Vilniaus universitetas
Matematikos ir informatikos fakultetas
Informatikos katedra

Saviorganizuojantys neuroniniai tinklai (žemėlapiai)

prof. dr. Olga Kurasova
Olga.Kurasova@mii.vu.lt

2018

Saviorganizuojantys neuroniniai tinklai (1)

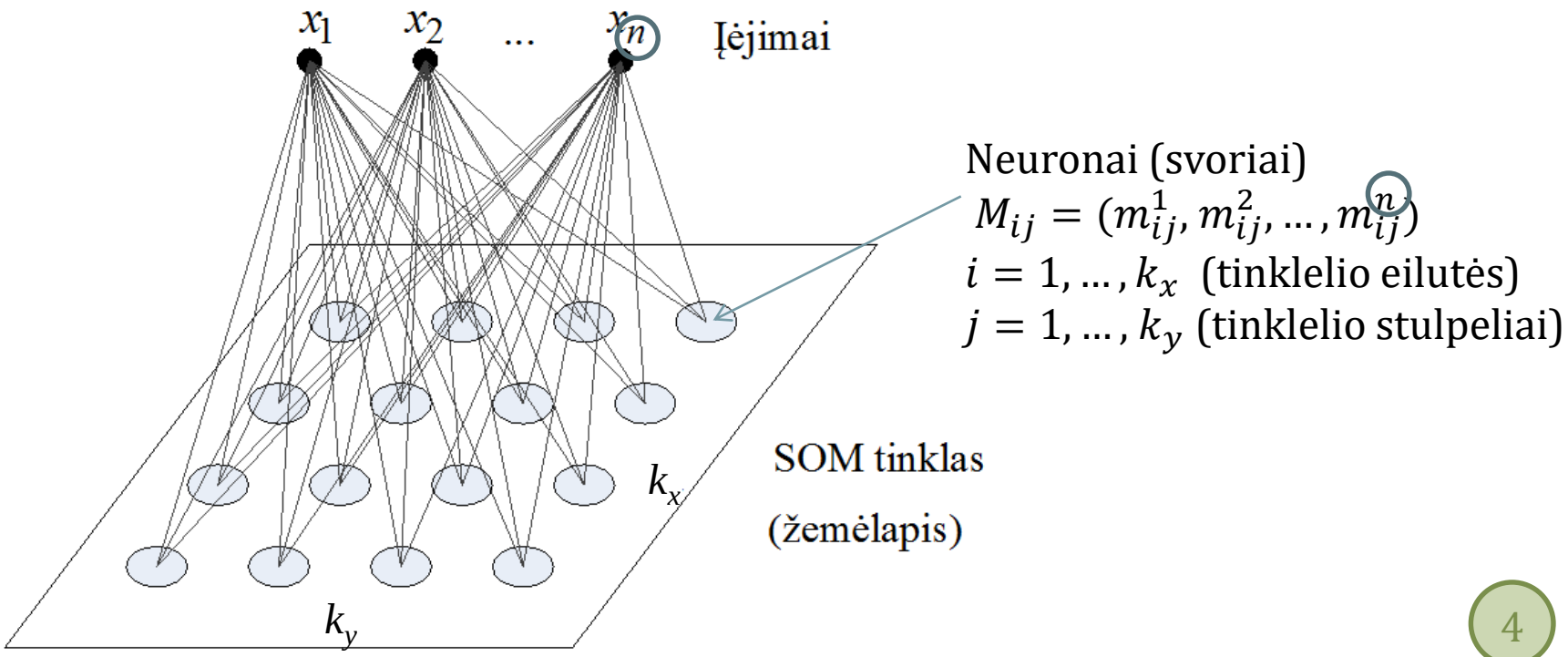
- **Saviorganizuojančius neuroninius tinklus** (žemėlapius, *self-organizing maps*, **SOM**) Suomijos mokslininkas T. Kohonenas pradėjo tyrinėti apie 1982 m., todėl jie dar vadinami Kohoneno neuroniniais tinklais, arba **Kohoneno saviorganizuojančiais žemėlapiais**.
- Iki šiol jie yra **nagrinėjami** daugelio pasaulio mokslininkų ir **plačiai taikomi** įvairiose srityse.
- **Pavadinimas kilo** iš to, kad saviorganizuojantis žemėlapis, naudodamas mokymo (įėjimo) aibę, pats save sukuria (**save organizuoja**).

Saviorganizuojantys neuroniniai tinklai (2)

- Pagrindinis **SOM tinklo tikslas** – išlaikyti duomenų topologiją.
- Taškai, **esantys arti** įėjimo vektorių erdvėje, yra atvaizduojami **arti vieni kitų ir SOM** tinkle.
- SOM tinklai gali būti naudojami siekiant **vizualiai pateikti duomenų klasterius** (grupes) ir ieškant daugiamačių duomenų **projekcijų** į mažesnio skaičiaus matmenų erdvę, paprastai į plokštumą.
- Todėl SOM yra ir **klasterizavimo**, ir **vizualizavimo** metodas.

Saviorganizuojantys neuroniniai tinklai (3)

Saviorganizuojantis neuroninis tinklas yra neuronų, paprastai išdėstytų **dvimačio tinklelio**, dar vadinamo **žemėlapiu** arba **lentele**, mazguose, **masyvas** $M = \{M_{ij}, i = 1, \ldots, k_x, j = 1, \ldots, k_y\}$.



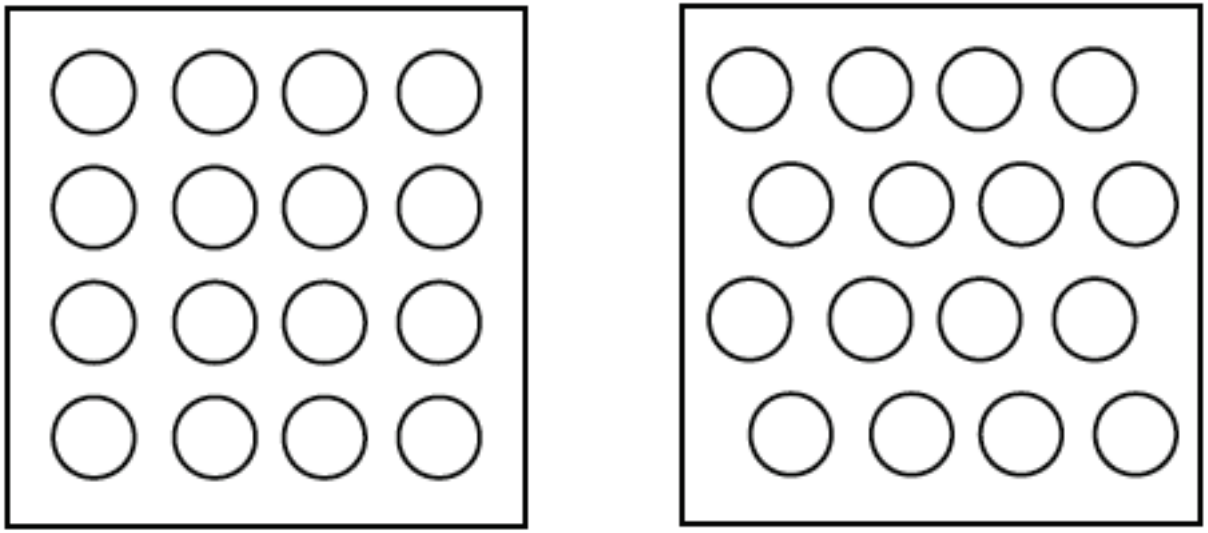
Saviorganizuojantys neuroniniai tinklai (4)

- Būtina suprasti, kad po kiekvienu SOM tinklo neuronu (paveiksle pažymėtu apskritimu) **„slepiasi“ vektorius** (*codebook vector*), kurio komponenčių skaičius sutampa su analizuojamų duomenų požymių skaičiumi.
- Priešingai nei anksčiau nagrinėti tiesioginio sklidimo neuroniniai tinklai, **nulinio įėjimo** SOM tinklas neturi.

SOM tinklo struktūra

Galimos SOM tinklo struktūros:

- **stačiakampė** (*rectangular*) (a)
- arba **šešiakampė** (*hexagonal*) (b)



a) b)

SOM tinklo mokymas (1)

- SOM tinklas mokomas **mokymo be mokytojo** būdu.
- Vektorių, apibūdinantį *i*-osios eilutės *j*-ajame stulpelyje esantį **neuroną**, pažymėkime $M_{ij} = (m_1^{ij}, m_2^{ij}, \ldots, m_n^{ij})$
- Mokymo pradžioje neuronų (vektorių) M_{ij} komponenčių $m_1^{ij}, m_2^{ij}, \ldots, m_n^{ij}$ **pradinės** reikšmės dažniausiai **nustatomos atsitiktinai** intervale (0, 1).
- Neuroniniam tinklui **daug kartų pateikiama** skirtingų objektų, nusakomų *n*-mačiais vektoriais $X_1, X_2, \ldots, X_m$.

SOM tinklo mokymas (2)

- Kiekviename mokymo žingsnyje (iteracijoje) **vienas** mokymo aibės **vektorius** X_k **pateikiamas** į tinklą.
- Vektorius X_k **palyginamas** su visais neuronais M_{ij}: dažniausiai skaičiuojamas **Euklido atstumas** tarp šio vektoriaus ir kiekvieno neurono ($||X_k - M_{ij}||$).
- Randama, iki kurio neurono $M_c \in \{M_{ij}\}$ atstumas yra mažiausias; rastas neuronas vadinamas **neuronu (vektoriumi) nugalėtoju** (*neuron (vector) winner*).

SOM tinklo mokymas (3)

- Visų tinklo neuronų **komponentės keičiamos** naudojantis iteracine formule:

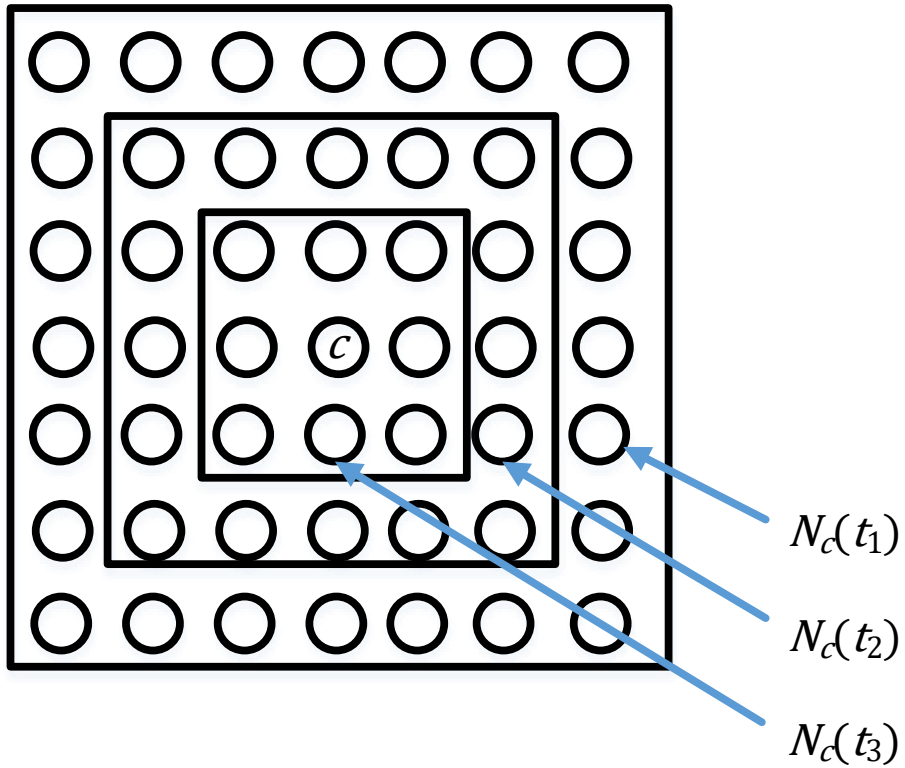
$$M_{ij}(t+1) = M_{ij}(t) + h_{ij}^{c}(X_k - M_{ij}(t))$$

- Čia c nurodo **neurono nugalėtojo indeksus** SOM žemėlapyje.
- t – **iteracijos** numeris,
- h_{ij}^{c} – **kaimynystės funkcija**, kuri yra mažėjanti funkcija ir artėjanti į 0, kai iteracijų skaičius artėja į begalybę; be to jos reikšmės **priklauso ir nuo neurono nugalėtojo vietos** perskaičiuojamo neurono atžvilgiu.

SOM kaimynystės funkcijos

- Galimos įvairios SOM **kaimynystės funkcijos** $h_{ij}{}^{c}$. Populiariosios šios:
  - Burbuliuko: $h_{ij}^{c}(t) = \begin{cases} \alpha(t), (i,j) \in N_c \\ 0, (i,j) \notin N_c \end{cases}$
  - Gauso: $h_{ij}^{c}(t) = \alpha(t) \cdot \exp\left(\frac{-\|R_c - R_{ij}\|^2}{2\left(\eta_{ij}^{c}(t)\right)^2}\right)$
- Čia N_c yra kaimyninių neuronų indeksų aibė aplink neuroną su indeksu c. Dvimačiai vektoriai R_c ir R_{ij} yra neuronų M_c ir M_{ij} indeksai. Indeksai parodo SOM žemėlapyje esančio neurono vietą (eilutės ir stulpelio numerį).
- Parametras η_{ij}^{c} yra neurono M_{ij} kaimynystės eilės numeris neurono-nugalėtojo M_c atžvilgiu.

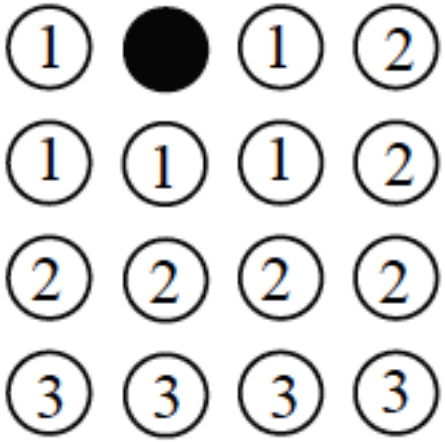
SOM kaimynystės aibės

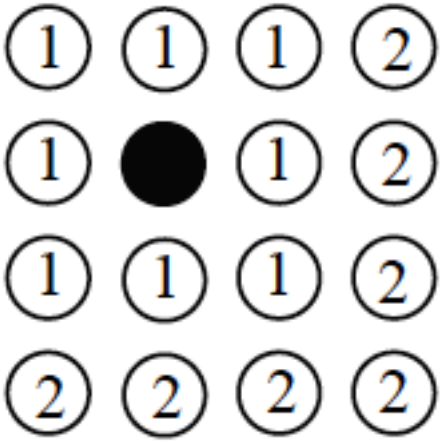


Kaimynystės aibės $N_c(t_1), N_c(t_2), N_c(t_3)$, čia $(t_1 < t_2 < t_3)$

SOM kaimynystės eilė

- Greta vektoriaus nugalėtojo M_c esantys neuronai vadinami **pirmosios eilės kaimynais** (kaimynystės eilė $\eta_{ij}^{c} = 1$).
- Greta pirmosios eilės kaimynų esantys neuronai, išskyrus jau paminėtus, vadinami **antrosios eilės kaimynais** (kaimynystės eilė $\eta_{ij}^{c} = 2$) ir t. t.





SOM mokymo parametras

- Kaimynystės funkcija $\alpha(t)$ yra **mokymo parametras**.
- Dažniausiai:
  - $\alpha(t) = \left(1 - \frac{t}{T}\right)$
  - $\alpha(t) = \frac{1}{t}$
  - $\alpha(t) = (0{,}005)^{\frac{t}{T}}$
- T yra iteracijų (epochų) skaičius.

function SOM_training($X, M, \hat{e}, k_x, k_y$)

// įvestis: X – duomenų aibė, M – pradiniai neuronai, $\hat{e}$ – tinklo mokymo epochų skaičius,

// k_x, k_y – eilučių ir stulpelių skaičius

// išvestis: M – neuronai

BEGIN

FOR $t = 1$ **TO** $\hat{e}$

FOR $l = 1$ **TO** m // *duomenų aibės vektorius* X_l *pateikiamas į neuroninį tinklą*

FOR $i = 1$ TO k_x

FOR $j = 1$ TO k_y

$$\|M_{ij} - X_l\| := \sqrt{\sum_{p=1}^{n} (m_p^{ij} - x_{lp})^2}$$ // *skaičiuojamas Euklido atstumas*

END

END

$c := \arg\min_{i,j} \{\|X_l - M_{ij}\|\}$ // $\hat{M}_c$ – *vektoriaus* X_l *neuronas nugalėtojas*

FOR $i = 1$ TO k_x

FOR $j = 1$ TO k_y

$$M_{ij}(t+1) := M_{ij}(t) + h_{ij}^{c}(t)(X_l - M_{ij}(t))$$ // *SOM mokymo taisyklė*

END

END

END // *visų vektorių peržiūrėjimo pabaiga*

END // *mokymo pabaiga*

RETURN M

END

SOM tinklo rezultatas (1)

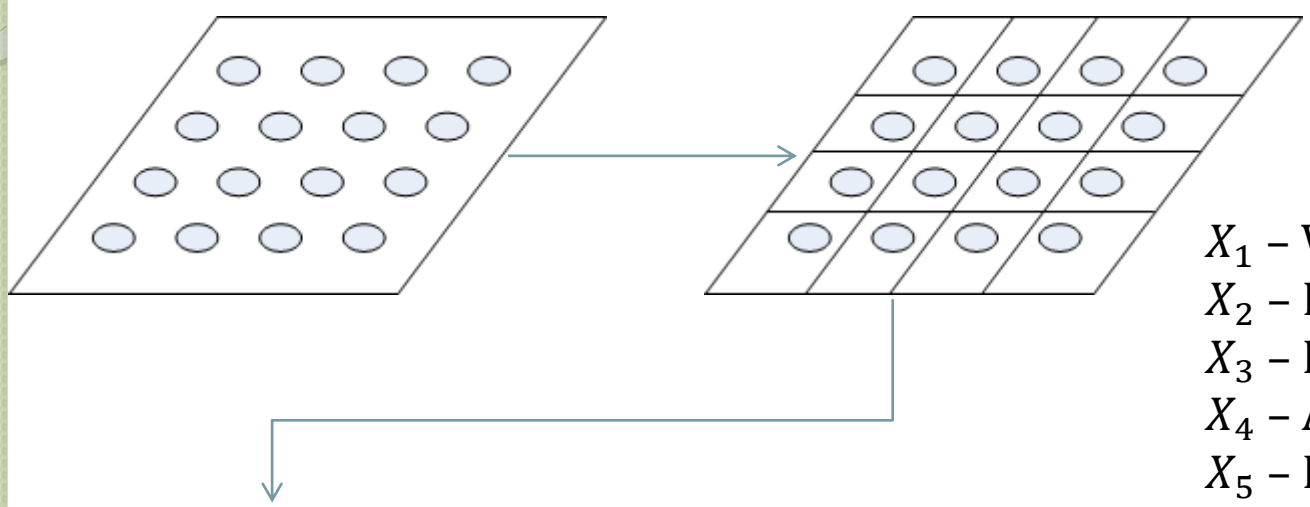
- **Po SOM tinklo** mokymo į tinklą pateikiami mokymo aibės arba nauji, dar tinklui „nematyti“, duomenų vektoriai.
- Randamas kiekvieno vektoriaus **neuronas nugalėtojas** ir jis pažymimas SOM žemėlapyje neurono nugalėtojo vietoje.
- Tokiu būdu vektoriai **išsidėsto tarp žemėlapio** (lentelės) elementų.

SOM tinklo rezultatas (2)

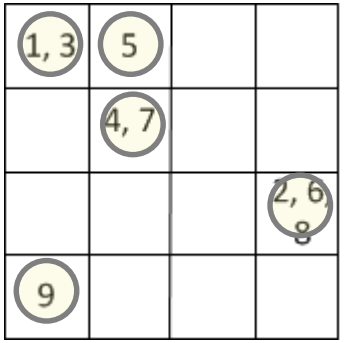
Irisų duomenys, pateikti **SOM** tinkle [10 x 10]. Čia pavaizduojami klasių numeriai, bet gali būti pavaizduojami ir duomenų numeriai.

3		3	3	2				1	1
3				2				1	1
3	3		2	2				1	1
3	3	3		2					1
3		3		2					1
		3	2	2	2	2			
3	3	3			2				
2,3	3			2		2			2
2				2		2	2		2
3		2	2	3	2	2	2	2	2

SOM rezultatas



M_{11}	M_{12}	M_{13}	M_{14}
M_{21}	M_{22}	M_{23}	M_{24}
M_{31}	M_{32}	M_{33}	M_{34}
M_{41}	M_{42}	M_{43}	M_{44}



X_1 – Vokietija
X_2 – Lietuva
X_3 – Prancūzija
X_4 – Austrija
X_5 – Didžioji Britanija
X_6 – Latvija
X_7 – Norvegija
X_8 – Lenkija
X_9 – Graikija

– neuronai-nugalėtojai

SOM tinklo kokybės nustatymas (1)

- Baigus SOM tinklo mokymus, būtina nustatyti jo kokybę.
- Dažniausiai vertinamos dvi paklaidos:
  - **kvantavimo** (*quantization error*)
  - ir **topografinė** (*topographic error*).
- **Kvantavimo paklaida** parodo, kaip tiksliai jau išmokyto tinklo neuronai prisiderina prie mokymo aibės vektorių.
- Tai **vidutinis atstumas** tarp duomenų vektorių ir jų vektorių nugalėtojų:

$$E_{SOM(q)} = \frac{1}{m} \sum_{k=1}^{m} \| X_k - M_{c(k)} \|$$

SOM tinklo kokybės nustatymas (2)

- **Topografinė paklaida** parodo, kaip gerai SOM tinklas išlaiko analizuojamų duomenų **topografiją**, t. y. tarpusavio išsidėstymą.
- Ji skaičiuojama pagal formulę:

$$E_{SOM(t)} = \frac{1}{m} \sum_{k=1}^{m} u(X_k)$$

- Jeigu SOM žemėlapyje vektoriaus X_k neuronas nugalėtojas **yra šalia neurono**, iki kurio atstumas nuo X_k yra mažiausias, neskaičiuojant iki neurono nugalėtojo, tai formulėje $u(X_k) = 0$, priešingu atveju $u(X_k) = 1$.

SOM vizualizavimo būdai

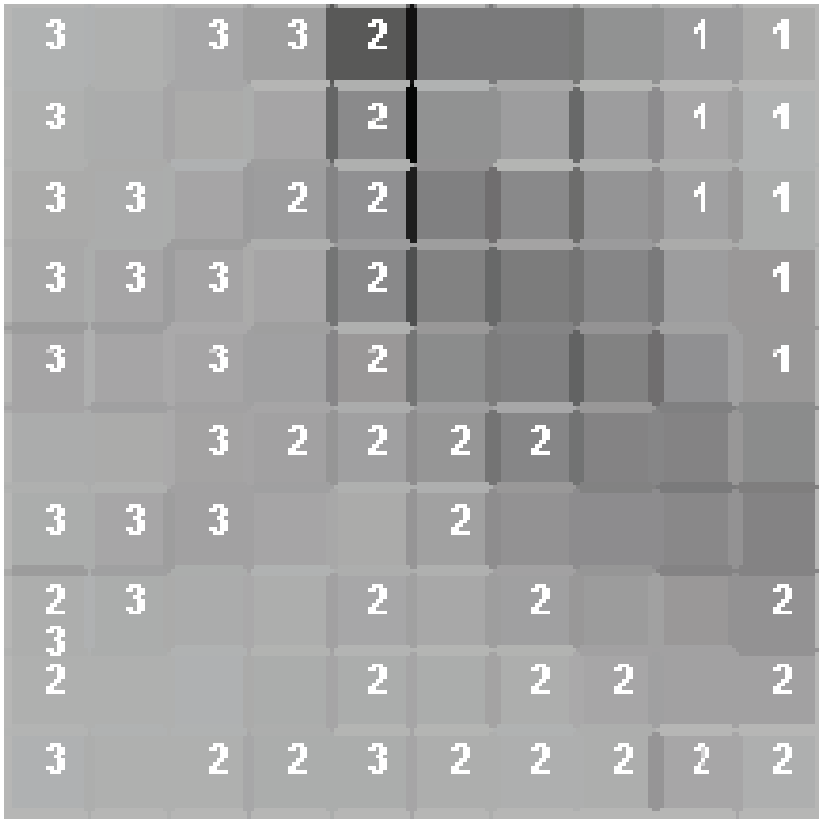
- Paprasčiausia SOM lentelė nėra labai informatyvi, **sunku pasakyti**, kaip toli yra duomenys, esantys gretimuose SOM lentelės langeliuose.
- Todėl **būtina ieškoti** būdų, kaip pagerinti tokio vizualizavimo kokybę.
- **Unifikuota atstumų matrica** (U-matrica, *unified distance matrix*) yra vienas populiariausių SOM vizualizavimo būdų.
- U-matricą sudaro atstumai tarp **kaimynynių** SOM neuronų.

U-matrica

- Paprastumo dėlei nagrinėkime **vienmatį SOM** [1x5] $(M_1, M_2, \dots, M_5)$.
- **U-matrica** bus vienos eilutės ir devynių stulpelių $(u_1, u_{12}, u_2, u_{23}, u_3, u_{34}, u_4, u_{45}, u_5)$.
- Čia $u_{ij} = ||M_i - M_j||$ yra **atstumas** tarp kaimyninių neuronų M_i ir M_j, o u_i yra tam tikra reikšmė, pavyzdžiui, vidutinis atstumas tarp kaimyninių reikšmių

$$u_i = \frac{u_{(i-1)i} + u_{i(i+1)}}{2}$$

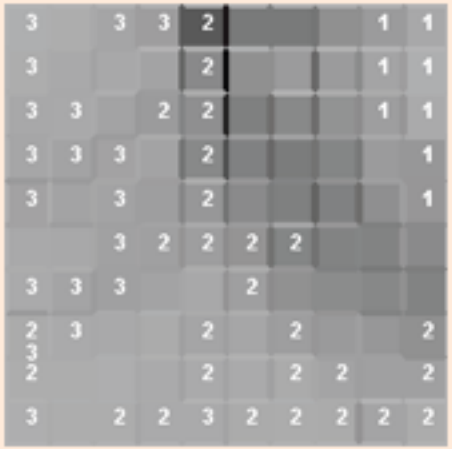
Irisų duomenys, vizualizuoti u-matrica



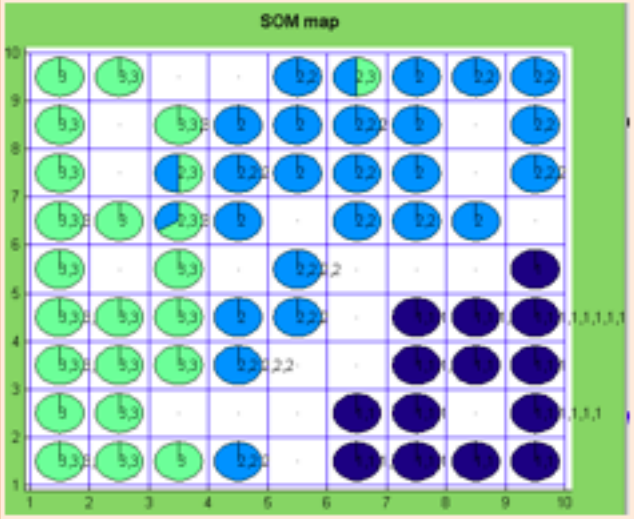
Irisai įvairiais vizualizavimo būdais

3		3	3	2				1	1
3				2				1	1
3	3		2	2				1	1
3	3	3		2					1
3		3		2					1
		3	2	2	2	2			
3	3	3			2				
2,3	3			2		2			2
2				2		2	2		2
3		2	2	3	2	2	2	2	2

paprasčiausia SOM lentelė



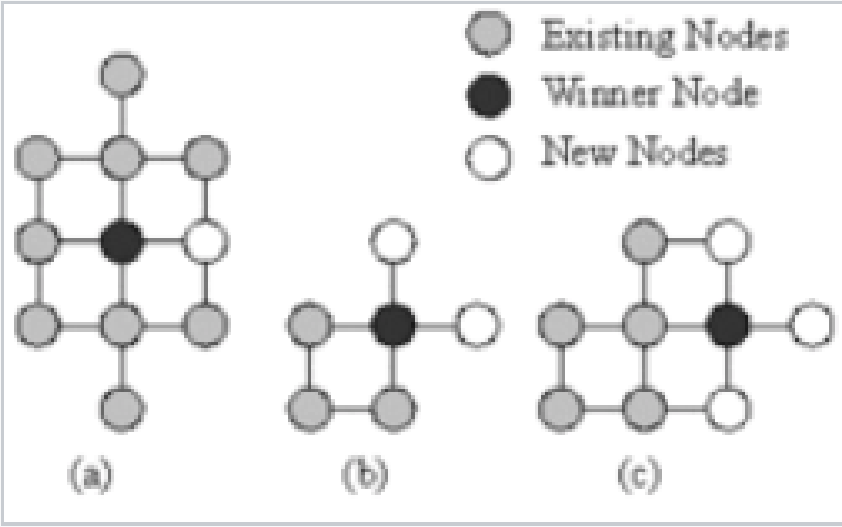
SOM lentelė vizualizuota U-matricos pagalba



SOM lentelė su skritulinėmis diagramomis

SOM praplėtimai

Augantis SOM (*growing self-organizing map*, GSOM).



Node growth options in GSOM: (a) one new node, (b) two new nodes and (c) three new nodes.

SOM praplėtimai

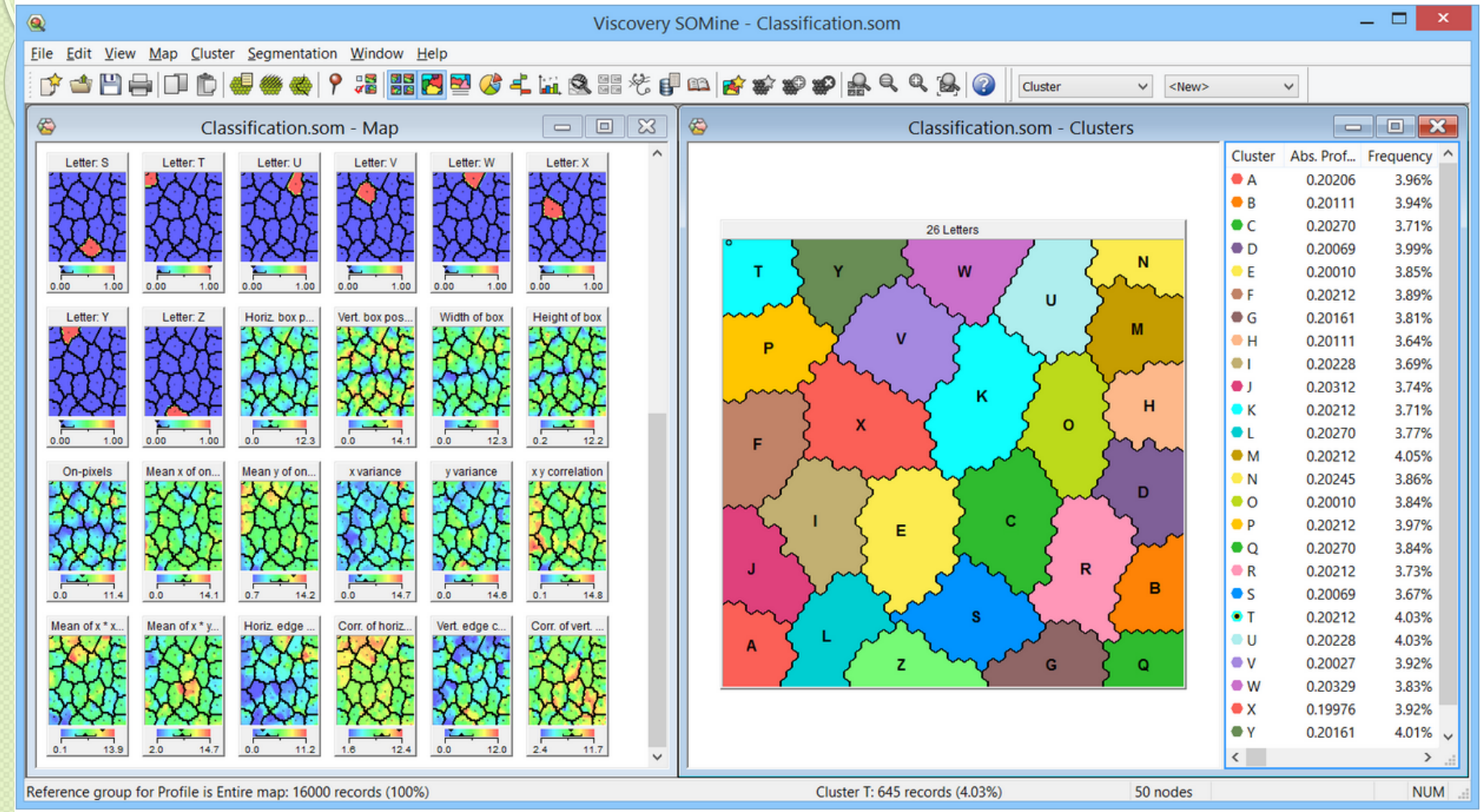
- **Laike prisitaikantis SOM** (*time adaptive self-organizing map*, TASOM), kur kiekvienas neuronas turi prisitaikantį savo mokymo parametrą ir kaimynystės dydį.
- **Generatyvinis topografinis žemėlapis** (*generative topographic map*, GTM) – tai alternatyva SOM'ui.

Viscovery SOMine

- **Viscovery SOMine** is a workflow-oriented software suite based on **self-organizing maps** (SOM) and multivariate statistics for explorative data mining and predictive modeling.
- Main functions and features:
  - **Creation** of self-organizing map models using predefined schedules
  - Interactive **SOM visualization** and exploration
  - Visual cluster analysis with integrated visualization of **cluster boundaries** and inner structures
  - **Statistical functions**, such as descriptive statistics, histograms, correlations, PCA, and scatter plots

From: https://www.viscovery.net/somine/

Viscovery SOMine



Optical character recognition

From: https://www.viscovery.net/somine/