A clustering neural network model of insect olfaction

Cengiz Pehlevan[1], Alexander Genkin[2], and Dmitri B. Chklovskii[1,2]

[1]Center for Computational Biology, Flatiron Institute, New York, NY
[2]NYU Langone Medical Center, New York, NY

***Abstract*— A key step in insect olfaction is the transformation of a dense representation of odors in a small population of neurons - projection neurons (PNs) of the antennal lobe - into a sparse representation in a much larger population of neurons - Kenyon cells (KCs) of the mushroom body. What computational purpose does this transformation serve? We propose that the PN-KC network implements an online clustering algorithm which we derive from the k-means cost function. The vector of PN-KC synaptic weights converging onto a given KC represents the corresponding cluster centroid. KC activities represent attribution indices, i.e. the degree to which a given odor presentation is attributed to each cluster. Remarkably, such clustering view of the PN-KC circuit naturally accounts for several of its salient features. First, attribution indices are nonnegative thus rationalizing rectification in KCs. Second, the constraint on the total sum of attribution indices for each presentation is enforced by a Lagrange multiplier identified with the activity of a single inhibitory interneuron reciprocally connected with KCs. Third, the soft-clustering version of our algorithm reproduces observed sparsity and overcompleteness of the KC representation which may optimize supervised classification downstream.**

I. Introduction

In the quest to understand neural computation, olfaction presents an attractive target. Thanks to recent technological developments, there is a wealth of experimental data on olfactory processing. Stereotypy of early olfactory processing between vertebrates and invertebrates suggests the existence of universal principles. Yet, a comprehensive algorithmic theory of olfaction remains elusive.

Here, we attempt to understand the computational task solved by one key olfactory circuit in invertebrates [1], Figure 1, top: antennal lobe (AL) projection neurons (PNs) synapsing onto a much larger population of Kenyon cells (KCs) in the mushroom body (MB) which in turn are reciprocally inhibited by a single giant interneuron (GI, which is called GGN in the locust [2], [3] and APL in *Drosophila* [4], [5]). The PN-KC-GI circuit transforms a dense PN activity vector into a sparse representation in KCs [6], [7], [8]. The lack of a teaching signal suggests that this transformation is learned in an unsupervised setting. In contrast, learning the weights of KC synapses onto MB output neurons (MBONs) is thought to rely on reinforcement [9].

II. Related work

Many existing publications proposed mechanistic models of the PN-KC-GI circuit and attempted to explain known structural and functional features. In particular, they aimed to explain the sparseness of the KC activity [10], [2], [11], [12], [13] and studied how sparseness and other features affect olfactory learning and generalization at the output of KCs [14], [15], [16], [17]. A recent work proposed that KC representations solve a similarity search problem by representing similar odors with similar KC responses [18], following the similarity matching principle [19].

However, these papers did not attempt to derive the observed network structure and function from a principled computational objective. Such derivation is the goal of the present paper.

A. Demixing network models

Previous attempts to derive network structure and function assumed that the computational task of olfactory processing, in general [20], [21], [22], and the PN-KC-GI circuit, in particular [23], [24], [25], is compressive sensing or sparse autoencoding. In this view, odors are sparse vectors in the space of concentrations of all possible molecules. ORNs project these vectors onto the vectors of receptor affinities resulting in (normalized) vectors of AL PN activity. Then, the PN-KC projection recovers the concentration of each constituent molecule in the activity of a corresponding KC.

However, modeling olfaction with a compressive sensing or autoencoding computational objective has three problems. First, experimentally measured KC responses seem to contradict the demixing theory. Even though many KCs often respond selectively to a single component in an odor mixture [21], some KCs respond to specific mixtures but not to their components by themselves [21].

Second, circuits proposed for computing sparse representations contradict the PN-KC-GI circuit architecture. They require one of the following: an all-to-all KC connectivity [26], [27], [28], [29], many inhibitory interneurons [30], [31], [32], [33], [34], [35], or strictly feedforward architecture [23], [24], [20], all three inconsistent with the single inhibitory interneuron in the MB.

Third, demixing models often interpret nonnegative neural activity in KCs (spiking neurons cannot have negative firing rates) as a manifestation of the physical nonnegativity of molecular concentrations (counts of molecules cannot be negative) [36]. However, such explanation does not generalize to most other neurons in the brain whose activity is nevertheless nonnegative. Hence, there must be another reason.

cpehlevan@flatironinstitute.org
alexander.genkin@gmail.com
mitya@flatironinstitute.org

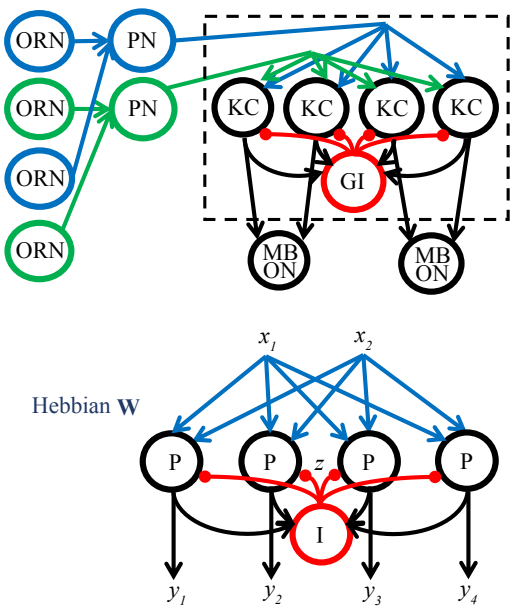

Fig. 1. Top: Schematic of the insect olfactory system. Odors are transduced in the olfactory receptor neurons (ORNs). Each projection neurons (PN) receives inputs from ORNs of the same class in the AL glomerulus. Then, PNs (numbering $\approx$ 800 in the locust and $\approx$ 150 in *Drosophila*) synapse onto Kenyon cells (KCs) (numbering $\approx$ 50,000 in the locust [37] and $\approx$ 2,000 in *Drosophila* [9]) of the mushroom body (MB). A single giant interneuron (GI) provides reciprocal inhibition to KCs [4], [5]. KCs project onto a smaller number (34 in *Drosophila* [9]) of MB output neurons (MBONs). The dashed box delineates the circuit modeled in this paper. Bottom: A biologically plausible clustering network whose architecture is remarkably similar to the insect MB (P-principal neuron, I-interneuron).

B. Classic online k-means algorithm and neural networks

In this paper, we explore the hypothesis that the computational task of the PN-KC-GI circuit is to cluster streaming olfactory stimuli. In this subsection, we review a classic online clustering algorithm and clustering neural networks.

The goal of clustering is to segregate data into k classes. A popular clustering algorithm, k-means, [38], [39] achieves this goal by minimizing the sum of squared distances between data points, $\{\mathbf{x}_1, ..., \mathbf{x}_T\}$, and cluster centroids, $\{\mathbf{w}_1, ..., \mathbf{w}_k\}$, specified by the attribution indices $y_{i,t},\ i \in [1,k],\ t \in [1,T]$:

$$\min_{\mathbf{y}_{t=1..T}, \mathbf{w}_{i=1..k}} \sum_t \sum_i y_{i,t} ||\mathbf{x}_t - \mathbf{w}_i||^2, \qquad (1)$$
$$\text{s.t.} \quad y_{i,t} \in \{0,1\} \text{ and } \sum_i y_{i,t} = 1.$$

Here and below, vectors are lowercase boldface and matrices capital boldface.

Importantly, a biologically plausible algorithm must minimize Eq. (1) in the online (or streaming) setting. Indeed, sensory organs stream data to neural networks sequentially, one datum at a time, and the network has to compute the corresponding attribution indices on the fly without seeing the full dataset or storing any significant fraction of past inputs in memory.

The classic online k-means algorithm [38] performs alternating minimization of Eq. (1) by, first, assigning current datum, $\mathbf{x}_t$, to the cluster with the closest centroid, $\mathbf{w}_c$, i.e. "winner-take-all" (WTA), and, second, updating the number of data points in the winning cluster, n_c, and the centroid of that cluster:

Algorithm 1 Classic online algorithm for k-means clustering

1: Initialize $\mathbf{w}_1$, and $\mathbf{n} = \mathbf{1}$
2: Repeat for each $t = 1, \ldots$
3: Set

$$c = \arg\min_i ||\mathbf{x}_t - \mathbf{w}_i||^2, \qquad y_{i,t} \leftarrow \delta_{i,c} \qquad (2)$$

4: Update

$$n_c \leftarrow n_c + 1, \qquad \mathbf{w}_c \leftarrow \mathbf{w}_c + \frac{1}{n_c}(\mathbf{x}_t - \mathbf{w}_c), \qquad (3)$$

The online k-means algorithm is often given a neural interpretation [40], where $y_{i,t}$ is the activity of the ith output neuron (KC) in response to input $\mathbf{x}_t$ and $\mathbf{w}_i$ is the synaptic weight vector impinging onto that neuron. Yet, such interpretation is difficult to reconcile with known biology. First, computing squared distances in Eq. (2) seems to require the availability of $\mathbf{x}_t$ and $\mathbf{w}_i$ at the same location in a neuron. Contrary to that, while $\mathbf{x}_t$ is available only to the upstream neurons (PNs) and their synapses, the KC soma sees only $\mathbf{x}_t^\top \mathbf{w}_i$. This difficulty could be circumvented by expanding the square and dropping the term, $||\mathbf{x}_t||^2$, constant among all KCs. Then, the argument becomes $-2\mathbf{x}_t^\top \mathbf{w}_i + ||\mathbf{w}_i||^2$ which could be computed in each KC's soma. Second, and more serious, difficulty is to actually find the minimum, (2), over all KCs. To compare $-2\mathbf{x}_t^\top \mathbf{w}_i + ||\mathbf{w}_i||^2$ among KCs, the KCs would have to output that quantity, contradicting the assumption that they output $y_{i,t}$.

Whereas neural implementations of clustering have been proposed before (reviewed in [40]), they either lack biological plausibility or are not derived from a principled objective. The central idea behind clustering neural network is competitive learning where neurons compete to assign a datum to their associated cluster by WTA dynamics. The "winner" neuron's synaptic weights, encoding that cluster's centroid, are updated using a Hebbian rule. Examples of such algorithms include the classic k-means algorithm (see Section 2.3) [38], [39], the self-organizing maps (SOM) [41], the neural gas [42], the adaptive resonance theory (ART) [43], and the nonnegative similarity matching (NSM) network [28]. A probabilistic interpretation of WTA networks with Hebbian learning is given in [44].

III. Our contributions

This paper makes four main contributions: i) Starting from the classic k-means objective function, we derive an online algorithm that can be implemented by activity dynamics and synaptic plasticity rules in a biologically plausible neural network. In contrast, ART, SOM, the neural gas, and the classic k-means algorithms all assume that WTA operation

is implemented in some biologically plausible way without specifying the network architecture or dynamics [45]. Hebbian plasticity, but not WTA dynamics, has been related to a clustering cost function in [46]. ii) The derived clustering network bears a striking resemblance to the architecture and dynamics of the PN-KC-GI circuit. Specifically, it accounts for the rectification in KCs as their activity represents nonnegative attribution indices and the existence of a single interneuron as its activity represents the Lagrange multiplier arising from the norm constraint (1). iii) In contrast with existing demixing models, we interpret activity of a KC as the presence of an olfactory object, which may be a complex odor made of many components or a simple odor of a single component, consistent with experimental findings [21]. iv) We extend our model to the soft-clustering scenario in which there could be multiple "winner" neurons. Such sparse, overcomplete representation is both reminiscent of the insect mushroom body and may be optimal for learning supervised classification downstream.

IV. A NEURAL ONLINE k-MEANS ALGORITHM

In this Section, we present a new online k-means algorithm that overcomes both biological plausibility difficulties mentioned above.

A. Derivation of a neurally inspired online algorithm

We start by relaxing the constraint, $y_{it} \in \{0,1\}$, in (1):

$$\min_{\mathbf{y}_{t=1..T}, \mathbf{w}_{i=1..k}} \sum_t \sum_i y_{i,t} ||\mathbf{x}_t - \mathbf{w}_i||^2, \quad \text{s.t.} \quad \sum_i y_{i,t} = 1, \;\; y_{i,t} \geq 0. \tag{4}$$

To see that, despite relaxation, the optimum of Eq. (4) has the property $y_{it} \in \{0,1\}$, note that, for any value of $\mathbf{w}_i$s, the problem separates by data points. Then, for each t, we get:

$$\min_{\mathbf{y}_t} \sum_i y_{i,t} ||\mathbf{x}_t - \mathbf{w}_i||^2, \;\; \text{s.t.} \;\; \sum_i y_{i,t} = 1, \;\; y_{i,t} \geq 0. \tag{5}$$

The solution of this problem is obviously, $y_{i^*,t} = \delta_{i,i^*}$ where $i^* = \arg\min_i ||\mathbf{x}_t - \mathbf{w}_i||^2$, and δ_{i,i^*} is the Kronecker delta, i.e., the same as for (1).

Next, we eliminate cluster centroids, $\mathbf{w}_i$s, from the objective. By taking a derivative of Eq. (4) with respect to $\mathbf{w}_i$ and setting it to zero we find that the optimal cluster centroid is given by the center of mass, $\mathbf{w}_i = \sum_t y_{i,t}\mathbf{x}_t / \sum_t y_{i,t}$. After substituting this expression into Eq. (4), some algebra, and using the Lagrangian form for the constraint, we obtain:

$$\min_{\mathbf{y}_{t=1..T} \geq 0} \max_{z_{t=1..T}} - \sum_i \frac{\sum_t \sum_{t'} y_{i,t} y_{i,t'} \mathbf{x}_t^\top \mathbf{x}_{t'}}{\sum_t y_{i,t}} + \sum_t z_t \left(\sum_i y_{i,t} - 1 \right). \tag{6}$$

To derive an online algorithm, we reason that future inputs have not been seen and past outputs and Lagrange multipliers cannot be modified. Therefore, at each time step, T, we optimize Eq. (6) only with respect to the latest outputs $\mathbf{y}_T := [y_{1,T}, \ldots, y_{k,T}]^\top$ and the Lagrange multiplier, z_T. Specifically, we solve:

$$\min_{\mathbf{y}_T \geq 0} \max_{z_T} - \sum_i \frac{\sum_t \sum_{t'} y_{i,t} y_{i,t'} \mathbf{x}_t^\top \mathbf{x}_{t'}}{\sum_t y_{i,t}} + \sum_t z_t \left(\frac{\sum_{t'} y_{i,t'}}{\sum_{t'} y_{i,t'}} \sum_i y_{i,t} - 1 \right), \tag{7}$$

where we introduced the factor $1 = \sum_{t'} y_{i,t'} / \sum_{t'} y_{i,t'}$ inside the parenthesis, which will be crucial for deriving an algorithm asymptotically equivalent to k-means (see Appendix I). Next, we drop i) the negligible current values from the sums in the denominators by invoking the large-T limit, ii) $y_{i,T}$ or $z_{i,T}$ independent terms, and iii) the negligible terms $||\mathbf{x}_T||^2 \mathbf{y}_T^\top \mathbf{N}_T^{-1} \mathbf{y}_T$ and $z_T \mathbf{y}_T^\top \mathbf{N}_T^{-1} \mathbf{y}_T$ by again invoking the large-T limit. Finally, we arrive at:

$$\min_{\mathbf{y}_T \geq 0} \max_{z_T} \mathbf{a}_T^\top \mathbf{y}_T + z_T \left(\mathbf{1}^\top \mathbf{y}_T - 1 \right), \tag{8}$$

where

$$\mathbf{a}_T := -\mathbf{W}_T \mathbf{x}_T + \boldsymbol{\theta}_T, \qquad \mathbf{N}_T := \text{diag}\left(\sum_{t<T} \mathbf{y}_t \right),$$
$$\mathbf{W}_T := 2\mathbf{N}_T^{-1} \sum_{t<T} \mathbf{y}_t \mathbf{x}_t^\top, \qquad \boldsymbol{\theta}_T := \mathbf{N}_T^{-1} \sum_{t<T} z_t \mathbf{y}_t. \tag{9}$$

To guarantee invertibility of $\mathbf{N}_T$ we initialize it with an identity matrix. We use a Matlab-inspired notation where, for any matrix $\mathbf{M}$, diag($\mathbf{M}$) is a vector equal to $\mathbf{M}$'s main diagonal and, for any vector $\mathbf{v}$, diag($\mathbf{v}$) is a diagonal matrix with $\mathbf{v}$ on the main diagonal.

For each datum, our neurally-inspired Algorithm 2 performs two steps: i) Optimizes (8), ii) Recursively updates $\mathbf{N}_T$, $\mathbf{W}_T$ and $\boldsymbol{\theta}_T$ (Eq. (11) below) using optimal $\mathbf{y}_T$ and z_T. Because the optimal $y_{i,T} \in \{0,1\}$, diag($\mathbf{N}_T$) = $\mathbf{n}_T$ counts data points assigned to clusters until time, T.

Algorithm 2 Neural algorithm for k-means clustering

1: Initialize $\mathbf{W}_1$, $\boldsymbol{\theta}_1$ and $\mathbf{n}_1 = \mathbf{1}$.
2: Repeat for each $T = 1, \ldots$
3: Set

$$\{\mathbf{y}_T, z_T\} \leftarrow \arg\min_{\mathbf{y}_T \geq 0} \arg\max_{z_T} \mathbf{a}_T^\top \mathbf{y}_T + z_T \left(\mathbf{1}^\top \mathbf{y}_T - 1 \right). \tag{10}$$

where $\mathbf{a}_T = -\mathbf{W}_T \mathbf{x}_T + \boldsymbol{\theta}_T$.

4: Update

$$\begin{aligned} n_{i,T+1} &= n_{i,T} + y_{i,T} \\ W_{ij,T+1} &= W_{ij,T} + \frac{1}{n_{i,T+1}} \left(2 y_{i,T} x_{j,T} - y_{i,T} W_{ij,T} \right) \\ \theta_{i,T+1} &= \theta_{i,T} + \frac{1}{n_{i,T+1}} \left(y_{i,T} z_T - y_{i,T} \theta_{i,T} \right). \end{aligned} \tag{11}$$

Step 3 of Algorithm 2, i.e. Eq.(8), is the Lagrange form (with z_T as the Lagrange multiplier) of the following

optimization problem:

$$\min_{\mathbf{y}_T \geq 0} \mathbf{a}_T^\top \mathbf{y}_T, \quad \text{s.t. } \mathbf{1}^\top \mathbf{y}_T = 1, \tag{12}$$

As Eq.(12) is a linear program on the simplex, it is optimized at the vertex with the smallest inner product value. Let $c = \arg\min_i a_{i,T}$, then $y_{i,T} = \delta_{i,c}$, recovering the WTA rule.

To find the value of z_T, we write the unconstrained Lagrangian: $L(\mathbf{y}_T, z_T, \boldsymbol{\lambda}) = \mathbf{a}_T^\top \mathbf{y}_T + z_T(\mathbf{1}^\top \mathbf{y}_T - 1) + \boldsymbol{\lambda}^\top \mathbf{y}_T$. From complementary slackness, $\lambda_c = 0$ and, from stationarity, $\mathbf{a}_T + z_T \mathbf{1} + \boldsymbol{\lambda} = 0$ [47]. Therefore, $z_T = -a_{c,T}$ and we arrive at the following simplification:

Step 3 of Algorithm 2

$$c = \arg\min_i a_{i,T}, \qquad y_{i,t} \leftarrow \delta_{i,c} \qquad z_T = -a_{c,T}. \tag{13}$$

Comparing Algorithm 2 with classic k-means (Algorithm 1) we note several similarities. Cluster centroids (rows of $\frac{1}{2}\mathbf{W}_T$) are centers of mass of the data points assigned to the corresponding clusters. Their update, (11), is identical to that in the classic online k-means algorithm (3). Both algorithms employ WTA dynamics.

Algorithm 2 differs from Algorithm 1 in that the "winner" is not determined by the Euclidean distance from cluster centroid to data point but by solving a linear program Eq. (12). Yet, despite this difference, Algorithm 2 is asymptotically equivalent to Algorithm 1 because $a_{i,t} \approx -2\mathbf{x}_t^\top \mathbf{w}_i + ||\mathbf{w}_i||^2$ (see Appendix I). This solves the first difficulty with implementing k-means using neural networks.

Whereas Algorithm 2 with (13) gets us closer to a biological implementation by dropping the requirement for the availability of the terms $||\mathbf{x}_t||^2$ and $||\mathbf{w}_i||^2$ at each output neuron's soma, it is still not clear how to implement the $\min$ operation biologically. The utility of Algorithm 2 with (13), which can be run on the CPU architecture, is due to its speed.

B. A neural circuit implementation

In order to implement the online algorithm 2 by a biologically plausible neural network, we solve the augmented Lagrangian of the linear program in Eq. (12) by primal-dual descent-ascent algorithm [47] (ρ is the augmented Lagrangian coefficient and η is the step size.):

Biologically plausible Step 3 of Algorithm 2
Iterate until convergence:

$$\begin{aligned} \mathbf{y}_T &\leftarrow \left[\mathbf{y}_T + \eta(\mathbf{W}_T \mathbf{x}_T - \mathbf{1}_k z_T - \boldsymbol{\theta}_T - \rho\left(\mathbf{1}_k^\top \mathbf{y}_T - 1\right))\right]_+, \\ z_T &\leftarrow \left[z_T + \eta(\mathbf{1}_k^\top \mathbf{y}_T - 1)\right] \end{aligned} \tag{14}$$

Using the dynamics in (14), Algorithm 2 has a biologically plausible neural network interpretation, Fig. 1, bottom. Each principal neuron (KC) receives input projected onto a synaptic weight vector (rows of $\mathbf{W}$) representing the corresponding cluster centroid and the output activity of that neuron, $y_{i,T}$, represents the corresponding cluster attribution index. θ_is are neural firing thresholds. The activity of a single inhibitory interneuron (GI) reciprocally connected with principal neurons represents the Lagrange multiplier, z_T. The $\mathbf{W}$ update in Eq. (11) is a local Hebbian synaptic plasticity rule. The update of threshold, $\boldsymbol{\theta}$, is carried out by homeostatic plasticity. Finally, the augmented Lagrangian term may be implemented by the interaction via the common extracellular space shared by multiple principal neurons (KCs) [48], [49].

C. Numerical experiments

To evaluate the neural Algorithm 2 we compared it with the classic Algorithm 1 on the famous Fisher's Iris dataset [50]. This dataset contains 150 samples equally representing three species of Iris: setosa, virginica, and versicolor. Each sample has four measurements: sepal length, sepal width, petals length, petal width (all in centimeters). We initialized the algorithms with three random points from a normal distribution with the mean and standard deviations of the whole data set, with independent coordinates. To emulate online behavior, the samples were permuted randomly. The same initial points and permutation were used for both algorithms.

The two algorithms perform similarly according to two indicators. Table I presents confusion matrices and Rand index values for the classic and neural algorithms, mean and standard deviation over 10 repetitions. Fig. 2, top, shows traces of cluster centroids evolution along the online execution of the two algorithms.

Fig. 2, bottom, gives an example of internal dynamics of the neural Algorithm 2, step 3 with Eq. (14), for one datum.

V. A NEURAL SOFT k-MEANS ALGORITHM

Whereas our model explains the existence of a single inhibitory interneuron, it does not account for the observed activity of KCs. Indeed, in k-means clustering only one attribution index is non-zero for every datum implying that only one KC is active for each odor presentation. In contrast, experiments suggest that about 5-10% of KCs are active in the same time window [8].

A. Soft-clustering online algorithm and neural network

Our framework can explain this observation if, instead of hard clustering, we consider soft clustering where a datum may have multiple non-zero attribution indices. One may think that this problem can be solved by using the fuzzy k-means algorithm [52], where the cost function of Eq. (4) is generalized to

$$\begin{aligned} &\min_{\mathbf{y}_{t=1..T}, \mathbf{w}_{i=1..k}} \sum_t \sum_i y_{i,t}^m ||\mathbf{x}_t - \mathbf{w}_i||^2, \\ &\text{s.t.} \quad \sum_i y_{i,t} = 1, \;\; y_{i,t} \geq 0, \end{aligned} \tag{15}$$

where $m \geq 1$ ($m = 1$ limit recovers the hard k-means cost). However, fuzzy k-means does not produce sparse output, see Appendix II, in contradiction with experimental observations [6], [7], [8].

Therefore, we propose an alternative formulation which yields soft and sparse clustering and can be related to

TABLE I

CONFUSION MATRICES ON FISHER'S IRIS DATASET.

	Classic Algorithm 1			Neural Algorithm 2		
	Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3
setosa	50.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	50.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
versicolor	0.00 ± 0.00	41.12 ± 8.18	8.88 ± 8.18	0.00 ± 0.00	34.38 ± 4.75	15.62 ± 4.75
virginica	0.00 ± 0.00	7.75 ± 5.95	42.25 ± 5.95	0.00 ± 0.00	2.12 ± 2.23	47.88 ± 2.23
Rand index [51]	0.878 ± 0.016			0.872 ± 0.022		

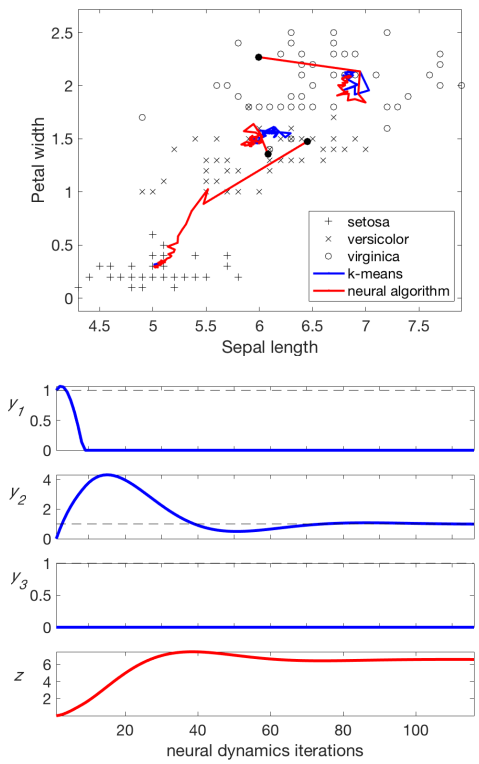

Fig. 2. Performance of the neural Algorithm 2 on Fisher's Iris data. Top: Trajectories of the three cluster centroids traced during the execution of the classic Algorithm 1 (blue) and the neural Algorithm 2 (red) initialized identically (solid black circles). In one case the traces almost coincide, in two other cases they deviate but end up in the similar positions. Shown is a projection of a 4D space onto 2D. Bottom: Neural dynamics during the execution of step 3 with Eq. (14), Algorithm 2, for one datum.

a semidefinite program (SDP) previously considered for clustering [53], [54] and manifold learning [55]. To this end, we add to the k-means cost function an l_2-norm penalty:

$$\min_{\mathbf{y}_{t=1..T},\mathbf{w}_{i=1..k}} \sum_t \sum_i y_{i,t}||\mathbf{x}_t - \mathbf{w}_i||^2 + \alpha T \sum_t \sum_i \frac{y_{i,t}^2}{\sum_{t'} y_{i,t'}},$$
$$\text{s.t.} \quad \sum_i y_{i,t} = 1, \quad y_{i,t} \geq 0, \tag{16}$$

where the additional term in the objective is the only difference from (4), and $\alpha > 0$ is a penalty parameter.

Following the steps in Section IV-A, we arrive at the online optimization problem for every time step:

$$\min_{\mathbf{y}_T \geq 0} \max_{z_T} \mathbf{a}_T^\top \mathbf{y}_T + \alpha T \mathbf{y}_T^\top \mathbf{N}_T^{-1} \mathbf{y}_T + z_T \left(\mathbf{1}^\top \mathbf{y}_T - 1\right). \tag{17}$$

Solving optimization problem 17 and updating variables $\mathbf{W}$, $\boldsymbol{\theta}$ and $\mathbf{N}$ recursively, we arrive at the online neural Algorithm 3.

Algorithm 3 Neural algorithm for soft k-means clustering

1: Initialize $\mathbf{W}_1$, $\boldsymbol{\theta}_1$ and $\mathbf{n}_1 = \mathbf{1}$ ($\mathbf{N}_T = \text{diag}(\mathbf{n}_T)$).

2: Repeat for each $T = 1, \ldots$

3: Set

$$\{\mathbf{y}_T, z_T\} \leftarrow \arg\min_{\mathbf{y}_T \geq 0} \arg\max_{z_T} \mathbf{a}_T^\top \mathbf{y}_T + \alpha T \mathbf{y}_T^\top \mathbf{N}_T^{-1} \mathbf{y}_T + z_T \left(\mathbf{1}^\top \mathbf{y}_T - 1\right). \tag{18}$$

where $\mathbf{a}_T = -\mathbf{W}_T \mathbf{x}_T + \boldsymbol{\theta}_T$.

4: Update

$$\begin{aligned} n_{i,T+1} &= n_{i,T} + y_{i,T} \\ W_{ij,T+1} &= W_{ij,T} + \frac{1}{n_{i,T+1}} \left(2 y_{i,T} x_{j,T} - y_{i,T} W_{ij,T}\right) \\ \theta_{i,T+1} &= \theta_{i,T} + \frac{1}{n_{i,T+1}} \left(y_{i,T} z_T - y_{i,T} \theta_{i,T}\right). \end{aligned} \tag{19}$$

Unlike Eq. (8), Eq. (17) is the Lagrangian form of a quadratic problem with linear constraints (with z_T being the Lagrange multiplier):

$$\min_{\mathbf{y}_T \geq 0} \mathbf{a}_T^\top \mathbf{y}_T + \alpha T \mathbf{y}_T^\top \mathbf{N}_T^{-1} \mathbf{y}_T, \quad \text{s.t.} \ \mathbf{1}^\top \mathbf{y}_T = 1. \tag{20}$$

Step 3 of Algorithm 3 can be carried out by any quadratic program solver. To see why optimal solutions to (20) are sparse, note that for every $y_{i,T} > 0$ the Lagrange multiplier for the non-negativity constraint is zero, so that the stationary condition takes the form:

$$a_{i,T} + 2\alpha T y_{i,T}/n_{i,T} + z_T = 0, \tag{21}$$

from which we have: $y_{i,T} = (-z_T - a_{i,T}) n_{i,T}/2\alpha T$. If all $y_{i,T} > 0$, then using the equality constraint we can define: $z_T = -(\sum_i n_{i,T} a_{i,T} + 2\alpha T)/\sum_i n_{i,T}$. Using positivity of $y_{i,T}$ we have $z_T < -a_{i,T}$, or

$$(\sum_i n_{i,T} a_{i,T} + 2\alpha T)/\sum_i n_{i,T} > a_{i,T}. \tag{22}$$

This inequality contains input variables only and cannot always be guaranteed to hold, so, in those cases, $y_{i,T} = 0$ for some i.

In a biologically plausible neural circuit, the quadratic program, Eq. (20), in the augmented Lagrangian [47] form

can be solved using primal-dual descent-ascent algorithm (ρ - the augmented Lagrangian coefficient and η - the step size):

Biologically plausible Step 3 of Algorithm 3

Iterate until convergence:

$$\mathbf{y}_T \leftarrow \left[\mathbf{y}_T + \eta\left(-\alpha T \mathbf{N}_T^{-1}\mathbf{y}_T + \mathbf{W}_T\mathbf{x}_T - \mathbf{1}_k z_T - \boldsymbol{\theta}_T - \rho\left(\mathbf{1}_k^\top \mathbf{y}_T - 1\right)\right)\right]_+,$$

$$z_T \leftarrow \left[z_T + \eta(\mathbf{1}_k^\top \mathbf{y}_T - 1)\right] \tag{23}$$

The only difference with the hard-clustering case, Eq.(14), is that the principal neurons are now leaky integrators with adaptive time constant, $\alpha T/n_{i,T}$.

B. Numerical experiments

We tested the utility of the representation learned by Algorithm 3 on the MNIST handwritten digits dataset by running a supervised classification algorithm on the vectors of attribution indices. The experiment was performed on a small-scale dataset (5,000 images were used in unsupervised training, last 1,000 images from unsupervised were used in supervised training, and another 1,000 images were used in testing) and on a large-scale dataset (60,000, 10,000, and 10,000 images, correspondingly). The results were averaged over 10 experiments with different random initializations.

After cluster centroids were learned by Algorithm 3 we classified the vectors of attribution indices using two versions of supervised learning that yielded similar results. The first was multiclass linear SVM, Figure 3. The second - directly connected each cluster to the most fitting target class (results not shown). Specifically, we defined T_d as the set of all t values where the true digit is d. Then for each cluster i we assigned the best matching digit label by taking $\arg\max_d \sum_{t\in T_d} y_{it}$. Then, we classified each image according to the digit which had the maximum sum of attribution indices.

We found that larger numbers of KCs led to better performance, Figure 3, top. Also, Figure 3, top, shows that nonzero values of the penalty parameter, α, resulted in better performance on small-scale (left) but not large-scale (right) datasets. Figure 3, bottom, shows that greater values of the penalty parameter, α, resulted in more active channels. This is consistent with viewing α as a regularization coefficient.

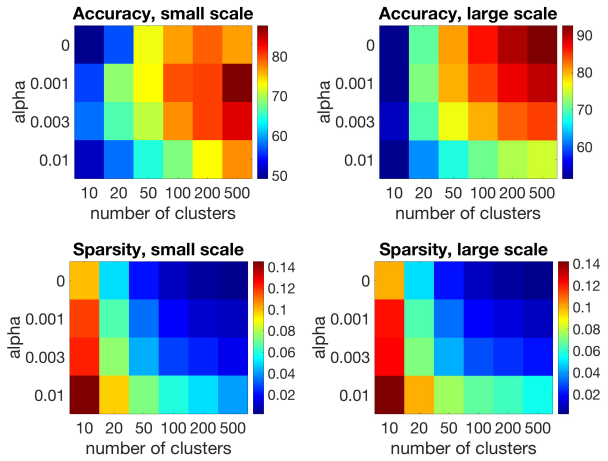

Fig. 3. Performance of the soft clustering Algorithm 3 on the MNIST hand-written digits data set for different α and number of clusters (special case, $\alpha = 0$, corresponds to the hard-clustering Algorithm 2). Top: Accuracy of supervised digit classification based on attribution indices. Bottom: Sparseness of the attribution indices (average fraction of non-zero indices per datum).

VI. Discussion

The proposed clustering neural network has a striking resemblance to the PN-KC-GI circuit in the insect olfactory system, Figure 1. We reproduced the general architecture including the divergence in the PN-KC circuit and, for the first time, derived the existence of a single inhibitory interneuron from a computational objective. The circuit dynamics in the model requires rectification in KCs, Eqs. (14,23), which occurs naturally in spiking neurons. The model does not require rectification in the inhibitory interneuron, (14,23), in agreement with the lack of spiking in giant interneurons [2].

According to our model, PN-KC synapses should obey Hebbian plasticity rules, Eqs. (11,19). KC-GI reciprocal synapses do not require plasticity, Eqs. (11,19). We predict interactions among KCs, Eqs. (14,23), that could be electrically mediated via shared extracellular space [48], [49].

In contrast to demixing models in which KCs signal the presence of constituent molecules in a complex odor, our model suggests that KCs represent olfactory objects which could be complex or simple odors. Our model predicts the existence of KCs selective to odor mixtures but not to the components alone. While preliminary evidence of such KC responses exists [21], they need to be investigated further.

Our clustering network has an advantage compared to existing random projection models [13], [17], [15], [18] in that it requires fewer KCs to represent olfactory stimuli. To span the PN activity vector space, random projections would require the number of KCs exponential in the number of dimensions. In contrast, our network is capable of reducing the required number of KCs by learning to represent only those parts of the PN activity space that are populated by data. In addition to minimizing volume [56] and energy [57] costs, reducing the number of KCs facilitates supervised learning.

In our model, nonnegativity of KC activity arises from the natural property of attribution indices (a datum cannot belong to a cluster with a negative attribution index). In contrast to the physical nonnegativity explanation of demixing models applicable only to neurons that code for molecular concentrations such as KCs, our interpretation can be applied, in addition, to most other spiking neurons encountered throughout the brain. Therefore, our interpretation is more general.

Appendix I
Neural clustering algorithm is asymptotically equivalent to classic k-means

Here, we demonstrate that Algorithm 2 asymptotically approaches classic k-means. To see this, let us focus on one

cluster i and denote $t_1, t_2, ...$ the sequence of all time indices such that $y_{i,t_j} = 1$. From Eq. (9) we have:

$$a_{i,t_j} = -2\mathbf{x}_{t_j}^\top \mathbf{w}_{i,t_j} + \theta_{i,t_j},\ N_{i,t_j} = j,\ \theta_{i,t_j} = -\frac{1}{j}\sum_{h<j} a_{i,t_h}, \tag{24}$$

where we substituted the value of Lagrange multiplier as derived in the main text. Also, we denoted by $\mathbf{w}_{i,t_j}$ the centroid of the i-th cluster, noting that the i-th row of matrix $\mathbf{W}_{t_j}$ is double that vector. From the recursive relation between a_{i,t_j} and θ_{i,t_j} we can infer:

$$\theta_{i,t_j} = \frac{2}{j(j-1)}\sum_{h<j} h\mathbf{x}_{t_h}^\top \mathbf{w}_{i,t_h}. \tag{25}$$

As the matrix, $\mathbf{W}$, stabilizes with time, we make an approximation and replace the running cluster centroids with the latest value, and then take the expectation:

$$E(\theta_{i,t_j}) \approx \frac{2}{j(j-1)}\mathbf{w}_{i,t_j}^\top E(\mathbf{x}_{t_h})\sum_{h<j} h = \mathbf{w}_{i,t_j}^\top E(\mathbf{x}_{t_h}) \approx ||\mathbf{w}_{i,t_j}||^2, \tag{26}$$

where for the last step we recall that vectors $\mathbf{x}_{t_h}$ are all and only those that constitute i-th cluster. Then

$$a_{i,t_j} \approx -2\mathbf{x}_{t_j}^\top \mathbf{w}_{i,t_j} + ||\mathbf{w}_{i,t_j}||^2 = ||\mathbf{x}_{t_j} - \mathbf{w}_{i,t_j}||^2 - ||\mathbf{x}_{t_j}||^2. \tag{27}$$

Therefore, asymptotically, the linear program solution of WTA in Algorithm 2 assigns each datum to the nearest cluster, just like k-means would, without requiring neurons to output Euclidean distances. Since the last term in (27) does not depend on i, the optimum in (6) only depends on distances between data points and, therefore, just like k-means, exhibits translational invariance (approximately), i.e. attribution indices depend only on the distance between data points, not on their absolute coordinates. In experiments, we observe the approximation to work well, so that θ_{i,t_j} quickly approaches and, then, follows $\mathbf{w}_{i,t_j}$ closely.

Appendix II
Fuzzy k-means produces dense output

To see why fuzzy k-means does not produce sparse output, write the Lagrangian for seeking $y_{i,t}$ under fixed $\mathbf{w}_i$:

$$\sum_i y_{i,t}^m ||\mathbf{x}_t - \mathbf{w}_i||^2 - \lambda(\sum_i y_{i,t} - 1) - \mu_i y_{i,t},$$
$$\text{s.t.} \quad \lambda \geq 0, \quad \mu_i \geq 0 \tag{28}$$

We write minus in front of λ because, since the objective is increasing, the values of $y_{i,t}$ need to be constrained only from below. Stationarity condition gives:

$$m y_{i,t}^{m-1} ||\mathbf{x}_t - \mathbf{w}_i||^2 - \lambda - \mu_i = 0. \tag{29}$$

If $y_{k,t} = 0$, then stationarity becomes $-\lambda = \mu_k$, and due to non-negativity of both variables: $\mu_k = \lambda = 0$. Then for all $i \neq k$ stationarity becomes $m y_{i,t}^{m-1} ||\mathbf{x}_t - \mathbf{w}_i||^2 = \mu_i$, which, using complementary slackness, leads to all $y_{i,t} = 0$ in violation of constraint unless for some j it happens that $||\mathbf{x}_t - \mathbf{w}_j|| = 0$. More generally, using some other than degree function $f(y_{i,t})$ in the objective, a necessary condition for sparsity is $f'(0) > 0$.

Appendix III
Relation of l_2-regularized k-means to an SDP for clustering and manifold learning

Here we show that the SDP used for clustering [53], [54] and manifold learning [55] is a convex relaxation of the l_2-regularized k-means cost function (16).

To see this connection, we rewrite the optimization problem (16) in terms of new variables. Plugging in for optimal $\mathbf{w}_i = \sum_t y_{i,t}\mathbf{x}_t / \sum_t y_{i,t}$, and defining new variables

$$\tilde{y}_{i,t} := \frac{y_{i,t}}{\sqrt{\sum_t y_{i,t}}},\ Q_{tt'} := \sum_i \tilde{y}_{i,t}\tilde{y}_{i,t'},\ D_{tt'} := \sum_i x_{i,t}x_{i,t'}. \tag{30}$$

the cost (16) becomes $-\mathrm{Tr}\,(\mathbf{DQ}) + \alpha T\,\mathrm{Tr}\,\mathbf{Q}$. By construction, $\mathbf{Q1} = \mathbf{1}$ and $\mathbf{Q} \geq 0$. Therefore, we can relax the objective (16) to an SDP, where optimization is with respect to the positive semidefinite matrix $\mathbf{Q}$:

$$\min_{\mathbf{Q}} -\mathrm{Tr}\,(\mathbf{DQ}) + \alpha T\,\mathrm{Tr}\,\mathbf{Q},$$
$$\text{s.t.}\ \mathbf{Q} \succeq 0,\ \ Q \geq 0,\ \ \mathbf{Q1} = \mathbf{1}. \tag{31}$$

The SDP of [53], [54], [55] is

$$\min_{\mathbf{Q}} -\mathrm{Tr}\,(\mathbf{DQ}),$$
$$\text{s.t.}\ \mathbf{Q} \succeq 0,\ \ Q \geq 0,\ \ \mathbf{Q1} = \mathbf{1},\ \ \mathrm{Tr}\,\mathbf{Q} = k, \tag{32}$$

from which (31) can be recovered by replacing the last constraint by a penalty term in the cost.

Acknowledgements

We thank Anirvan Sengupta, Mariano Tepper and Evgeniy Bauman for helpful discussions.

References

[1] N. Y. Masse, G. C. Turner, and G. S. Jefferis, "Olfactory information processing in drosophila," *Current Biology*, vol. 19, no. 16, pp. R700–R713, 2009.

[2] M. Papadopoulou, S. Cassenaer, T. Nowotny, and G. Laurent, "Normalization for sparse encoding of odors by a wide-field interneuron," *Science*, vol. 332, no. 6030, pp. 721–725, 2011.

[3] B. Leitch, G. Laurent *et al.*, "Gabaergic synapses in the antennal lobe and mushroom body of the locust olfactory system," *J Comp Neurol*, vol. 372, no. 4, pp. 487–514, 1996.

[4] X. Liu and R. L. Davis, "The gabaergic anterior paired lateral neuron suppresses and is suppressed by olfactory learning," *Nat Neurosci*, vol. 12, no. 1, pp. 53–59, 2009.

[5] A. C. Lin, A. M. Bygrave, A. De Calignon, T. Lee, and G. Miesenböck, "Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination," *Nat Neurosci*, vol. 17(4), p. 559, 2014.

[6] J. Perez-Orive, O. Mazor, G. C. Turner, S. Cassenaer, R. I. Wilson, and G. Laurent, "Oscillations and sparsening of odor representations in the mushroom body," *Science*, vol. 297, no. 5580, pp. 359–365, 2002.

[7] G. C. Turner, M. Bazhenov, and G. Laurent, "Olfactory representations by drosophila mushroom body neurons," *J Neurophysiol*, vol. 99, no. 2, pp. 734–746, 2008.

[8] K. S. Honegger, R. A. Campbell, and G. C. Turner, "Cellular-resolution population imaging reveals robust sparse coding in the drosophila mushroom body," *Journal of Neuroscience*, vol. 31, no. 33, pp. 11 772–11 785, 2011.

[9] Y. Aso, D. Sitaraman, T. Ichinose, K. R. Kaun, K. Vogt, G. Belliart-Guérin, P.-Y. Plaçais, A. A. Robie, N. Yamagata *et al.*, "Mushroom body output neurons encode valence and guide memory-based action selection in drosophila," *Elife*, vol. 3, p. e04580, 2014.

[10] H. Demmer and P. Kloppenburg, "Intrinsic membrane properties and inhibitory synaptic input of kenyon cells as mechanisms for sparse coding?" *J Neurophysiol*, vol. 102, no. 3, pp. 1538–1550, 2009.

[11] F. Farkhooi, A. Froese, E. Muller, R. Menzel, and M. P. Nawrot, "Cellular adaptation facilitates sparse and reliable coding in sensory pathways," *PLoS Comput Biol*, vol. 9, no. 10, p. e1003251, 2013.

[12] T. Kee, P. Sanda, N. Gupta, M. Stopfer, and M. Bazhenov, "Feed-forward versus feedback inhibition in a basic olfactory circuit," *PLoS Comput Biol*, vol. 11, no. 10, p. e1004531, 2015.

[13] S. Luo, R. Axel, and L. Abbott, "Generating sparse and selective third-order responses in the olfactory system of the fly," *PNAS*, vol. 107, no. 23, pp. 10 713–10 718, 2010.

[14] J. Wessnitzer, J. M. Young, J. D. Armstrong, and B. Webb, "A model of non-elemental olfactory learning in drosophila," *J Comput Neurosci*, vol. 32, no. 2, pp. 197–212, 2012.

[15] A. Litwin-Kumar, K. D. Harris, R. Axel, H. Sompolinsky, and L. Abbott, "Optimal degrees of synaptic connectivity," *Neuron*, vol. 93, no. 5, pp. 1153–1164, 2017.

[16] F. Peng and L. Chittka, "A simple computational model of the bee mushroom body can explain seemingly complex forms of olfactory learning and memory," *Curr Biol*, 2016.

[17] K. Krishnamurthy, A. M. Hermundstad, T. Mora, A. M. Walczak, and V. Balasubramanian, "Disorder and the neural representation of complex odors: smelling in the real world," *bioRxiv*, p. 160382, 2017.

[18] S. Dasgupta, C. F. Stevens, and S. Navlakha, "A neural algorithm for a fundamental computing problem," *Science*, vol. 358, no. 6364, pp. 793–796, 2017.

[19] C. Pehlevan, T. Hu, and D. Chklovskii, "A hebbian/anti-hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data," *Neural Comput*, vol. 27, pp. 1461–1495, 2015.

[20] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput*, vol. 7, no. 6, pp. 1129–1159, 1995.

[21] K. Shen, S. Tootoonian, and G. Laurent, "Encoding of mixtures in a simple olfactory system," *Neuron*, vol. 80, no. 5, pp. 1246–1262, 2013.

[22] A. Grabska-Barwińska, S. Barthelmé, J. Beck, Z. F. Mainen, A. Pouget, and P. E. Latham, "A probabilistic approach to demixing odors," *Nat Neurosci*, 2016.

[23] S. Tootoonian and M. Lengyel, "A dual algorithm for olfactory computation in the locust brain," in *NIPS*, 2014, pp. 2276–2284.

[24] Y. Zhang and T. O. Sharpee, "A robust feedforward model of the olfactory system," *PLoS Comput Biol*, vol. 12, no. 4, p. e1004850, 2016.

[25] C. Stevens, "What the fly's nose tells the fly's brain," *PNAS*, vol. 112, no. 30, pp. 9460–9465, 2015.

[26] P. Földiak, "Forming sparse representations by local anti-hebbian learning," *Biol Cyb*, vol. 64, no. 2, pp. 165–170, 1990.

[27] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[28] C. Pehlevan and D. Chklovskii, "A hebbian/anti-hebbian network derived from online non-negative matrix factorization can cluster and discover sparse features," in *ACSSC*. IEEE, 2014, pp. 769–775.

[29] T. Hu, C. Pehlevan, and D. B. Chklovskii, "A hebbian/anti-hebbian network for online sparse dictionary learning derived from symmetric matrix factorization," in *ACSSC*. IEEE, 2014, pp. 613–619.

[30] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision Res*, vol. 37, no. 23, pp. 3311–3325, 1997.

[31] J. Zylberberg, J. T. Murphy, and M. R. DeWeese, "A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of v1..." *PLoS Comp Biol*, vol. 7, no. 10, p. e1002250, 2011.

[32] A. Koulakov and D. Rinberg, "Sparse incomplete representations: a potential role of olfactory granule cells," *Neuron*, vol. 72, no. 1, pp. 124–136, 2011.

[33] S. Druckmann, T. Hu, and D. Chklovskii, "A mechanistic model of early sensory processing based on subtracting sparse representations," in *NIPS*, 2012, pp. 1979–1987.

[34] M. Zhu and C. Rozell, "Modeling inhibitory interneurons in efficient sensory coding models," *PLoS Comput Biol*, vol. 11, no. 7, p. e1004353, 2015.

[35] C. Pehlevan and D. Chklovskii, "A normative theory of adaptive dimensionality reduction in neural networks," in *NIPS*, 2015, pp. 2260–2268.

[36] D. Kepple, H. Giaffar, D. Rinberg, and A. Koulakov, "Deconstructing odorant identity via primacy in dual networks," *arXiv preprint arXiv:1609.02202*, 2016.

[37] C. Masson, H. Mustaparta *et al.*, "Chemical information processing in the olfactory system of insects." *Physiol Rev*, vol. 70, no. 1, pp. 199–245, 1990.

[38] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc of the 5th Berkeley symposium on mathematical statistics and probability*, 1967, pp. 281–297.

[39] S. Lloyd, "Least squares quantization in pcm," *IEEE Trans Inf Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[40] K.-L. Du, "Clustering: A neural network approach," *Neural networks*, vol. 23, no. 1, pp. 89–107, 2010.

[41] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol Cyb*, vol. 43, no. 1, pp. 59–69, 1982.

[42] T. Martinetz and K. Schulten, "A "neural-gas" network learns topologies," 1991.

[43] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput Vision Graph*, vol. 37, no. 1, pp. 54–115, 1987.

[44] C. Keck, C. Savin, and J. Lücke, "Feedforward inhibition and synaptic scaling–two sides of the same coin?" *PLoS computational biology*, vol. 8, no. 3, p. e1002432, 2012.

[45] A. Yuille and D. Geiger, "Winner-take-all networks," *The handbook of brain theory and neural networks*, pp. 1228–1231, 2003.

[46] H. Ritter and K. Schulten, "Kohonen's self-organizing maps: exploring their computational capabilities," in *Proc IEEE Int Conf Neur Net*, vol. 1, 1988, pp. 109–116.

[47] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[48] M. Weckström and S. Laughlin, "Extracellular potentials modify the transfer of information at photoreceptor output synapses in the blowfly compound eye," *Journal of Neuroscience*, vol. 30, no. 28, pp. 9557–9566, 2010.

[49] W. B. Thoreson and S. C. Mangel, "Lateral interactions in the outer retina," *Progress in retinal and eye research*, vol. 31, no. 5, pp. 407–441, 2012.

[50] R. A. FISHER, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936. [Online]. Available: http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x

[51] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J Am Stat Assoc*, vol. 336, p. 846–850, 1971.

[52] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Comput Geosci*, vol. 10(2-3), pp. 191–203, 1984.

[53] B. Kulis, A. C. Surendran, and J. C. Platt, "Fast low-rank semidefinite programming for embedding and clustering," in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 235–242.

[54] P. Awasthi, A. S. Bandeira, M. Charikar, R. Krishnaswamy, S. Villar, and R. Ward, "Relax, no need to round: Integrality of clustering formulations," in *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*. ACM, 2015, pp. 191–200.

[55] M. Tepper, A. M. Sengupta, and D. Chklovskii, "The surprising secret identity of the semidefinite relaxation of k-means: manifold learning," *arXiv preprint arXiv:1706.06028*, 2017.

[56] M. Rivera-Alba, H. Peng, G. G. de Polavieja, and D. B. Chklovskii, "Wiring economy can account for cell body placement across species and brain areas," *Current Biology*, vol. 24(3), pp. R109–R110, 2014.

[57] S. B. Laughlin, "Energy as a constraint on the coding and processing of sensory information," *Current opinion in neurobiology*, vol. 11, no. 4, pp. 475–480, 2001.