university of groningen

Unsupervised Learning: Vector Quantization Competitive Learning and K-means algorithm

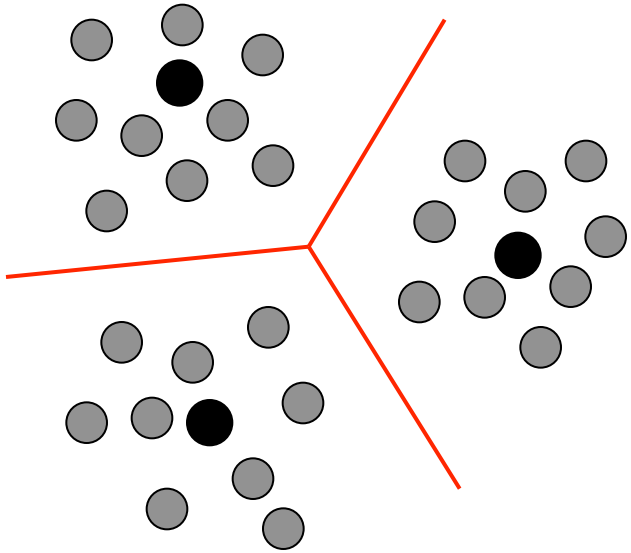
Introduction to Intelligent Systems
Michael Biehl

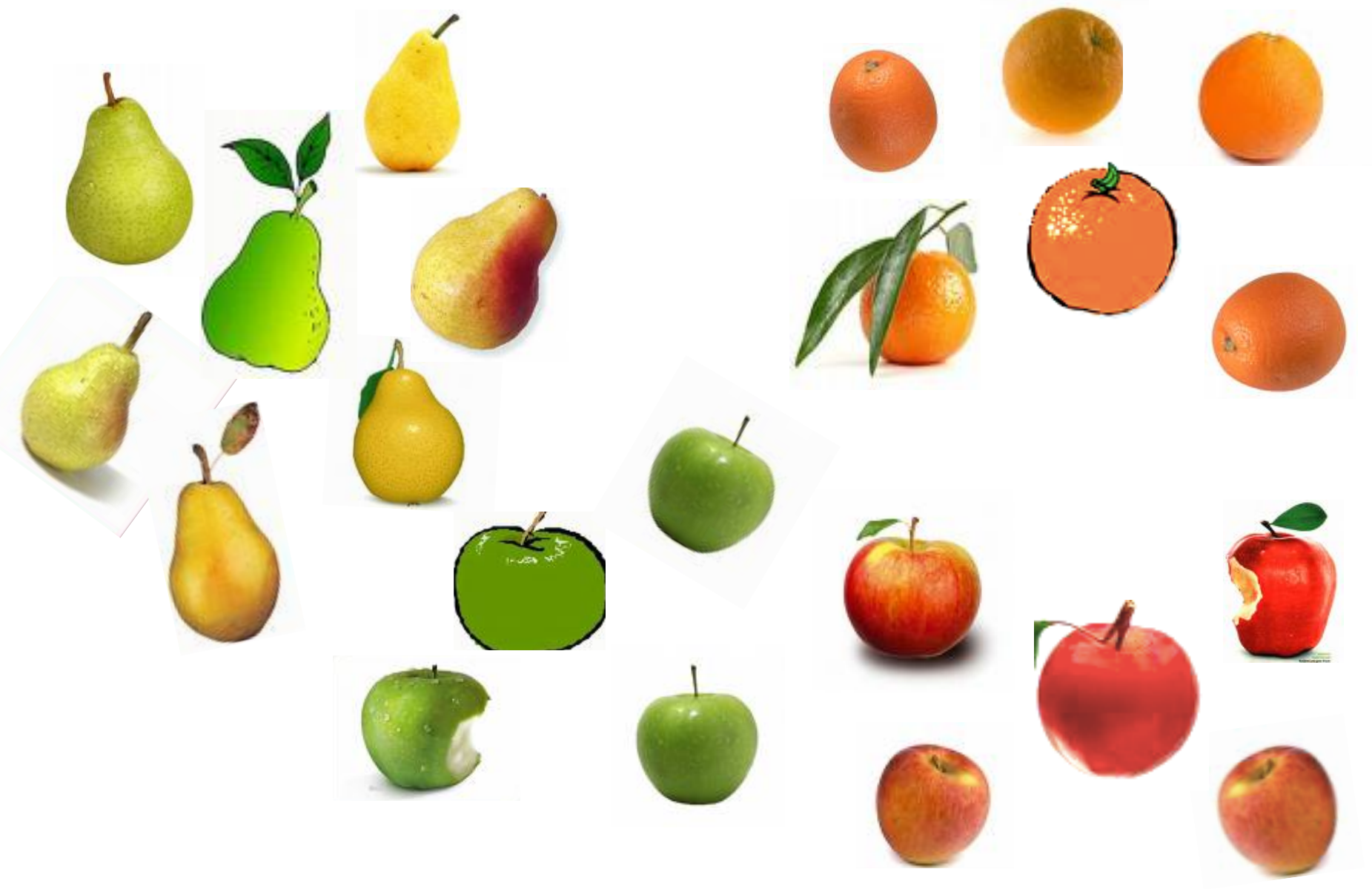
Vector Quantization (VQ)

aim:
representation of large amounts
of data by (few) **prototype vectors**

often used for:
identification and grouping
in *clusters* of similar data

assignment of feature vector ξ
to the **closest prototype w**
(similarity or distance measure,
e.g. Euclidean distance)

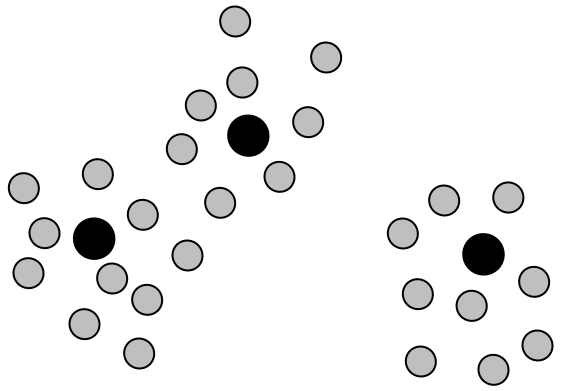




unsupervised competitive learning

- initialize K prototype vectors

- present a single example
- identify the closest prototype, i.e the so-called ***winner***
- move the winner even closer towards the example



intuitively clear, plausible procedure

- places prototypes in areas with high density of data
- identifies typical representatives of similar data
- similar to the K-means algorithm (see lectures by Nicolai Petkov)

Vector Quantization

VQ system: set of **prototypes** $w^1, w^2, \ldots, w^K \qquad w^k \in \mathbb{R}^N$

data: set of **feature vectors** $\xi^1, \xi^2, \ldots, \xi^P \qquad \xi^i \in \mathbb{R}^N$

assignment to prototypes:

based on dis-similarity/distance measure $d[w, \xi] \geq 0$

given feature vector ξ, determine the *winner* $w^{i^*} = \text{argmin}_j \{ d[w^j, \xi] \}$

→ assign ξ to prototype i^*

just one popular example: (squared) **Euclidean distance**

$$d[w, \xi] = \sum_{j=1}^{N} (w_j - \xi_j)^2$$

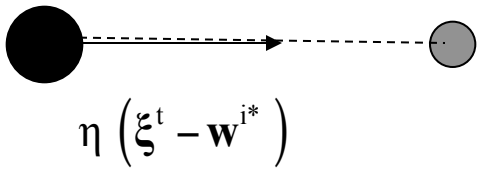
Vector Quantization

initially: randomized w^k, e.g. equal to randomly selected data points

sequential presentation of example data $\{\xi^t\} \quad t = 1,2,\ldots,P,1,2,\ldots$

… the winner takes it all: $w^{i^*} = \underset{j}{\operatorname{argmin}} \{ d[w^j, \xi^t] \}$

$$\mathbf{w}^{i^*} \rightarrow \mathbf{w}^{i^*} + \eta \left(\xi^t - \mathbf{w}^{i^*} \right)$$



η (<1): learning rate, step size of update

quantization error

competitive VQ (and K-means) aim at optimizing a **cost function**:

$$H_{VQ} = \sum_{j=1}^{K} \sum_{\mu=1}^{P} \underbrace{\left(\xi^{\mu} - w_j\right)^2}_{d_j^{\mu}} \underbrace{\prod_{k \neq j}^{K} \Theta\left(d_k^{\mu} - d_j^{\mu}\right)}_{w_j \text{ is the winner !}}$$

prototypes data

here:
Euclidean distance

- assign each data to one prototype
- measure the corresponding (squared) distance

$$\Theta(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{else} \end{cases}$$

- **quantization error** (sum over all data points)
  quantifies the quality of the representation

defines a (one) criterion to evaluate / compare the quality of different prototype configurations (important: only for fixed K, see below!)

Lloyd's algorithm (*K-means*)

(0) initialization:

place vectors w^k, e.g. equal to randomly selected points

(1) assignment of data to centers:

assign every data point to nearest center
(e.g. according to Euclidean distance)

(2) re-compute centers:

compute the means of the K clusters

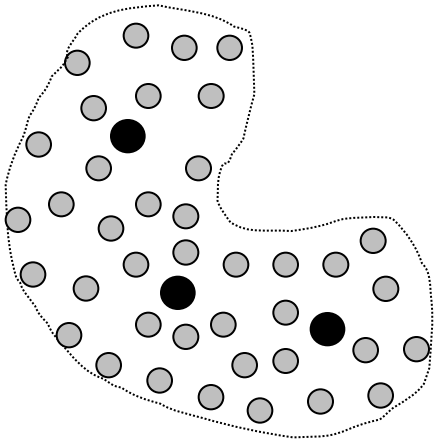
comparison:

K-means: updates all prototypes, considers all data at a time
(batch- or offline-optimization)

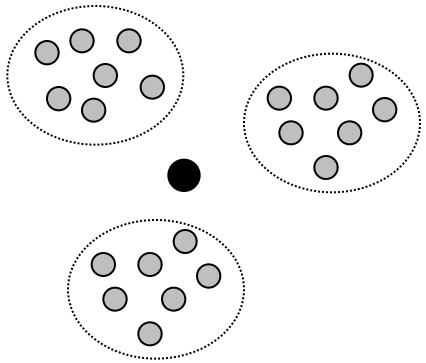
VQ-alg. : updates only the winner, random sequential presentation of single examples (stochastic or on-line optimization)

quantization and clustering

Remark 1: VQ / K-means may be related to clustering
but it is not quite the same

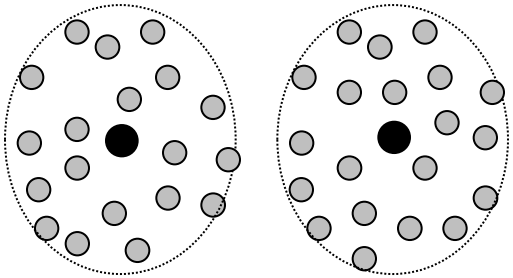


one “cluster” represented
by several prototypes

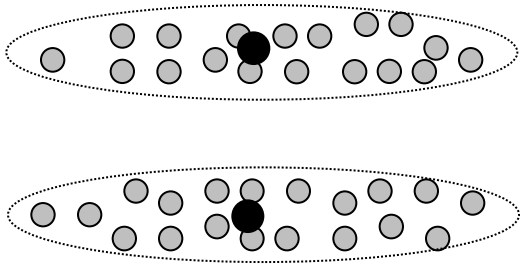


several “clusters”
represented
by one prototype

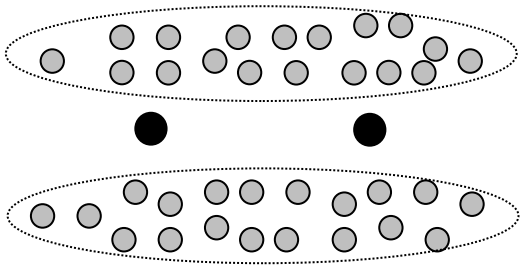
quantization and clustering



intuitive clustering
= low H_{VQ}



intuitive clustering



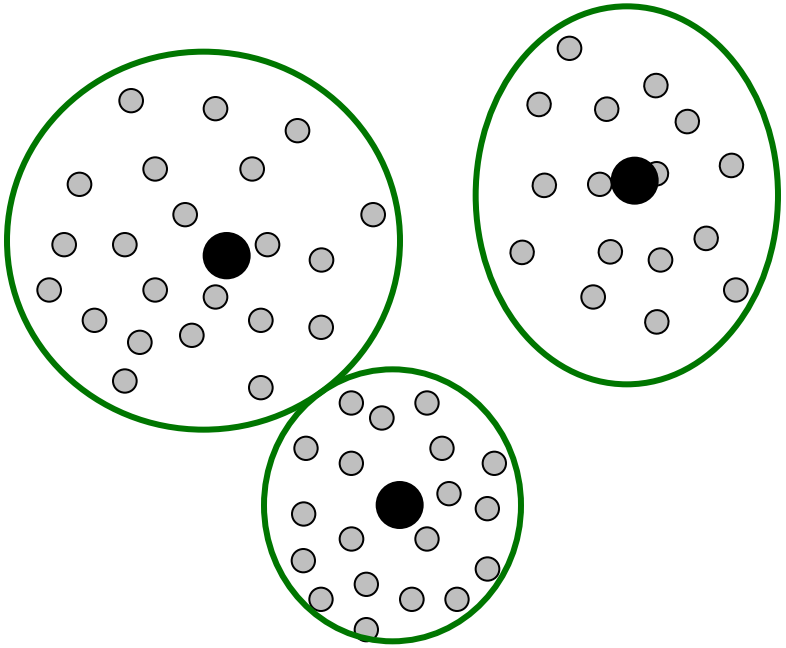
lower H_{VQ} !

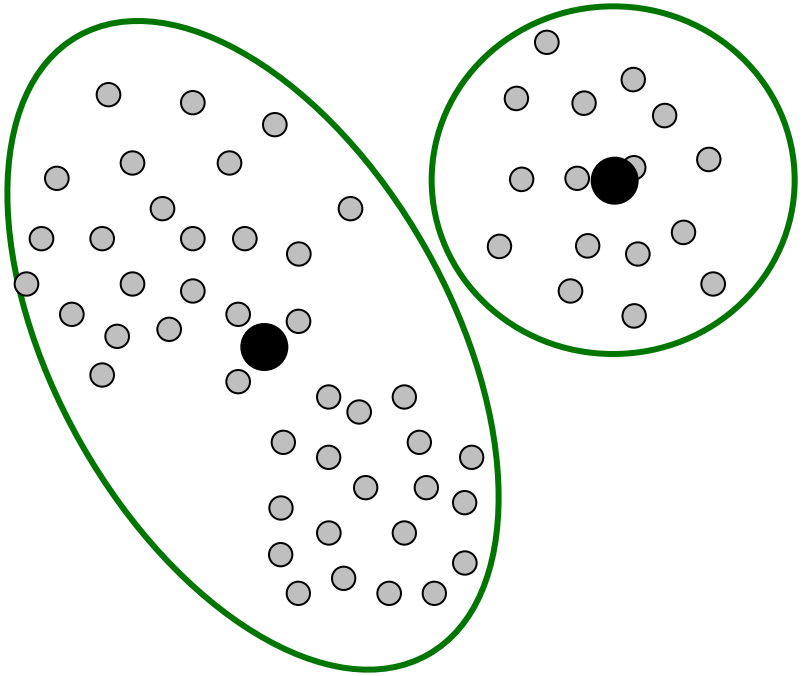
VQ/K-means result depends on

- the **number of prototypes** (predefined!)
- shape and relative position of clusters
- the **distance measure** / metric used
- **representation of the data**, e.g. scaling, transformations, normalizations etc. !!!

quantization and clustering

Remark 2: clustering is an ill-defined problem!





"obviously three clusters"

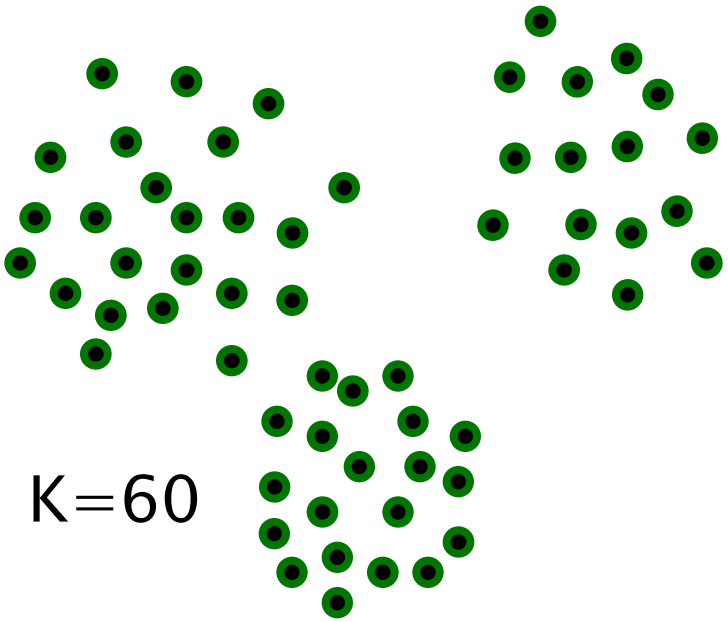
our **criterion**: lower H_{VQ}

→ " better clustering " ???

"well, maybe only two?"

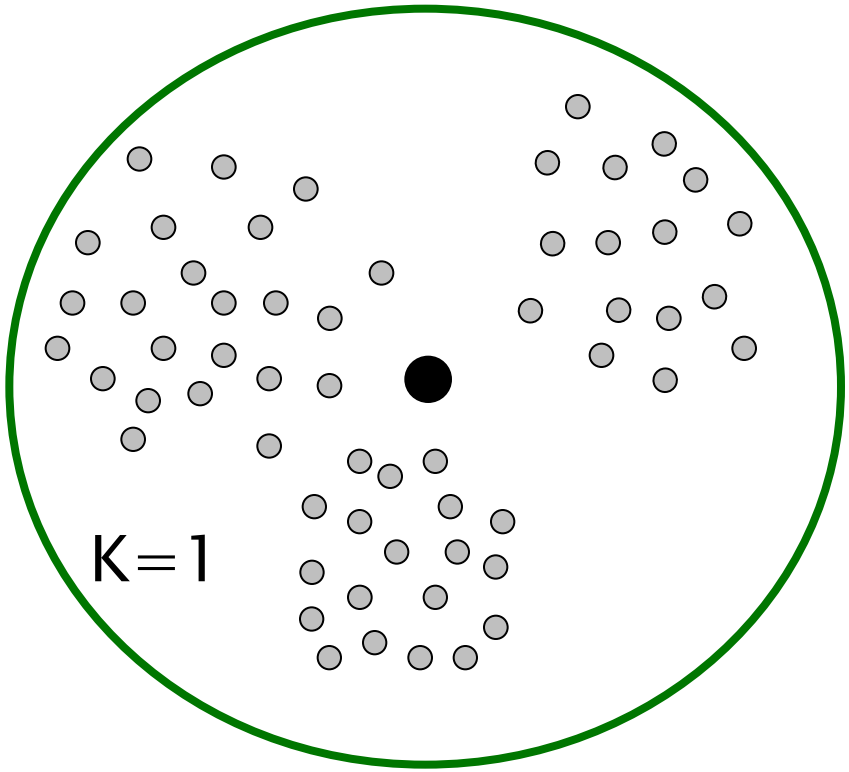
higher H_{VQ}

quantization and clustering



$H_{VQ} = 0$

→ " the best clustering " ???



the simplest "clustering" …

H_{VQ} (and similar criteria) allow only to compare VQ with the same K !

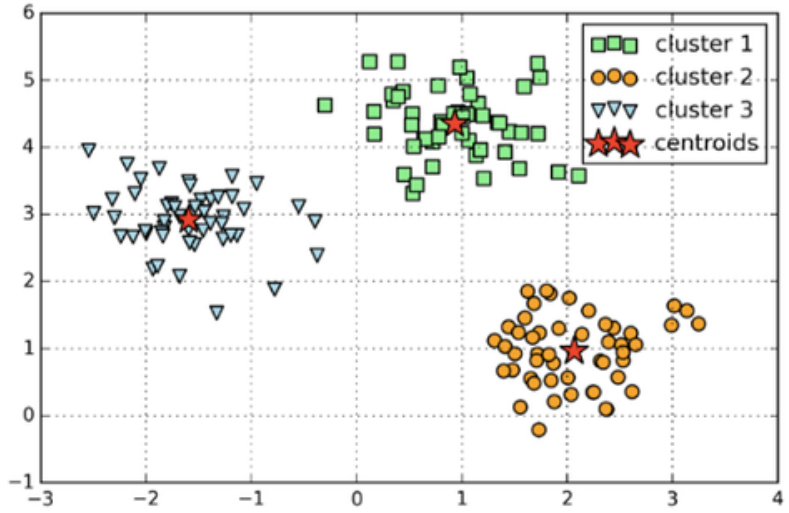
more general: heuristic compromise between "error" and "simplicity"

quantization and clustering

popular heuristics: **elbow method**

- run VQ for different K to convergence
- determine quantization error as a function of K
- identify "**elbow**" – characteristic values of K

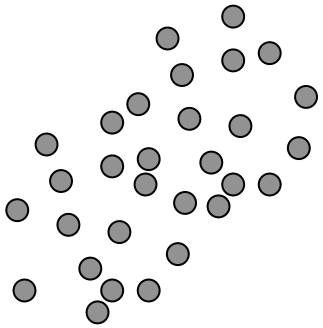
very clear example (in practice usually less pronounced)



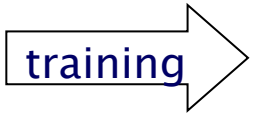
http://sebastianraschka.com
Machine Learning Blog

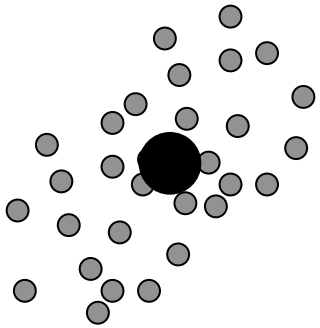
problems

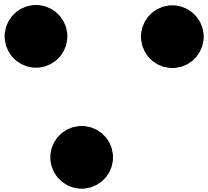
one problem of *winner takes all* prescriptions



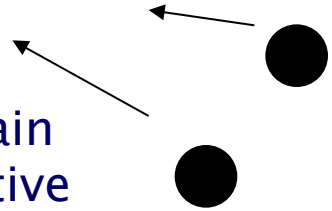
data







initial
prototypes



remain
inactive

solution: rank-based updates (winner, second, third,…)

more general problem (also K-means!)

- **local minima** of the quantization error
- inialization-dependent outcome of training