Lecture 38.
Learning Vector Quantization (LVQ)

Outline

- Vector Quantization: A Brief Introduction
- Vector Quantization: Properties
- Learning Vector Quantization
- Applications of SOM and LVQ

VQ Problem Statement

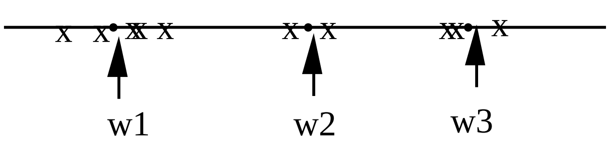
Given a set of vectors {v} drawn from a distribution f(v). The goal of vector quantization is to find an encoding scheme, which is a mapping from v to a code word w = c(v) such that the average distortion

$$D = \int d(v, w) f(v) dv$$

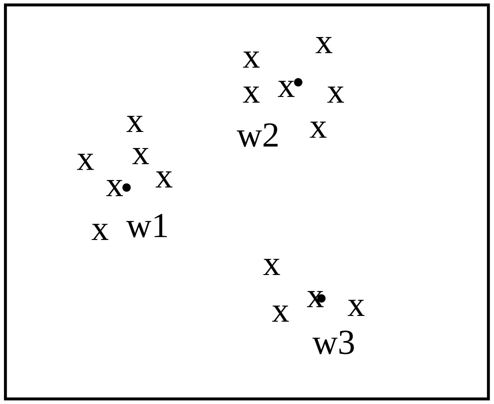
is minimized, where d(v,w) is a distortion measure chosen appropriately according to specific applications.

Vector Quantization = Clustering

- Given a set of vectors {x}, find a set of representative vectors $\{w_m; 1 \le m \le M\}$ such that each x is *quantized* into a particular w_m.



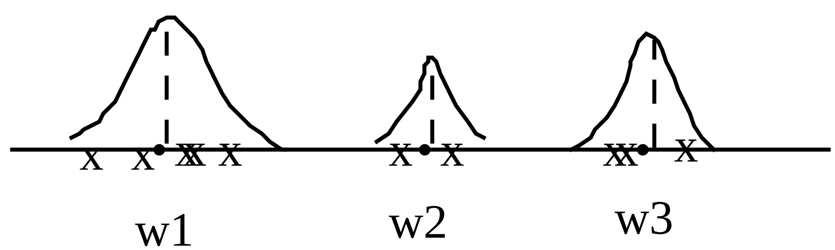
1-D (scaler) quantization



2-D vector quantization

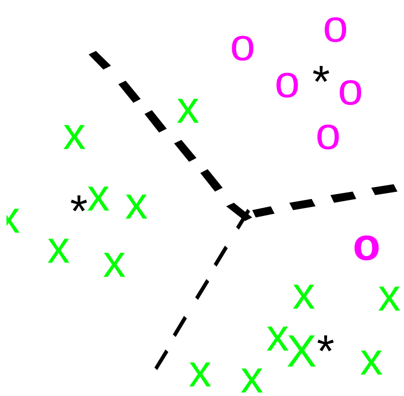
Vector Quantization

- VQ is data dependent. $\{w_m\}$ locate at the mean (centroid) of the density distribution of each cluster.



VQ and SOM

- <u>Relation between VQ and SOM</u>: SOM is a special VQ method with a constraint on spatial ordering.
- <u>Relation between VQ and pattern classification</u>: VQ is an unsupervised pattern classifier where the actual class membership information is not used.



SOM

Closest distance
$\Rightarrow$ correct classification.

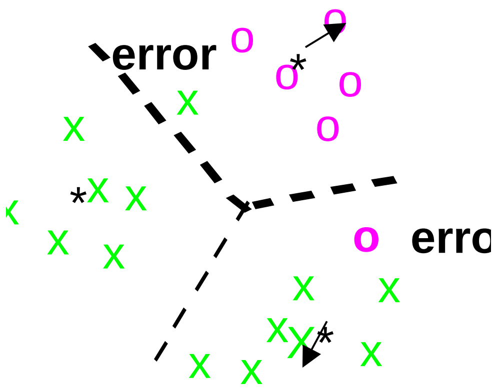
Learning Vector Quantization (LVQ)

- Fine tune SOM result to perform supervised pattern classification by fine tuning the decision boundary.
- LVQ1: First, perform SOM. Then, assign each code word to a particular class (class # < codebook size). Correct mis-classification by pushing code word away from current data vector:

$$w_{m*}(t+1) = w_{m*}(t) + \eta(t)\,(x - w_{m*}(t))$$
if x and $w_{m*}(t)$ in the same class.

$$w_{m*}(t+1) = w_{m*}(t) - \eta(t)\,(x - w_{m*}(t))$$
if x and $w_{m*}(t)$ in different classes.

$$w_m(t+1) = w_m(t) \qquad \text{if } m \neq m^*.$$



LVQ1

Learning Vector Quantization (LVQ2)

- LVQ2 – Update nearest code word and the second nearest (runner-up) code word with different classes. Denote the indices of them to be i and j:

$w_i(t+1) = w_i(t) + \eta(t)\,(x - w_i(t))$

if x and wi(t) in the same class and x in a window.

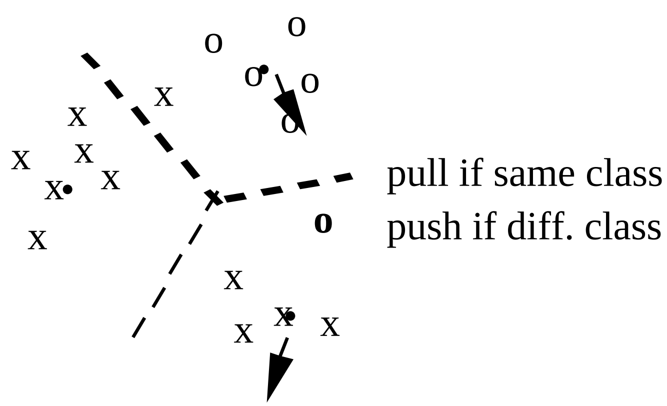
$w_j(t+1) = w_j(t) - \eta(t)\,(x - w_j(t))$

if x and wj(t) in different classes and in a window.

$w_m(t+1) = w_m(t)$ Otherwise.

LVQ2 Continued

- The window is a neighborhood near the decision boundary



LVQ 2

Learning Vector Quantization–3

- LVQ3 – i,j are the indices of the first two nearest codewords. If x and wi(t) are in the same class, x and wj(t) are in different classes, then when x falls within a predefined window, (same as LVQ2)

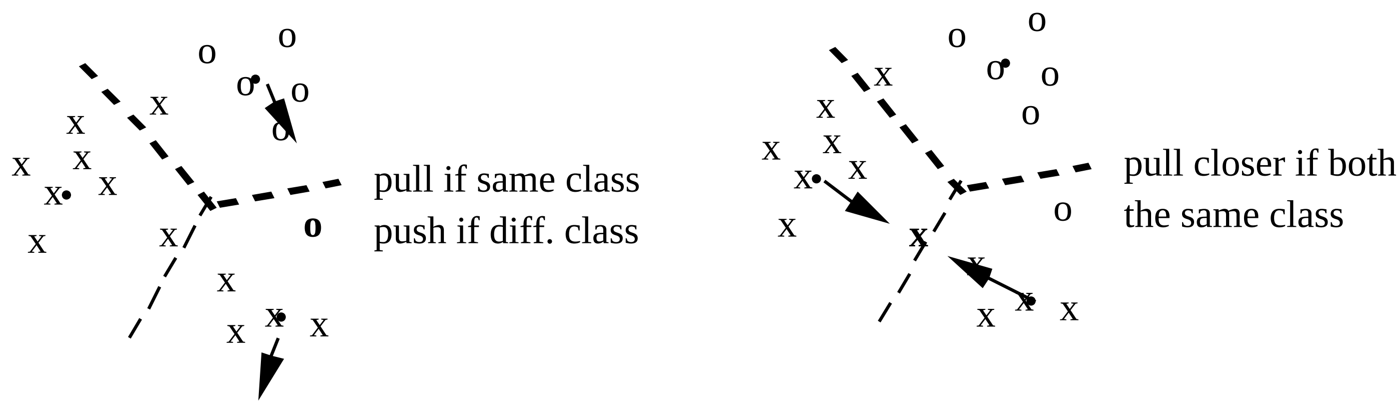
$w_i(t+1) = w_i(t) + \eta(t)\ (x - w_i(t))$

$w_j(t+1) = w_j(t) - \eta(t)\ (x - w_j(t))$

Otherwise, if x, $w_i(t)$, and $w_j(t)$ are in the same class, (different from LVQ2)

$w_k(t+1) = w_k(t) + \varepsilon\ \eta(t)\ (x - w_k(t))$ $k = i, j,$ $0.1 < \varepsilon < 0.5$

LVQ-3 Continued



LVQ 3

- Update one code word at a time.
- $\eta(t) < 0.02$, and should be decreased to zero in, say, 100,000 steps.

Applications of SOM and LVQ

- Speech Recognition
- Robot Arm control
- Industrial process control
- automated synthesis of digital systems
- channel equalization for telecommunication
- image compression
- radar classification of sea-ice
- optimization problems
- sentence understanding
- classification of insect courtship songs