Clustering
Hypothesis: Hebbian synaptic plasticity enables a perceptron to compute the mean of its preferred stimuli.
Unsupervised learning

• Sequence of data vectors
• Learn something about their structure
• Multivariate statistics
• Neural network algorithms
• Brain models
Data can be summarized by a few prototypes.
Vector quantization

- Many telecom applications
- Codebook of prototypes
- Send index of prototype rather than whole vector
- Lossy encoding
A single prototype

• Summarize all data with the sample mean.

\[ \mu = \frac{1}{m} \sum_{a=1}^{m} x_a \]
Multiple prototypes

• Each prototype is the mean of a subset of the data.
• Divide data into $k$ clusters.
  – One prototype for each cluster.
Assignment matrix

\[
A_{a\alpha} = \begin{cases}
1, & x_a \in \text{cluster } \alpha \\
0, & \text{otherwise}
\end{cases}
\]

- Data structure for cluster memberships.
**k-means algorithm**

- Alternate between computing means and computing assignments.

\[
\mu_\alpha = \frac{1}{m} \sum_{i=1}^{m} x_a A_{a\alpha} \quad \sum_{b=1}^{m} A_{b\alpha} = 1 \text{ for } \alpha = \arg \min_b |x_b - \mu_\alpha|
\]
Objective function

• Why does it work?
• Method of minimizing an objective function.
Rubber band computer

\[
\frac{1}{2} \sum_{a=1}^{m} \left| x_a - \mu \right|^2
\]

- Attach rubber band from each data vector to the prototype vector.
- The prototype will converge to the sample mean.
The sample mean maximizes likelihood

- Gaussian distribution

\[ P_\mu(x) \propto \exp\left(-\frac{1}{2}|x - \mu|^2\right) \]

- Maximize

\[ P_\mu(x_1)P_\mu(x_2)\cdots P_\mu(x_m) \]
Objective function for $k$-means

$$E(A, \mu) = \frac{1}{2} \sum_{a=1}^{m} \sum_{\alpha=1}^{k} A_{a\alpha} |x_a - \mu_\alpha|^2$$

$$\mu = \arg \min_{\mu} E(A, \mu)$$

$$A = \arg \min_{\overline{A}} E(\overline{A}, \mu)$$
Local minima can exist
Model selection

• How to choose the number of clusters?
• Tradeoff between model complexity and objective function.
Neural implementation

• A single perceptron can learn the mean in its weight vector.

• Many competing perceptrons can learn prototypes for clustering data.
Batch vs. online learning

- **Batch**
  - Store all data vectors in memory explicitly.

- **Online**
  - Data vectors appear sequentially.
  - Use one, then discard it.
  - Only memory is in learned parameters.
Learning rule 1

\[ w_t = w_{t-1} + \eta x_t \]
Learning rule 2

\[ w_t = w_{t-1} + \eta_t (x_t - w_{t-1}) \]

\[ = (1 - \eta_t) w_{t-1} + \eta_t x_t \]

• “weight decay”
Learning rule 2 again

$$\Delta w = -\eta \frac{\partial}{\partial w} \frac{1}{2} |x - w|^2$$

- Is there an objective function?
Stochastic gradient following

The average of the update is in the direction of the gradient.
Stochastic gradient descent

\[ \Delta w = -\eta \frac{\partial}{\partial w} e(w, x) \]

\[ \langle \Delta w \rangle = -\eta \frac{\partial E}{\partial w} \quad E(w) = \langle e(w, x) \rangle \]
Convergence conditions

• Learning rate vanishes
  – slowly \( \sum_{t} \eta_{t} = \infty \)
  – but not too slowly \( \sum_{t} \eta_{t}^2 < \infty \)

• Every limit point of the sequence \( w_{t} \) is a stationary point of \( E(w) \)
Competitive learning

• Online version of $k$-means

$$y_a = \begin{cases} 1, & \text{minimal } |x - w_a| \\ 0, & \text{other clusters} \end{cases}$$

$$\Delta w_a = \eta y_a (x - w_a)$$
Competition with WTA

• If the $w_a$ are normalized

$$\arg\min_a |x - w_a| = \max_a w_a \cdot x$$
Objective function

\[ \left\langle \min_a \frac{1}{2} |x - w_a|^2 \right\rangle \]
Cortical maps

Images removed due to copyright reasons.
Ocular dominance columns

Images removed due to copyright reasons.
Orientation map

Images removed due to copyright reasons.
Kohonen feature map

\[ y_a = \begin{cases} 
1, & \text{neighborhood of closest cluster} \\
0, & \text{elsewhere} 
\end{cases} \]

\[ \Delta w_a = \eta y_a (x - w_a) \]
Hypothesis: Receptive fields are learned by computing the mean of a subset of images
Nature vs. nurture

• Cortical maps
  – dependent on visual experience?
  – preprogrammed?