Predictive Learning from Data

LECTURE SET 6

Methods for Data Reduction and Dimensionality Reduction

Cherkassky, Vladimir, and Filip M. Mulier. *Learning from data: concepts, theory, and methods*. John Wiley & Sons, 2007.

Source: Dr. Vladimir Cherkassky (revised by Dr. Hsiang-Han Chen)

OUTLINE

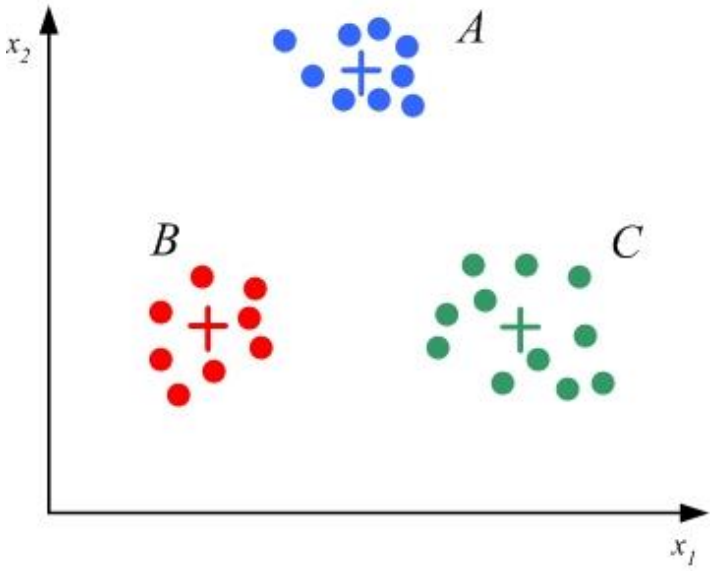
- **Motivation for unsupervised learning**
  - Goals of modeling
  - Overview of artificial neural networks
- Vector quantization and unsupervised learning
- Statistical methods for dim. reduction
- Methods for multivariate data analysis
- Summary and discussion

MOTIVATION

Recall from *Lecture Set 2*:

- unsupervised learning
- data reduction and dimensionality reduction

- Example: Training data represented by 3 ‘centers’



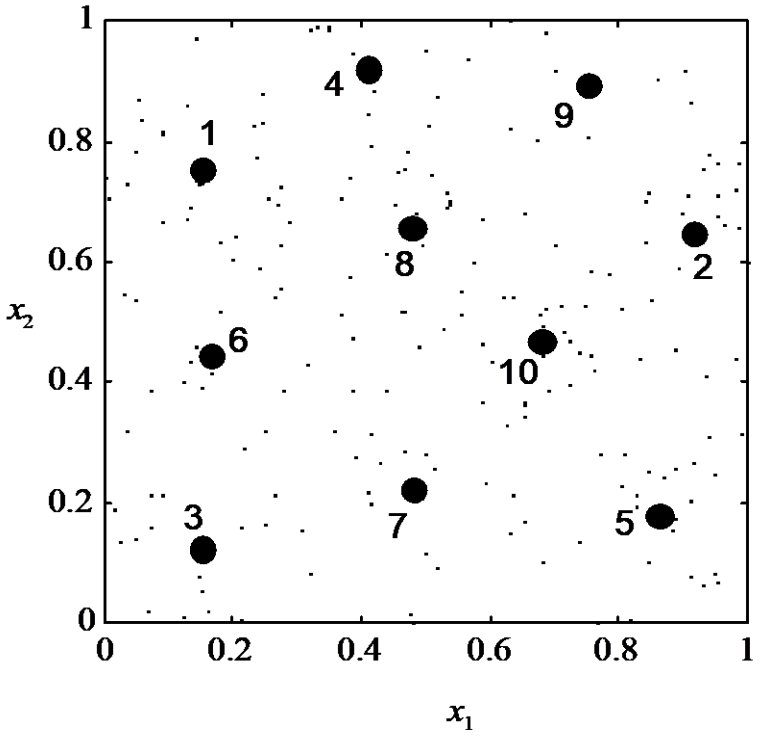
Two types of problems

1. Data reduction:

VQ + clustering

Vector Quantizer Q:

$$f(\mathbf{x},\omega) = Q(\mathbf{x}) = \sum_{j=1}^{m} \mathbf{c}_j I\left(\mathbf{x} \in \mathbf{R}_j\right)$$

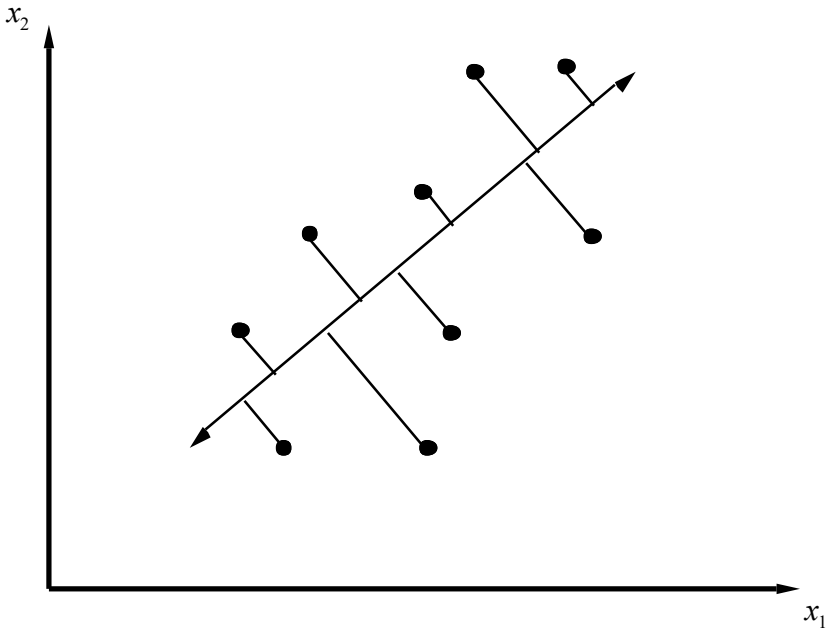


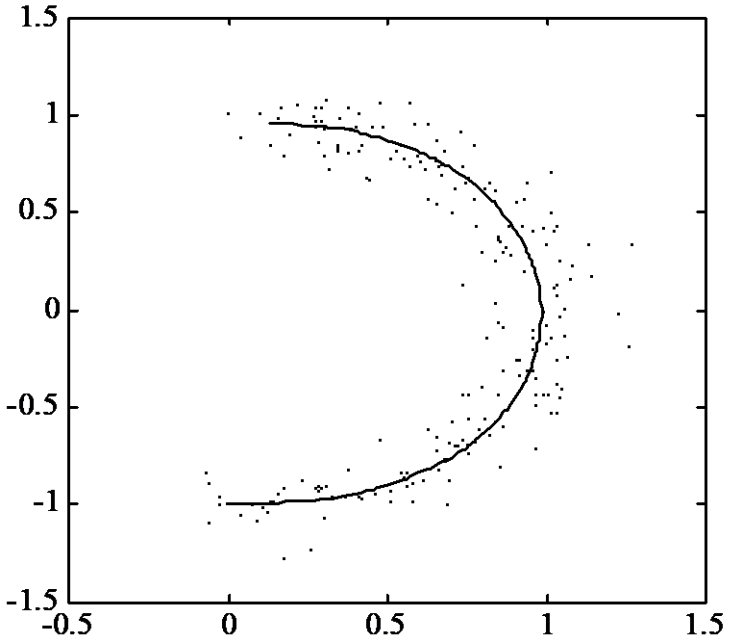
VQ setting: given *n* training samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ find the coordinates $\mathbf{c}_j$ of *m* centers (prototypes) such that the total squared error distortion is minimized

$$R(\omega) = \int \left\| \mathbf{x} - f(\mathbf{x},\omega) \right\|^2 p(\mathbf{x}) d\mathbf{x}$$

2. Dimensionality reduction:

linear ***nonlinear***





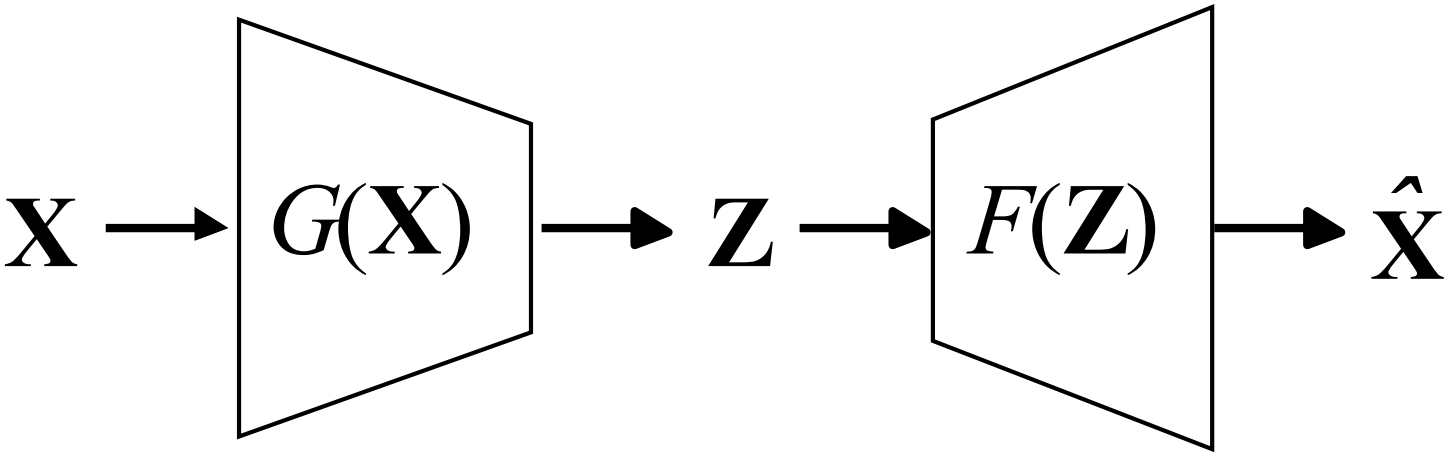
Note: the goal is to estimate a mapping from *d*-dimensional input space (*d=2*) to low-dim. feature space (*m=1*)

$$R(\omega) = \int \left\| \mathbf{x} - \boldsymbol{f}(\mathbf{x}, \omega) \right\|^2 \boldsymbol{p}(\mathbf{x}) d\mathbf{x}$$

Dimensionality reduction

- **Dimensionality reduction** as information bottleneck
  (= data reduction)
- The goal of learning is to find a mapping $\boldsymbol{f}(\mathbf{x},\omega)=F(G(\mathbf{x}))$ minimizing prediction risk $R(\omega)=\int L(\mathbf{x},\boldsymbol{f}(\mathbf{x},\omega))\boldsymbol{p}(\mathbf{x})d\mathbf{x}$

  Note $\mathbf{z}=G(\mathbf{x})$ provides low-dimensional encoding of the original high-dimensional data



Goals of Unsupervised Learning

- Usually, *not prediction*
- **Understanding** of multivariate data via

  - data reduction (clustering)

  - dimensionality reduction
- Only input (**x**) samples are available
- **Preprocessing and feature selection** preceding supervised learning
- Methods originate from information theory, statistics, neural networks, sociology etc.
- May be difficult to assess objectively

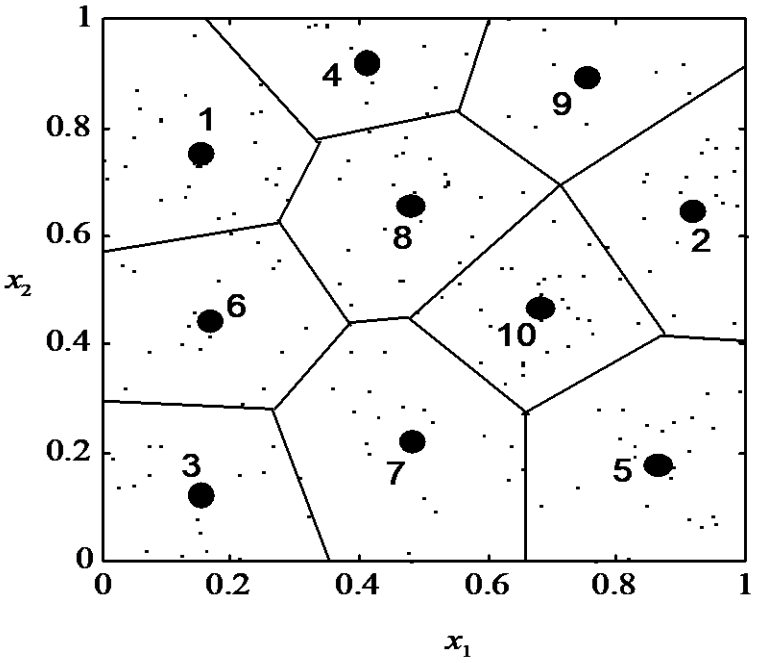
OUTLINE

- Motivation for unsupervised learning
- **Vector quantization and unsupervised learning**
  - Vector quantization and clustering
  - Self-Organizing Maps (SOM)
  - MLP for data compression
- Statistical methods for dim. reduction
- Methods for multivariate data analysis
- Summary and discussion

Vector Quantization and Clustering

- Two complementary goals of VQ:

  1. partition the input space into disjoint regions
  2. find positions of units (coordinates of prototypes)



Note: optimal partitioning into regions is according to the nearest-neighbor rule (~ the Voronoi regions)

Generalized Lloyd Algorithm(GLA) for VQ

Given data points $\mathbf{x}(k) \quad k = 1,2,\ldots$, loss function *L* (i.e., squared loss) and initial centers $\mathbf{c}_j(0) \quad j = 1,\ldots,m$

Perform the following updates upon presentation of $\mathbf{x}(k)$

1. **Find the nearest center** to the data point (the winning unit):

$$j = \underset{i}{\boldsymbol{arg\,min}} \left\| \mathbf{x}(k) - \mathbf{c}_i(k) \right\|$$

2. **Update the winning unit** coordinates (only) via

$$\mathbf{c}_j(k+1) = \mathbf{c}_j(k) + \gamma(k)\left[\mathbf{x}(k) - \mathbf{c}_j(k)\right]$$

Increment *k* and iterate steps (1) – (2) above

Note: the learning rate decreases with iteration number *k*

Batch version of GLA

Given data points $\mathbf{x}_i \quad i = 1,\ldots,n$, loss function *L* (i.e., squared loss) and initial centers $\mathbf{c}_j(0) \quad j = 1,\ldots,m$

Iterate the following two steps

1. **Partition the data** (assign sample $\mathbf{x}_i$ to unit *j*) using the nearest neighbor rule. Partitioning matrix Q:

$$q_{ij} = \begin{cases} 1 \text{ if } L\left(\mathbf{x}_i, \mathbf{c}_j(k)\right) = \min\limits_l L\left(\mathbf{x}_i, \mathbf{c}_l(k)\right) \\ 0 \text{ otherwise} \end{cases}$$

2. **Update unit coordinates** as centroids of the data:

$$\mathbf{c}_j(k+1) = \frac{\sum_{i=1}^{n} q_{ij}\mathbf{x}_i}{\sum_{i=1}^{n} q_{ij}}, \; j = 1,\ldots,m$$

Note: final solution may depend on initialization (local min) – potential problem for both on-line and batch GLA

Numeric Example of univariate VQ

Given data: {2,4,10,12,3,20,30,11,25}, set m=2

- Initialization (random): $c_1=3, c_2=4$
- Iteration 1

  Projection: $P_1=\{2,3\}$ $P_2=\{4,10,12,20,30,11,25\}$

  Expectation (averaging): $c_1=2.5$, $c_2=16$
- Iteration 2

  Projection: $P_1=\{2,3,4\}$, $P_2=\{10,12,20,30,11,25\}$
  Expectation(averaging): $c_1=3$, $c_2=18$
- Iteration 3

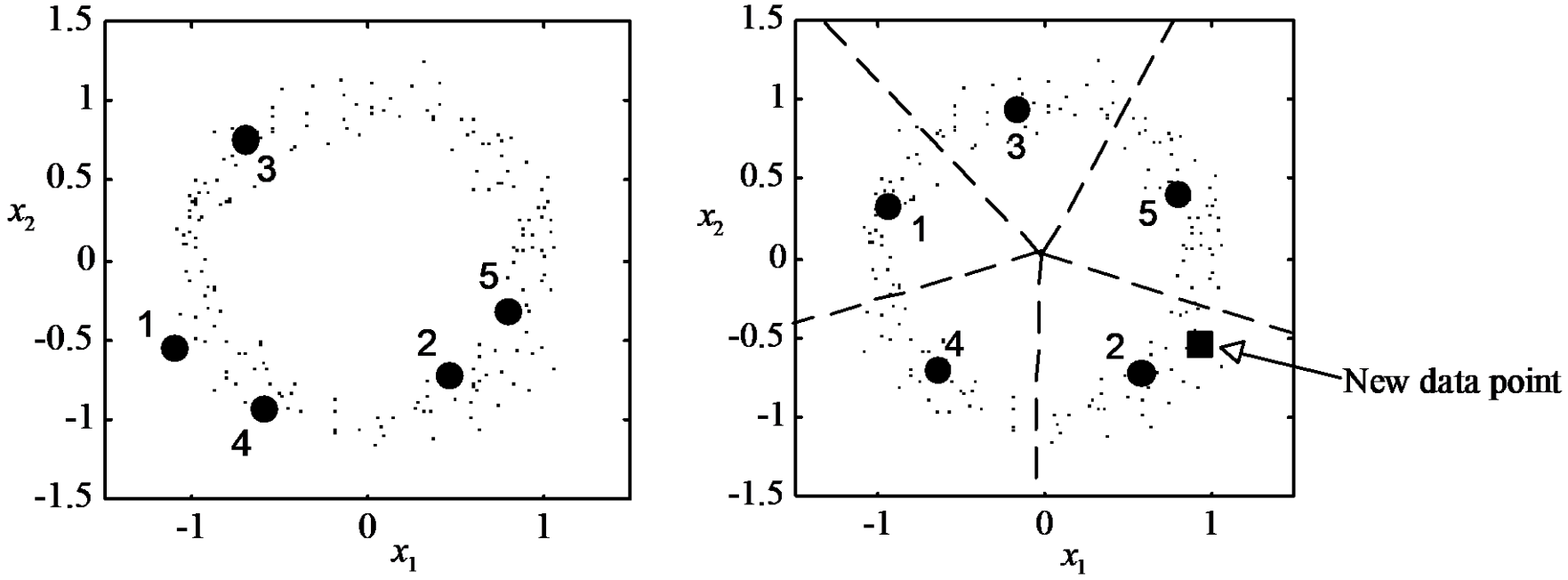
  Projection: $P_1=\{2,3,4,10\}$,$P_2=\{12,20,30,11,25\}$
  Expectation(averaging): $c_1=4.75$, $c_2=19.6$
- Iteration 4

  Projection: $P_1=\{2,3,4,10,11,12\}$, $P_2=\{20,30,25\}$
  Expectation(averaging): $c_1=7$, $c_2=25$
- Stop as the algorithm is stabilized with these values

GLA Example 1

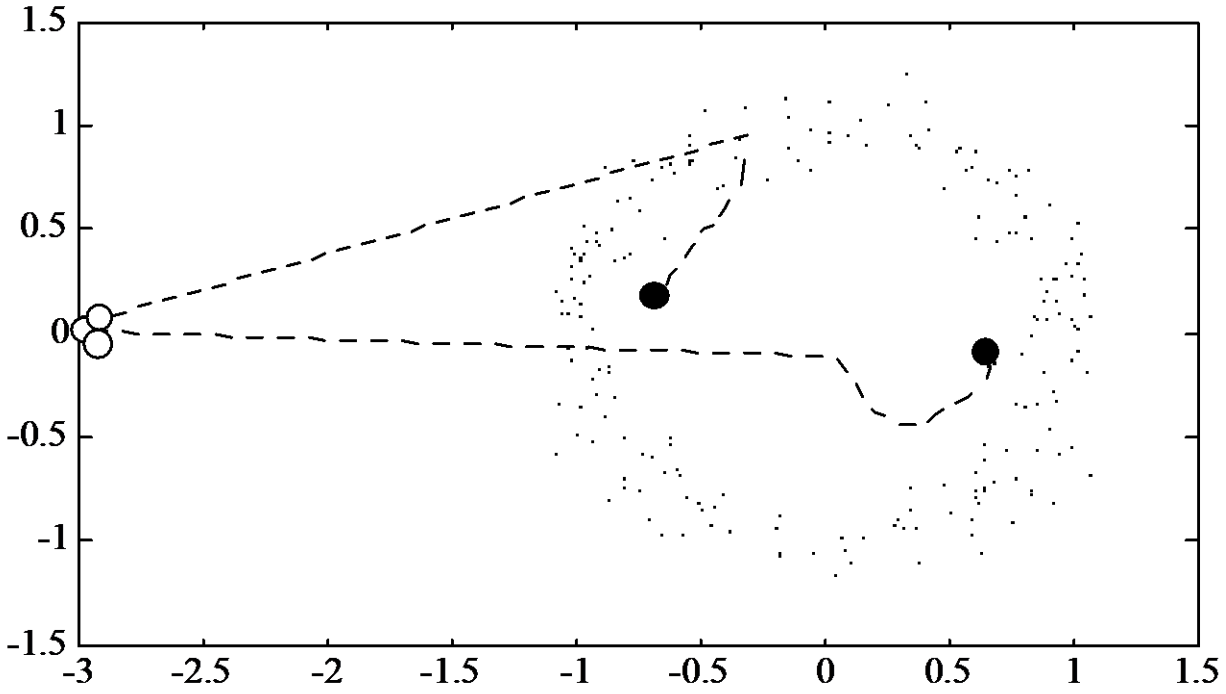
- Modeling doughnut distribution using 5 units

(a) initialization (b) final position (of units)



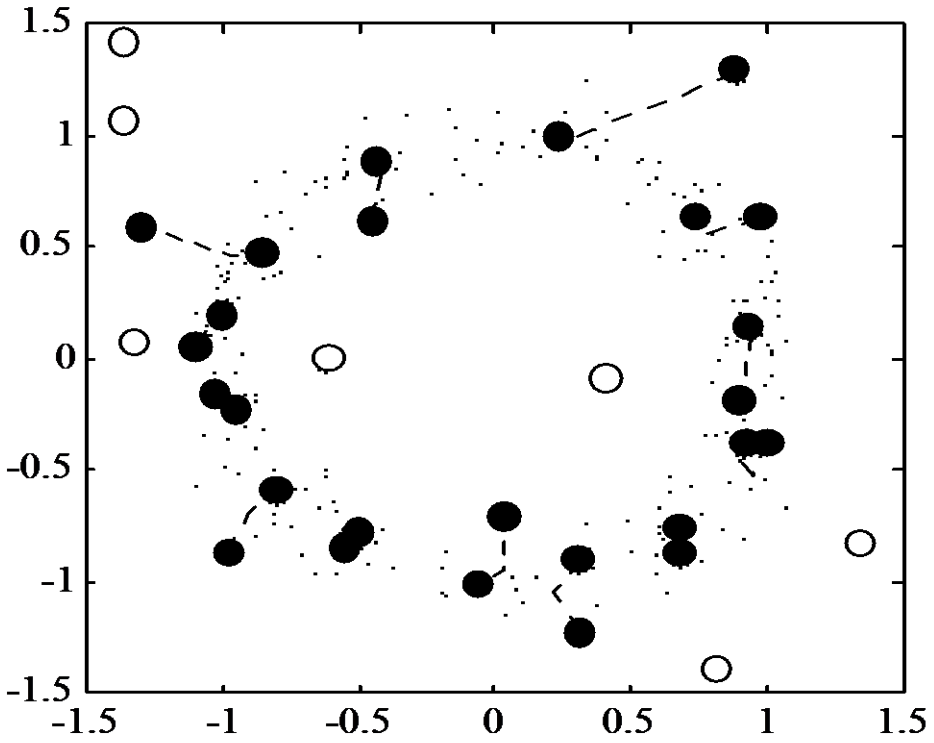
GLA Example 2

- Modeling doughnut distribution using 3 units:
  Bad initialization → poor local minimum



GLA Example 3

- Modeling doughnut distribution using 20 units:
  7 units were never moved by the GLA
  → the problem of unused units (dead units)



Avoiding local minima with GLA

- **Starting with many random initializations**, and then choosing the best GLA solution
- **Conscience mechanism**: forcing 'dead' units to participate in competition, by keeping the frequency count (of past winnings) for each unit,

  i.e. for on-line version of GLA in Step 1

$$j = \arg\min_i \left\| \mathbf{x}(k) - \mathbf{c}_i(k) \right\| freq_i(k)$$

- **Self-Organizing Map**: introduce topological relationship (map), thus forcing the *neighbors* of the winning unit to move towards the data.

Clustering methods

- **Clustering**: separating a data set into several groups (clusters) according to some measure of similarity
- **Goals of clustering**:

  interpretation (of resulting clusters)

  exploratory data analysis

  preprocessing for supervised learning

  often the goal is not formally stated
- **VQ-style methods** (GLA) often used for clustering, aka *k-means* or *c-means*
- Many other clustering methods as well

Clustering (cont'd)

- **Clustering**: partition a set of n objects (samples) into k disjoint groups, based on some similarity measure. Assumptions:

  - similarity ~ distance metric *dist (i,j)*

  - usually k given a priori (but not always!)

- **Intuitive motivation**:

  similar objects into one cluster

  dissimilar objects into different clusters

  → the goal is not formally stated

- **Similarity (distance) measure is critical**

  but usually hard to define (~ feature selection). Distance may need to be defined for different types of input variables.

Overview of Clustering Methods

- **Hierarchical Clustering**:

  tree-structured clusters

- **Partitional methods**:

  typically variations of GLA known as *k-means* or *c-means*, where clusters can merge and split dynamically

- Partitional methods can be divided into

  - **crisp clustering** ~ each sample belongs to only one cluster

  - **fuzzy clustering** ~ each sample may belong to several clusters

Applications of clustering

- **Marketing**:

  explore customers data to identify buying patterns for targeted marketing (Amazon.com)
- **Economic data**:

  identify similarity between different countries, states, regions, companies, mutual funds etc.
- **Web data:**

  cluster web pages or web users to discover groups of similar access patterns
- Etc., etc.

K-means clustering

Given a data set of *n* samples $\mathbf{x}_i$ and the value of k:

Step 0: (arbitrarily) initialize cluster centers

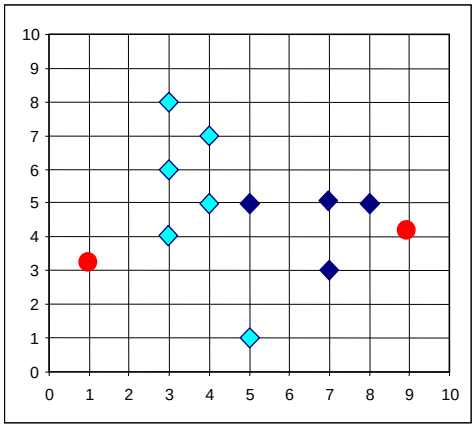
Step 1: assign each data point (object) to the cluster with the closest cluster center

Step 2: calculate the mean (centroid) of data points in each cluster as estimated cluster centers

Iterate steps 1 and 2 until the cluster membership is stabilized

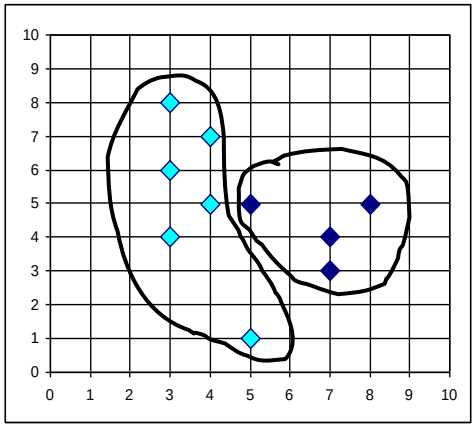
The *K-Means* Clustering Method

- Example

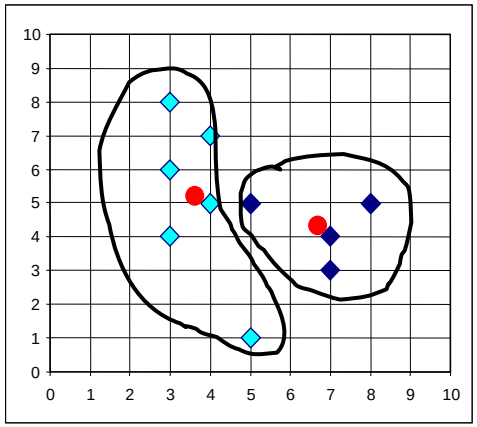


K=2

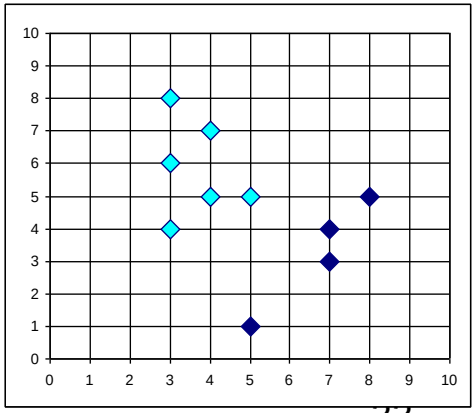
Arbitrarily choose K points as initial cluster centers



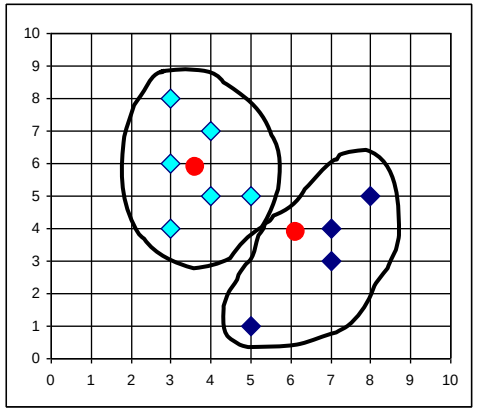
Assign each objects to most similar center



Update the cluster means



reassign



reassign

Update the cluster means

Self-Organizing Maps

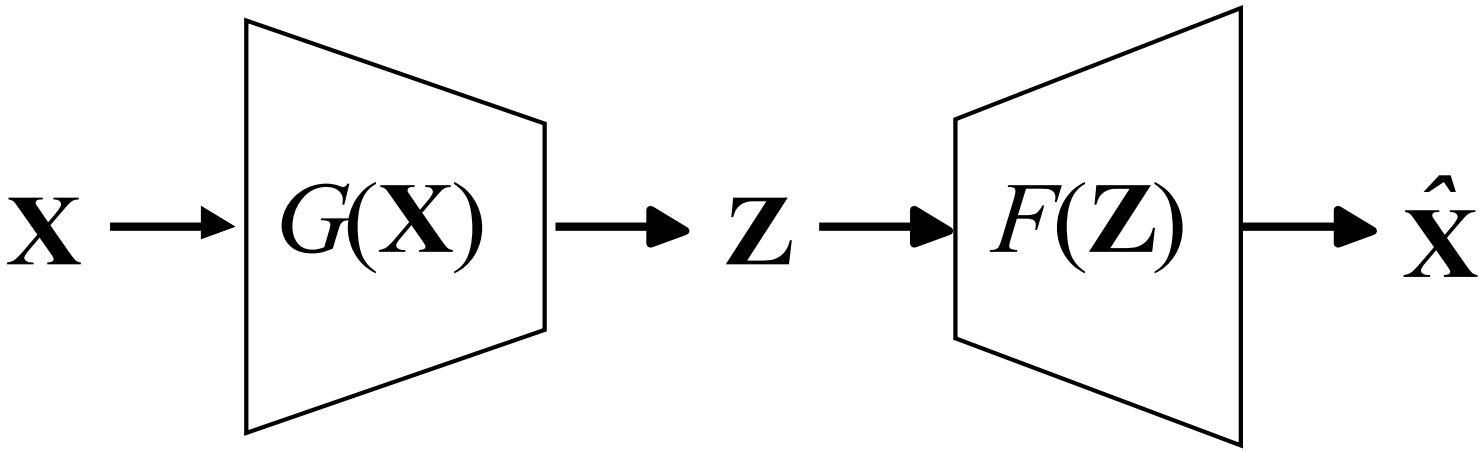
History and biological motivation

- Brain changes its internal structure to reflect life experiences → interaction with environment is critical at early stages of brain development (first 2-3 years of life)
- Existence of various regions (maps) in the brain
- How these maps may be formed?

  i.e. information-processing model leading to map formation
- T. Kohonen (early 1980's) proposed SOM
- Original flow-through SOM version reminds VQ-style algorithm

SOM and dimensionality reduction

- **Dimensionality reduction:** project given (high-dimensional) data onto low-dimensional space (map)
- Feature space (**Z**-space) is 1D or 2D and is discretized as a number of units, i.e., 10x10 map
- Z-space has distance metric → **ordering of units**
- *Encoder mapping* **z** = G(**x**) *Decoder mapping* **x'** = F(**z**)
- Seek a function *f*(**x**,**w**) = F(G(**x**)) minimizing the risk R(**w**) = L(**x**, **x'**) p(**x**) d**x** = L(**x**, *f*(**x**,**w**)) p(**x**) d**x**

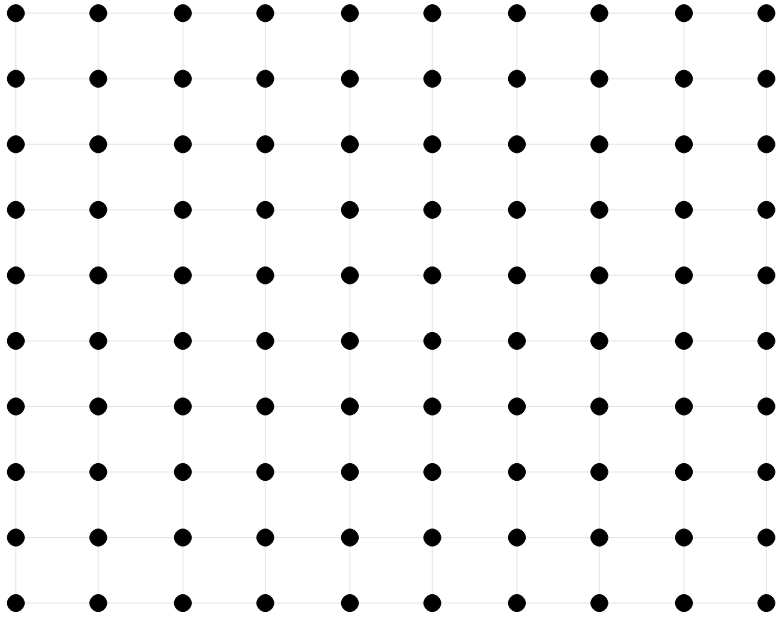


Self-Organizing Map

Discretization of 2D space via 10x10 map. In this discrete space, distance relations exist between all pairs of units.

Distance relation ~ **map topology**

Units in 2D feature space



SOM Algorithm (flow through)

Given data points $\mathbf{x}(k) \quad k = 1, 2, \ldots$, distance metric in the input space (~ Euclidean), map topology (in **z**-space), initial position of units (in **x**-space) $\mathbf{c}_j(0) \quad j = 1, \ldots, m$

Perform the following updates upon presentation of $\mathbf{x}(k)$

1. **Find the nearest center** to the data point (the winning unit):

$$\mathbf{z}^*(k) = \arg\min_i \left\| \mathbf{x}(k) - \mathbf{c}_i(k-1) \right\|$$

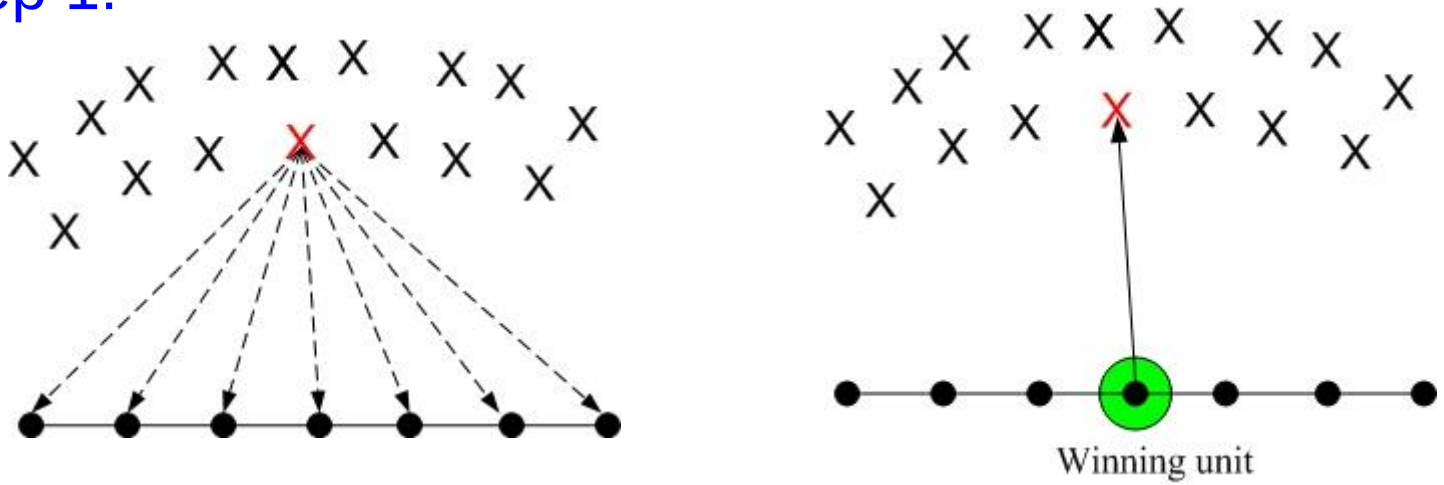
2. **Update all units around the winning unit** via

$$\mathbf{c}_j(k) = \mathbf{c}_j(k-1) + \beta(k) K_{\alpha(k)}(\mathbf{z}, \mathbf{z}^*)\left(\mathbf{x}(k) - \mathbf{c}_j(k-1)\right)$$

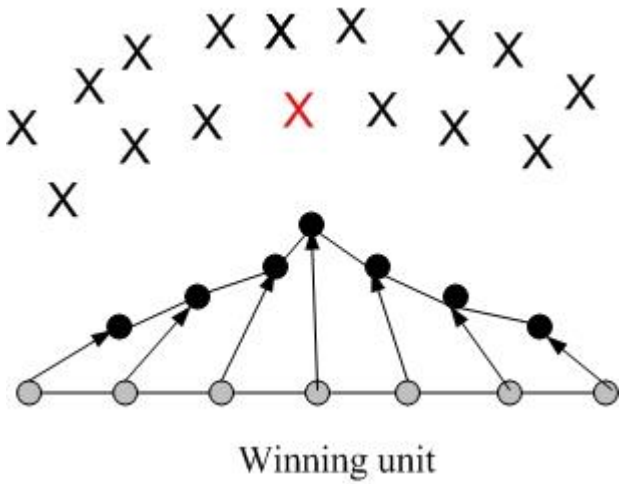
Increment *k*, *decrease the learning rate and the neighborhood width*, and repeat steps (1) – (2) above

SOM example (1-st iteration)

Step 1:

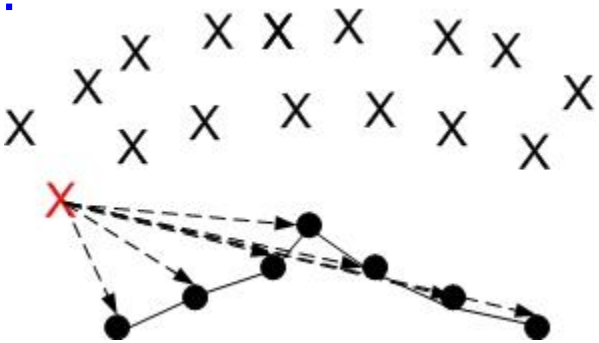


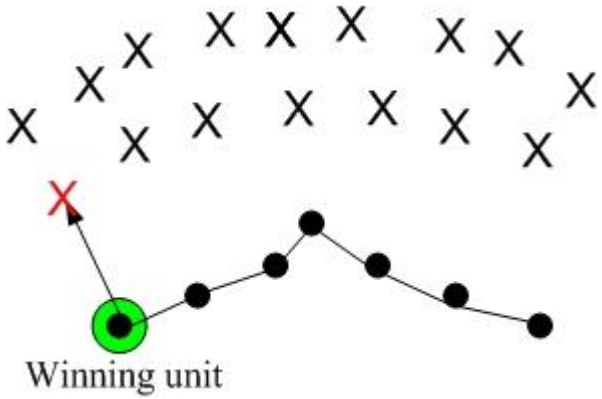
Step 2:



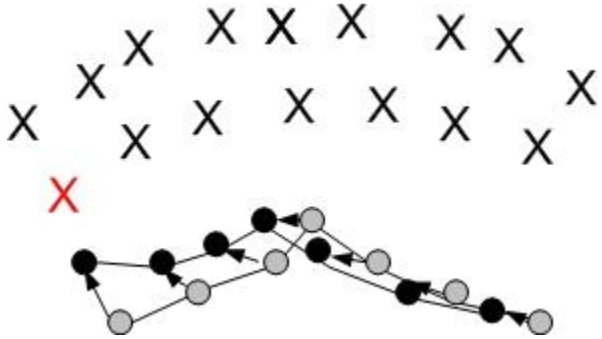
SOM example (next iteration)

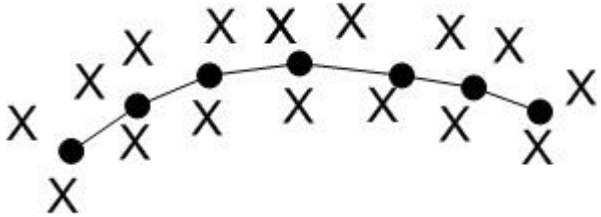
Step 1:





Step 2:





Final map

Hyper-parameters of SOM

SOM performance depends on parameters (~ user-defined):

- **Map dimension and topology** (usually 1D or 2D)
- **Number of SOM units** ~ quantization level (of **z**-space)
- **Neighborhood function** ~ rectangular or gaussian (not important)
- **Neighborhood width decrease schedule** (important), i.e. exponential decrease for Gaussian

$$\alpha(k) = \alpha_{initial}\left(\alpha_{final}/\alpha_{initial}\right)^{k/k_{max}} \qquad K_{\alpha(k)}(\mathbf{z},\mathbf{z}') = \exp\left(-\frac{\|\mathbf{z}-\mathbf{z}'\|^2}{2\alpha^2(k)}\right)$$

with user defined: $k_{\max}$ $\alpha_{initial}$ α_{final}

Also linear decrease of neighborhood width

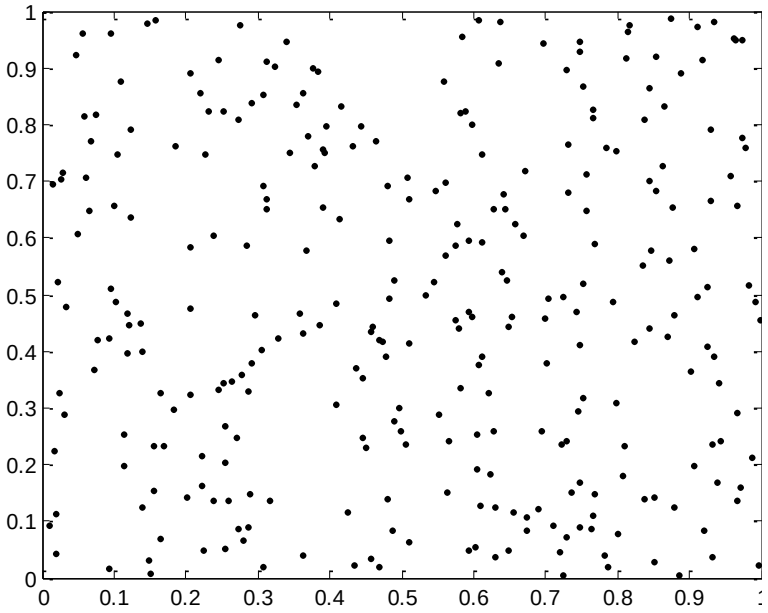
- **Learning rate schedule** (important) $\beta(k) = \beta_{initial}\left(\frac{\beta_{final}}{\beta_{initial}}\right)^{k/k_{max}}$

  (also linear decrease)

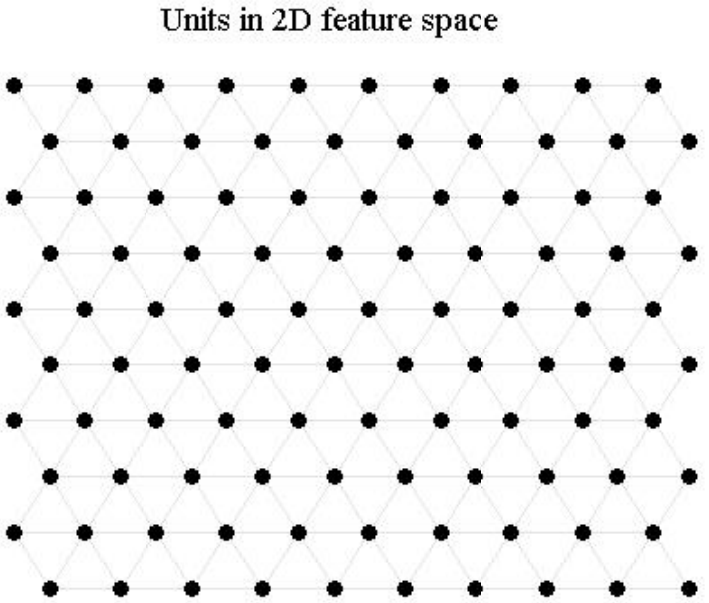
Note: learning rate and neighborhood decrease should be set jointly

Modeling uniform distribution via SOM

(a) 300 random samples



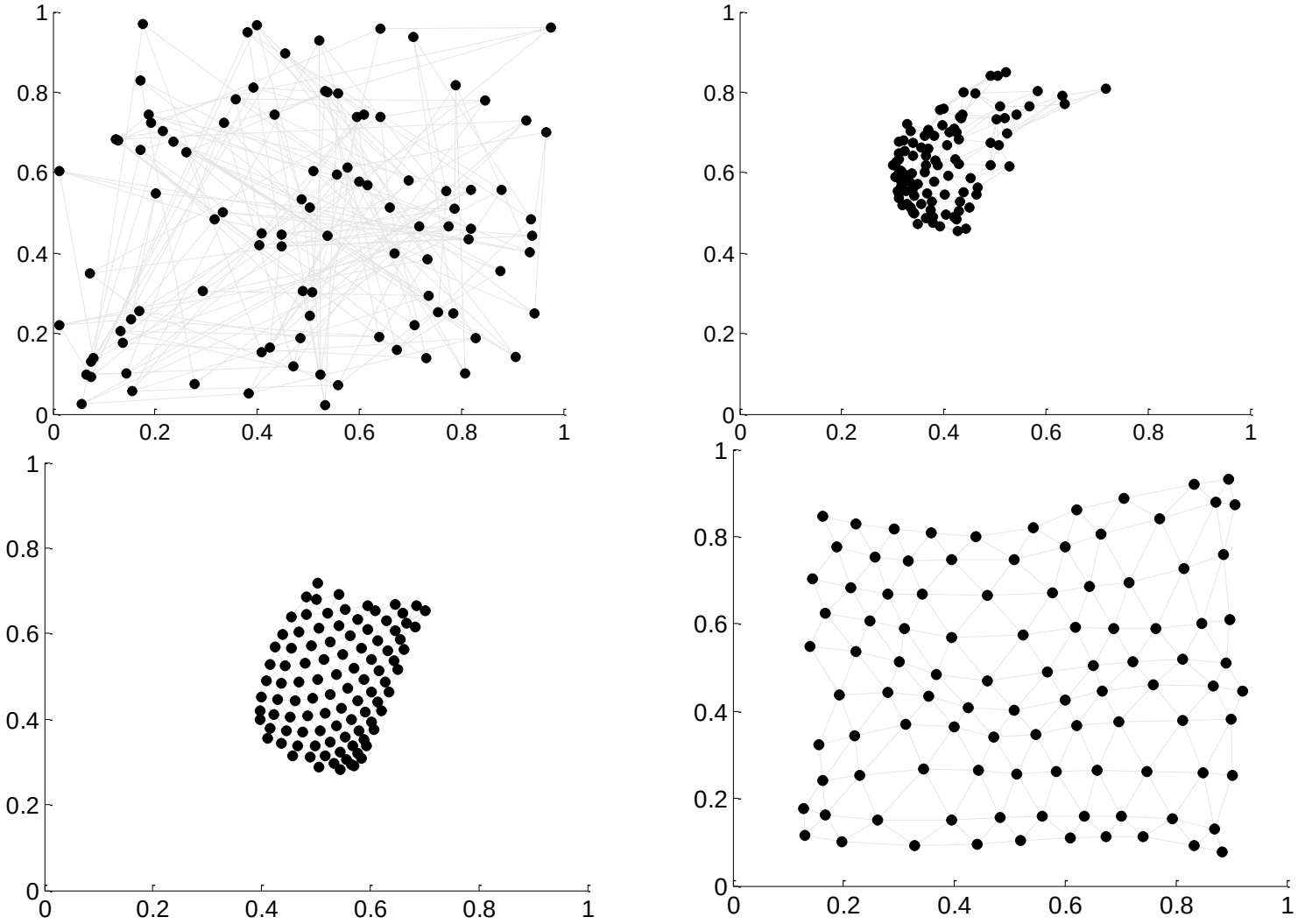
(b) 10X10 map



SOM neighborhood: Gaussian

Learning rate: linear decrease $\beta(k) = 0.1(1 - k / k_{\max})$

Position of SOM units: (a) initial, (b) after 50 iterations, (c) after 100 iterations, (d) after 10,000 iterations



Batch SOM (similar to Batch GLA)

Given data points $\mathbf{x}_i$, distance metric (i.e., Euclidian), map topology and initial centers $\mathbf{c}_j(0) \quad j = 1,\ldots,m$

Iterate the following two steps

1. **Project the data** onto map space (discretized **Z**-space) using the nearest distance rule:

Encoder G(**x**): $\hat{\mathbf{z}}_i = \Psi\left(\underset{j}{\boldsymbol{argmin}} \left\| \mathbf{c}_j - \mathbf{x}_i \right\|^2 \right)$

2. **Update unit coordinates** = **kernel smoothing** (in **Z**-space):

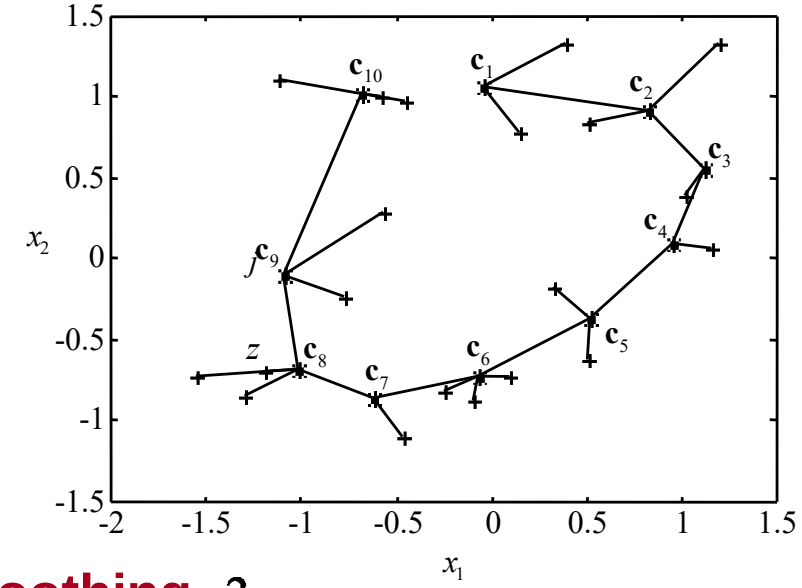
$$F(\mathbf{z}, \alpha) = \frac{\sum_{i=1}^{n} \mathbf{x}_i K_\alpha(\mathbf{z}, \mathbf{z}_i)}{\sum_{i=1}^{n} K_\alpha(\mathbf{z}, \mathbf{z}_i)} \qquad \mathbf{c}_j = F(\Psi(j), \alpha), j = 1,\ldots,b$$

Decrease the neighborhood, and iterate.

NOTE: solution is (usually) *insensitive to poor initialization*

Example: one iteration of batch SOM

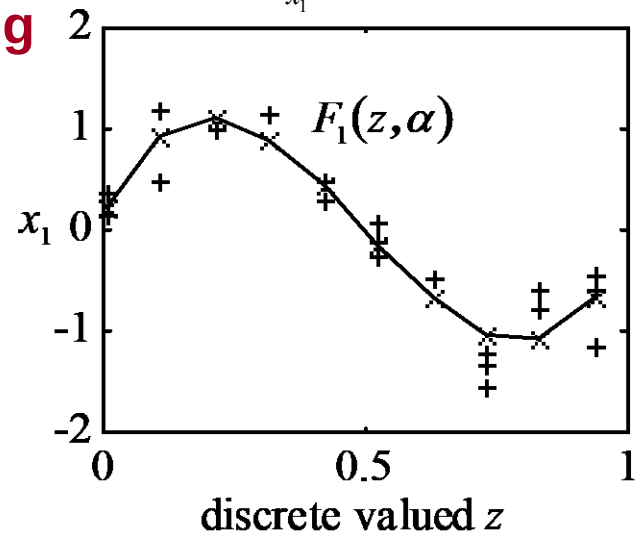
Projection step

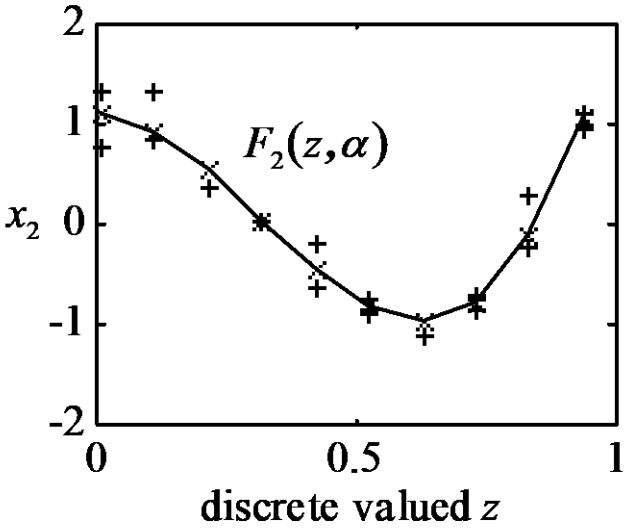


Discretization:

j	z
1	0.0
2	0.1
3	0.2
4	0.3
5	0.4
6	0.5
7	0.6
8	0.7
9	0.8
10	0.9

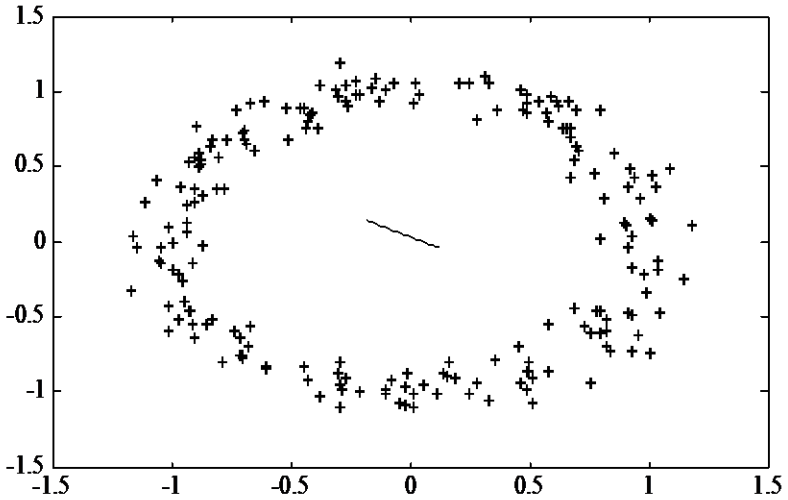
Smoothing



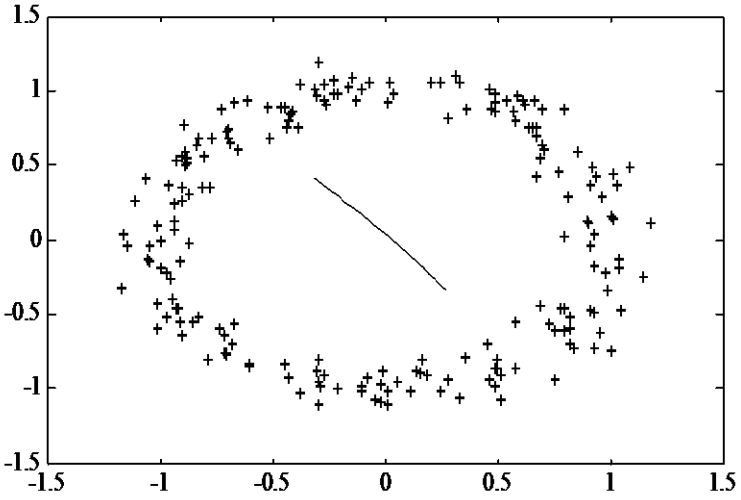


Example: effect of the final neighborhood width

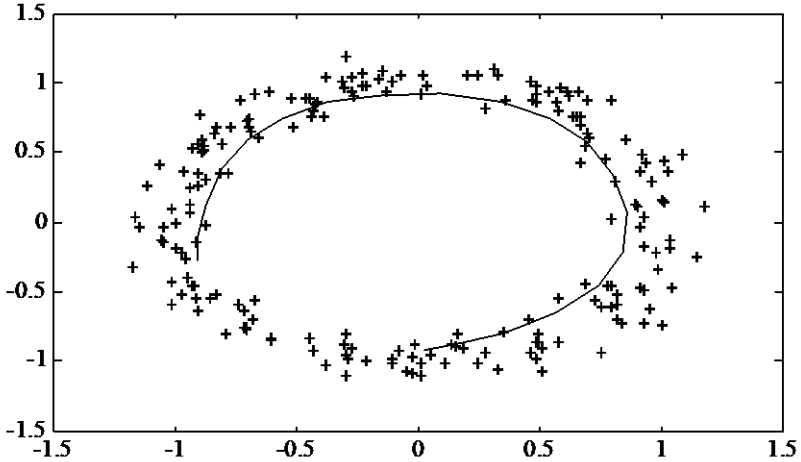
90%



50%



10%

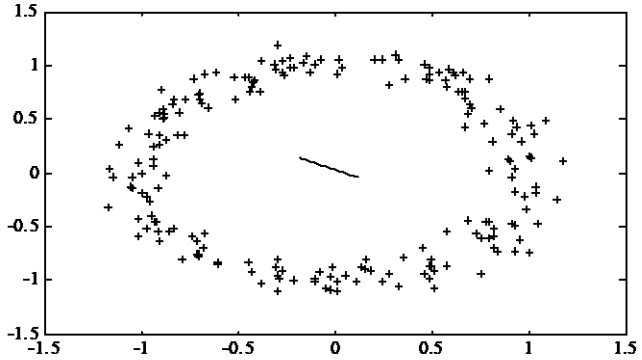


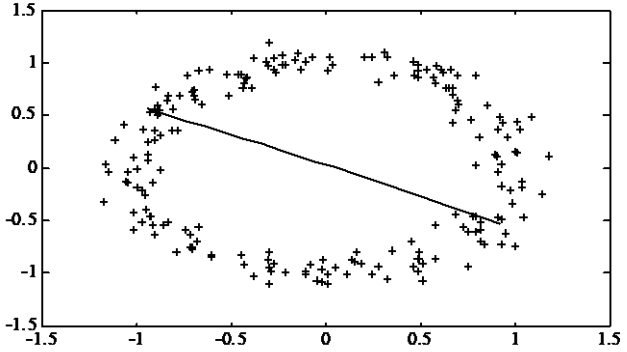
Statistical Interpretation of SOM

- New approach to dimensionality reduction: kernel smoothing in a map space
- Local averaging vs local linear smoothing.

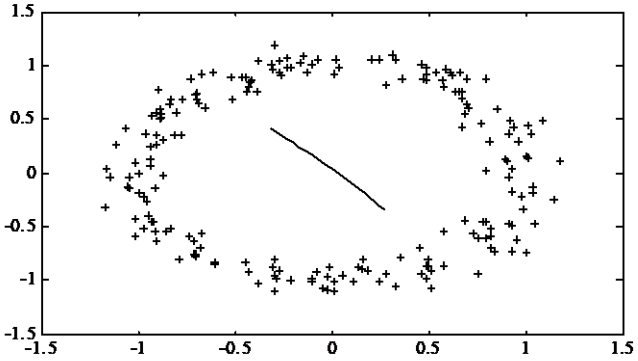
Local Average **Local Linear**

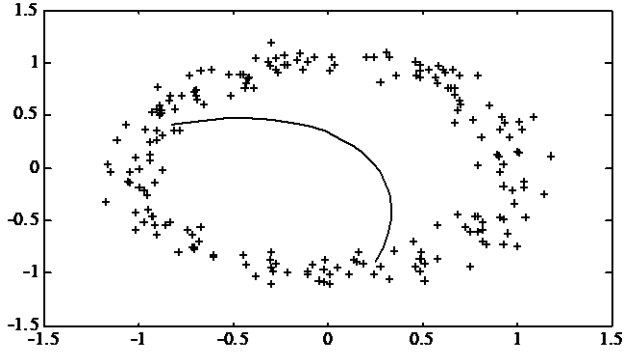
90%





50%





Practical Issues for SOM

- Pre-scaling of inputs, usually to [0, 1] range. Why? for distance measuremetn
- Map topology: usually 1D or 2D
- Number of map units (per dimension)
- Learning rate schedule (for on-line version)
- Neighborhood type and schedule:

  Initial size (~1), final size

  Final neighborhood size and the number of units determine model complexity.

SOM Similarity Ranking of US States

Each state ~ a multivariate sample of several socio-economic inputs for 2000:

- OBE obesity index (see Table 1)
- EL election results (EL=0~Democrat, =1 ~ Republican)-see Table 1
- MI median income (see Table 2)
- NAEP score ~ national assessment of educational progress
- IQ score

- Each input pre-scaled to (0,1) range
- Model using 1D SOM with 9 units

TABLE 1

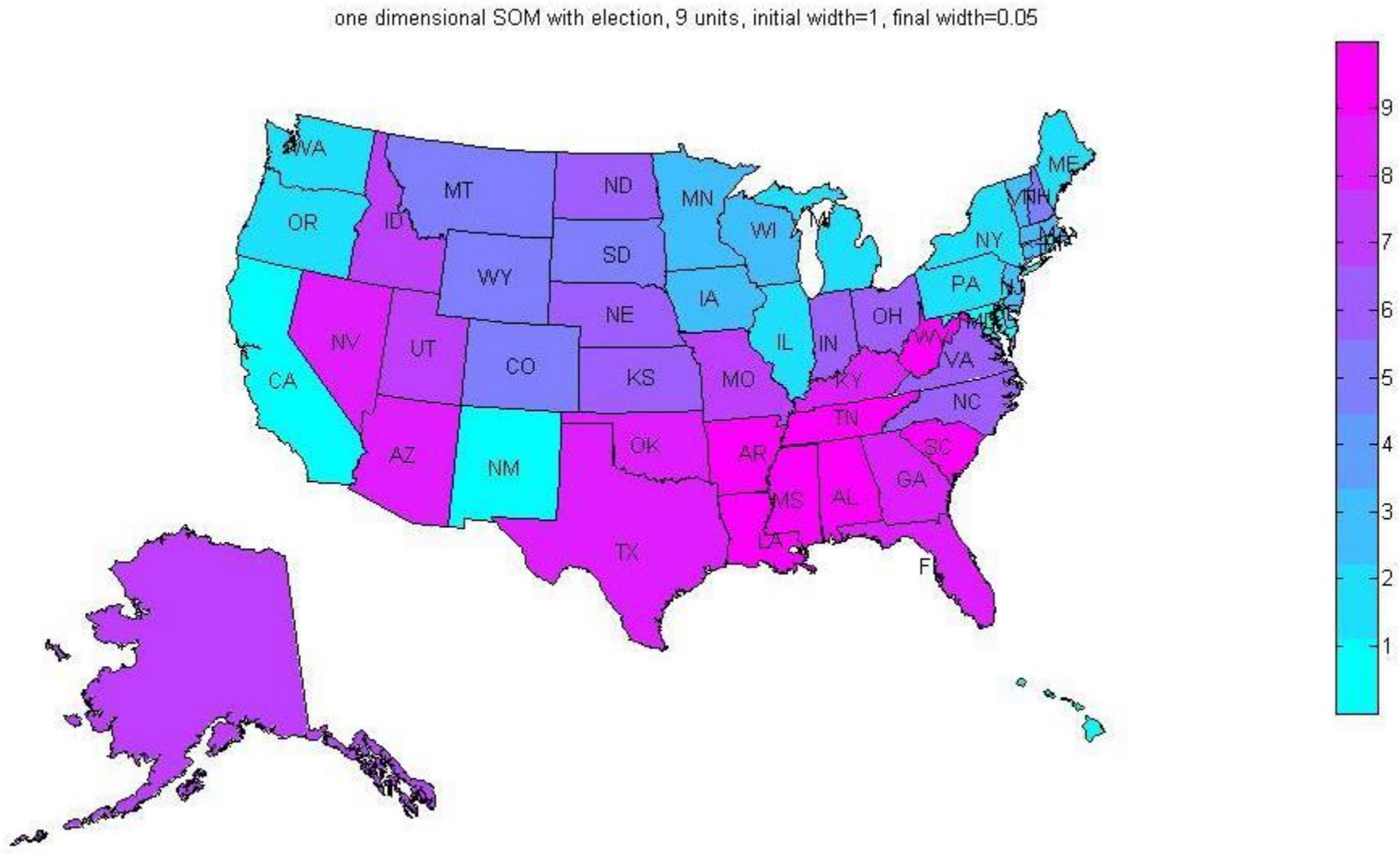
STATE	% Obese 2000	Election
Hawaii	17	D
Wisconsin	22	D
Colorado	17	R
Nevada	22	R
Connecticut	18	D
Alaska	23	R

……

TABLE 2

STATE	MI	NAEP	IQ
Massachusets	$50,587	257	111
New Hampshire	$53,549	257	102
Vermont	$41,929	256	103
Minnesota	$54,939	256	113
……..			

SOM Modeling Result 1 (by Feng Cai)



SOM Modeling Result 1

- Out of 9 units total:

  - units 1-3 ~ Democratic states

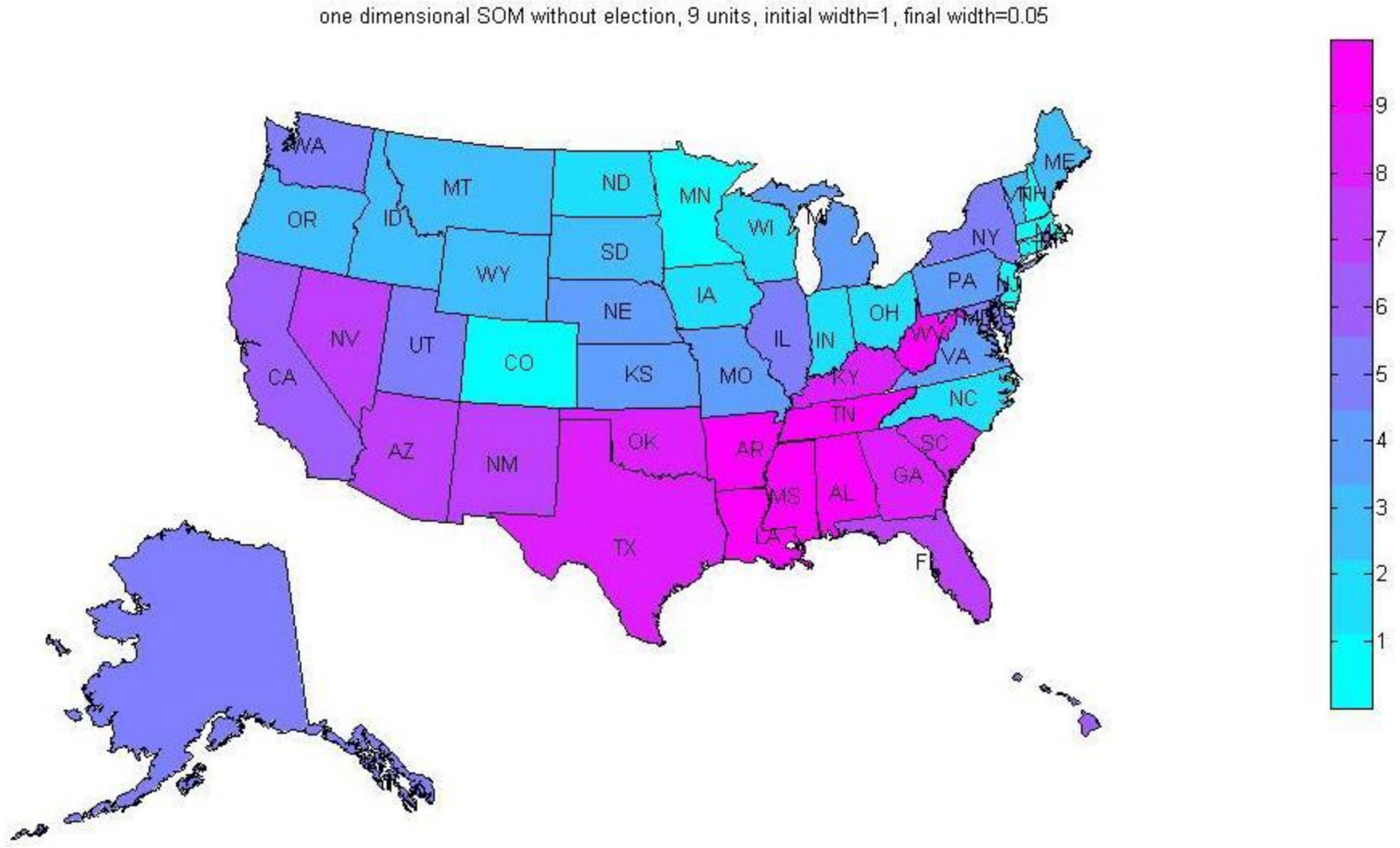
  - unit 4 – no states (?)

  - units 5-9 ~ Republican states

- Explanation:

  election input has two extreme values (0/1) and tends to dominate in distance calculation

SOM Modeling Result 2: no voting input



SOM Applications and Variations

Main web site: http://www.cis.hut.fi/research/som-research

Public domain SW

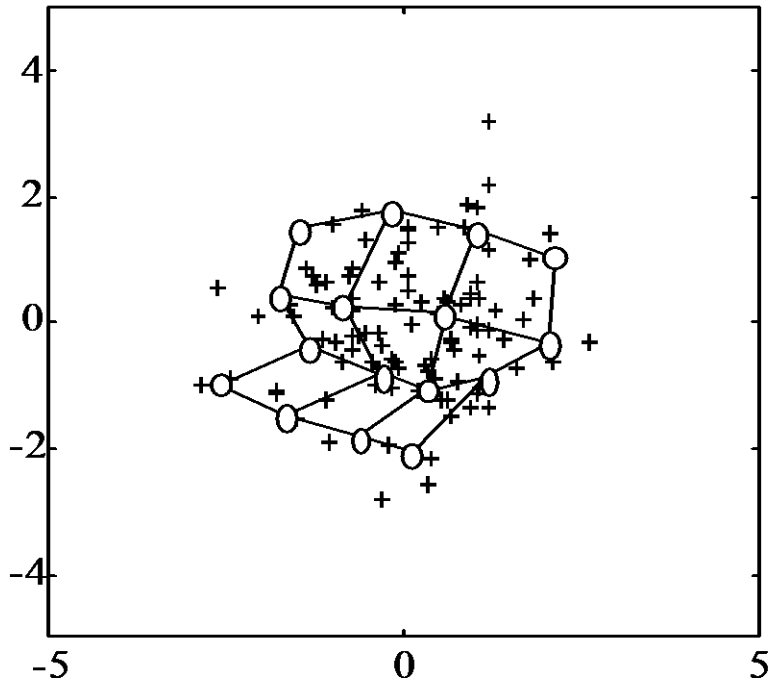
Numerous Applications

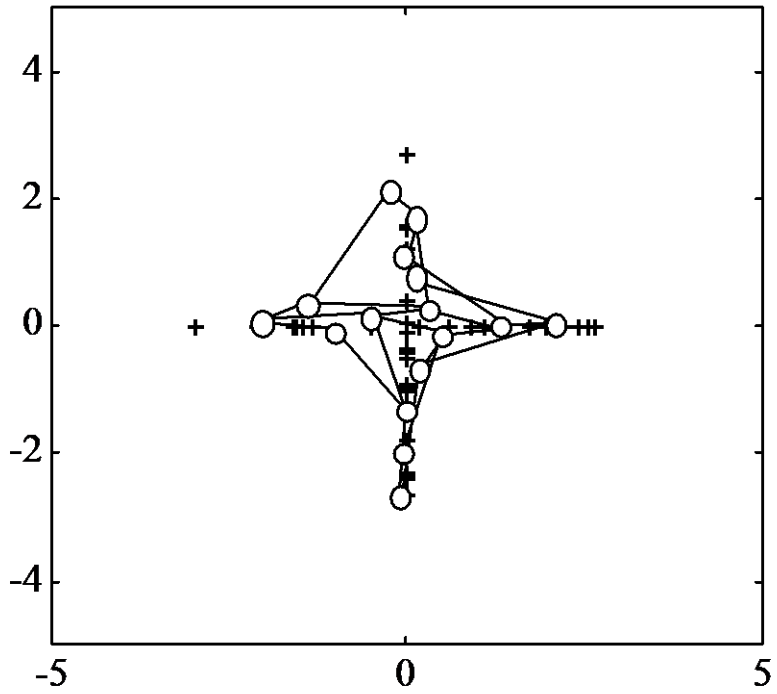
- Marketing surveys/ segmentation
- Financial/ stock market data
- Text data / document map – WEBSOM
- Image data / picture map - PicSOM

  see HUT web site
- Semantic maps ~ category formation
  http://link.springer.com/article/10.1007/BF00203171
- SOM for Traveling Salesman Problem

Tree-structured SOM

Fixed SOM topology provides poor modeling for structured distributions:

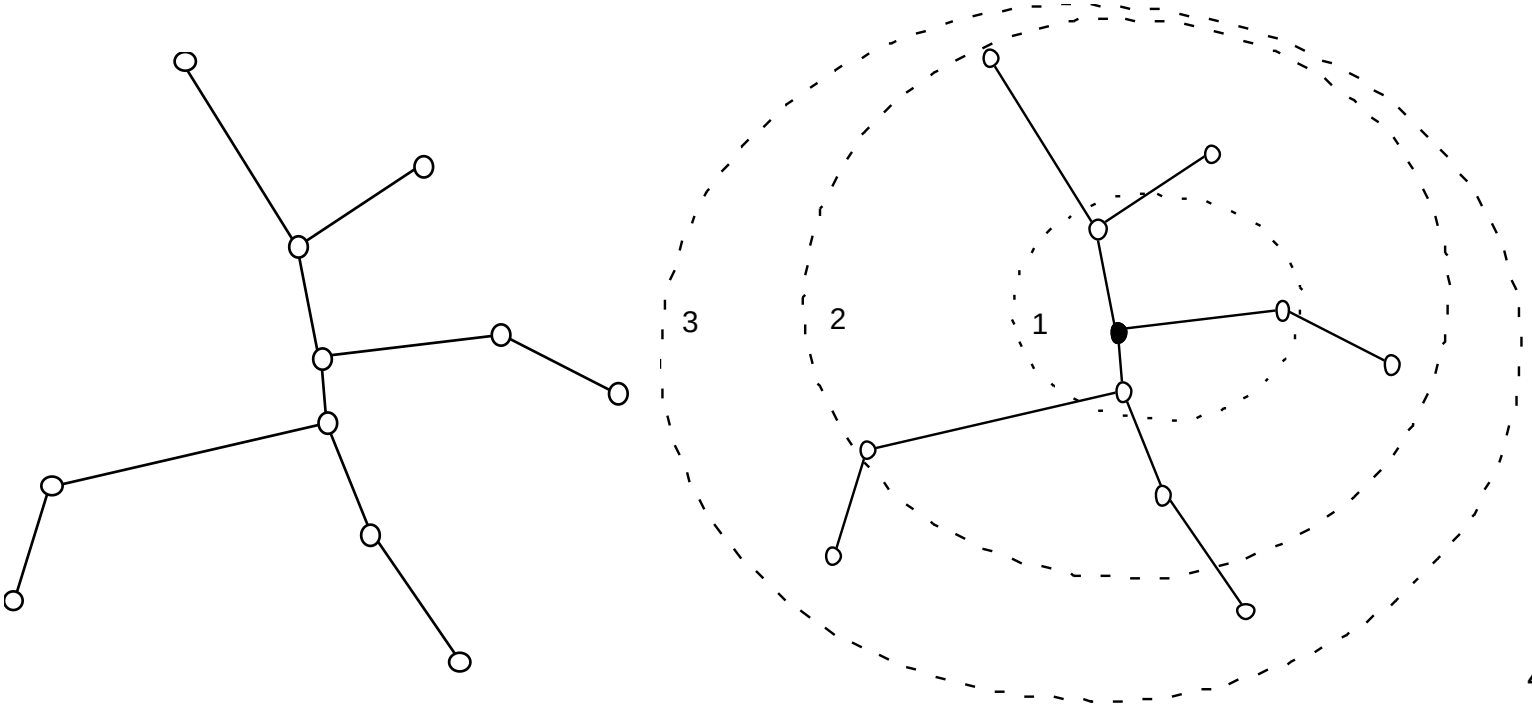




Minimum Spanning Tree SOM

- Define SOM topology adaptively during each iteration of SOM algorithm
- Minimum Spanning Tree (MST) topology ~ according to distance between units (in the input space)

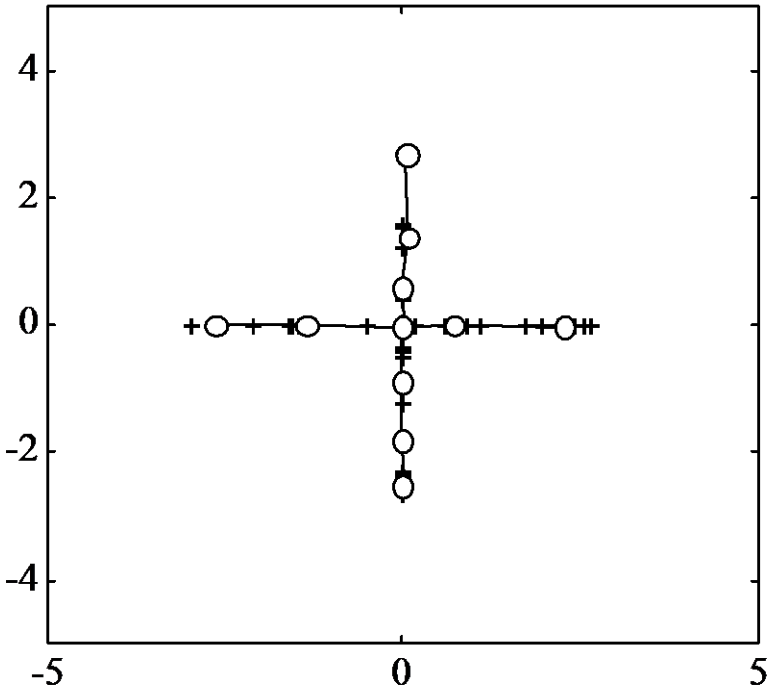
→ Topological distance ~ number of hops in MST

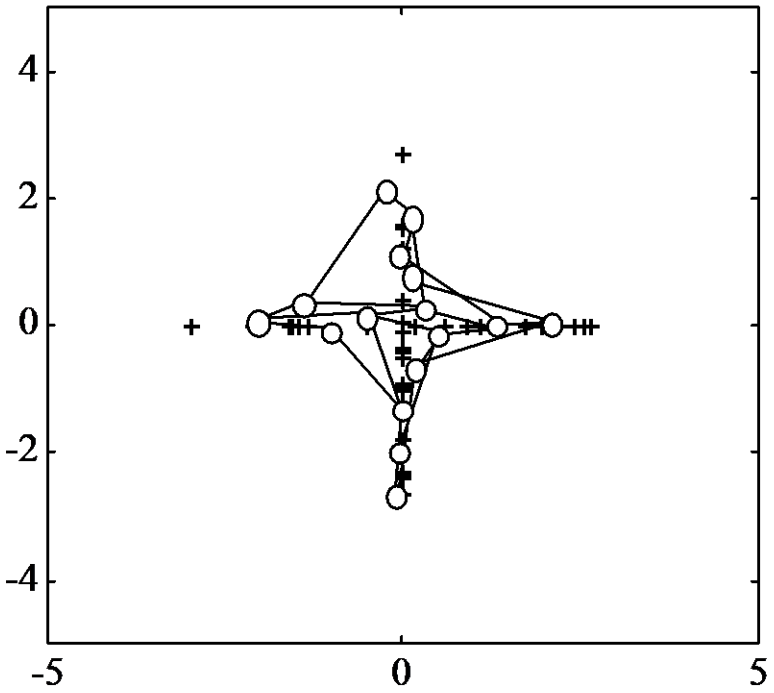


Example of using MST SOM

- Modeling cross distribution

  MST topology vs. fixed 2D grid map



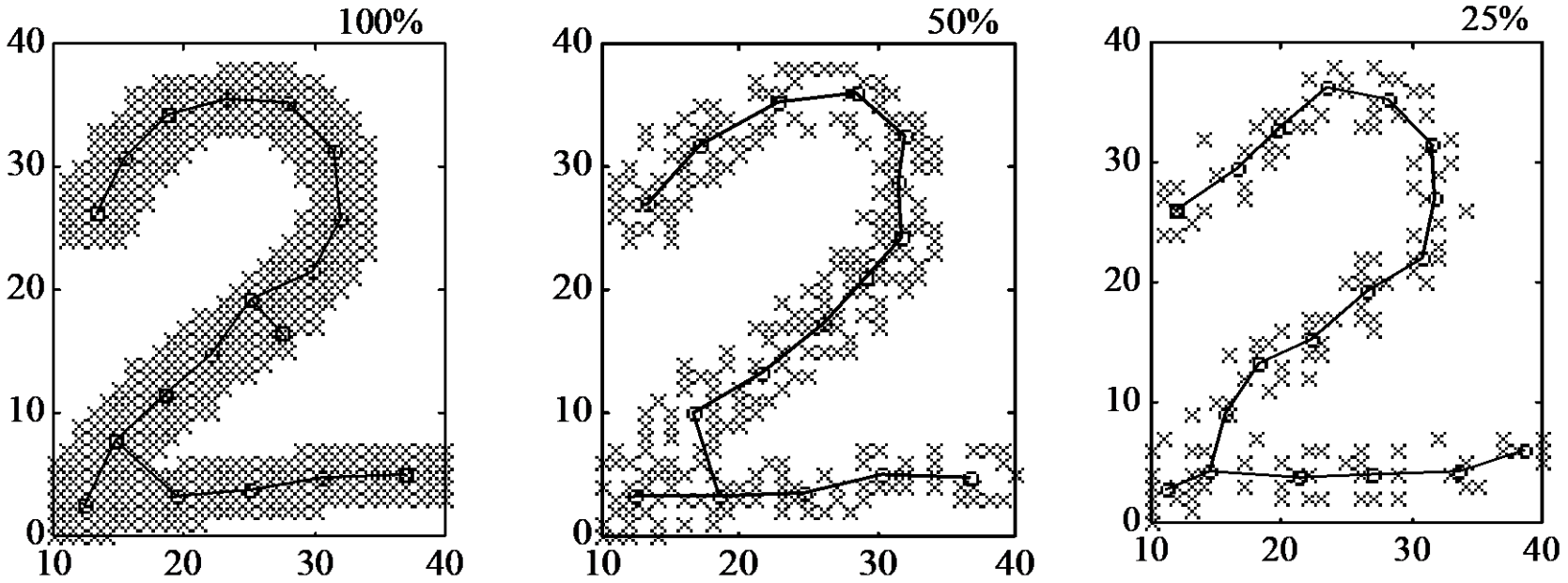


Application: skeletonization of images

Singh at al, Self-organizing maps for skeletonization of sparse shapes, IEEE TNN, 2000

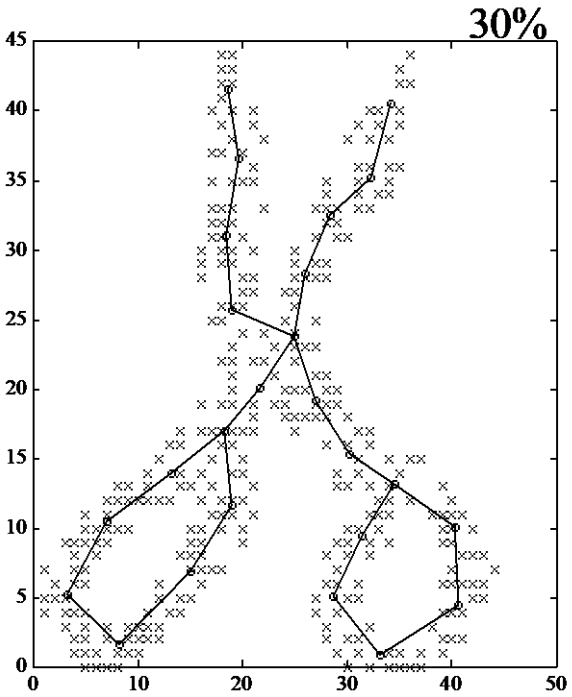
Skeletonization Problem:

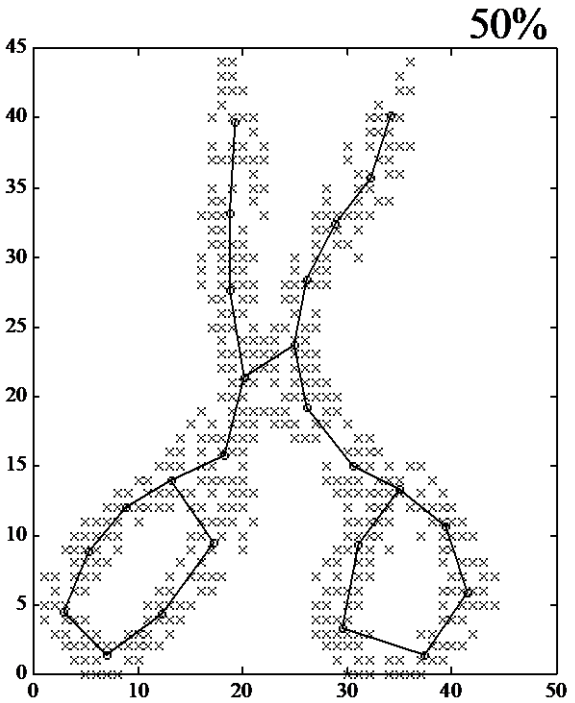
- Easy problem if image boundaries are known
- Skeletonization of noisy images is hard
- Application of MST SOM: robust with respect to noise



Modification of MST to represent loops

- Postprocessing: in the trained SOM identify a pair of units close in the input space but far in the map space (more than 3 hops apart). Connect these units.
- **Example:** Percentage indicates the proportion of data used for approximation from original image.





Clustering of European Languages

- Background historical linguistics studies relatedness btwn languages based on

  phonology, morphology, syntax and lexicon
- Difficulty of the problem: due to evolving nature of human languages and globalization.
- Hypothesis: similarity based on analysis of a small ‘stable’ word set.

See glottochronology, Swadesh list, at

http://en.wikipedia.org/wiki/Glottochronology

SOM for clustering European languages

Modeling approach: language ~ 10 word set.

Assuming words in different languages are encoded in *the same alphabet*, it is possible to perform clustering using some distance measure.

- Issues:

  selection of stable word set

  data encoding + distance metric

- Stable word set: numbers 1 to 10
- Data encoding: Latin alphabet, use 3 first letters (in each word)

Numbers word set in 18 European languages

Each language is a feature vector encoding 10 words

English	Norwegian	Polish	Czech	Slovakian	Flemish	Croatian	Portuguese	French	Spanish	Italian	Swedish	Danish	Finnish	Estonian	Dutch	German	Hungarian
one	en	jeden	jeden	jeden	ien	jedan	um	un	uno	uno	en	en	yksi	uks	een	erins	egy
two	to	dwa	dva	dva	twie	dva	dois	deux	dos	due	tva	to	kaksi	kaks	twee	zwei	ketto
three	tre	trzy	tri	tri	drie	tri	tres	trois	tres	tre	tre	tre	kolme	kolme	drie	drie	harom
four	fire	cztery	ctyri	styri	viere	cetiri	quarto	quatre	cuatro	quattro	fyra	fire	nelja	neli	vier	vier	negy
five	fem	piec	pet	pat	vuvve	pet	cinco	cinq	cinco	cinque	fem	fem	viisi	viis	vijf	funf	ot
six	seks	szesc	sest	sest	zesse	sest	seis	six	seis	sei	sex	seks	kuusi	kuus	zes	sechs	hat
seven	sju	sediem	sedm	sedem	zevne	sedam	sete	sept	siete	sette	sju	syv	seitseman	seitse	zeven	sieben	het
eight	atte	osiem	osm	osem	achte	osam	oito	huit	ocho	otto	atta	otte	kahdeksan	kaheksa	acht	acht	nyolc
nine	ni	dziewiec	devet	devat	negne	devet	nove	neuf	nueve	nove	nio	ni	yhdeksan	uheksa	negen	neun	kilenc
ten	ti	dziesiec	deset	desat	tiene	deset	dez	dix	dies	dieci	tio	ti	kymmenen	kumme	tien	zehn	tiz

Data Encoding

- Word ~ feature vector encoding 3 first letters
- Alphabet ~ 26 letters + 1 symbol ‘BLANK’

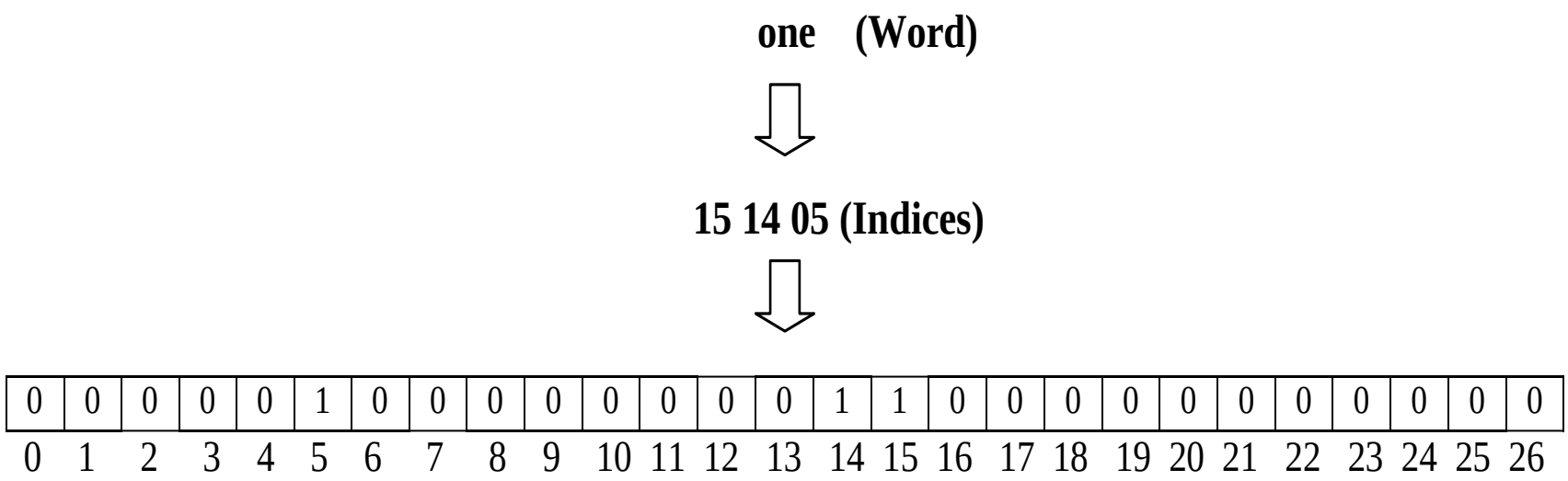
vector encoding:

ALPHABET	INDEX
‘BLANK’	00
A	01
B	02
C	03
D	04
…	…
X	24
Y	25
Z	26

For example, ONE : ‘O’~14 ‘N’~15 ‘E’~05

Word Encoding (cont'd)

- Word → 27-dimensional feature vector



- Encoding is insensitive to order (of 3 letters)
- Encoding of 10-word set: concatenate feature vectors of all words: 'one' + 'two' + …+ 'ten'

  → **word set** encoded as vector of dim. [1 X 270]

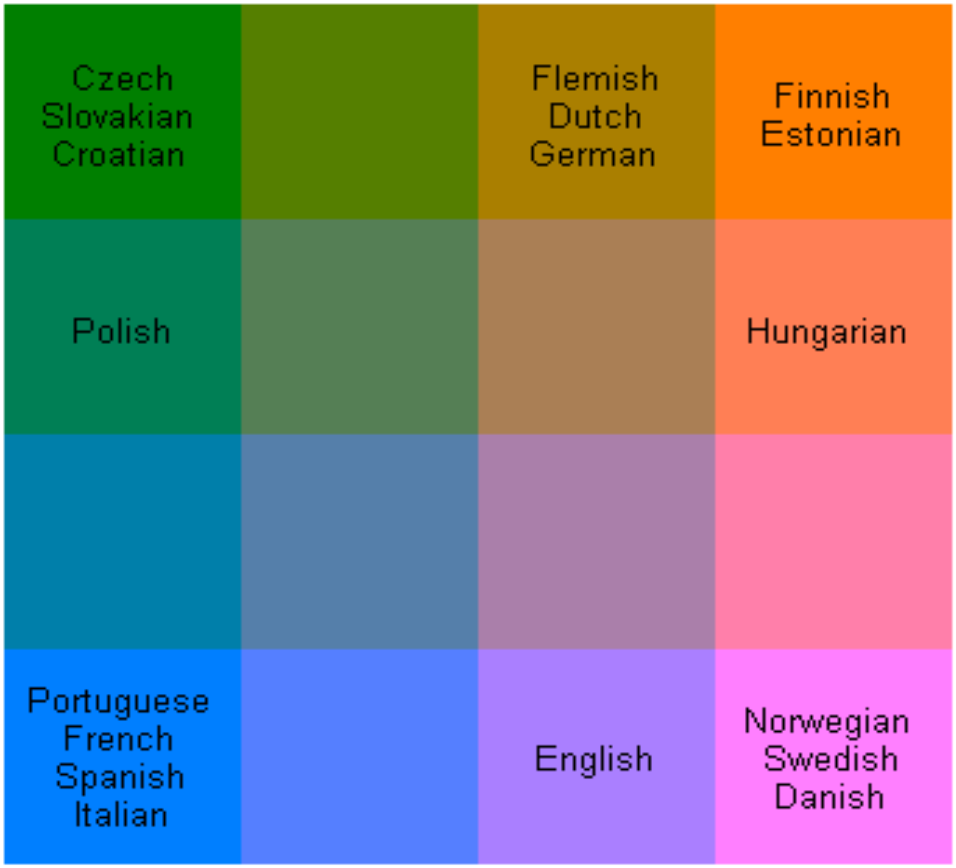
SOM Modeling Approach

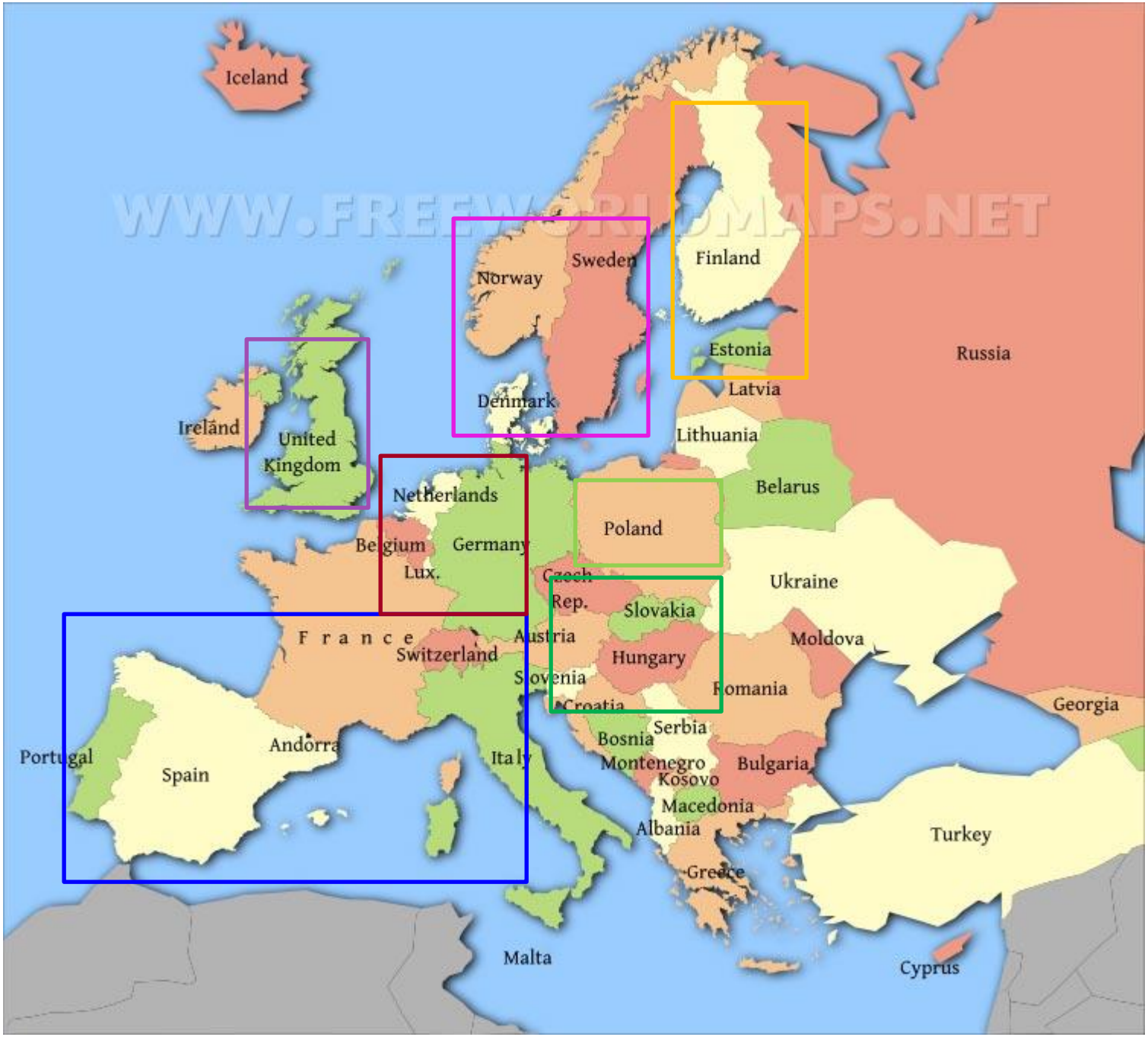
- 2-Dimensional SOM (Batch Algorithm)

  Number of Knots per dimension=4

  Initial Neighborhood =1 Final Neighborhood = 0.15

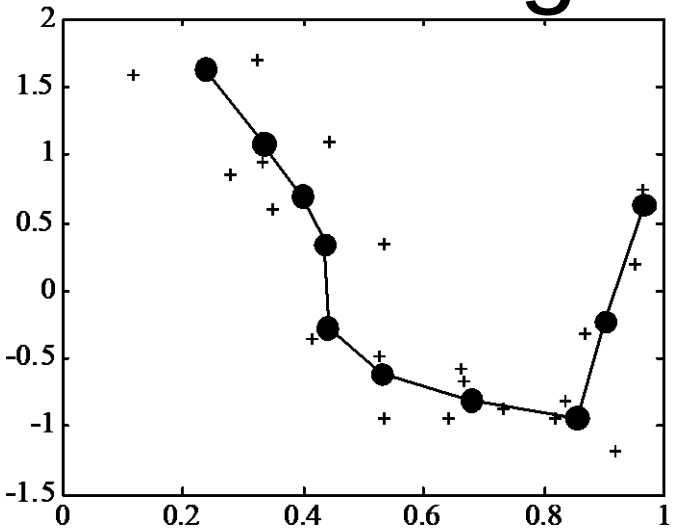
  Total Number of Iterations= 70




Iceland
Norway
Sweden
Finland
Russia
Estonia
Latvia
Lithuania
Belarus
Denmark
Ireland
United Kingdom
Netherlands
Belgium
Germany
Lux.
Poland
Czech Rep.
Slovakia
Ukraine
France
Switzerland
Austria
Hungary
Moldova
Slovenia
Romania
Croatia
Georgia
Serbia
Bosnia
Montenegro
Kosovo
Bulgaria
Portugal
Spain
Andorra
Italy
Macedonia
Albania
Turkey
Greece
Malta
Cyprus

SOM for Supervised Learning

How to apply SOM
to regression problem?



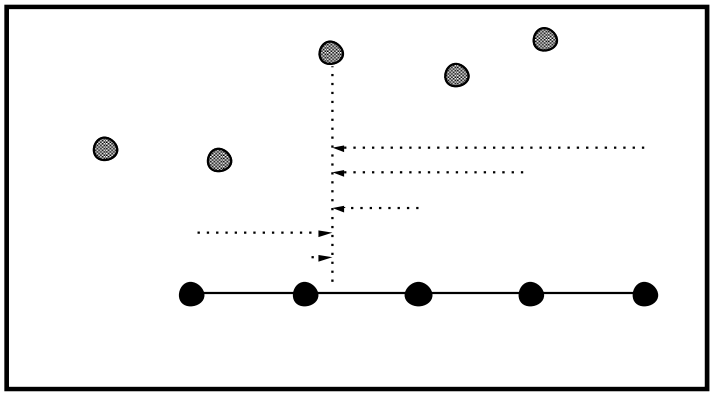
CTM Approach: (Constrained Topological Mapping)

Given training data (**x**,y) perform

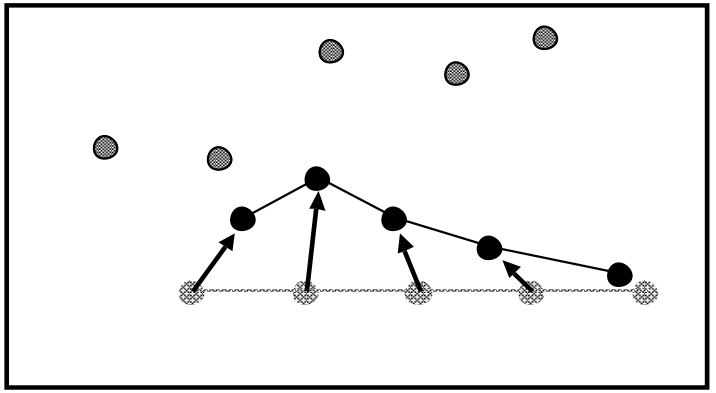
1. Dimensionality reduction **x**→**z**
   (Apply SOM to **x**-values of training data)
2. Apply kernel regression to estimate
   y=*f*(z) at discrete points in z-space

Constrained Topological Mapping: search for nearest unit (winning unit) in **x**-space

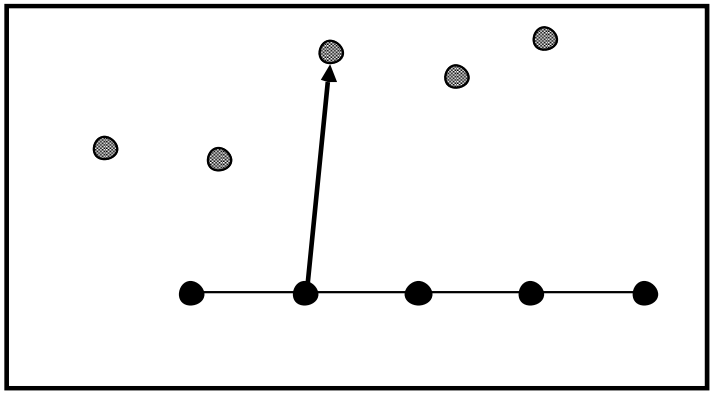
CTM Algorithm



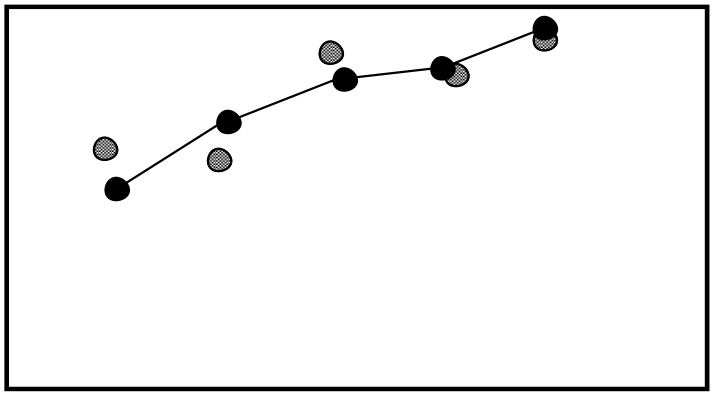
Find Winning Unit



Move Neighborhood



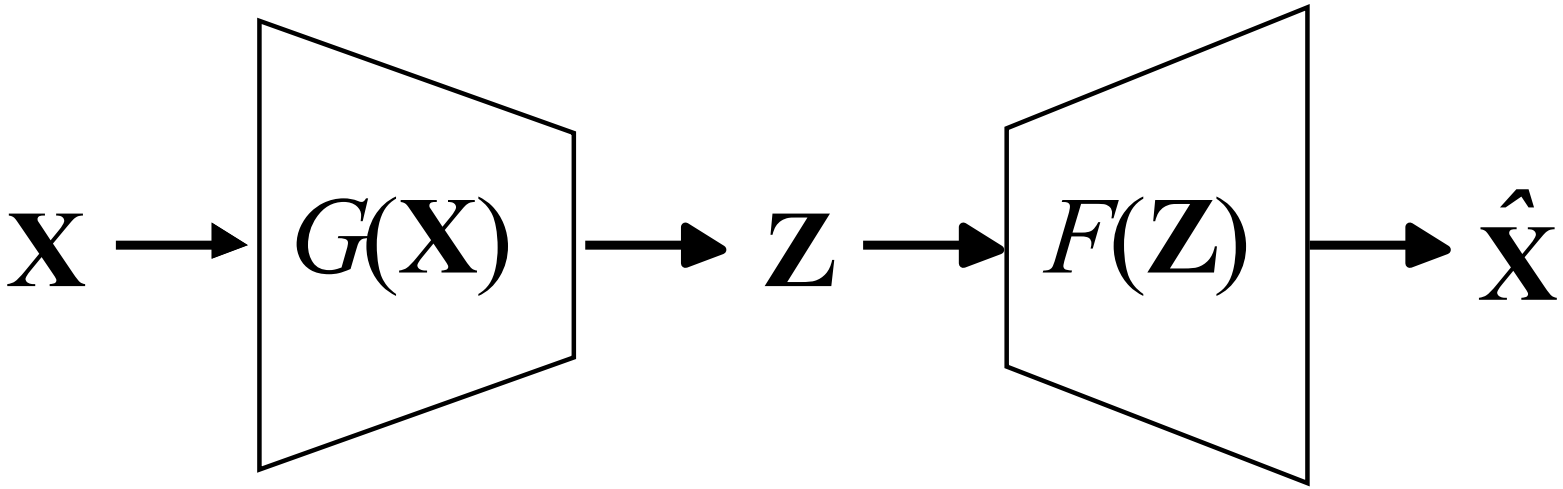
Winning Unit Found



After Many Iterations

MLP for data compression

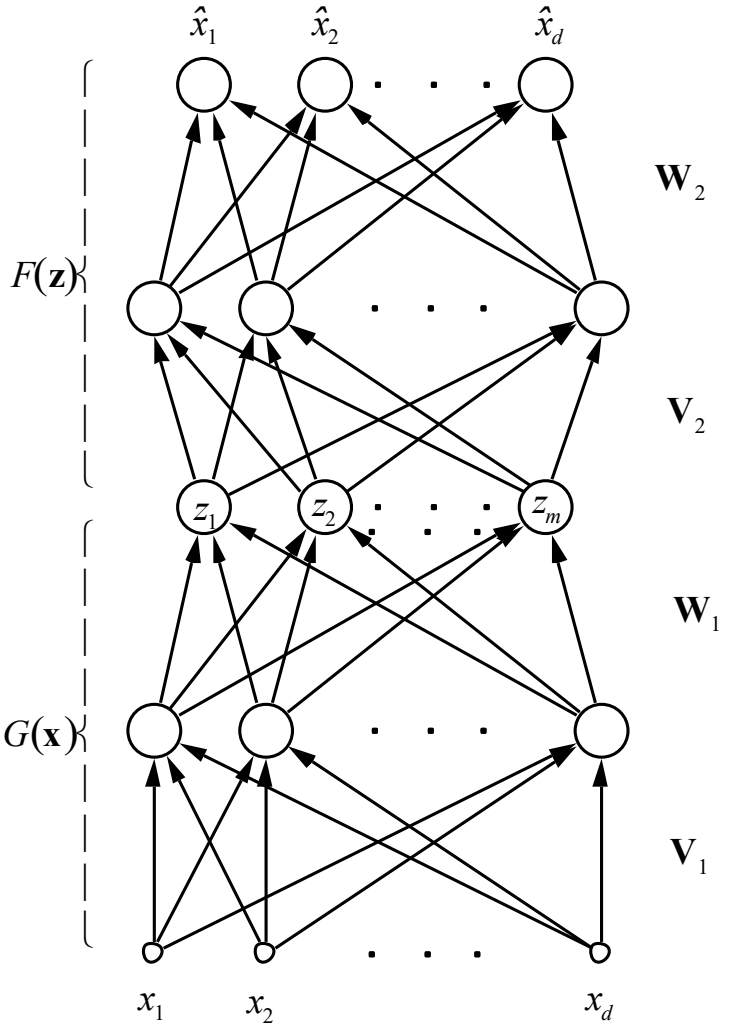
- Recall general setting for data compression and dimensionality reduction
- How to implement it via MLP?
- Can we use single hidden layer MLP?



- Need multiple hidden layers to implement nonlinear dimensionality reduction:

  both *F* and *G* are nonlinear

- Many problems (with implementation)

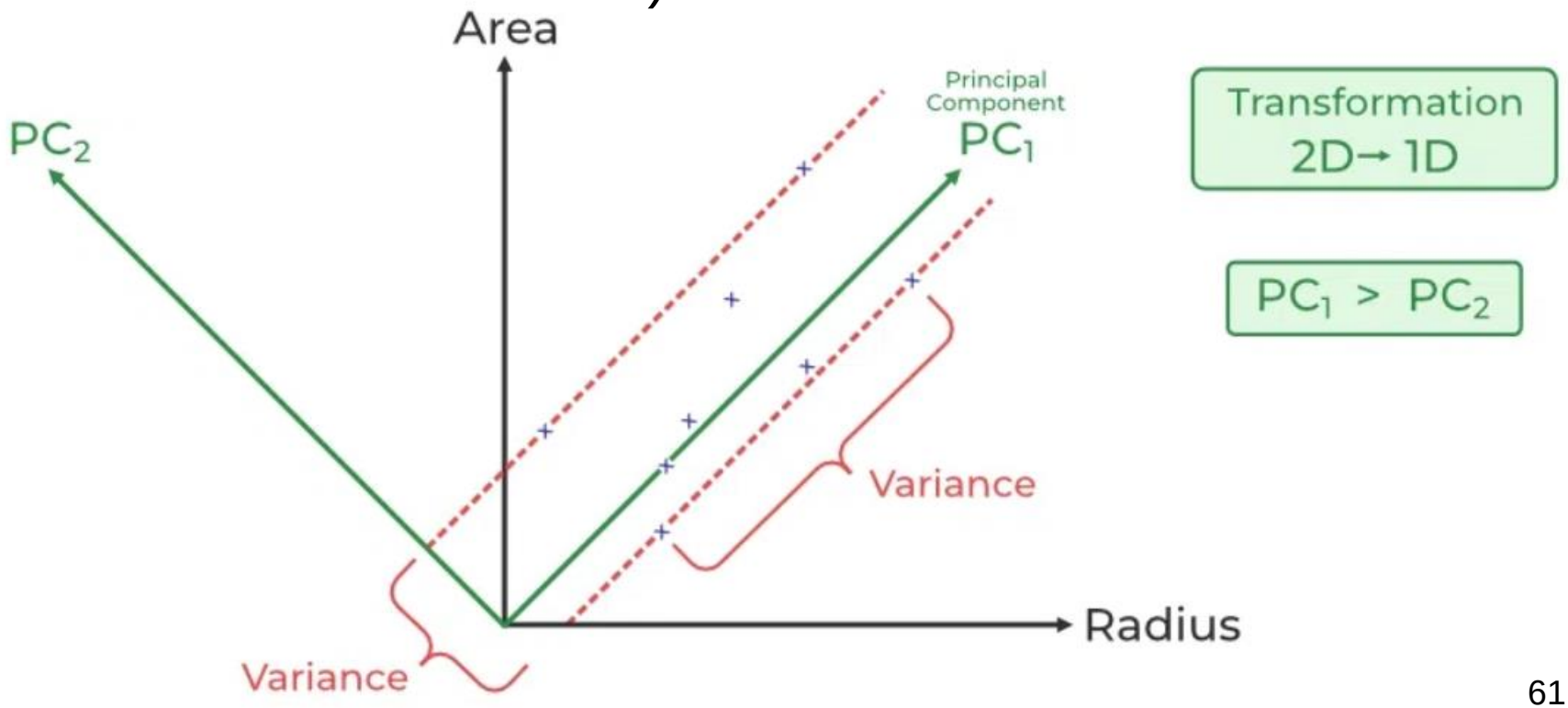


OUTLINE

- Motivation for unsupervised learning
- Vector quantization and unsupervised learning
- **Statistical methods for dimensionality reduction**
  - **Principal components (PCA)**
  - **Principal curves**
  - **Multidimensional scaling (MDS)**
- Methods for multivariate data analysis
- Summary and discussion

Principal Component Analysis(PCA)

- PCA is used to reduce the dimensionality of a data set by finding a new set of variables.
- Smaller than the original set of variables
- Retaining most of the sample's information (capturing the maximum variance).



https://www.geeksforgeeks.org/principal-component-analysis-pca/

Step-By-Step Explanation of PCA

- Step 1: Standardization

$$z = \frac{x - \mu}{\sigma}$$

- Step2: Covariance Matrix Computation

$$cov(x_1, x_2) = \frac{\sum_{i=1}^{n}(x_1 - \bar{x}_1)(x_2 - \bar{x}_2)}{n - 1}$$

- Step 3: Compute Eigenvalues and Eigenvectors of Covariance Matrix to Identify Principal Components

$$(A - \lambda I)X = 0$$

https://www.geeksforgeeks.org/principal-component-analysis-pca/

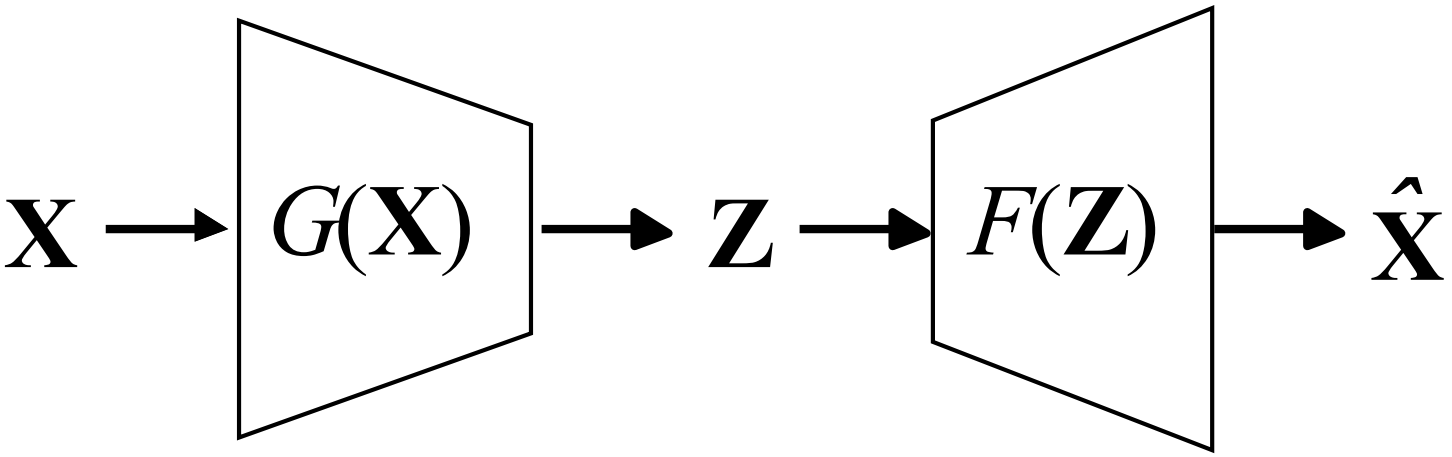
Step-By-Step Explanation of PCA

- Step 4: Sort the eigenvalues in descending order and sort the corresponding eigenvectors accordingly.
- Step 5: Project the Data onto the Selected Principal Components
- Find the projection matrix, It is a matrix of eigenvectors corresponding to the largest eigenvalues of the covariance matrix of the data.

Dimensionality reduction

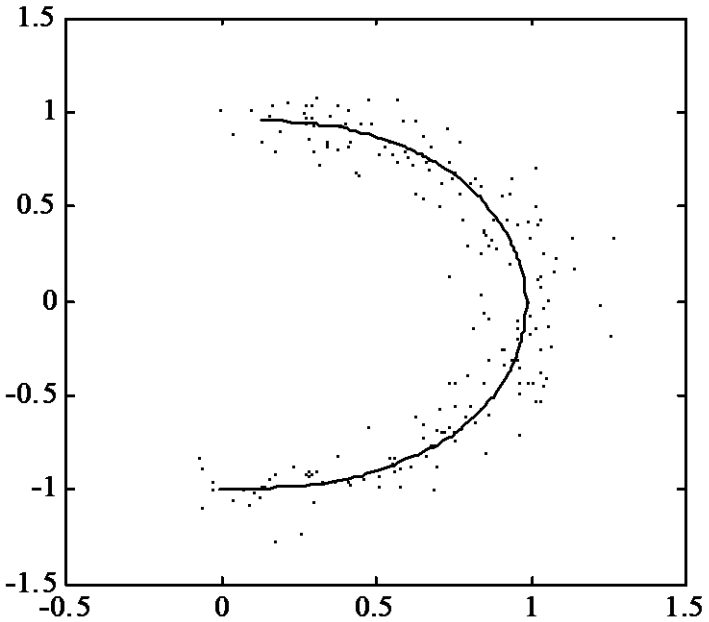
- **Recall dimensionality reduction** ~ estimation of two mappings G(**x**) and F(**z**)
- The goal of learning is to find a mapping $\boldsymbol{f}(\mathbf{x},\omega)=F(G(\mathbf{x}))$ minimizing prediction risk $R(\omega)=\int L(\mathbf{x},\boldsymbol{f}(\mathbf{x},\omega))\boldsymbol{p}(\mathbf{x})d\mathbf{x}$

- **Two approaches:** linear G(x) or nonlinear G(x)



Principal Curves

- **Principal Curve:** Generalization of the first linear PC i.e., the curve that passes through the 'middle' of the data



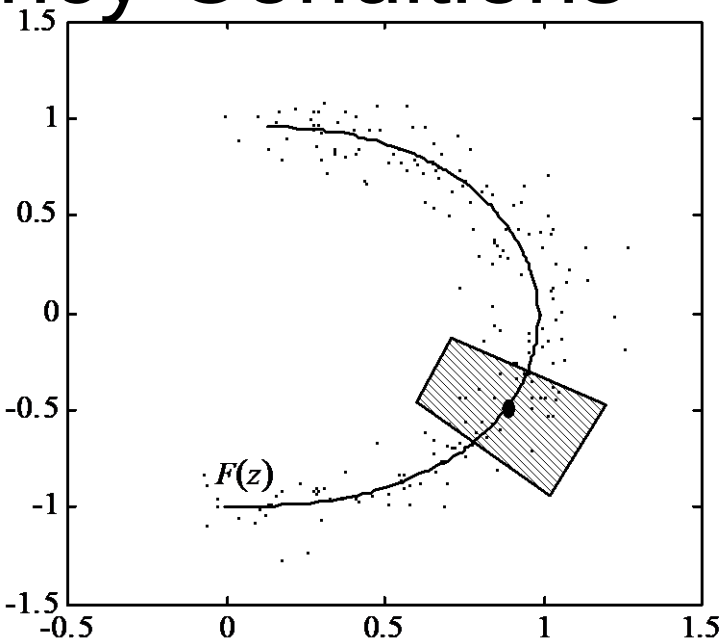
- The Principal Curve (manifold) is a vector-valued function $F(\mathbf{z},\omega)$ that minimizes the empirical risk

$$R_{emp} = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{x}_i - F\left(G(\mathbf{x}_i),\omega\right)\right\|^2$$

Subject to smoothness constraint on $F(\mathbf{z},\omega)$

Self Consistency Conditions

The point of the curve ~ the mean of all points that 'project' on the curve



Necessary Conditions for Optimality

- Encoder mapping

$$G(\mathbf{x}) = \arg\min_{\mathbf{z}} \left\| F(\mathbf{z}) - \mathbf{x} \right\|^2$$

- Decoder mapping (Smoothing/ conditional expectation)

$$F(\mathbf{z}) = E(\mathbf{x} / \mathbf{z})$$

Algorithm for estimating PC (manifold)

Given data points $\mathbf{x}_i$, distance metric, and initial estimate of d-valued function $\hat{F}(\mathbf{z})$

Iterate the following two steps (until empirical risk < THR)

Projection for each data point, find its closest projected point on the curve (manifold)

$$\hat{\mathbf{z}}_i = \arg\min_{\mathbf{z}} \left\| \hat{F}(\mathbf{z}) - \mathbf{x}_i \right\|$$

Conditional expectation = **kernel smoothing**

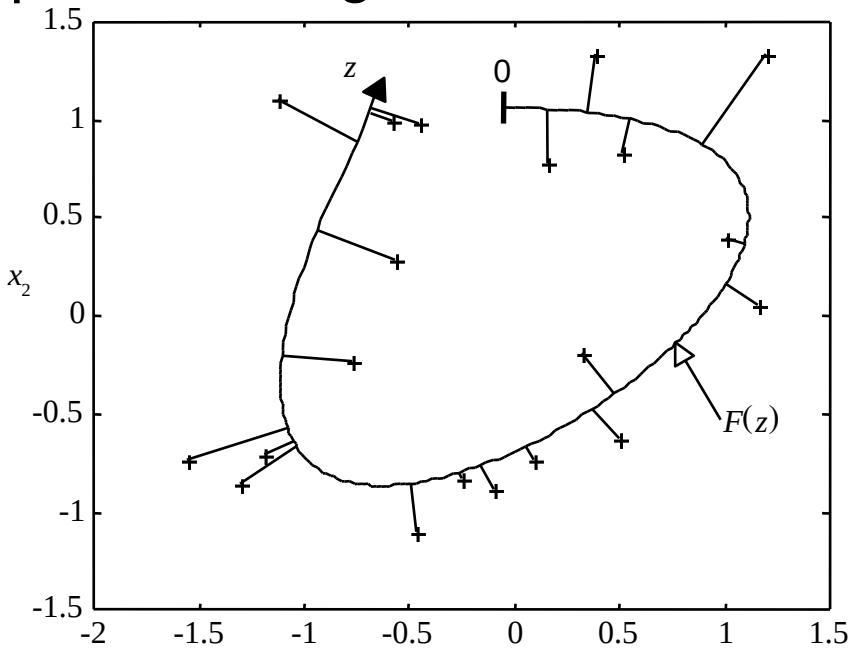
~ use $\{\hat{\mathbf{z}}_i, \mathbf{x}_i\}$ as training data for multiple-output regression problem. The resulting estimates $\hat{F}_j(\mathbf{z})$ are components of the d-dimensional function describing principal curve

Increase flexibility: decrease the smoothing parameter of the regression estimator

Example: one iteration of principal curve algorithm

20 samples generated according to

$$\mathbf{x} = \left[\cos(2\pi z), \sin(2\pi z)\right] + \varepsilon$$

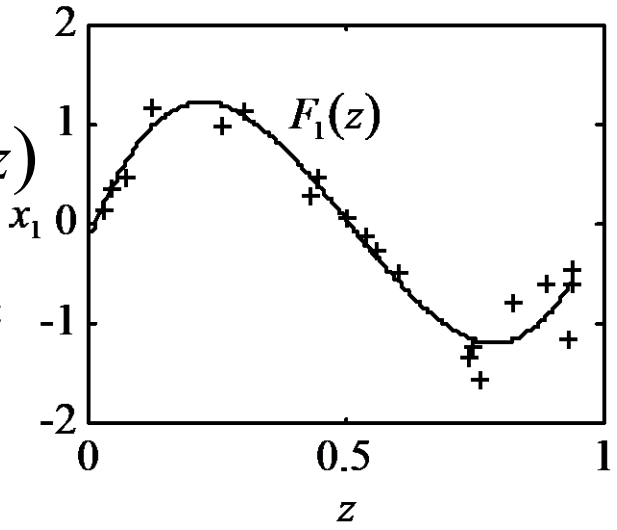


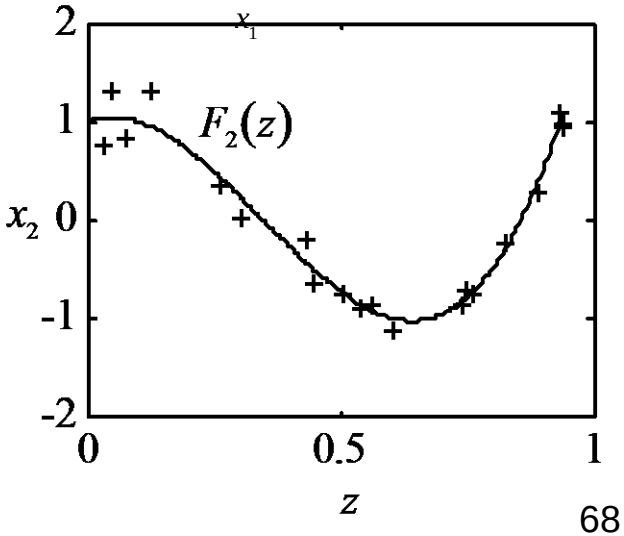
Projection step

Data points projected on the curve to estimate $z = G(x_1, x_2)$

Smoothing step

Estimates $F_1(z)\, F_2(z)$ describe principal curve in a parametric form





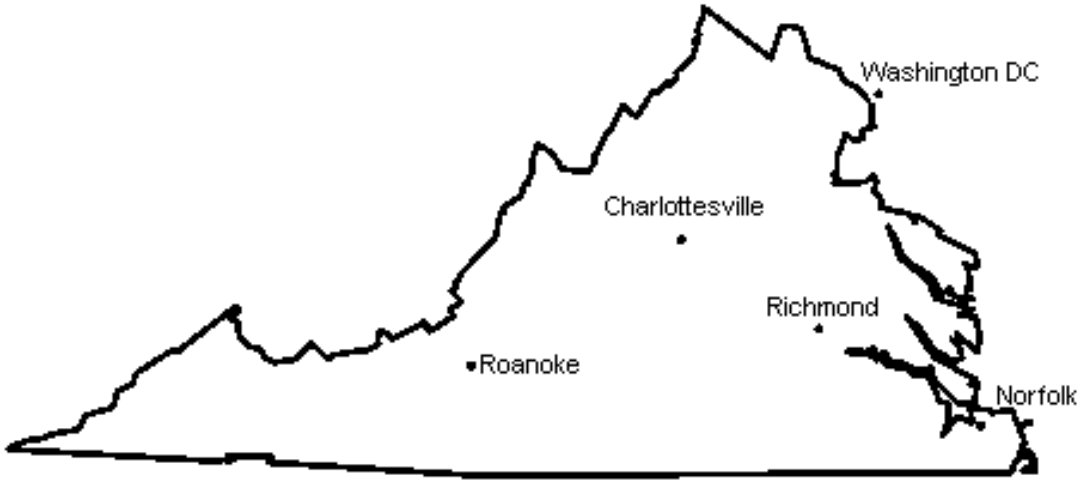
Multidimensional Scaling (MDS)

- Mapping *n* input samples onto a set of points $\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_n]$ in a low-dim. space that preserves the interpoint distances δ_{ij} of inputs $\mathbf{x}_i$ and $\mathbf{x}_j$
- MDS minimizes **the stress function**

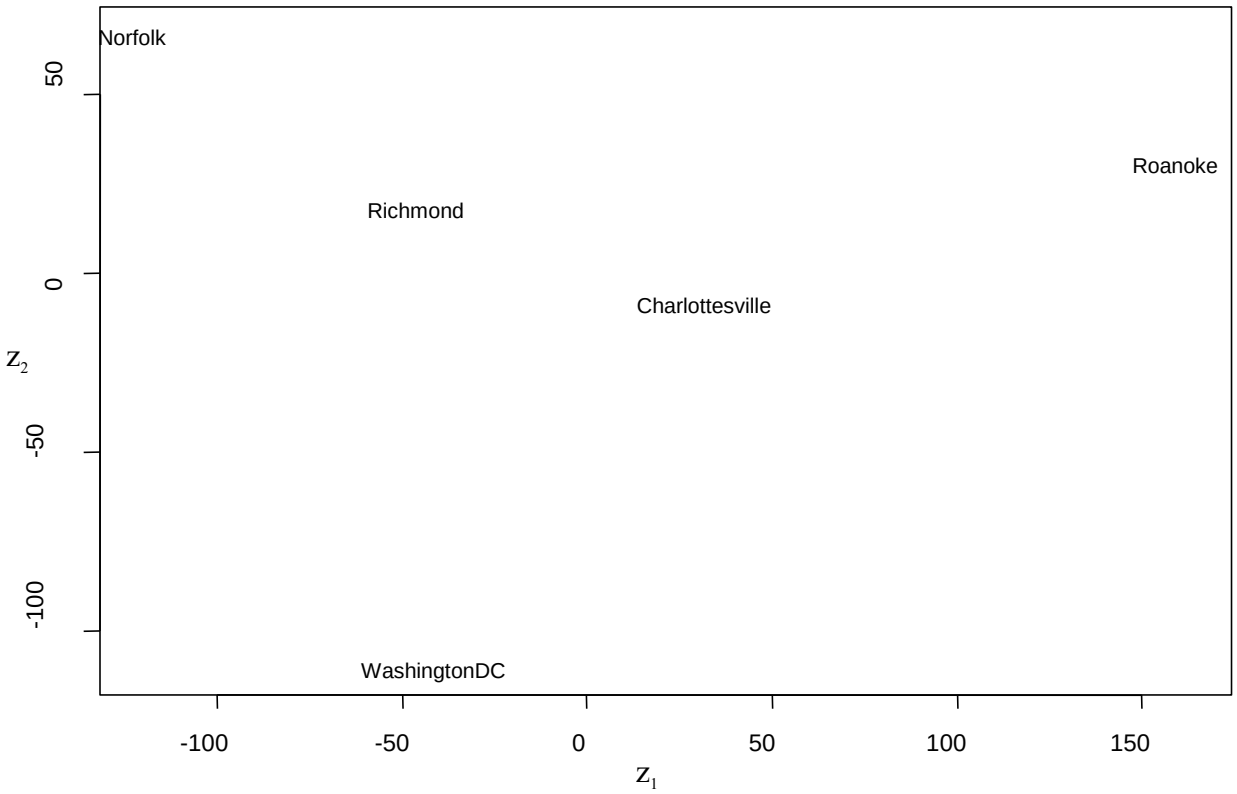
$$S(\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n) = \sqrt{\sum_{i \neq j} \left(\delta_{ij} - \|\mathbf{z}_i - \mathbf{z}_j\|\right)^2}$$

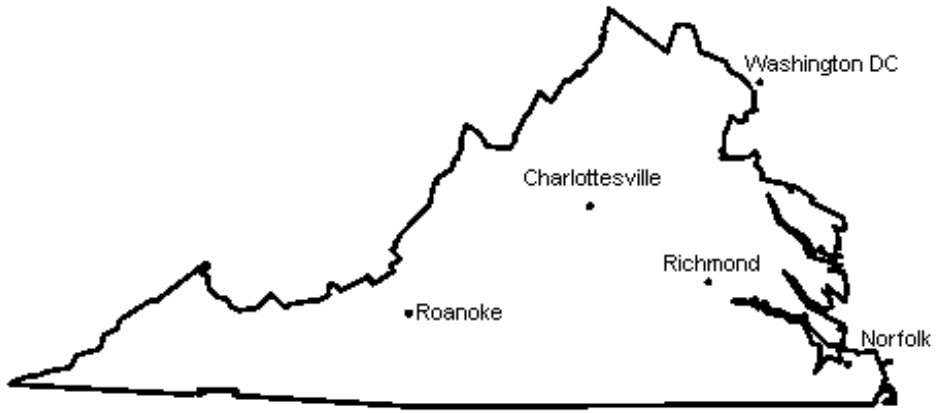
Note: MDS uses only interpoint distances. Therefore, it is applicable in situations where the input coordinate locations are not available.

Example



Traveling distance	Washington DC	Charlottesville	Norfolk	Richmond	Roanoke
WashingtonDC	0	118	196	108	245
Charlottesville	118	0	164	71	123
Norfolk	196	164	0	24	285
Richmond	108	71	24	0	192
Roanoke	245	123	285	192	0


Norfolk
Roanoke
Richmond
Charlottesville
WashingtonDC
50
0
-50
-100
Z_2
-100
-50
0
50
100
150
Z_1


Washington DC
Charlottesville
Richmond
Roanoke
Norfolk

MDS for clustering European languages

Modeling approach: language ~ 10 word set.

Assuming words in different languages are encoded in *the same alphabet*, it is possible to perform clustering using some distance measure.

- Issues:

  selection of stable word set

  data encoding + distance metric

- Stable word set: numbers 1 to 10
- Data encoding: Latin alphabet, use 3 first letters (in each word) – the same as was used for SOM → 270-dimensional vector

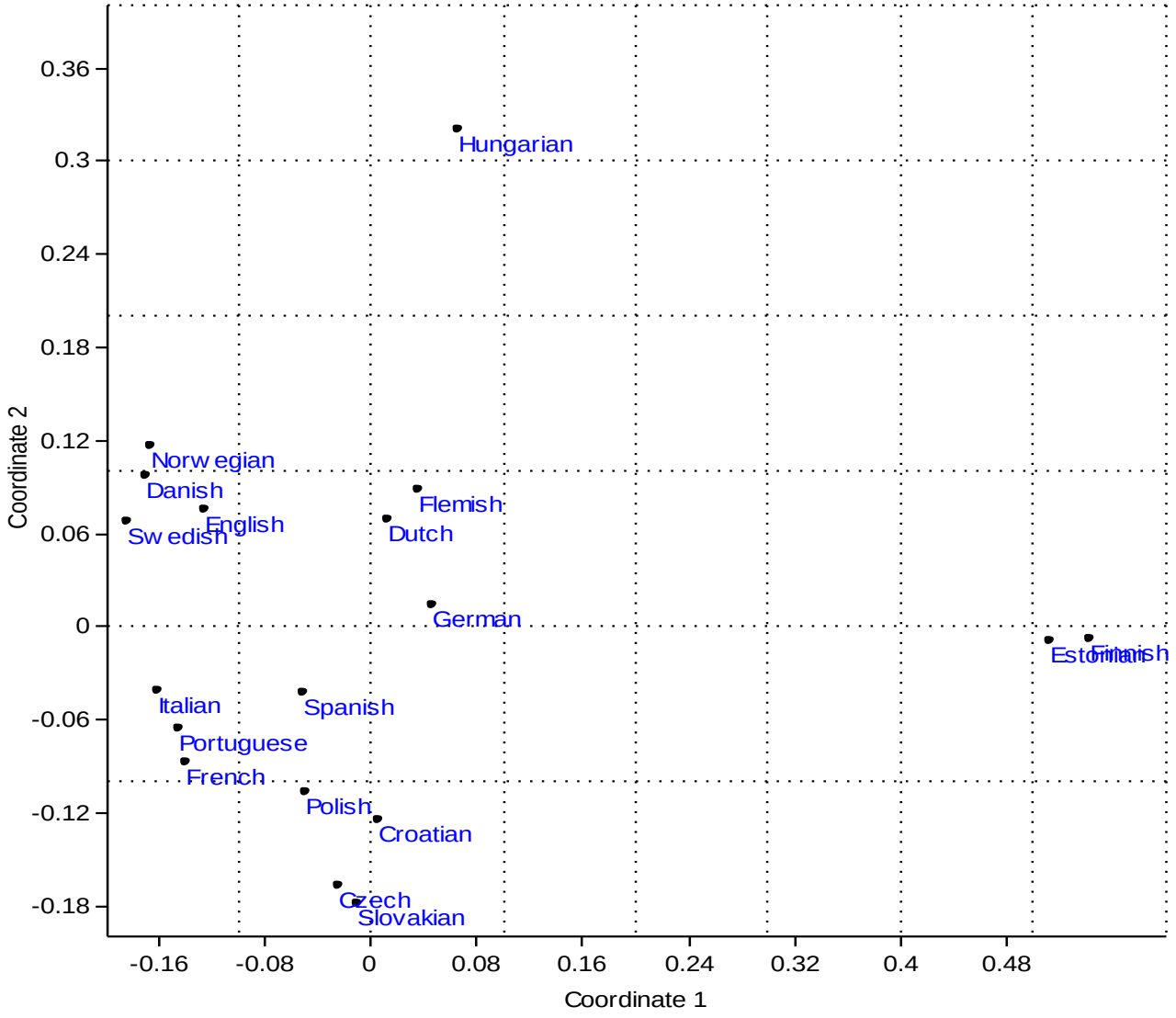
MDS Modeling Approach:

- calculate interpoint distances (Euclidean) btwn feature vectors
- map data points onto 2D space using software package Past

http://folk.uio.no/ohammer/past/

	English	Norwegian	Polish	Czech	Slovakian	Flemish	Croatian	Portuguese	French	Spanish	Italian	Swedish	Danish	Finnish	Estonian	Dutch	German	Hungarian
English	0.00	6.00	6.63	7.07	6.93	6.32	7.07	6.00	6.16	5.83	6.16	6.00	5.83	7.62	7.62	6.00	6.63	7.07
Norwegian	6.00	0.00	6.63	6.93	7.07	6.48	6.78	6.32	6.78	6.48	6.16	3.46	2.45	7.87	7.87	6.32	6.63	7.07
Polish	6.63	6.63	0.00	4.69	4.90	6.78	4.47	6.00	6.32	6.00	6.32	6.48	6.48	7.35	7.48	6.48	6.63	6.78
Czech	7.07	6.93	4.69	0.00	3.16	6.78	2.45	6.16	6.63	6.00	6.32	6.63	6.78	7.75	7.75	6.63	6.48	7.21
Slovakian	6.93	7.07	4.90	3.16	0.00	6.78	3.16	6.32	6.63	6.16	6.48	6.78	6.93	7.62	7.75	6.63	6.48	7.21
Flemish	6.32	6.48	6.78	6.78	6.78	0.00	6.48	6.93	6.78	6.32	6.93	6.63	6.48	7.48	7.35	3.16	4.90	6.93
Croatian	7.07	6.78	4.47	2.45	3.16	6.48	0.00	6.16	6.63	6.00	6.32	6.63	6.78	7.48	7.48	6.32	6.16	7.07
Portuguese	6.00	6.32	6.00	6.16	6.32	6.93	6.16	0.00	4.90	4.47	3.46	6.32	6.16	7.62	7.48	6.93	6.48	7.07
French	6.16	6.78	6.32	6.63	6.63	6.78	6.63	4.90	0.00	4.47	4.69	6.63	6.78	7.35	6.93	6.48	6.32	7.21
Spanish	5.83	6.48	6.00	6.00	6.16	6.32	6.00	4.47	4.47	0.00	4.24	6.63	6.32	7.35	6.93	6.16	5.83	7.21
Italian	6.16	6.16	6.32	6.32	6.48	6.93	6.32	3.46	4.69	4.24	0.00	6.00	6.00	7.75	7.62	6.63	6.63	7.07
Swedish	6.00	3.46	6.48	6.63	6.78	6.63	6.63	6.32	6.63	6.63	6.00	0.00	4.24	7.87	7.87	6.48	6.78	7.07
Danish	5.83	2.45	6.48	6.78	6.93	6.48	6.78	6.16	6.78	6.32	6.00	4.24	0.00	8.00	8.00	6.32	6.78	6.93
Finnish	7.62	7.87	7.35	7.75	7.62	7.48	7.48	7.62	7.35	7.35	7.75	7.87	8.00	0.00	2.83	7.35	7.35	7.35
Estonian	7.62	7.87	7.48	7.75	7.75	7.35	7.48	7.48	6.93	6.93	7.62	7.87	8.00	2.83	0.00	7.21	7.07	7.48
Dutch	6.00	6.32	6.48	6.63	6.63	3.16	6.32	6.93	6.48	6.16	6.63	6.48	6.32	7.35	7.21	0.00	4.90	6.63
German	6.63	6.63	6.63	6.48	6.48	4.90	6.16	6.48	6.32	5.83	6.63	6.78	6.78	7.35	7.07	4.90	0.00	6.93
Hungarian	7.07	7.07	6.78	7.21	7.21	6.93	7.07	7.07	7.21	7.21	7.07	7.07	6.93	7.35	7.48	6.63	6.93	0.00

MDS Modeling Approach: Map 270-dimensional data samples onto 2D space while preserving interpoint distances



OUTLINE

- Motivation for unsupervised learning
- NN methods for unsupervised learning
- Statistical methods for dimensionality reduction
- **Methods for multivariate data analysis**
- Summary and discussion

Methods for multivariate data analysis

Motivation: in many applications, observed (correlated) variables are assumed to depend on a small number of hidden or latent variables

$$\mathbf{x}_i = F_{true}(\mathbf{t}_i) + \xi_i$$

The goal is to model the system as

$$\mathbf{x}_i = F_{model}(\mathbf{z}_i, \omega) + \xi_i$$

where **z** is a set of factors of dim. *m*

Note: identifiability issue, the setting is not predictive

- **Approaches:** PCA, Factor Analysis, ICA (check course material for the details.)

$$\mathbf{x} = \mathbf{A}\mathbf{z} + \xi$$

Factor Analysis

- Motivation from psychology, aptitude tests
- Assumed model $\mathbf{x} = \mathbf{A}\mathbf{z} + \mathbf{u}$

  where ***x***, ***z*** and ***u*** are column vectors.

  z ~ common factor(s), ***u*** ~ unique factors
- Assumptions:

  Gaussian ***x***, ***z*** and ***u*** (zero-mean)

  Uncorrelated ***z*** and ***u*** $Cov(\mathbf{z}, \mathbf{u}) = \mathbf{0}$

  Unique factors ~ noise for each input variable (not seen in other variables)

Example: Measuring intelligence

- Aptitude tests: similarities, arithmetic, vocabulary, comprehension. Correlation btwn test scores

	Similarities test	Arithmetic test	Vocabulary test	Comprehension test
Similarities test	1.00			
Arithmetic test	0.55	1.00		
Vocabulary test	0.69	0.54	1.00	
Comprehension test	0.59	0.47	0.64	1.00

- FA result

$$similarities = (0.81)z + N(0,0.34)$$
$$arithmetic = (0.66)z + N(0,0.51)$$
$$vocabulary = (0.86)z + N(0,0.24)$$
$$comprehension = (0.73)z + N(0,0.45)$$

Factor Analysis (cont'd)

- FA vs PCA:

  - FA breaks down the covariance into two parts: common and unique factors

  - if unique factors are small (zero variance) then FA ~ PCA
- FA is designed for descriptive setting but used to infer causality (in social studies)
- However, it is dangerous to infer causality from correlations in the data

OUTLINE

- Motivation for unsupervised learning
- NN methods for unsupervised learning
- Statistical methods for dimensionality reduction
- Methods for multivariate data analysis
- **Summary and discussion**

Summary and Discussion

- Methods originate from: statistics, neural networks, signal processing, data mining, psychology etc.
- Methods pursue different goals:

  - data reduction, dimensionality reduction, data understanding, feature extraction
- Unsupervised learning is often used for supervised-learning problems if there exist many unlabeled samples.
- SOM ~ new approach to dim. reduction