

Machine Intelligence 2

4.3 Self-Organising Maps

Prof. Dr. Klaus Obermayer

Fachgebiet Neuronale Informationsverarbeitung (NI)

SS 2017

Self-Organising Maps

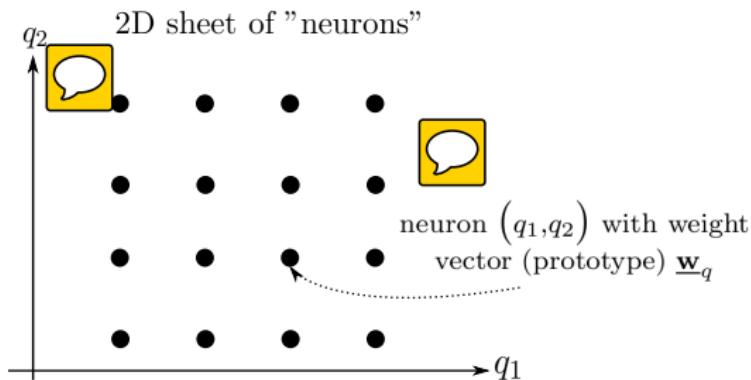
Self-organising maps (SOM)



- clustering & local embedding of vectorial data: $\underline{x}^{(\alpha)}, \alpha = 1, \dots, p$
- clustering of data based on similarity: squared Euclidean distance
- low-dimensional and **neighborhood preserving** representation for visualization

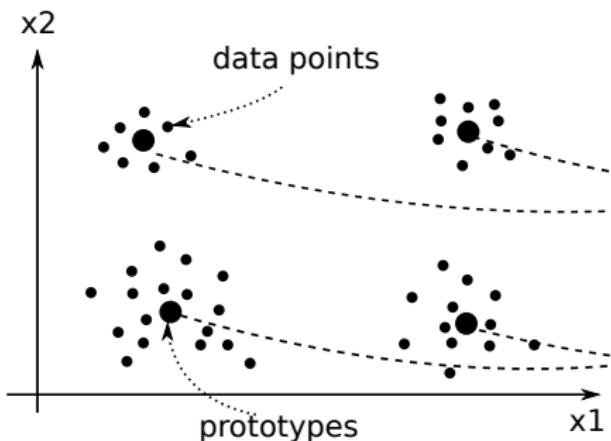
Model class

$$m_{\underline{\mathbf{q}}}^{(\alpha)} = \begin{cases} 1, & \text{if } \underline{\mathbf{q}} = \operatorname{argmin}_{\underline{\mathbf{r}}} |\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_{\underline{\mathbf{r}}}| \\ 0, & \text{else} \end{cases}$$

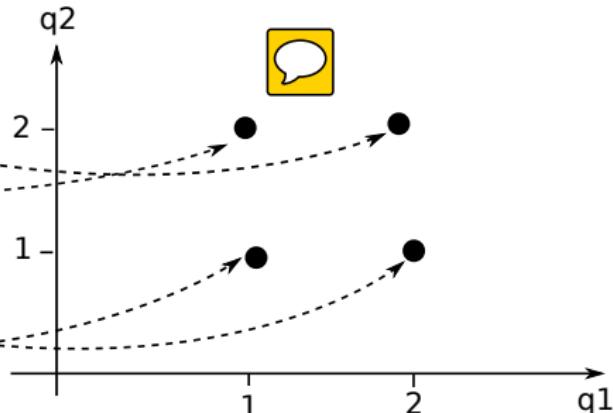


"Binary neurons" with index $\underline{\mathbf{q}} = (q_1, q_2)^T$ are assigned to prototypes $\underline{\mathbf{w}}_{\underline{\mathbf{q}}}$.

"Topographic" maps



Visualisation Space



"map" of data space for $\left\{ \begin{array}{l} \text{visualization} \\ \text{dimension reduction} \\ \text{preprocessing for prediction} \end{array} \right.$

Algorithm 1: On-line learning for SOM "almost kmeans clustering"**Initialization:**

- choose no. M of partitions ("neurons")
- choose annealing schedule for ε and σ
- initialize prototypes, $\underline{\mathbf{w}}_q = 1/p \sum_{\alpha} \underline{\mathbf{x}}^{(\alpha)} + \underline{\eta}$, $\underline{\eta}$ small noise vector

beginchoose a data point $\underline{\mathbf{x}}^{(\alpha)}$

determine the closest prototype:

$$\underline{\mathbf{p}} = \operatorname{argmin}_{\underline{\mathbf{r}}} |\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_{\underline{\mathbf{r}}}|$$

change all prototypes according $\underline{\mathbf{w}}_q$ to:

$$\Delta \underline{\mathbf{w}}_q = \varepsilon h_{\underline{\mathbf{q}}\underline{\mathbf{p}}} (\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_q) \quad \text{for all } \underline{\mathbf{q}}$$

end

Neighborhood function $h_{\underline{q}\underline{p}}$

$$\Delta \underline{\mathbf{w}}_{\underline{q}} = \varepsilon h_{\underline{q}\underline{p}} (\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_{\underline{q}}^{\text{old}}) \quad \text{for all } q$$

- $h_{\underline{q}\underline{p}}$ enforces similar learning steps for neighboring neurons
- common choice:

$$h_{\underline{q}\underline{p}} = \exp \left\{ -\frac{(\underline{\mathbf{q}} - \underline{\mathbf{p}})^2}{2\sigma^2} \right\}$$


- $\sigma = 0$: standard on-line K-means only (update $\underline{\mathbf{q}} = \underline{\mathbf{p}}$)

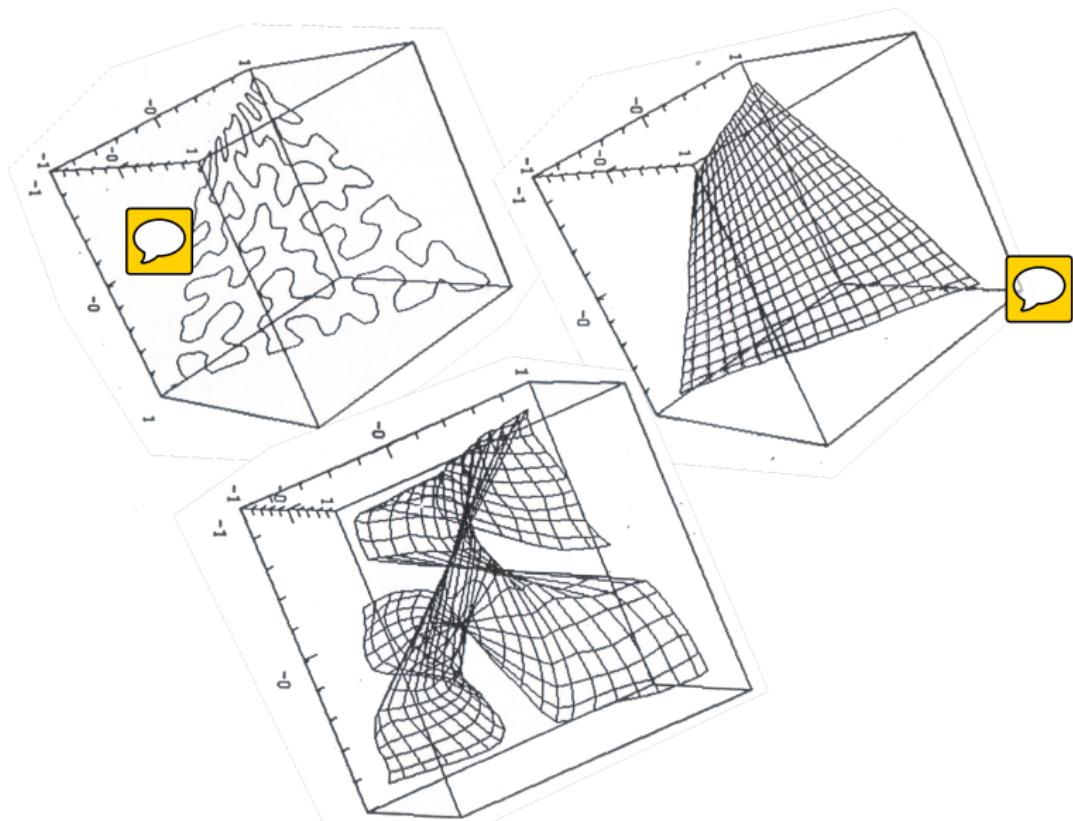
Neighborhood function $h_{\underline{\mathbf{q}}\underline{\mathbf{p}}}$


$$h_{\underline{\mathbf{q}}\underline{\mathbf{p}}} = \exp \left\{ -\frac{(\underline{\mathbf{q}} - \underline{\mathbf{p}})^2}{2\sigma^2} \right\}$$

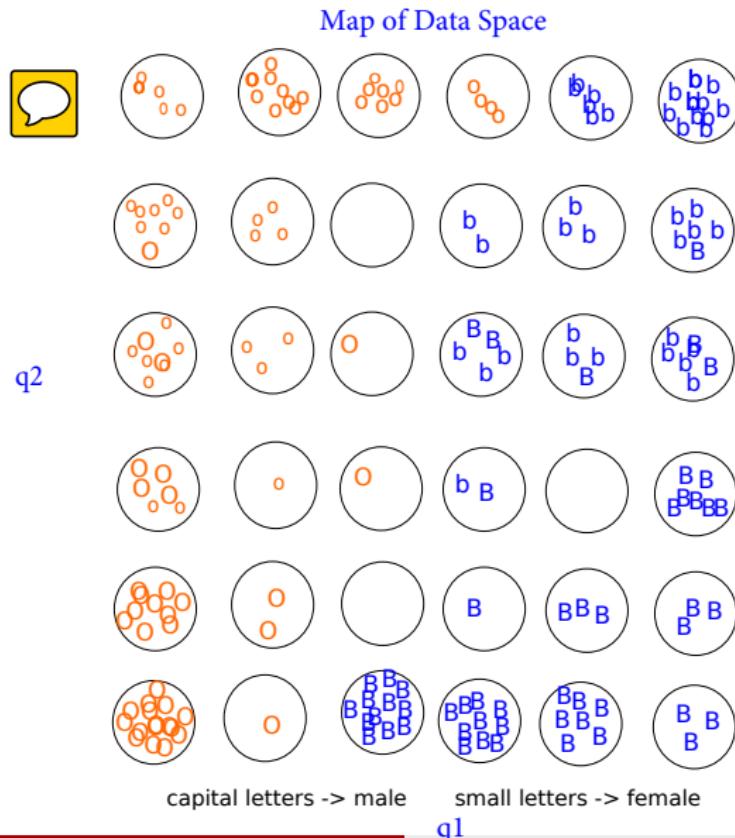
σ annealing

- start with large σ (\leadsto neighborhood function convex over its support)
- decrease linearly or exponentially (but "slow") during learning.
- solution depends on final value of σ
- $\sigma = 0$: minimum of the K-means clustering cost function but "*neighborhood preserving*"
- σ small : better representation capabilities at the expense of a non-optimal clustering cost

2d manifold example



Leptograpsus example

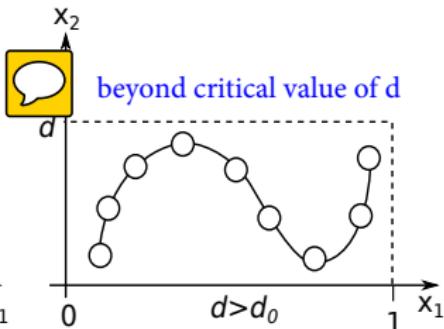
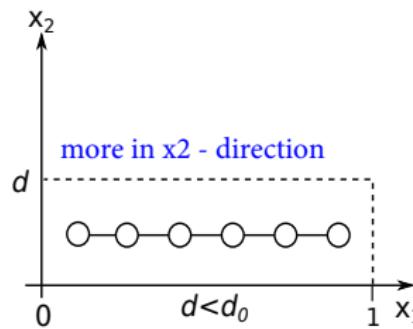
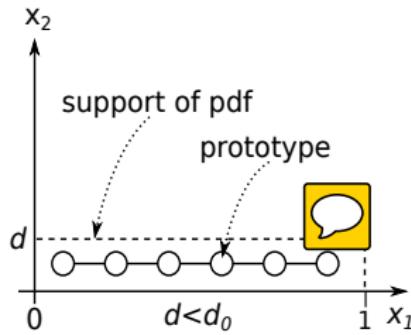


attributes

- X_1 : frontal lobe size (mm)
- X_2 : rear width (mm)
- X_3 : carapace length (mm)
- X_4 : carapace width (mm)
- X_5 : body depth (mm)

Dimension reducing mappings

what if dimensions dont match?



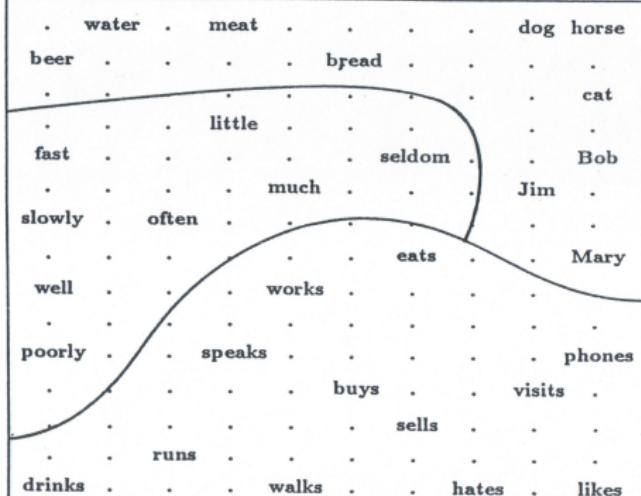
Semantic maps

Abstract Data: Verbal Statements

Bob/Jim/Mary	1
horse/dog/cat	2
beer/water	3
meat/bread	4
runs/walks	5
works/speaks	6
visits/phones	7
buys/sells	8
likes/hates	9
drinks/eats	10
much/little	11
fast/slowly	12
often/seldom	13
well/poorly	14

Sentence Patterns:		
1-5-12	1-9-2	2-5-14
1-5-13	1-9-3	2-9-1
1-5-14	1-9-4	2-9-2
1-6-12	1-10-3	2-9-3
1-6-13	1-11-4	2-9-4
1-6-14	1-10-12	2-10-3
1-6-15	1-10-13	2-10-12
1-7-14	1-10-14	2-10-13
1-8-12	1-11-12	2-10-14
1-8-2	1-11-13	1-11-4
1-8-3	1-11-14	1-11-12
1-8-4	2-5-12	2-11-13
1-9-1	2-5-13	2-11-14

Mary likes meat
 Jim speaks well
 Mary likes Jim
 Jim eats often
 Mary buys meat
 dog drinks fast
 horse hates meat
 Jim eats seldom
 Bob buys meat
 cat walks slowly
 Jim eats bread
 cat hates Jim
 Bob sells beer
 (etc.)

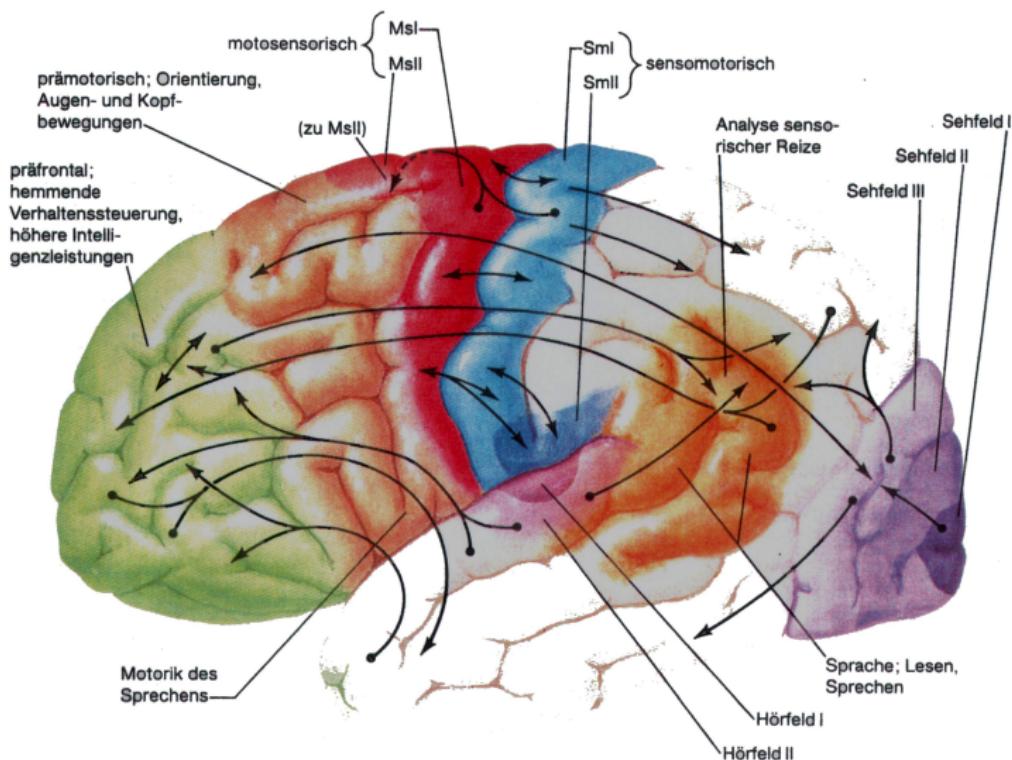


$$\begin{bmatrix} \vec{v}_s \\ \vec{v}_c \end{bmatrix} = \begin{bmatrix} \vec{0} \\ \vec{0} \end{bmatrix} + \begin{bmatrix} \vec{v}_s \\ \vec{v}_c \end{bmatrix}$$



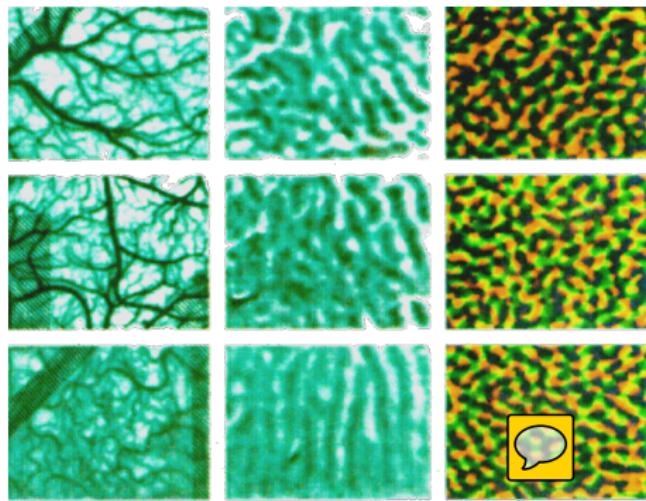
Ritter & Kohnen 1989

Orientation & ocular dominance maps



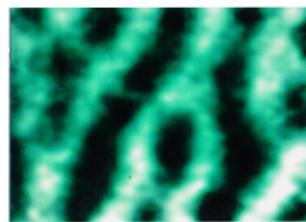
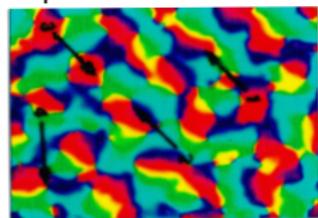
Orientation & ocular dominance maps

maps of feature vectors

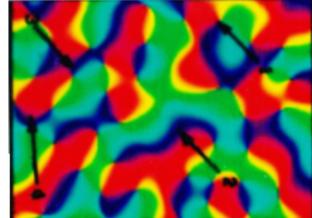


Orientation & ocular dominance maps

Experimental data:



mapping results:



5d feature vector

position: $x, y \in [0, d]$

orientation preference & specificity:

$$r \cos(2\phi), \phi \in [0, \pi[$$

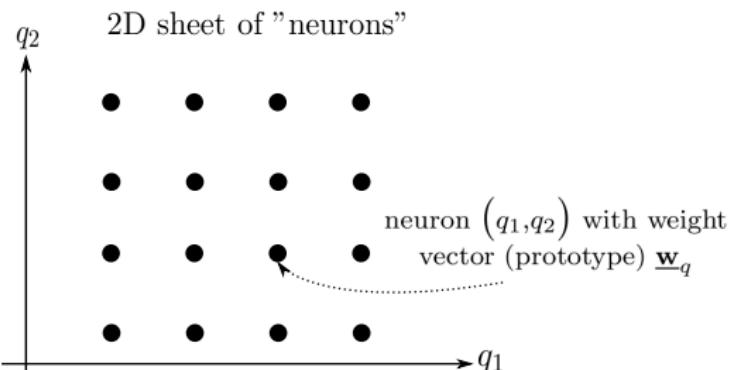
$$r \sin(2\phi), \phi \in [0, \pi[$$

ocular dominance: $z \in \{-1, 1\}$

Self-organizing maps for pairwise data

- distance matrix $\underline{\mathbf{D}}$ specifying the distances or 'dissimilarities' $d_{\alpha\alpha'}$ between p "objects" $\alpha = 1, \dots, p$
- set of M clusters (partitions) $\underline{\mathbf{q}}$ with a geometrical structure (e.g. 1-d line or 2-d grid).
- binary assignment variables (normalized):

$$m_{\underline{\mathbf{q}}}^{(\alpha)} = \begin{cases} 1, & \text{if object } \alpha \text{ belongs to cluster } \underline{\mathbf{q}} \\ 0, & \text{else} \end{cases}$$



	1	2	3	\dots	p
1	0	1.7	0.99	\dots	3.0
2	1.7	0	0.3	\dots	0.1
3	0.9	0.3	0	\dots	0.2
\vdots	\vdots	\vdots	\ddots	\ddots	\vdots
p	3.0	0.1	0.2	\dots	0

relational representation
"pairwise data"

Cost function & model selection

$$E\left[\{m_{\underline{\mathbf{q}}}^{(\alpha)}\}\right] = \frac{1}{p} \sum_{\underline{\mathbf{r}}} \frac{\sum_{\alpha, \alpha'} \left(\sum_{\underline{\mathbf{q}}} h_{\underline{\mathbf{r}}\underline{\mathbf{q}}} m_{\underline{\mathbf{q}}}^{(\alpha)} \right) \left(\sum_{\underline{\mathbf{q}}} h_{\underline{\mathbf{r}}\underline{\mathbf{q}}} m_{\underline{\mathbf{q}}}^{(\alpha')} \right) d_{\alpha\alpha'}}{\sum_{\alpha} \left(\sum_{\underline{\mathbf{q}}} h_{\underline{\mathbf{r}}\underline{\mathbf{q}}} m_{\underline{\mathbf{q}}}^{(\alpha)} \right)} \stackrel{!}{=} \min$$

- $m_q^{(\alpha)} \rightarrow \sum_{\underline{\mathbf{q}}} h_{\underline{\mathbf{r}}\underline{\mathbf{q}}} m_{\underline{\mathbf{q}}}^{(\alpha)}$
- "neighboring" clusters (w.r.t. $h_{\underline{\mathbf{r}}\underline{\mathbf{q}}}$) contribute to the total average distance
- "neighborhood preserving maps" induce lower cost

Algorithm 2: SOM for pairwise data (mean-field approximation)**Initialization:**

- choose no. M of partitions, initial (β_0) and final (β_f) noise parameters, annealing factor η , width σ of neighborhood function $h_{\underline{\mathbf{sq}}}$, tolerance θ
- initialize mean-fields $e_{\underline{\mathbf{q}}}^{(\alpha)}$: random numbers $\in [0, 1]$

$$\beta \leftarrow \beta_0$$

while $\beta < \beta_f$ **do** annealing

repeat EM

$$\text{compute assignment probabilities } \langle m_{\underline{\mathbf{q}}}^{(\alpha)} \rangle_Q = \frac{\exp \left\{ -\beta \left(e_{\underline{\mathbf{q}}}^{(\alpha)} \right)_{\text{old}} \right\}}{\sum_{\underline{\mathbf{r}}} \exp \left\{ -\beta \left(e_{\underline{\mathbf{r}}}^{(\alpha)} \right)_{\text{old}} \right\}} \quad \forall \underline{\mathbf{q}}, \alpha$$

compute new mean-fields

$$(e_{\underline{\mathbf{q}}}^{(\alpha)})_{\text{new}} = \frac{1}{p} \sum_{\underline{\mathbf{s}}} h_{\underline{\mathbf{sq}}} \left[\frac{1}{\sum_{\gamma} \left(\sum_{\underline{\mathbf{r}}} h_{\underline{\mathbf{sr}}} \langle m_{\underline{\mathbf{r}}}^{(\gamma)} \rangle_Q \right)} \sum_{\delta} \left(\sum_{\underline{\mathbf{r}}} h_{\underline{\mathbf{sr}}} \langle m_{\underline{\mathbf{r}}}^{(\delta)} \rangle_Q \right) \right. \\ \left. \cdot \left\{ d_{\delta\alpha} - \frac{1}{2} \frac{1}{\sum_{\gamma} \left(\sum_{\underline{\mathbf{r}}} h_{\underline{\mathbf{sr}}} \langle m_{\underline{\mathbf{r}}}^{(\gamma)} \rangle_Q \right)} \sum_{\varepsilon} \left(\sum_{\underline{\mathbf{r}}} h_{\underline{\mathbf{sr}}} \langle m_{\underline{\mathbf{r}}}^{(\varepsilon)} \rangle_Q \right) d_{\varepsilon\delta} \right\} \right] \quad \forall \underline{\mathbf{q}}, \alpha$$

until $| (e_{\underline{\mathbf{q}}}^{(\alpha)})_{\text{new}} - (e_{\underline{\mathbf{q}}}^{(\alpha)})_{\text{old}} | < \theta \quad \forall \underline{\mathbf{q}}, \alpha$

$$\beta \leftarrow \eta \beta$$

end

Comments

- replacing $h_{\underline{s}p}$ by $\delta_{\underline{s}p}$ recovers standard pairwise clustering
- "Kohonen-approximation": replace $h_{\underline{s}p}$ by $\delta_{\underline{s}p}$ for the neighborhood function (only) at \circledast in algorithm 2
 - reduction of computational cost
 - visualization properties remain
 - original algorithm suggested by T. Kohonen is recovered for squared Euclidean distances $d_{\alpha\alpha'}$ and $\beta \rightarrow \infty$
- no need for σ annealing

Toy example: noisy spiral

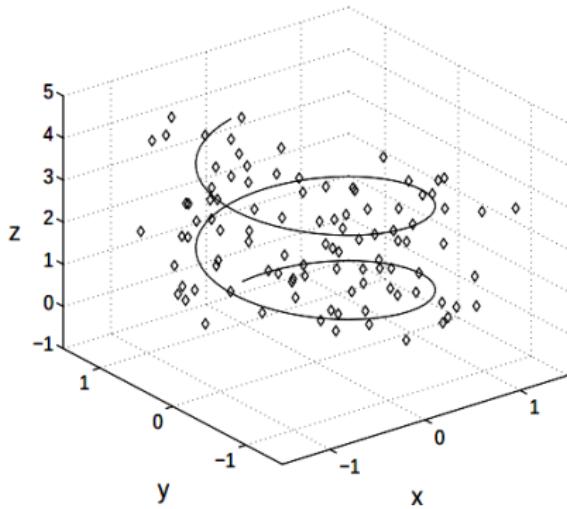
1000 data points:

$$x = \sin \theta + \eta_x$$



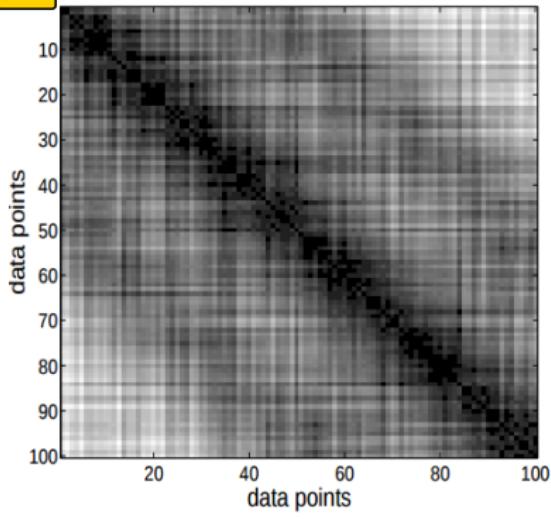
$$y = \cos \theta + \eta_y$$

$$z = \frac{\theta}{\pi} + \eta_z$$



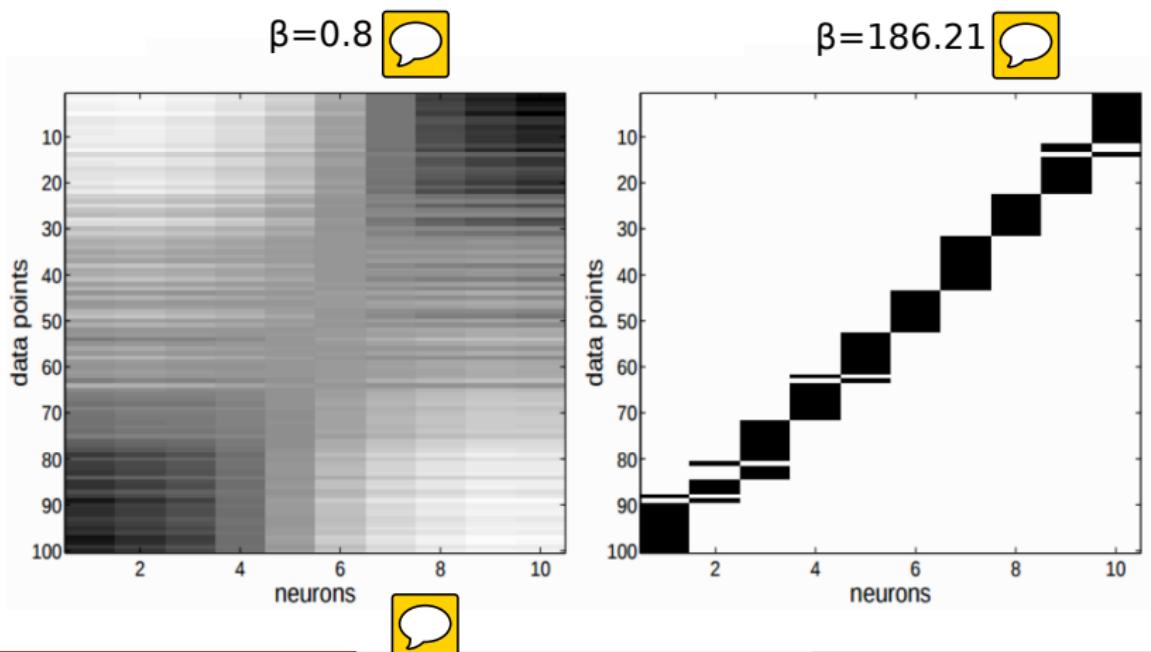
$$\theta \in [0, 4\pi]; \quad \eta \sim \mathcal{N}_{(0, 0.3)}$$

Euclidean squared distance



Toy example: noisy spiral

- 10 neurons in 1D
- Gaussian neighborhood function, $\sigma = 0.5$



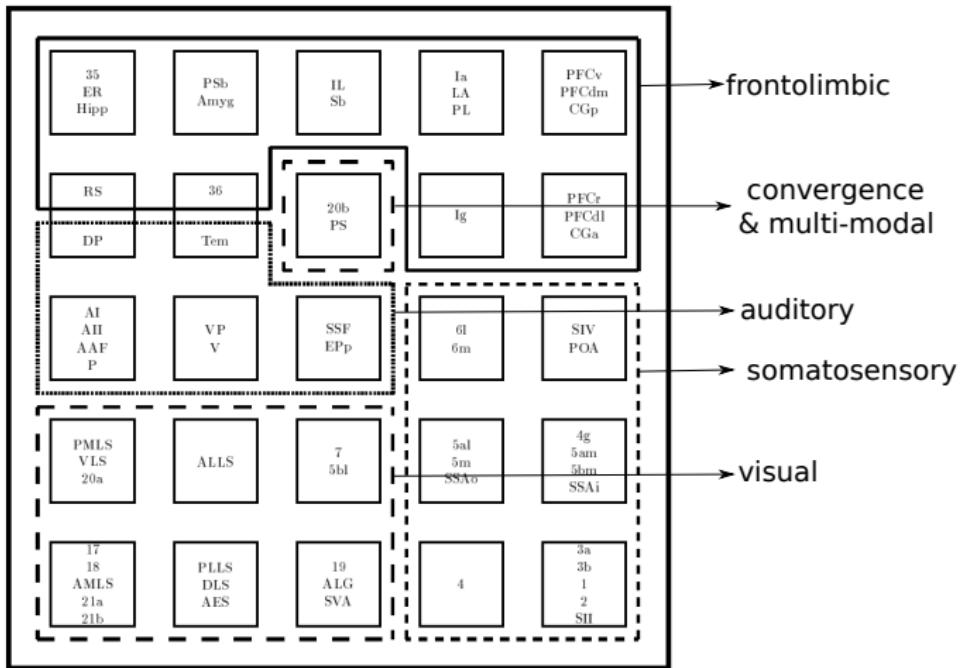
Mapping cat's cerebral cortex

	DFFERENT																								
AREA	17	18	19	PMLS	PLLS	ALLS	VLS	DLS	21a	21b	20a	20b	ALG	7	AES	SVA	PS	All	AAF	DP	VP	SIV	Hipp	Em	
17	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	17
18	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	18
19	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	19
PMLS	3	3	2	1	3	2	1	3	0	3	1	3	0	3	1	3	0	3	2	1	3	0	3	1	PMLS
PLLS	1	1	1	2	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	PLLS
ALLS	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ALLS
VLS	2	1	2	3	1	0	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	VLS
DLS	0	0	2	3	0	2	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DLS
21a	2	2	2	1	1	0	0	0	1	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	21a
21b	2	2	2	1	1	0	0	0	1	2	2	2	1	0	0	0	0	0	0	0	0	0	0	0	21b
20a	3	2	3	2	0	2	2	2	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	20a
20b	0	0	0	1	0	0	0	0	0	3	0	2	2	2	0	0	0	0	0	0	0	0	0	0	20b
ALG	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ALG
7	0	0	0	1	2	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
AES	0	0	1	3	1	2	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	AES
SVA	0	0	1	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SVA
PS	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PS
All	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	All
AllF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	AllF
DP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DP
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	P
VP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	VP
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	V
SIF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SIF
EIf	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EIf
Tem	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Tem
3a	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3a
3b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3b
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
SII	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SII
SIV	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SIV
4g	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4g
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
6l	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6l
6m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6m
POA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	POA
Sam	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Sam
Sal	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Sal
Sm	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Sm
Sst	0	1	2	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Sst
Sm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Sm
SSAo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSAo
SSAl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSAl
PPCg	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PPCg
PPCgl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PPCgl
PPCcv	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PPCcv
PPCdm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PPCdm
In	0	0	0	1	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	In
Ig	0	0	1	2	2	0	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ig
CGs	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CGs
LA	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	LA
RS	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS
PL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PL
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35
36	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	36
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37
PSb	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PSb
ER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ER
Hipp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Hipp
Amyg	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Amyg
	17	18	19	PMLS	PLLS	ALLS	VLS	DLS	21a	21b	20a	20b	ALG	7	AES	SVA	PS	All	AAF	DP	VP	SIV	Hipp	Em	

Mapping cat's cerebral cortex



"Kohonen"-map:
5x5 neurons,
Gaussian
neighborhood
 $\delta_n = 0.4$



Squared Euclidean distances

$$d_{\alpha\alpha'} = \frac{1}{2} (\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{x}}^{(\alpha')})^2$$


cost function:

$$E[\{m_{\underline{\mathbf{q}}}^{(\alpha)}\}] = \frac{1}{p} \sum_{\underline{\mathbf{q}}, \alpha} \left(\sum_{\underline{\mathbf{p}}} h_{\underline{\mathbf{q}}\underline{\mathbf{p}}} m_{\underline{\mathbf{p}}}^{(\alpha)} \right) (\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_q)^2 = \frac{1}{p} \sum_{\underline{\mathbf{p}}^\alpha} m_{\underline{\mathbf{p}}}^{(\alpha)} \sum_{\underline{\mathbf{q}}} h_{\underline{\mathbf{q}}\underline{\mathbf{p}}} (\underline{\mathbf{x}}^{(\alpha)} - \underline{\mathbf{w}}_{\underline{\mathbf{q}}})^2$$

$$\underline{\mathbf{w}}_{\underline{\mathbf{q}}} = \frac{\sum_{\alpha'} \left(\sum_{\underline{\mathbf{p}}} h_{\underline{\mathbf{q}}\underline{\mathbf{p}}} m_{\underline{\mathbf{p}}}^{(\alpha')} \right) \underline{\mathbf{x}}^{(\alpha')}}{\sum_{\alpha'} \left(\sum_{\underline{\mathbf{p}}} h_{\underline{\mathbf{q}}\underline{\mathbf{p}}} m_{\underline{\mathbf{p}}}^{(\alpha')} \right)}$$

$\underline{\mathbf{w}}_{\underline{\mathbf{q}}}$: center of mass of all data which belongs to cluster weighted by the neighborhood function $h_{\underline{\mathbf{q}}\underline{\mathbf{p}}}$

On-line minimization of E^T

Algorithm 3: Online learning for SOM with Euclidean distances

Initialization of prototypes

Select learning step ε

begin

choose a data point $\underline{x}^{(\alpha)}$

assign data point to the prototype with minimum assignment cost

$$\underline{p} = \operatorname{argmin}_{\underline{r}} \sum_{\substack{\underline{q} \\ (*)}} h_{\underline{r}\underline{q}} (\underline{x}^{(\alpha)} - \underline{w}_{\underline{q}}^{\text{old}})^2$$

change all prototypes according to

$$\Delta \underline{w}_{\underline{q}} = \varepsilon h_{\underline{p}\underline{q}} (\underline{x}^{(\alpha)} - \underline{w}_{\underline{q}}) \text{ for all } \underline{q}$$

end
