-t_0}{\tau}\right) & t_0\le t\le t_0+T \\ \beta\tau\left[\tan^{-1}\left(\frac{t-t_0}{\tau}\right)-\tan^{-1}\left(\frac{t-t_0-T}{\tau}\right)\right] & t>t_0+T, \end{cases}$$

where $r(t)$ stands for the time response of a certain sensor. The model employs four physically interpretable parameters: $\beta$ is a measure of the signal's amplitude, $\tau$ represents its typical decay time, $t_0$ is the time when the signal starts to rise, and $T$ is the time is takes to achieve the maximum. We have demonstrated [1], that any measured signal is described with high precision by this model. An example is brought in Figure 1, where a measured signal is plotted together with its corresponding Lorentzian model.

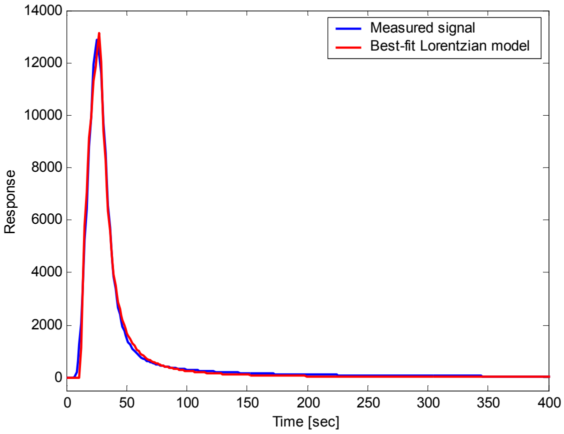

**Figure 1: A comparison between a typical signal (cis-3-hexenyl acetate measured with a QMB sensor) and the best-fit Lorentzian model. The differences between the measured signal and the model are hardly distinguishable. Here, $\beta$ = 1435.8, $\tau$ = 8.29, $t_0$ = 11.21, and $T$ = 16.6.**

The ***correlation function*** between two time signals $f(t)$ and $g(t)$ is given by:

$$\operatorname{corr}(f,g)=\int_{-\infty}^{\infty} f(\tau+t)g(\tau)d\tau\,.$$

Although not explicitly symbolized, $\operatorname{corr}(f,g)$ is a function of the time $t$. Let us denote by $c(f,g)$ the maximum of this function, $c(f,g)=\max(\operatorname{corr}(f,g))$. Intuitively, $c(f,g)$ expresses the highest possible match between the two functions $f(t)$ and $g(t)$, and thus is strongly associated with their similarity. Actually, it can be shown that we can define a normalized similarity index between the functions $f(t)$ and $g(t)$ as the ratio

$$\frac{c(f,g)}{\sqrt{c(f,f)\cdot c(g,g)}}\,.$$

A two-dimensional analog of this measure is oftentimes used in the field of image processing.

We can use this index together with the Lorentzian model as follows: Let $r_k^i(t)$ be the time response of the $k$'th sensor on the $i$'th measurement. Then, the similarity index (for sensor $k$ alone) between the two measurements $i$ and $j$ is:

$$S_{ij}^k=\frac{c(r_k^i,r_k^j)}{\sqrt{c(r_k^i,r_k^i)\cdot c(r_k^j,r_k^j)}}\,. \qquad (1)$$

We define the total similarity between these two measurements as the average over all the *m* sensors,

$$S_{ij} = \frac{1}{m}(S_{ij}^{1} + S_{ij}^{2} + \ldots + S_{ij}^{m}) . \qquad (2)$$

Our definitions impose unit self-similarities, i.e, $S_{ii} = 1$ for all *i*. However, for reasons to become clear in the next section, we deliberately force all the self-similarities to be zero, $S_{ii} = 0$ .

## Visualization

In order to use the similarity matrix for data visualization, we borrow tools from the field of graph drawing. A graph is usually written as $G(V, E)$ , where $V = \{1, \ldots, n\}$ is the set of *n* nodes, and *E* is the set of weighted edges, $w_{ij}$ being the non-negative weight of the edge connecting nodes *i* and *j*. For drawing purposes, the weights are interpreted as measures of similarity, such that more similar nodes are connected with larger weights. Henceforth, we will assume $w_{ij} = 0$ for any non-adjacent (disconnected) pair of nodes.

The weighted sum of edges connected to a particular node is defined as its ***degree***,

$$d_i = \sum_{k=1}^{n} w_{ik} \ .$$

A high degree node would probably be a "central" one, in the sense that it would probably be connected to many other nodes. The ***Laplacian*** of the graph is the symmetric $n \times n$ matrix *L*, where

$$L_{ij} = \begin{cases} -w_{ij} & i \neq j \\ d_i & i = j. \end{cases}$$

For some graphs, each node is also associated with a strictly positive number, $m_i$ , known as its ***mass***. The ***mass matrix*** *M* is the $n \times n$ diagonal matrix that satisfies $M_{ii} = m_i$ .

In [5] we describe an algorithm for drawing such a graph, using a technique of energy minimization. Here, we outline the algorithm in much brevity, and the interested reader is referred to [5] for more details. The idea is to draw the graph in one dimension, by assigning a coordinate $x_i$ to each node *i*, via the minimization of the ***Hall energy*** function

$$E(x) = \tfrac{1}{2} \sum_{i,j=1}^{n} w_{ij} (x_i - x_j)^2 \ ,$$

which strives to make edge lengths short. Using the formerly defined Laplacian, this energy function can be compactly written as $E(x) = x^T Lx$ , with *x* being the vector of coordinates, $x = (x_1, x_2, \ldots, x_n)^T$ . Actually, to avoid degenerate solutions, the following constrained minimization problem is the one that should be solved:

$$\begin{aligned} &\min_x \ x^T Lx \\ &\text{given } x^T Mx = 1 \\ &\text{in the subspace } x^T M \cdot 1_n = 0. \end{aligned}$$

Here $1_n$ is the *n*-vector $(1,1,\ldots,1)^T$ . The constraint $x^T Mx = 1$ poses an overall scaling to the drawing. For, if $x_0$ is a minimizer of the problem with energy $E_0 = x_0^T Lx_0$ , then $\sqrt{c}x_0$ (with energy $cE_0$ ) will be a minimizer of the same problem but with the constraint $x^T Mx = c$ . The constraint $x^T M \cdot 1_n = 0$ limits us only to solutions that obey $\sum_i m_i x_i = 0$ , thus avoiding the degenerate solution of putting all the nodes at the same location.

The extrema of the Hall energy under the above constraints are obtained for those *x* that are the solutions of the generalized eigenvalue problem of $(L, M)$ ,

$$Lx = \lambda Mx \ .$$

The value of the Hall energy at these extrema is simply the corresponding generalized eigenvalues *λ*. It can be proved that all the eigenvalues are non-negative, with a zero eigenvalue corresponding to a degenerate solution. Consequently, the optimal one-dimensional drawing of the graph is obtained by taking the vector of coordinates as the generalized eigenvector associated with the smallest positive generalized eigenvalue. If it is desired to plot the graph in more dimensions, subsequent generalized eigenvectors may be taken. Thus, a two-dimensional drawing is obtained by taking the *x*-coordinates of the nodes to be given by the smallest positive generalized eigenvector, and the *y*-coordinates to be given by the next smallest generalized eigenvector. The entire drawing technique is appropriately called the ***eigenprojection*** method.

This approach to graph drawing yields impressive drawings when applied to many kinds of graphs, see [5]. But how can it be used in the context of eNose data visualization? We can think of each measurement as a node in a graph, and identify the similarity indices with the edge weights. Moreover, to better reflect the relative dominance of the so produced nodes, we associate with each measurement a mass which is equal to its degree, $m_i = d_i$, see also [4]. The reason for this is that the drawing algorithm tends to push heavier nodes towards the origin of coordinates, thus placing presumably more central nodes in the center of the drawing. We then solve the generalized eigenvalue problem $Lx = \lambda Mx$ , taking the first two smallest positive generalized eigenvectors as the *x* and *y* coordinates of the nodes.

We seal this theoretical discussion by giving a simple-to-follow recipe for those who wish to implement our technique:

1. Calculate the similarities between the measurements using formulae (1) and (2).
2. Zero self-similarities, i.e., set $S_{ii} = 0$ for all $i$.
3. Construct the diagonal mass matrix $M$, where $M_{ii} = d_i = \sum_{k=1}^{n} S_{ik}$ .
4. Construct the Laplacian $L = M - S$ .
5. Solve the generalized eigenvalue problem $Lx = \lambda Mx$ .
6. Determine the $x$ and $y$ coordinates of the nodes as the two generalized eigenvectors with the smallest and next smallest positive generalized eigenvalues, respectively.

Matlab implementation and examples can be found in www.wisdom.weizmann.ac.il/~liran.

Figure 2 shows the two-dimensional drawing obtained by applying the eigenprojection method to our 30 chemical dataset. In the figure, each dot is a measurement, color-coded by the odor species. For comparison, the orthodox drawing technique of applying PCA to standardized height feature vectors is also brought, see Figure 3. On the upmost level, the two drawing techniques seem quite comparable. A closer look, however, reveals the following:

**Outlier detection**: The most prominent property of the eigenprojection drawing (Figure 2) is the sharp partition of the space into a large central region, and a small one, containing only three chemicals, near the upper right corner. An inspection of the three isolated chemicals reveals that they are true outliers, with some of their sensors showing anomalous double peak time response. Figure 4 shows an example to this behavior, probably caused by some kind of "chromatographic effect". The corresponding PCA drawing (Figure 3) does not distinguish these outliers at all. The combination of the Lorentzian model feature extraction and the eigenprojection visualization technique turns out, therefore, to form a powerful tool for outlier detection.

**General layout (qualitative)**: The way clusters are spread in space in the eigenprojection method is better observed if we zoom on the central region alone, as is done in Figure 5. It might be slightly difficult to see in the resolution of Figures 3 and 5, but the two layouts exhibit impressive discriminatory power, with the different clusters being hardly overlapping.

**Cluster separability (quantitative)**: So both drawings are discriminatory to some extent. But which is more so? We can obtain a crude estimate of the quality of a drawing by measuring how well separated are the different clusters. To this end let us define the separability index of the drawing as

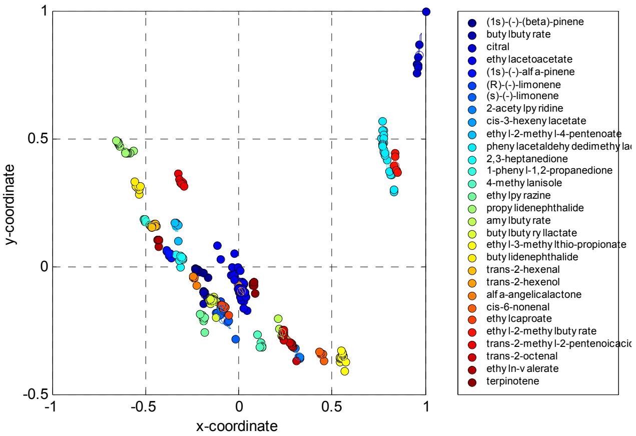

**Figure 2: Visualization of our 30 chemical dataset using the eigenprojection method.**

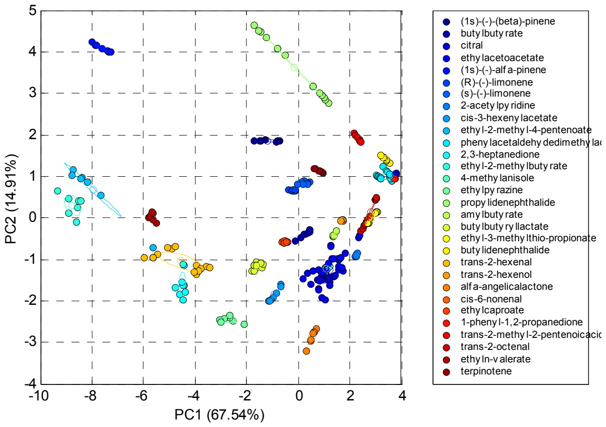

**Figure 3: Visualization of our 30 chemical dataset using PCA on standardized height feature vectors.**

$$I = \frac{\mathrm{Tr}(S_B)}{\mathrm{Tr}(S_W)} . \qquad (3)$$

Here $S_W$ is the average within-cluster covariance matrix (the average scatter of a cluster), and $S_B$ is the between-cluster covariance matrix (the scatter of the clusters themselves, where each cluster is represented by its centroid). For more information on these magnitudes see [8]. The reasoning for defining the separability index as in (3) stems from the fact that the trace of a covariance matrix is proportional to the average Euclidean distance between the cluster members. Thus, the higher is $I$, the denser the clusters, and better is their separation. The eigenprojection drawing (Figure 1) gives $I = 147.9$, while the PCA drawing (Figure 2) gives an index smaller by a factor of three, $I = 53.2$. We therefore expect the eigenprojection method to have more powerful discriminatory abilities, as is indeed supported by the results that we bring in the next section.

**Cluster shape (qualitative)**: Comparing Figures 3 and 5, one can notice that the clusters in the eigenprojection drawing tend to be circular, while those in the PCA drawing tend to be cigar-shaped. The most outstanding example of this is the propylidenephthalide cluster (uppermost cluster in Figure 3, uppermost and leftmost cluster in Figure 5), which is elongated in the PCA drawing, but much more compact in the eigenprojection drawing. Probably, this compactness of the eigenprojection clusters is one of the reasons for the high separability index in the eigenprojection drawing.

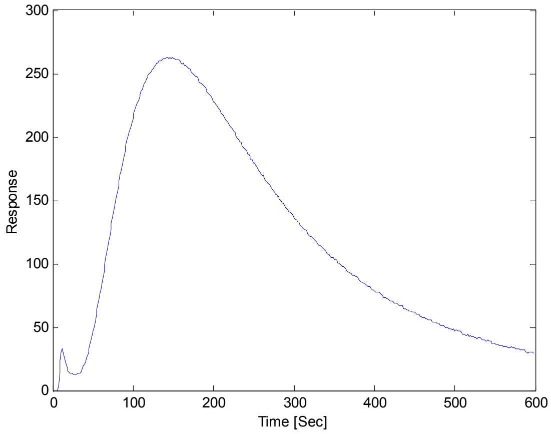

**Figure 4: A signal of one of the outliers (citral measured with a QMB sensor), showing the double-peaked behavior.**

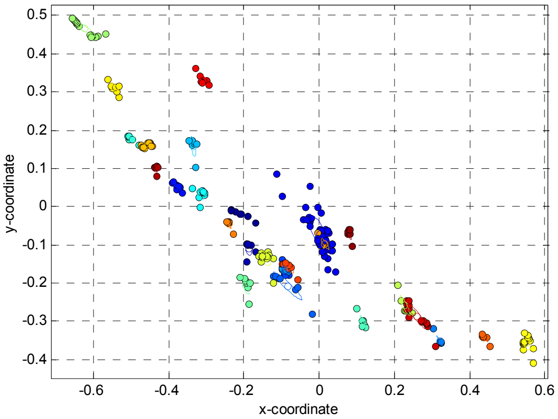

**Figure 5: Zoom on the central region of the eigenprojection drawing presented in Figure 1.**

## Classification

In analogy to the KNN algorithm, we propose to use a **K-most-similar** (KMS) algorithm, whose stages are:

1. Define a ***reference set***, which is the set of representative measurements for which the class association is known in advance.
2. Given an unknown sample, calculate its similarity index with respect to each of the measurements in the reference set.
3. Find the K most similar measurements from within the references set.
4. Associate the unknown sample with the cluster that contains the majority of the K most similar measurements.

The probability of classification error in this method appears to be very small. For our 30 chemical dataset, the classification success rate is 96.7%. Here we take the first 7 measurements of each odor as references, and treat subsequent measurements as unknown samples. Thus, 70% of the measurements were taken as references, and 30% as unknown samples. For comparison, a KNN algorithm in the original feature space yields in this case only a 72.2% rate of correct classification.

## Summary

Standard methodology of eNose data analysis dictates the use of visualization and classification algorithms directly on the feature vectors. We suggest here an alternative approach, utilizing the Lorentzian model feature extraction technique to obtain a similarity index for each pair of measurements. We present a visualization algorithm and a classification algorithm, both operate on these similarity indices.

The visualization algorithm uses tools borrowed from the field of graph drawing. It was shown to have powerful discrimination ability, stronger about three times than PCA. Moreover, it was demonstrated to be capable of efficiently screening out outliers.

The classification algorithm is a similarity index oriented variation on the KNN algorithm. However, when compared to KNN, it was shown to have significantly lower rate of classification errors.

## REFERENCES

[1] Carmel L., Levy S., Lancet D. and Harel D., "A New Feature Extraction Technique for Electronic Noses", *Proc. 9th International Meeting on Chemical Sensor,* 2002.

[2] Gardner J. W. and Bartlett P. N., *Electronic Noses, Principles and Applications*, Oxford University Press, New York, NY, 1999.

[3] Gutierrez-Osuna R., "Pattern Analysis for Machine Olfaction: A Review", IEEE Sensors Journal (June 2002).

[4] Koren Y., "On Spectral Graph Drawing", manuscript, (2002).

[5] Koren Y., Carmel L. and Harel D., "ACE: A Fast Multiscale Eigenvectors Computation for Drawing Huge Graphs", *Proc. IEEE Symposium on Information Visualization 2002* (*InfoVis 2002*), Boston, USA, to appear, 2002.

[6] Mitrovics J., Ulmer H., Weimar U. and Göpel W., "Modular Sensor Systems for Gas Sensing and

Odor Monitoring: the MOSES Concept", Acc. Chem. Res. **31** (1998) 307-315.

[7] H. T. Nagle, S. S. Schiffman, and R. Gutierrez-Osuna, "The how and why of electronic noses", IEEE Spectrum (Sept. 1998) 22-34.

[8] Webb A., *Statistical Pattern Recognition*, Arnold, 1999.