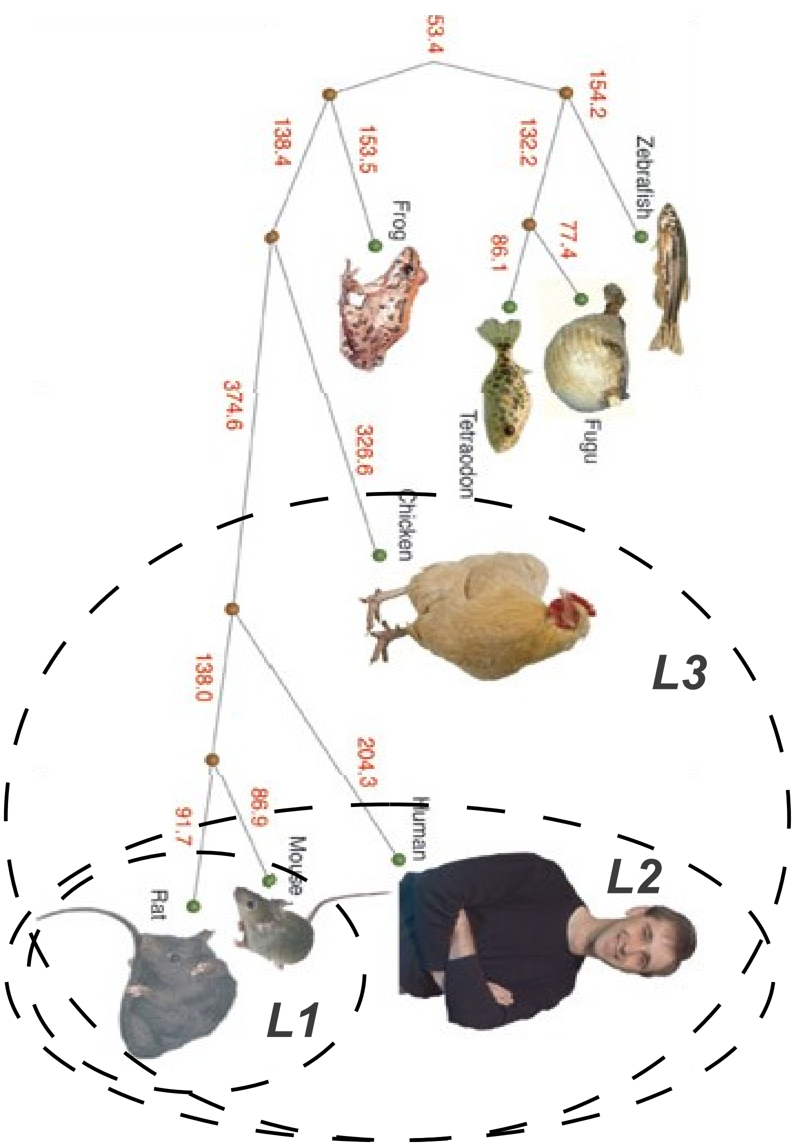
CIS501 – Lecture 14

Woon Wei Lee
Fall 2013, 10am-11:15am,
Sundays and Wednesdays

For today:

- Unsupervised learning
  - Hierarchical clustering
  - Visualization

Hierarchical clustering



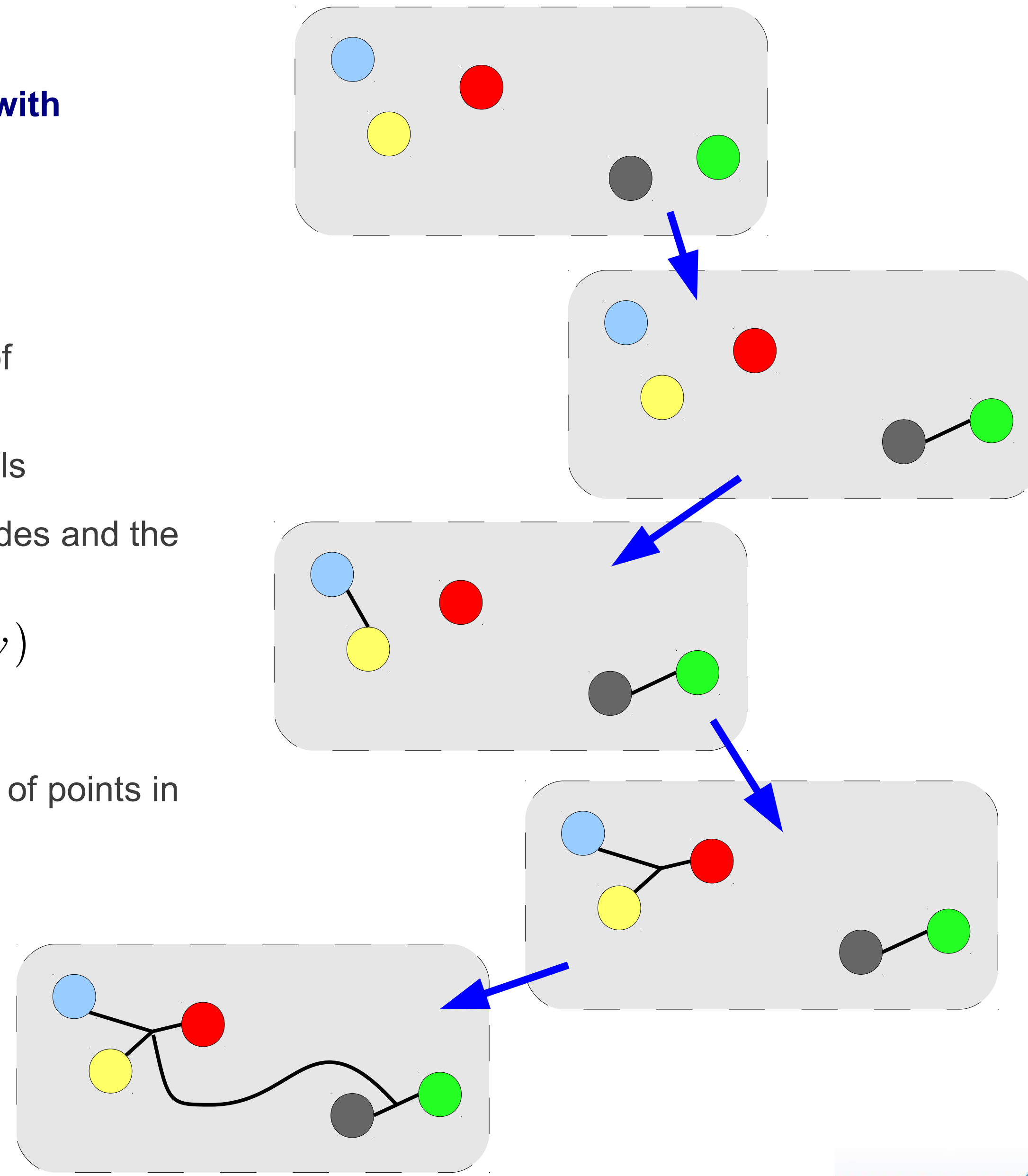
- **A form of clustering where the nodes are organized hierarchically**
  - Depiction can be rooted - as a “taxonomy”..
  - Or unrooted – simple depiction of relationships between nodes without super/sub-classes
- **Many hierarchical clustering algorithms have their roots in biology/bioinformatics**
  - Inference of *phylogenetic trees* (evolutionary history)
  - But also used for document clustering, image clustering, etc.
  - Typically works using only *distance* information
  - Clusters defined by number of levels in the tree.
- **Two classes**
  - Agglomerative clustering
  - Divisive clustering

Agglomerative clustering: UPGMA

- **Most basic of agglomerative techniques**
- **Stands for “Unweighted Pair Group Method with Arithmetic mean”**
  - “Greedy” algorithm
- **Algorithm:**
  - Inputs: Distance matrix between all pairs of nodes/individiuals in the group
  - *Combine nearest pairs of nodes/individuals
  - Recalculate new distances between all nodes and the cluster using:

$$\frac{1}{|A||B|}\cdot\sum_{x\in A}\sum_{y\in B} d(x,y)$$

  - i.e. the average distance between all pairs of points in the two groups
  - Generate new distance matrix
  - Repeat from (*)

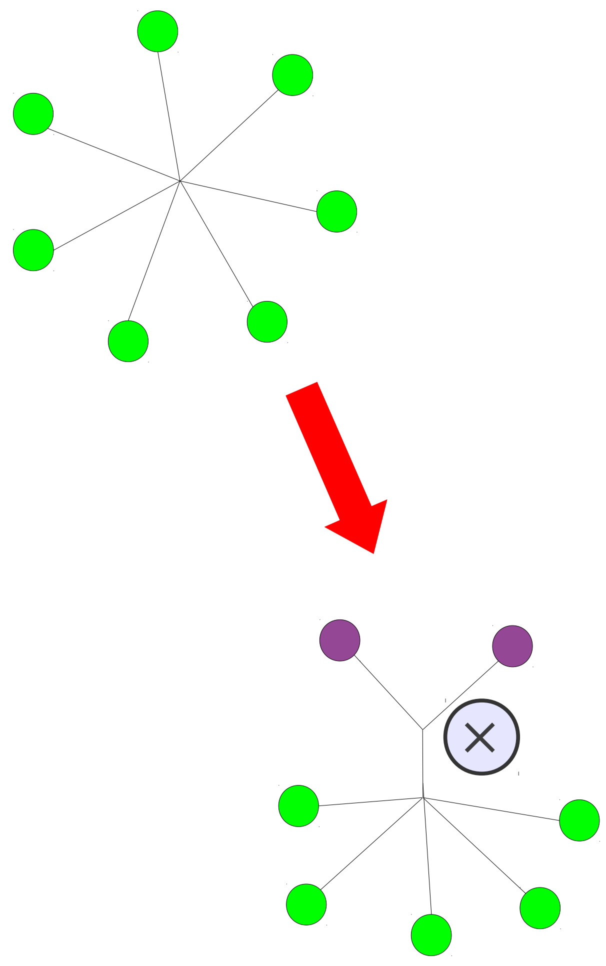


Agglomerative clustering: Neighbour Joining

- **UPGMA – good starting point but poor performance:**
  - “Molecular clock” problem
  - Heuristic technique → unreliable and frequently does not give good results
- **“Neighbour-Joining” algorithm**
  - Generates “unrooted” trees
  - Works via global estimates of branch length
- **Algorithm:**
  - Inputs: Distance matrix between all pairs of nodes/individiuals in the group
  - Initiate with “star topology”
  - Choose pair of nodes which minimize:

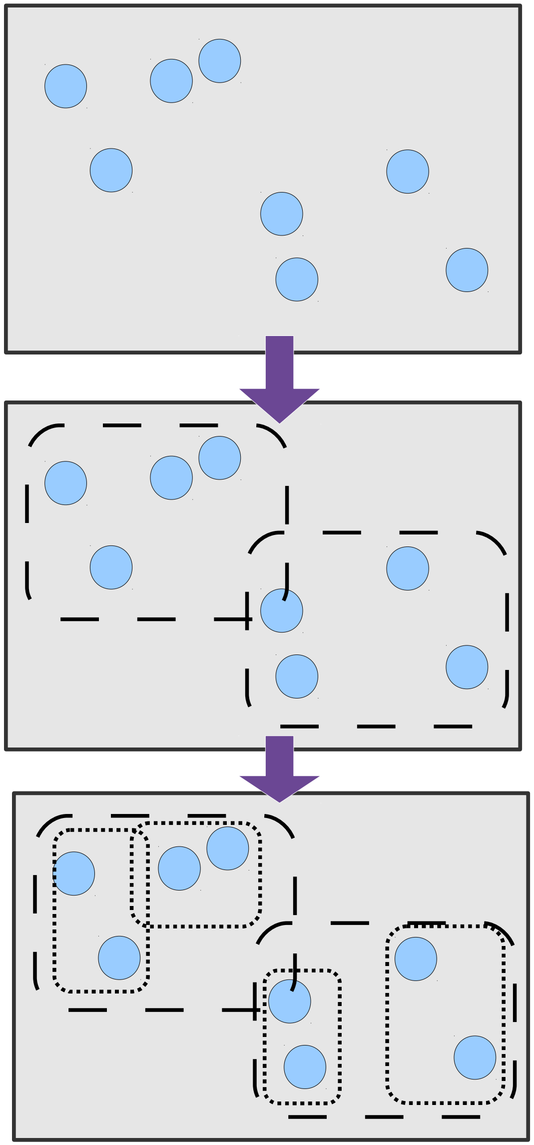
$$s_{12}=\frac{1}{2(N-2)}\sum_{k=3}^{N}\left(d(1,k)-d(2,k)\right)+\frac{1}{2}d(1,2)+\frac{1}{n-2}\sum_{3\leqslant i\leqslant j}d(i,j)$$

  - Pair up as shown on right
  - Remove pair from distance matrix and replace with branch point (marked “×”)
  - Repeat until only binary splits remain
  - S_{ij} is the *total branch length* in the tree, which we attempt to minimize (principle of parsimony)



Divisive clustering: Hierarchical *k*-means algorithm

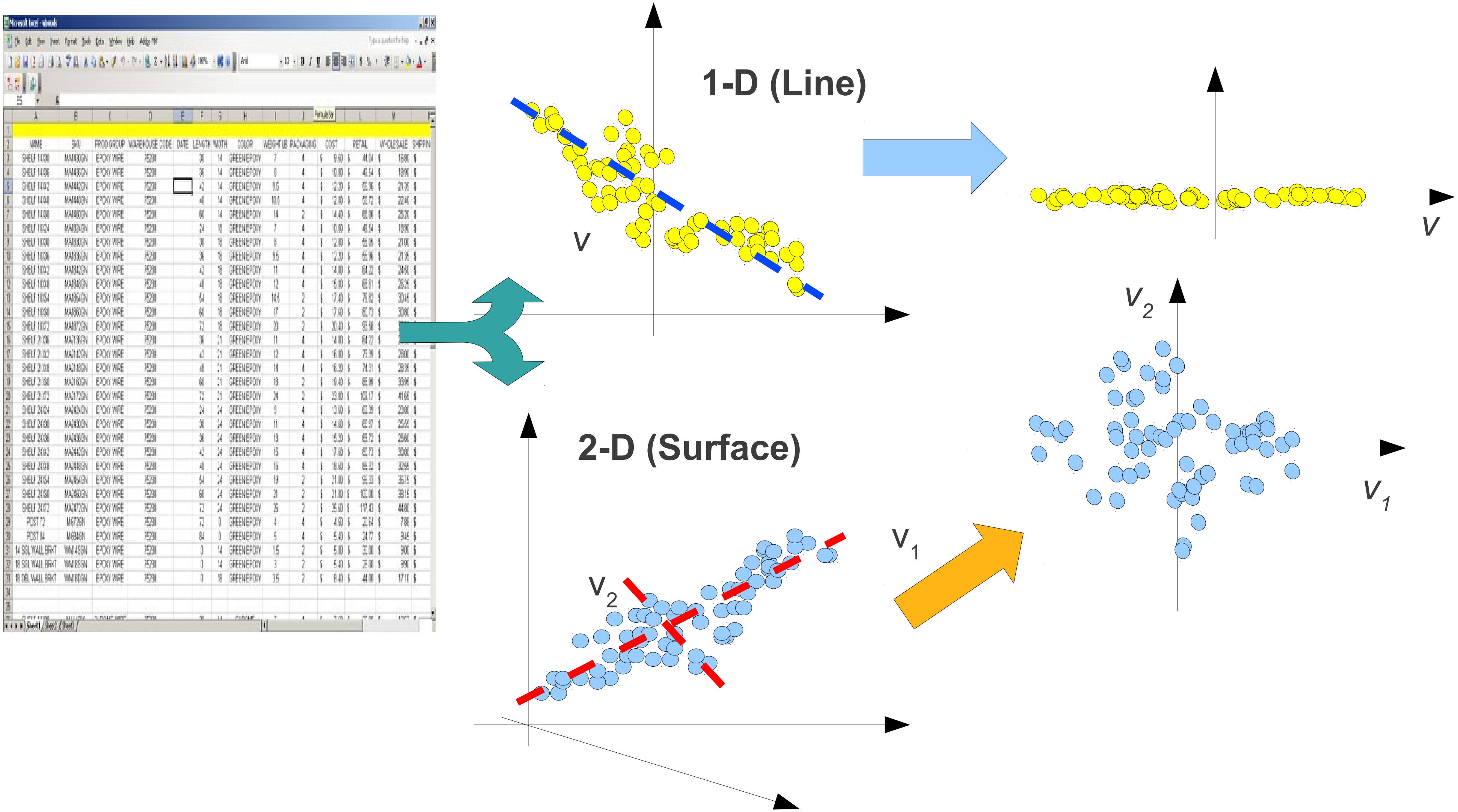
- **Also known as "top-down" clustering**
  - Any partitional clustering technique can be used as a divisive hierarchical clustering algorithm
- **Advantages:**
  - Many efficient/principled algorithms can be "recycled"
  - Clustering can be for a fixed number of levels
- **Example: "Hierarchical" k-means algorithm:**
  - Perform k-means as per normal
  - For each cluster with at least a minimum number of individuals, cluster using k-means again
  - Iterate until termination



Unsupervised learning: Visualization

- **A cross between feature selection and dimensionality reduction!**
- **Basic motivation**
  - Convert high dimensional data set into a representation that is easily visualized/comprehended
  - Allows for manual clustering, classification, etc.
- **Two basic categories:**
  - Linear projections
    - PCA
    - ICA
    - NMF, etc..
  - Nonlinear projections
    - SOM
    - Multi-dimensional scaling (MDS)
    - Sammon Mapping

Linear visualization techniques



Problem is: how to find the optimal projection line/surface?

Principle Component Analysis (PCA)

- **Basic idea – find direction which capture “most” of the data**
  - From figure on right, intuitively, v_a does this better
  - One metric – variance of the projection
  - Direction of maximal variance - known as the *principle components*
- **Finding the principle components**
  - Linear projection defined as:

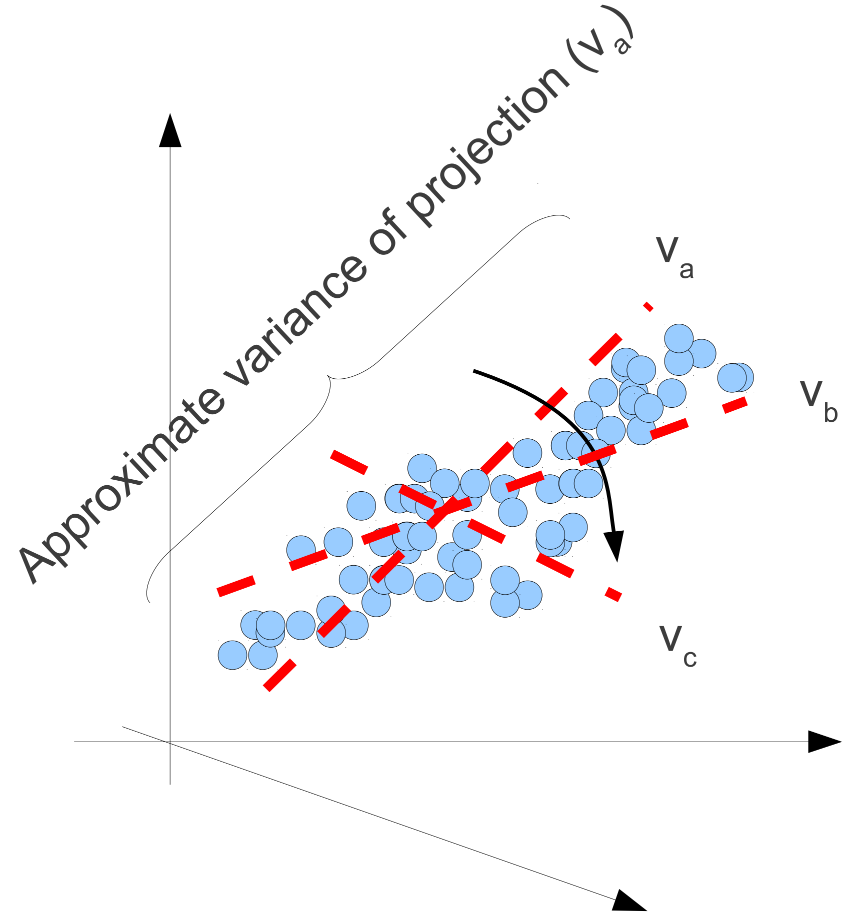
$$x' = v^T x$$

(*x'* is the transformed variable)

  - Variance of projection is given by:

$$\sigma^{2\prime} = x' x'^T$$
$$= v^T x (v^T x)^T$$
$$= v^T x x^T v = v^T \Sigma v$$

(where Σ is the data covariance matrix)



Principle Component Analysis (PCA)

- **Hence, to find the principle components,**

  Maximize: $v^T \Sigma v$ (w.r.t. *v)*

- **Trivial solution → set *v* to ∞**
  - Constraint needed:

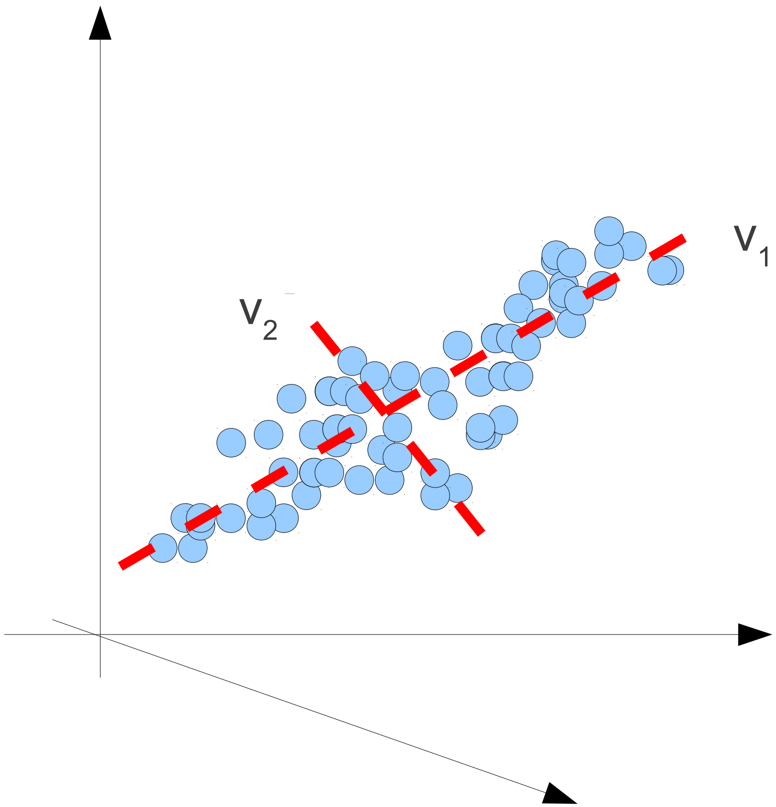
    Set: $v^T v = 1$

$$L(\sigma,\lambda) = v^T \Sigma v - \lambda(v^T v - 1)$$

Lagrange multiplier

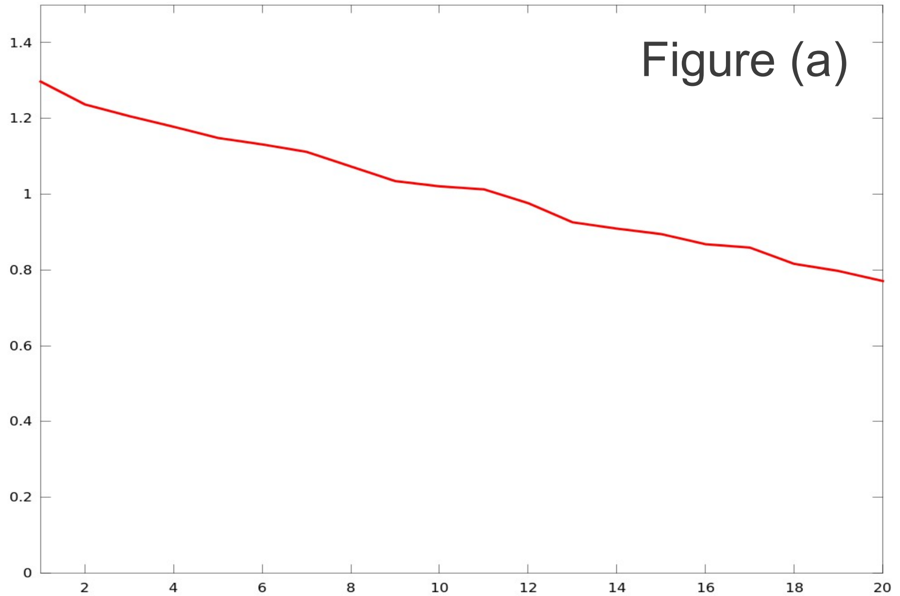
$$\frac{dL(\sigma,\lambda)}{dv} = 2\Sigma v - 2\lambda v = 0$$

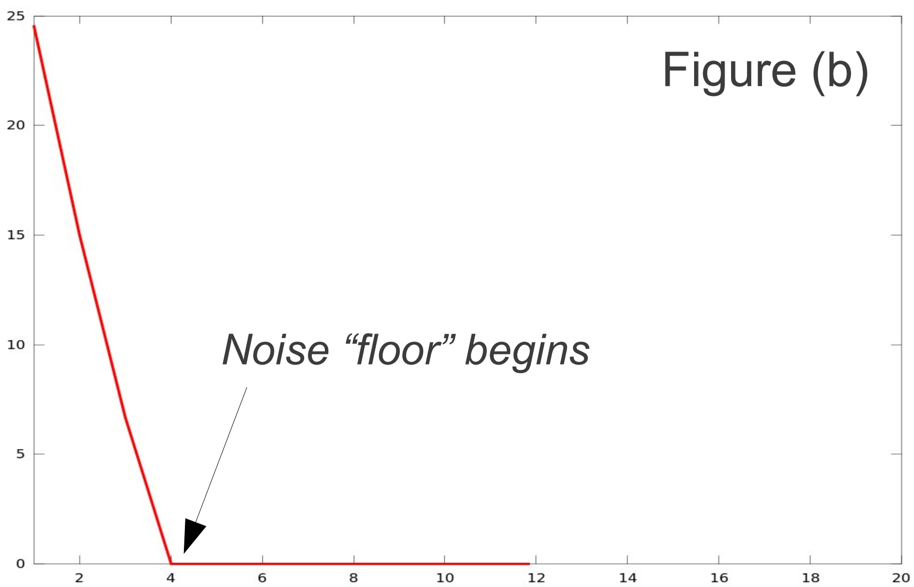
$$\Sigma v = \lambda v$$



- **This can be re-written in the form of the "characteristic equation"**
  - Solution given by the eigenvectors of the covariance matrix
  - The λ value gives the variance of the projection in this direction (the eigenvalues of the covariance matrix)

PCA (Cont'd)





i.e. The principle components are given by the *eigenvectors* of the covariance matrices

- For an *n*-dimensional dataset, there will be *n* such eigenvectors.
- Eigenvectors are mutually orthonormal
- Matrix of eigenvectors is hence a *rotation matrix*

Project upon a subset of these eigenvectors → dimensionality reduction

- The λ value → the eigenvalues of the covariance matrix
- Sorting these and plotting gives the *singular spectrum* (**SS**)

Figures (a) and (b):

(a) SS corresponding to 20 dimensional white noise

(b) SS for 3 dimensional white noise embedded in 20 dimensional space

- Note the noise “floor” in figure (b).