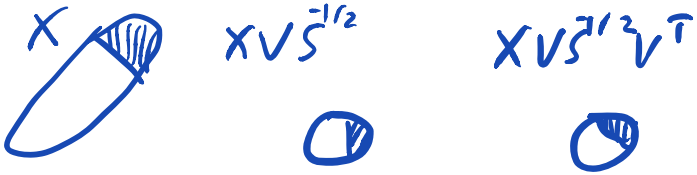
Shameless plug for my course next quarter

CSE 599: Online and Adaptive Methods for Machine Learning.

Webpage: https://courses.cs.washington.edu/courses/cse599i/18wi/
Non-CSE need add-codes: https://goo.gl/forms/G76D6cOKNtdBlbe62

The standard approach to machine learning uses a training set of labeled examples to learn a prediction rule that will predict the labels of new examples. Collecting such training sets can be expensive and time-consuming. This course will explore methods that leverage already-collected data to guide future measurements, in a closed loop, to best serve the task at hand. We focus on two paradigms: i) in pure-exploration we desire algorithms that identify or learn a good model using as few measurements as possible (e.g., classification, drug discovery, science), and ii) in regret minimization we desire algorithms that balance taking measurements to learn a model with taking measurements to exploit the model to obtain high reward outcomes (e.g., medical treatment design, ad-serving). The course will assume introductory machine learning (e.g., CSE 546) and maturity in topics like linear algebra, statistics, and calculus. The course will be analysis heavy, with a focus on methods that work well in practice.

Practice



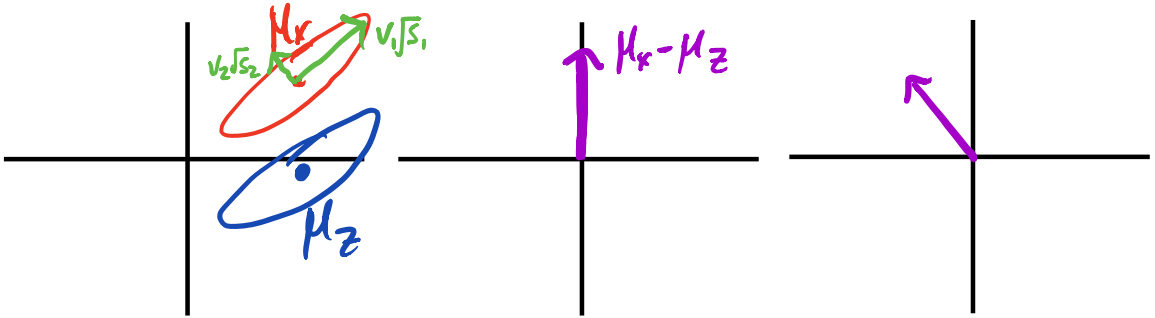
- Fill in the missing plots:

$$\Sigma = \mathbf{X}^T\mathbf{J}\mathbf{J}\mathbf{X} = \mathbf{Z}^T\mathbf{J}\mathbf{J}\mathbf{Z}$$

$$\mathbf{V}\mathbf{S}\mathbf{V}^T = \text{eig}(\Sigma) \qquad \mathbf{J} = I - \mathbf{1}\mathbf{1}^T/n$$

$$\mu_X = \mathbf{X}^T\mathbf{1}/n \qquad \mu_Z = \mathbf{Z}^T\mathbf{1}/n$$

$\mathbf{X}$ $\mathbf{Z}$ $\mu_X - \mu_Z$ $\mathbf{V}S^{-1/2}\mathbf{V}^T(\mu_X - \mu_Z) = \hat{w}$



$\hat{f}(x) = \hat{w}^T x + b$

Pick b to minimize classification loss

Principal Component Analysis (continued)

Machine Learning – CSE546

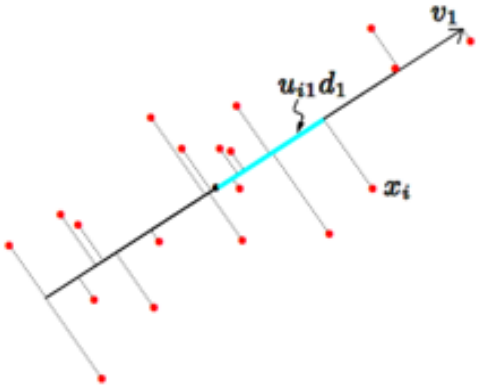
Kevin Jamieson
University of Washington

November 16, 2017

Linear projections

Given $x_i \in \mathbb{R}^d$ and some $q < d$ consider

$$\min_{\mathbf{V}_q} \sum_{i=1}^{N} ||(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})||^2.$$



where $\mathbf{V}_q = [v_1, v_2, \ldots, v_q]$ is orthonormal:

$$\mathbf{V}_q^T \mathbf{V}_q = I_q$$

$\mathbf{V}_q$ are the first q eigenvectors of Σ

$\mathbf{V}_q$ are the first q *principal components*

$$\Sigma := \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T$$

Principal Component Analysis (PCA) projects $(\mathbf{X} - \mathbf{1}\bar{x}^T)$ down onto $\mathbf{V}_q$

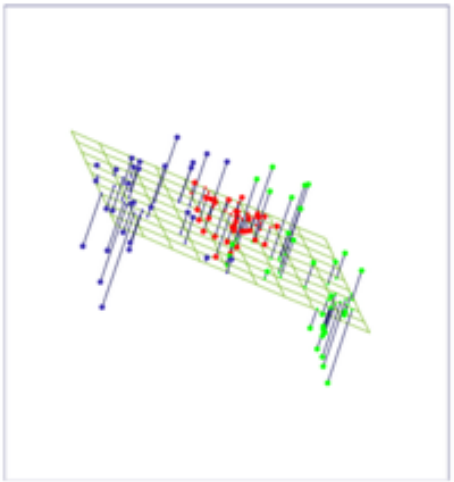
$$(\mathbf{X} - \mathbf{1}\bar{x}^T)\mathbf{V}_q = \mathbf{U}_q \mathrm{diag}(d_1, \ldots, d_q) \qquad \mathbf{U}_q^T \mathbf{U}_q = I_q$$

Singular Value Decomposition defined as

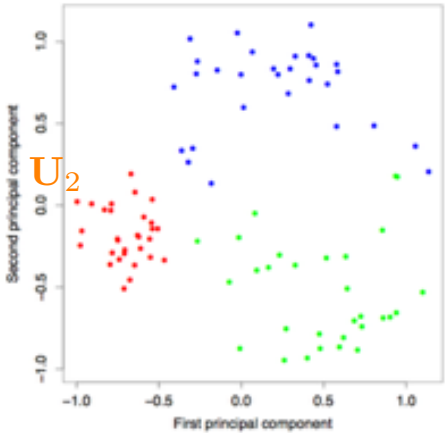
$$\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

Dimensionality reduction

$\mathbf{V}_q$ are the first q eigenvectors of Σ and *SVD* $\mathbf{X} - \mathbf{1}\bar{x}^T = \mathbf{USV}^T$



$\mathbf{X} - \mathbf{1}\bar{x}^T$



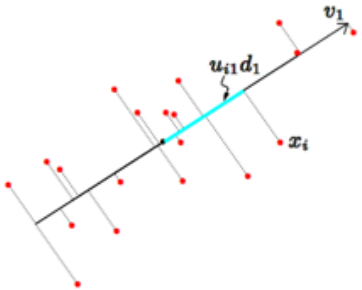
$\mathbf{U}_2$

$\mathbf{U}_1$

Power method - one at a time

$$\Sigma := \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T \qquad v_* = \arg\max_{v} \; v^T \Sigma v$$

$$v_{k+1} = \frac{\Sigma v_k}{||\Sigma v_k||} \qquad v_0 \sim \mathcal{N}(0, I)$$



Matrix completion

Given historical data on how users rated movies in past:

17,700 movies, 480,189 users, 99,072,112 ratings (Sparsity: 1.2%)

Predict how the same users will rate movies in the future (for $1 million prize)

Alice	1	?	?	4	?	
Bob	?	2	5	?	?	
Carol	?	?	4	5	?	
Dave	5	?	?	?	4	
⋮						

Matrix completion

Choose d by cross-valid.

σ^2 is chosen by best training loss

$X \in \mathbb{R}^{m \times n}$

n movies, m users, |S| ratings

$\ell(U,V)$

$U_t^{(i,j)} \sim \mathcal{N}(0,\sigma^2)$

$V_t^{(i,j)} \sim \mathcal{N}(0,\sigma^2)$

$$\arg\min_{U \in \mathbb{R}^{m \times d}, V \in \mathbb{R}^{n \times d}} \sum_{(i,j,s) \in \mathcal{S}} ||(UV^T)_{i,j} - s_{i,j}||_2^2$$

How do we solve it? With full information?

SGD: Sample (i,j,s) from training set

$$U_{t+1} = U_t - \nabla \ell_{ijs}(U_t, V_t)\, \eta_t$$

$$V_{t+1} = V_t - \nabla \ell_{ijs}(U_t, V_t)\, \eta_t$$

$$\ell_{ijs} = \left(\langle U_i, V_j \rangle - s_{ij}\right)^2$$

Alternating Min.

$$U_{t+1} = \arg\min_U \ell(U, V_t)$$

$$V_{t+1} = \arg\min_V \ell(U_{t+1}, V)$$

$X = \hat{U} \hat{S} \hat{V}^T$

$$\ell(U,V) = \|UV^T - X\|_F^2$$

$$= Tr\left((UV^T - X)^T (UV^T - X)\right)$$

$$\propto -Tr(U^T X V)$$

Matrix completion

n movies, m users, |*S*| ratings

$$\underset{U \in \mathbb{R}^{m \times d}, V \in \mathbb{R}^{n \times d}}{\arg\min} \sum_{(i,j,s) \in \mathcal{S}} ||(UV^T)_{i,j} - s_{i,j}||_2^2$$

Random projections

PCA finds a low-dimensional representation that reduces population variance

$$\min_{\mathbf{V}_q} \sum_{i=1}^{N} ||(x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x})||^2.$$

$\mathbf{V}_q \mathbf{V}_q^T$ is a *projection matrix* that minimizes error in basis of size q

$\mathbf{V}_q$ are the first q eigenvectors of Σ

$$\Sigma := \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T$$

But what if I care about the reconstruction of the *individual* points?

$$\min_{\mathbf{W}_q} \max_{i=1,\ldots,n} ||(x_i - \bar{x}) - \mathbf{W}_q \mathbf{W}_q^T (x_i - \bar{x})||^2$$

Random projections

$$\min_{\mathbf{W}_q} \max_{i=1,\ldots,n} ||(x_i - \bar{x}) - \mathbf{W}_q \mathbf{W}_q^T (x_i - \bar{x})||^2$$

Johnson-Lindenstrauss (1983)

Theorem 1.1. *(Johnson-Lindenstrauss) Let $\epsilon \in (0, 1/2)$. Let $Q \subset \mathbb{R}^d$ be a set of n points and $k = \frac{20 \log n}{\epsilon^2}$. There exists a Lipshcitz mapping $f : \mathbb{R}^d \to \mathbb{R}^k$ such that for all $u, v \in Q$:*

(independent of d)

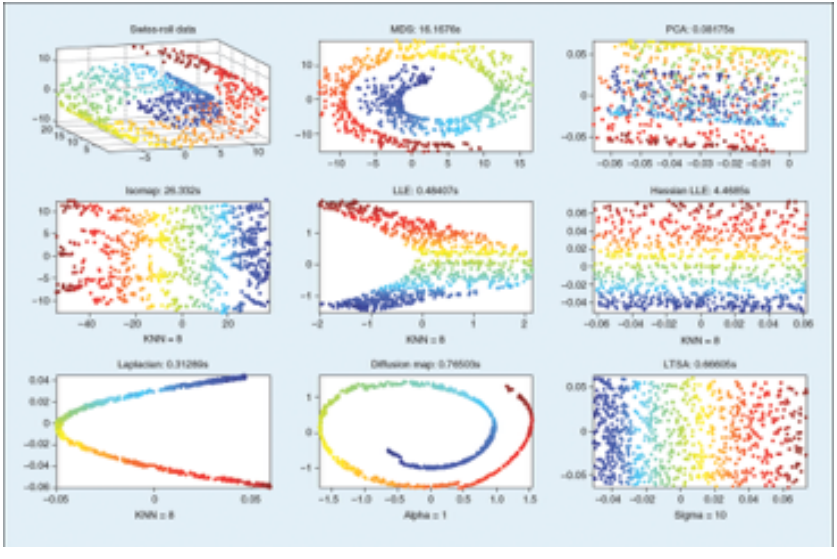
$$(1 - \epsilon)\|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \epsilon)\|u - v\|^2$$

Theorem 1.2. *(Norm preservation) Let $x \in \mathbb{R}^d$. Assume that the entries in $A \subset \mathbb{R}^{k \times d}$ are sampled independently from $N(0, 1)$. Then,*

$$\Pr((1 - \epsilon)\|x\|^2 \leq \|\frac{1}{\sqrt{k}} Ax\|^2 \leq (1 + \epsilon)\|x\|^2) \geq 1 - 2e^{-(\epsilon^2 - \epsilon^3)k/4}$$

Nonlinear dimensionality reduction

Find a low dimensional representation that respects “local distances” in the higher dimensional space



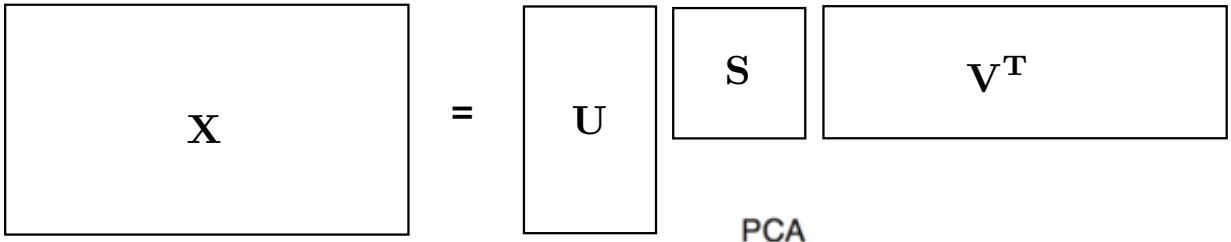
Zhang et al 2010

Many methods:

- Kernel PCA
- ISOMAP
- Local linear embedding
- Maximum volume unfolding
- Non-metric multidimensional scaling
- Laplacian
- Neural network auto encoder
- …

Due to lack of agreed upon metrics, it is very hard to judge which is best. Also, results from 3 to 2 dims is probably not representative of 1000 to 2 dimensions.

Other matrix factorizations



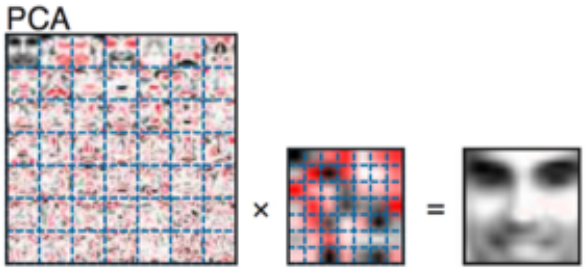
Singular value decomposition

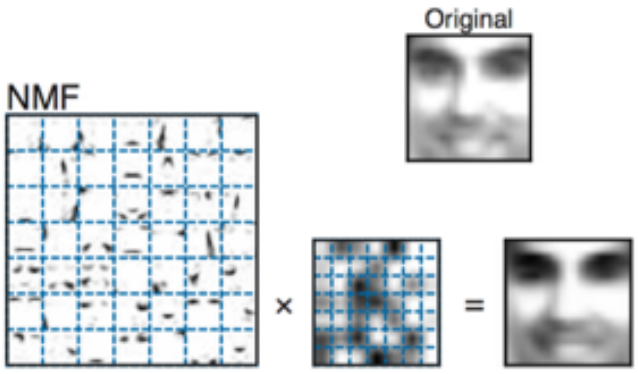
$$\mathbf{U}^T\mathbf{U} = I,\ \mathbf{V}^T\mathbf{V} = I,\ \mathbf{S} = \text{drag}(s)$$
$$\mathbf{U} \in \mathbb{R}^{n\times q},\ \mathbf{V} \in \mathbb{R}^{m\times q},\ s \in \mathbb{R}_+^q$$
$$\mathbf{X} \approx \mathbf{U}_q\mathbf{S}_q\mathbf{V}_q^T$$

Nonnegative matrix factorization (NMF)

$$\mathbf{W} \in \mathbb{R}_+^{n\times q} \text{ with } \mathbf{W1} = \mathbf{1}$$
$$\mathbf{B} \in \mathbb{R}_+^{q\times n} \text{ with } \mathbf{B1} = \mathbf{1}$$
$$\mathbf{X} \approx \mathbf{WBX}$$

H





Clustering

Machine Learning – CSE546

Kevin Jamieson
University of Washington

November 16, 2016

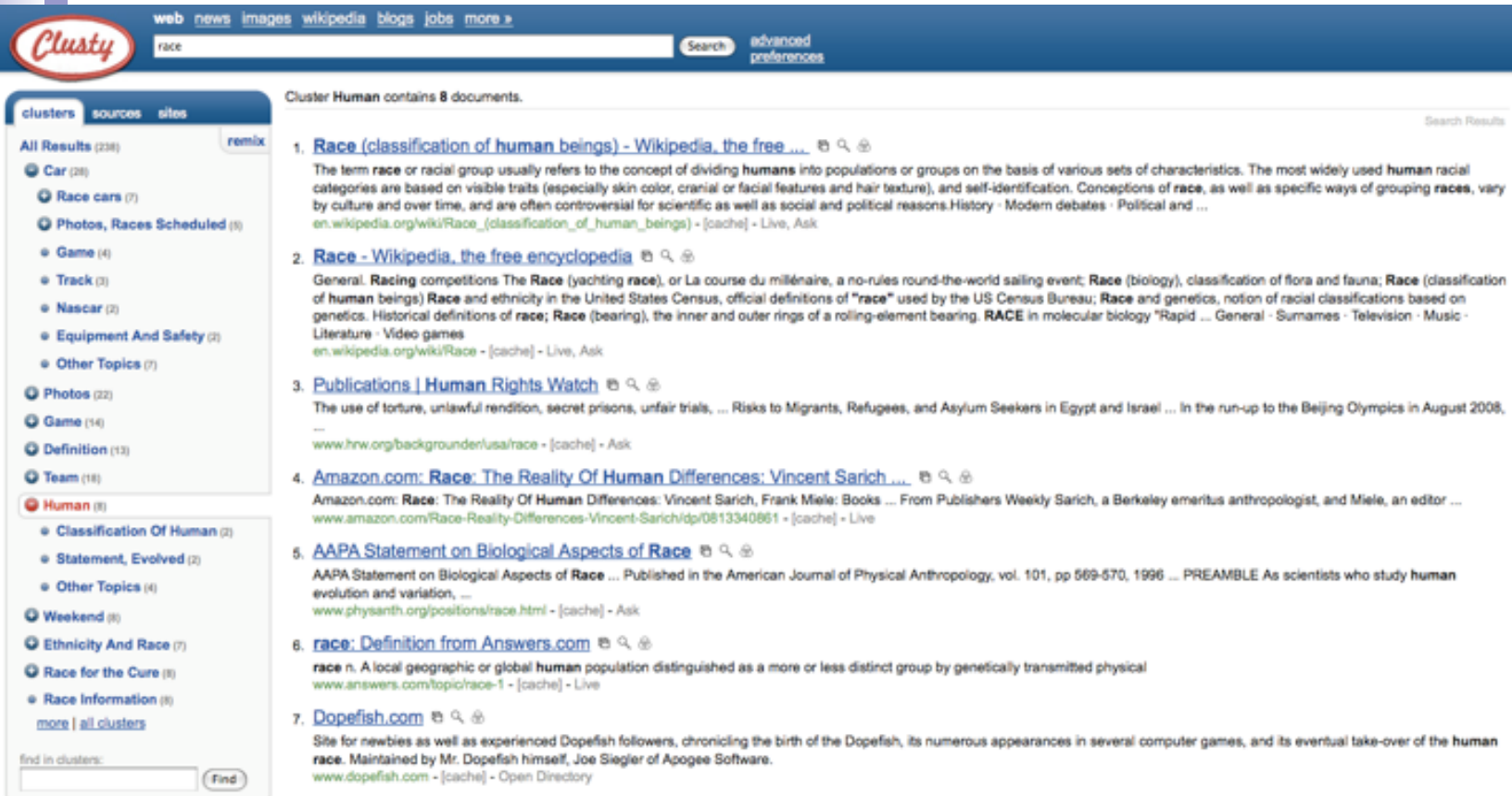
©Kevin Jamieson 2017

Clustering images

Set of Images

[Goldberger et al.]

Clustering web search results



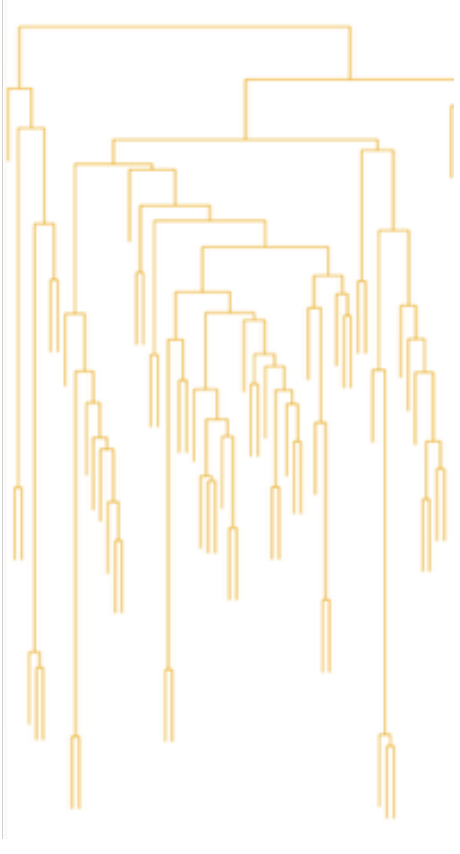
©Kevin Jamieson 2017

Hierarchical Clustering

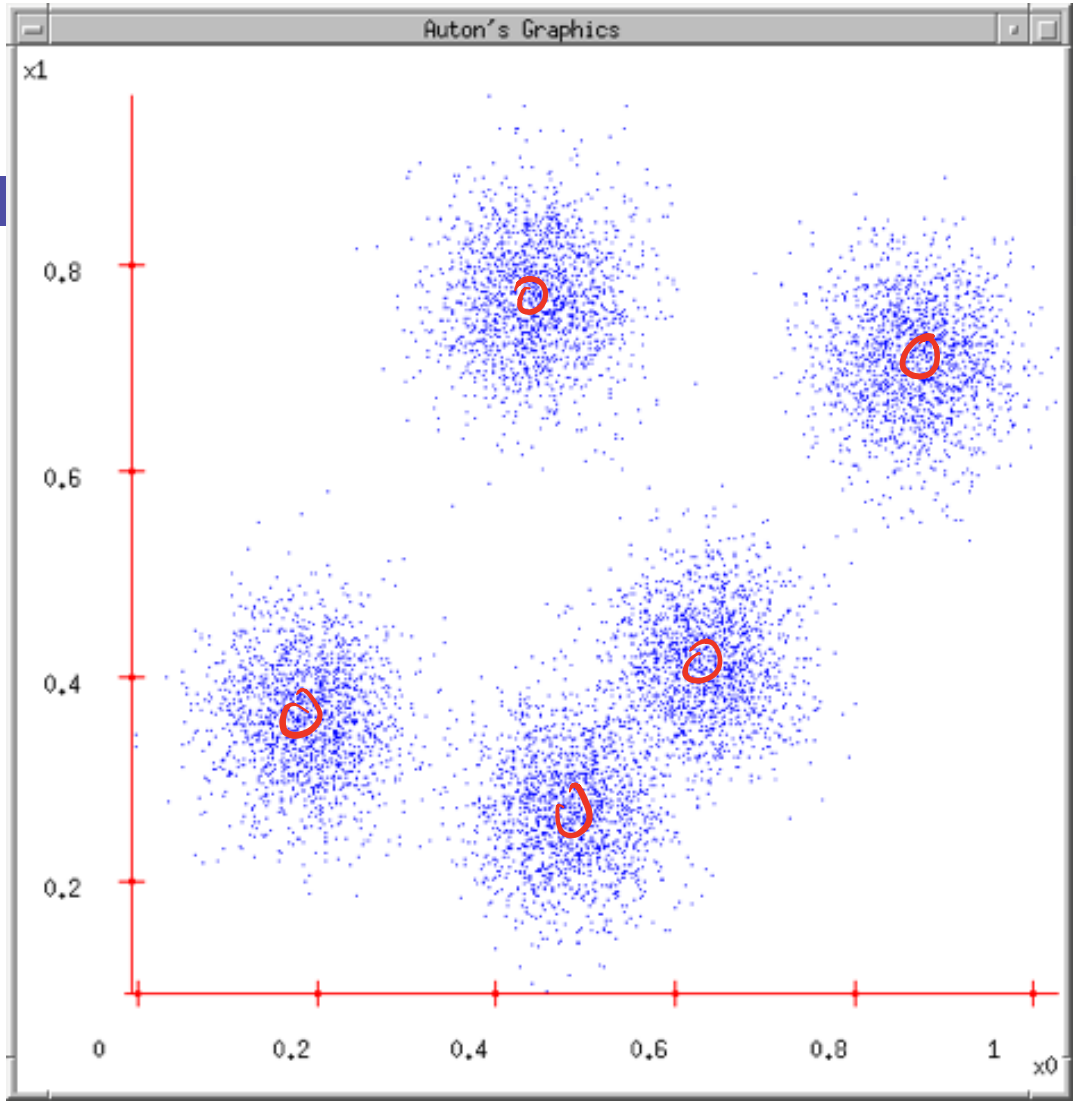
Pick one:
- Bottom up: start with every point as a cluster and merge
- Top down: start with a single cluster containing all points and split

Different rules for splitting/merging, no “right answer”

Gives apparently interpretable tree representation. However, warning: even random data with no structure will produce a tree that “appears” to be structured.



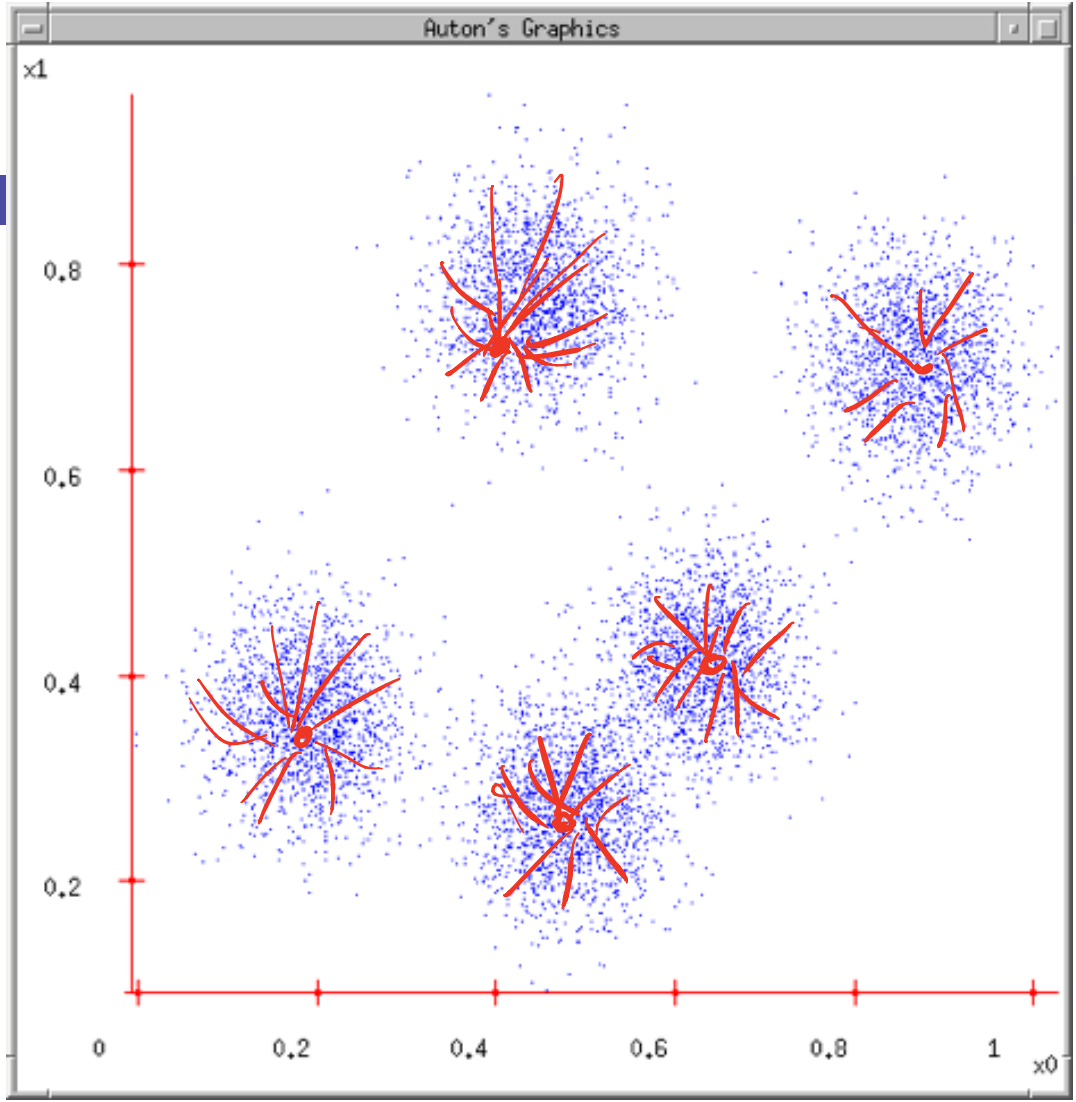
Some Data



©Kevin Jamieson 2017

K-means

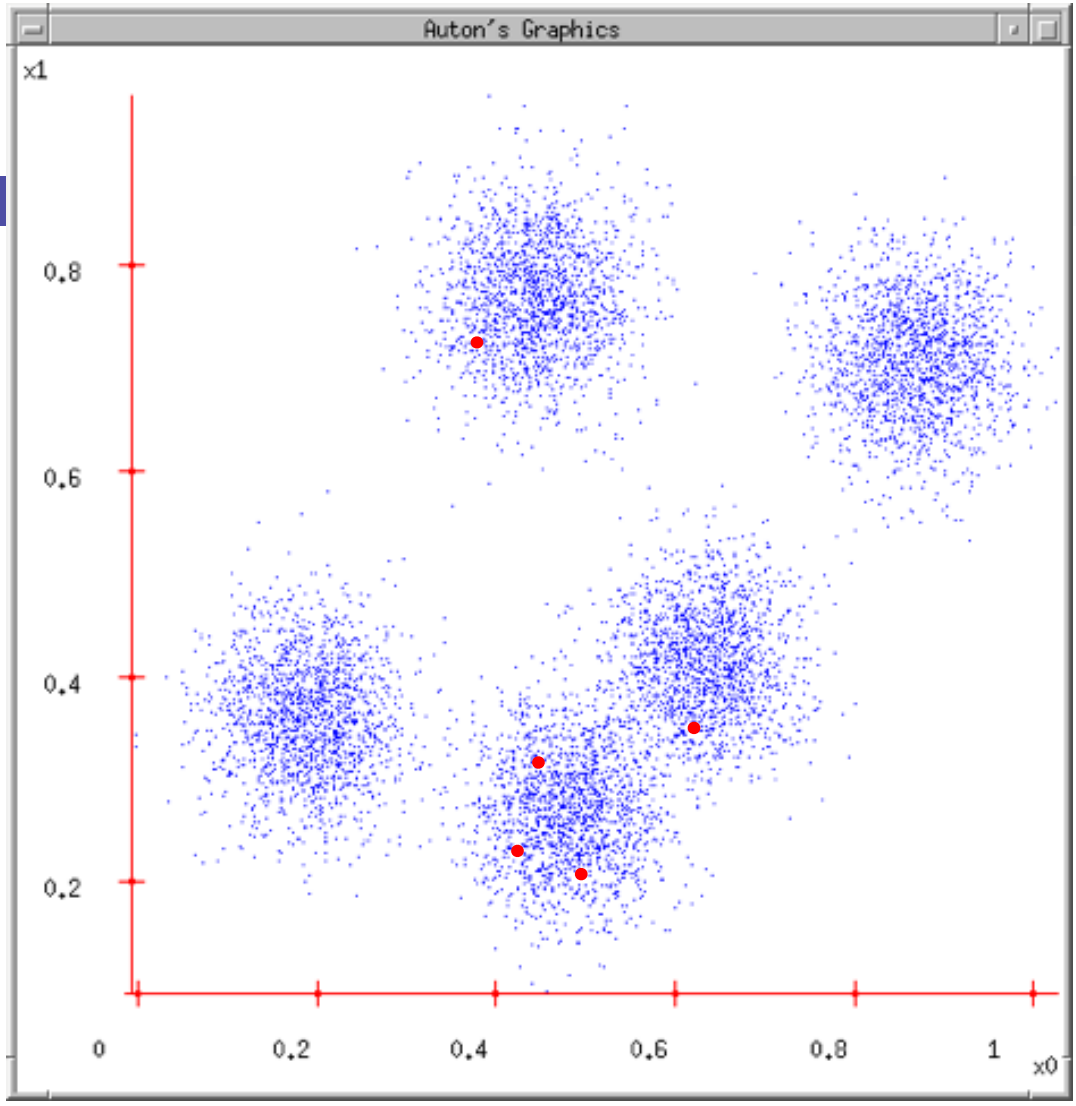
1. Ask user how many clusters they'd like. *(e.g. k=5)*



©Kevin Jamieson 2017

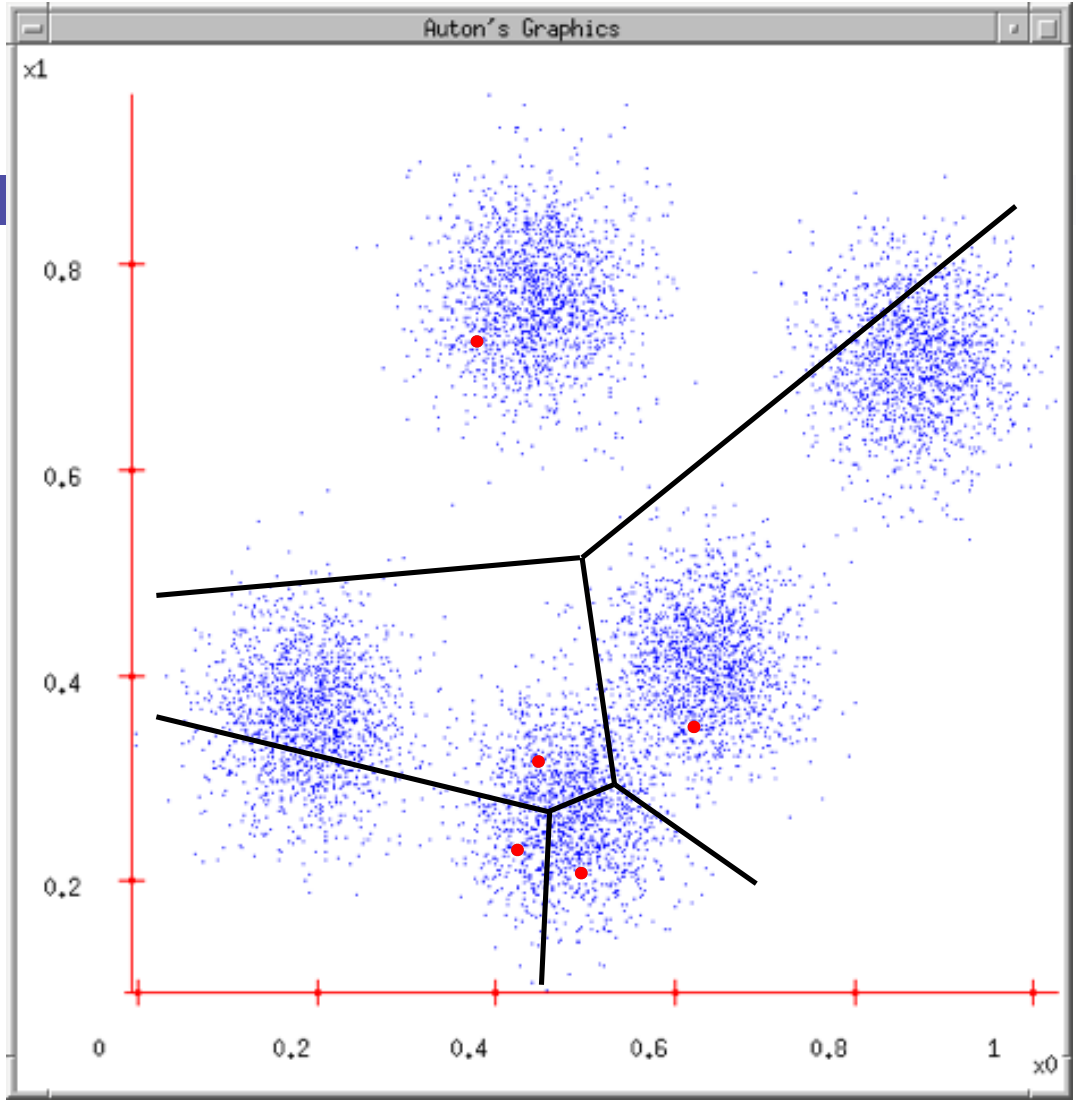
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



K-means

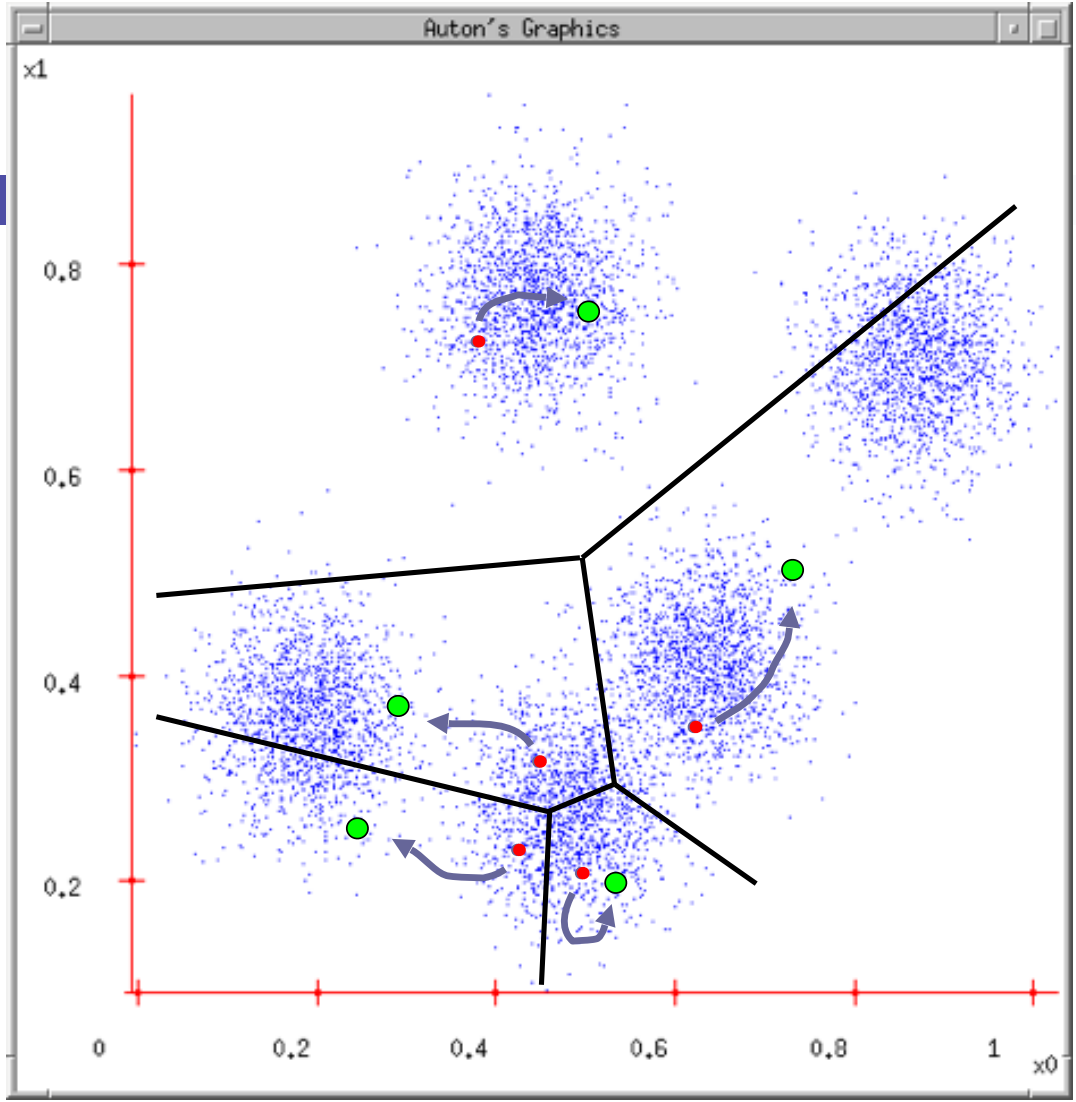
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



©Kevin Jamieson 2017

K-means

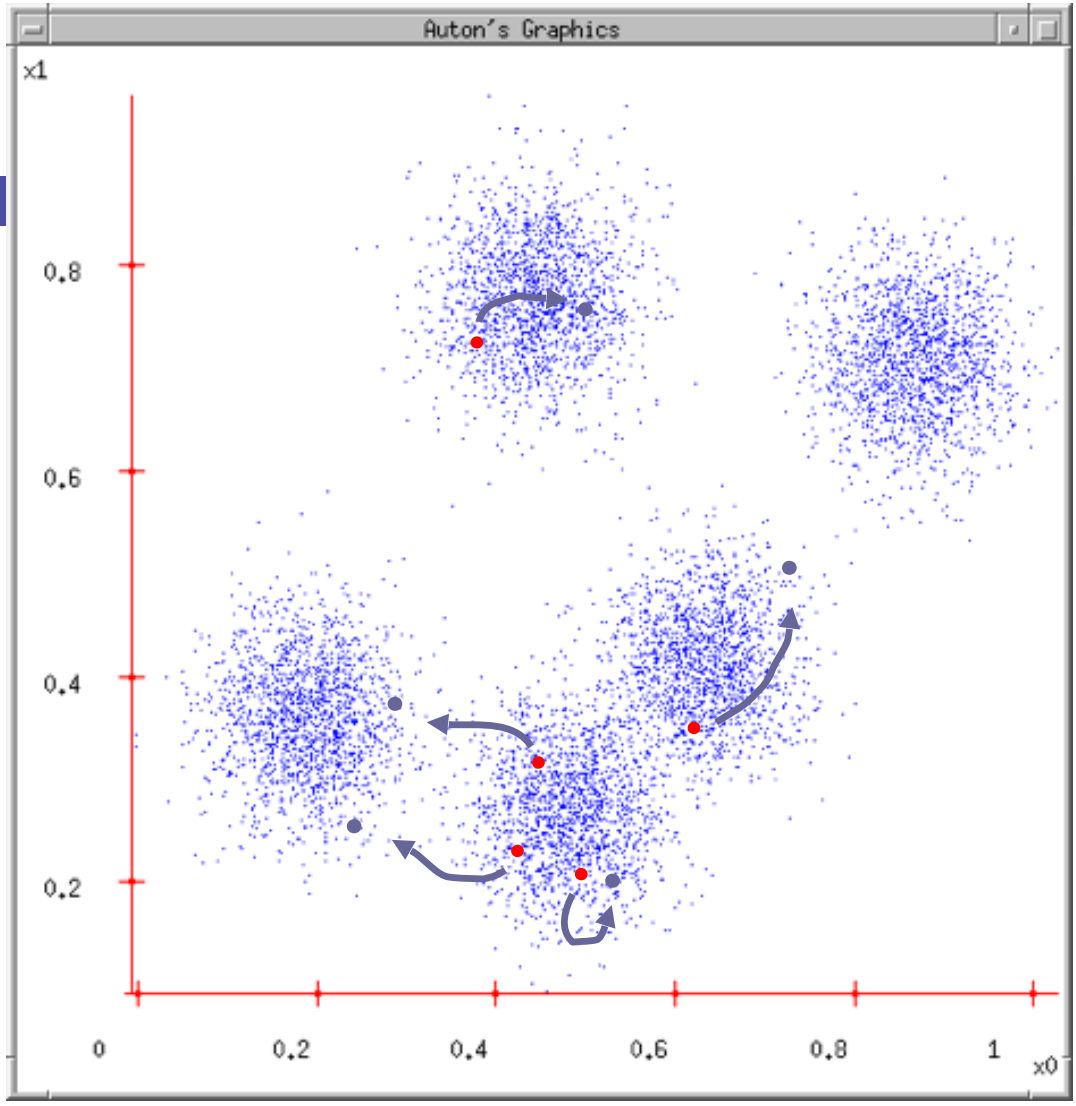
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



©Kevin Jamieson 2017

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



©Kevin Jamieson 2017

K-means

- Randomly initialize *k* centers
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$
- **Classify**: Assign each point j∈{1,…N} to nearest center:
  - $C^{(t)}(j) \leftarrow \arg\min_i ||\mu_i - x_j||^2$
- **Recenter**: μ_i becomes centroid of its point:
  - $\mu_i^{(t+1)} \leftarrow \arg\min_\mu \sum_{j:C(j)=i} ||\mu - x_j||^2$
  - Equivalent to $\mu_i \leftarrow$ average of its points!

What is K-means optimizing?

- Potential function F(μ,C) of centers μ and point allocations C:
  - $$F(\mu, C) = \sum_{j=1}^{N} ||\mu_{C(j)} - x_j||^2$$

- Optimal K-means:
  - $\min_{\mu} \min_{C} F(\mu, C)$

©Kevin Jamieson 2017

Does K-means converge??? Part 1

- Optimize potential function:

$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- Fix μ, optimize C

$$\min_{c_i} ||x_j - \mu_{c_i}||_2^2$$

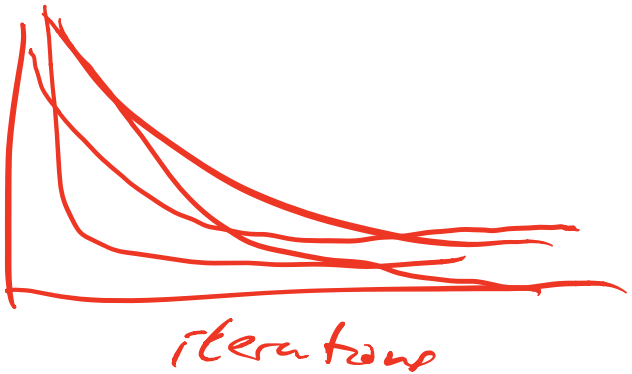
Does K-means converge??? Part 2

- Optimize potential function:

$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- Fix C, optimize μ

$$\text{argmin}_{\mu} \sum_{i \in C} ||\mu - x_i||_2^2 \Rightarrow \mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$



Vector Quantization, Fisher Vectors

Vector Quantization (for compression)

1. Represent image as grid of patches
2. Run k-means on the patches to build code book
3. Represent each patch as a code word.

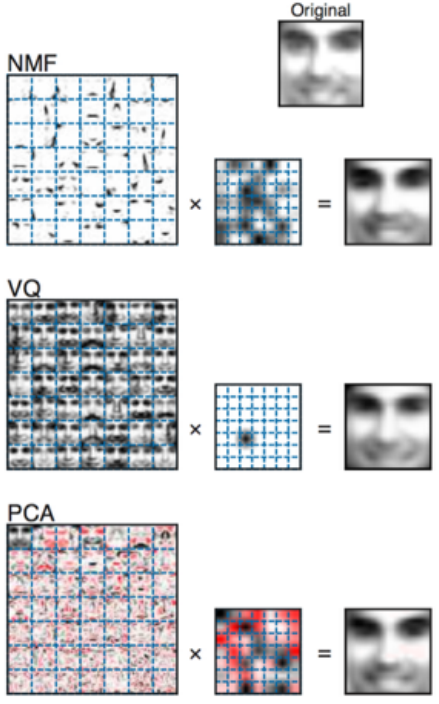
FIGURE 14.9. *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a 1024×1024 grayscale image at 8 bits per pixel. The center image is the result of 2 × 2 block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*

Vector Quantization, Fisher Vectors

Vector Quantization (for compression)

1. Represent image as grid of patches
2. Run k-means on the patches to build code book
3. Represent each patch as a code word.

FIGURE 14.9. *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a 1024×1024 grayscale image at 8 bits per pixel. The center image is the result of 2 × 2 block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*



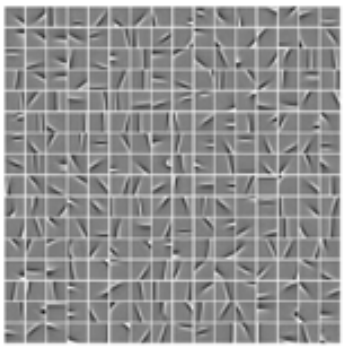
Vector Quantization, Fisher Vectors

Vector Quantization (for compression)

1. Represent image as grid of patches
2. Run k-means on the patches to build code book
3. Represent each patch as a code word.

FIGURE 14.9. *Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a 1024×1024 grayscale image at 8 bits per pixel. The center image is the result of 2 × 2 block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel*

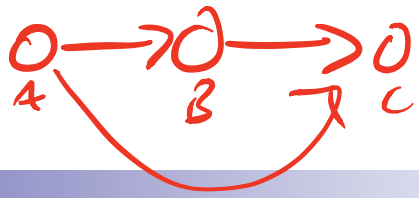
Typical output of k-means on patches



Similar reduced representation can be used as a feature vector

Coates, Ng, *Learning Feature Representations with K-means*, 2012

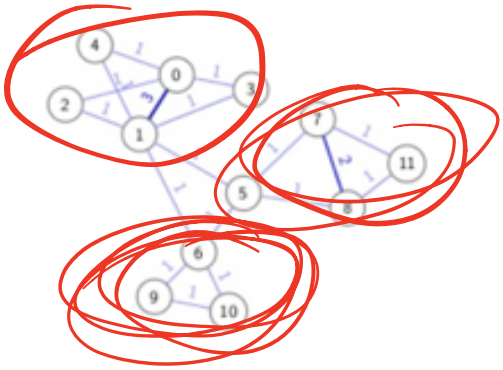
Spectral Clustering



Adjacency matrix: $\mathbf{W}$

$\mathbf{W}_{i,j} = \text{weight of edge } (i,j)$

$$\mathbf{D}_{i,i} = \sum_{j=1}^{n} \mathbf{W}_{i,j} \qquad \mathbf{L} = \mathbf{D} - \mathbf{W}$$



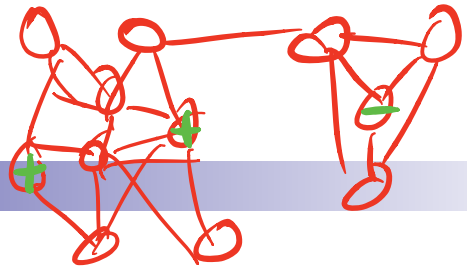
Given feature vectors, could construct:

- k-nearest neighbor graph with weights in {0,1}
- weighted graph with arbitrary *similarities* $\mathbf{W}_{i,j} = e^{-\gamma||x_i - x_j||^2}$

Let $f \in \mathbb{R}^n$ be a function over the nodes

$$\begin{aligned} \mathbf{f}^T\mathbf{L}\mathbf{f} &= \sum_{i=1}^{N} g_i f_i^2 - \sum_{i=1}^{N}\sum_{i'=1}^{N} f_i f_{i'} w_{ii'} \\ &= \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N} w_{ii'}(f_i - f_{i'})^2. \end{aligned}$$

Spectral Clustering



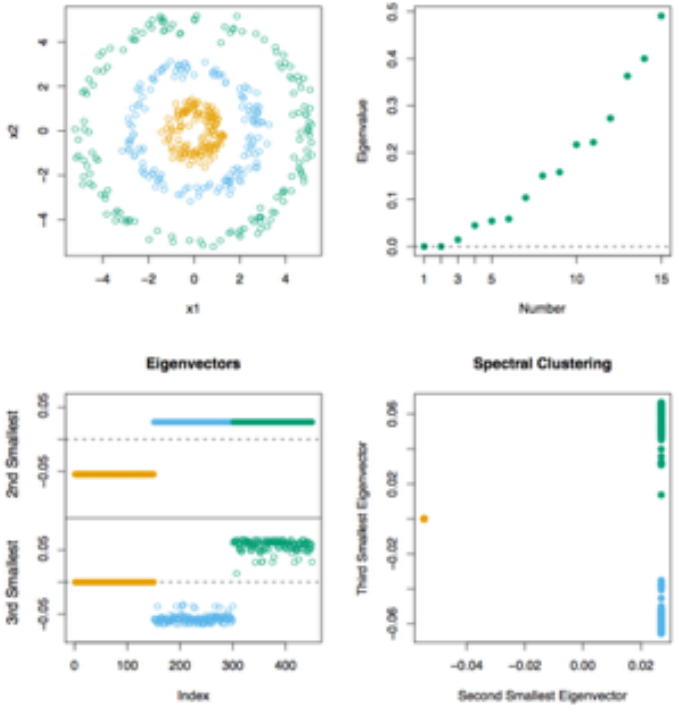
Adjacency matrix: $\mathbf{W}$

$\mathbf{W}_{i,j}$ = weight of edge (i,j)

$$\mathbf{D}_{i,i} = \sum_{j=1}^{n} \mathbf{W}_{i,j} \qquad \mathbf{L} = \mathbf{D} - \mathbf{W}$$

Given feature vectors, could construct:

- (k=10)-nearest neighbor graph with weights in {0,1}



Popular to use the Laplacian $\mathbf{L}$ or its normalized form $\widetilde{\mathbf{L}} = I - \mathbf{D}^{-1}\mathbf{W}$ as a regularizer for learning over graphs

$$\min_f \sum_{i \in \text{Observed}} (y_i - f_i)^2 + \lambda f^\top L f$$