

# Kernels

Machine Learning – CSE446  
Kevin Jamieson  
University of Washington

May 13, 2019

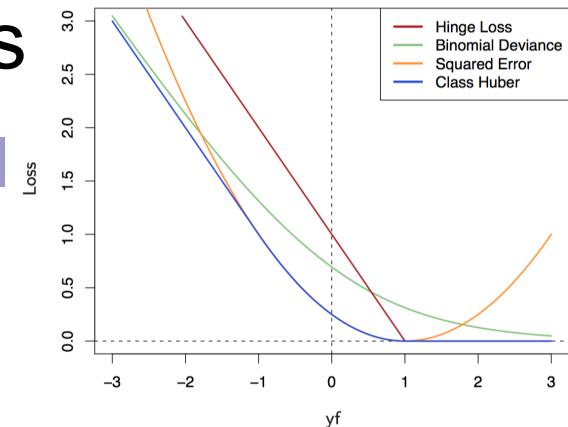
# Machine Learning Problems

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d \quad y_i \in \mathbb{R}$$

- Learning a model's parameters:

Each  $\ell_i(w)$  is convex.



$$\sum_{i=1}^n \ell_i(w)$$

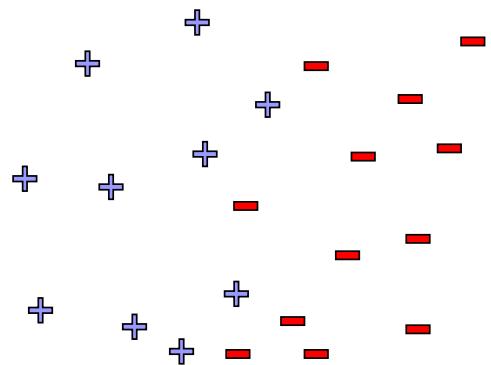
Hinge Loss:  $\ell_i(w) = \max\{0, 1 - y_i x_i^T w\}$

Logistic Loss:  $\ell_i(w) = \log(1 + \exp(-y_i x_i^T w))$

Squared error Loss:  $\ell_i(w) = (y_i - x_i^T w)^2$

All in terms of inner products! Even nearest neighbor can use inner products!

# What if the data is not linearly separable?



Use features of features  
of features of features....

$$\phi(x) : \mathbb{R}^d \rightarrow \mathbb{R}^p$$

Write least squares in  
terms of  $\Phi$ :

$$\hat{w} = \arg \min_w \sum_{i=1}^n (y_i - \phi(x_i)^T w)^2 + \lambda ||w||_2^2$$

# Kernel Trick

$$\underline{\phi} = [\phi(x_1) \dots \phi(x_n)]^T \in \mathbb{R}^{n \times p}$$

$$\widehat{w} = \arg \min_w \sum_{i=1}^n (y_i - \phi(x_i)^T w)^2 + \lambda \|w\|_2^2$$

$$\widehat{w}^T \phi(x)$$

$$= \arg \min_w \|y - \underline{\Phi}^T w\|_2^2 + \lambda \|w\|_2^2$$

$$w = \underline{\Phi}^T \alpha$$

**Claim:** We can write  $\widehat{w} = \underline{\Phi}^T \widehat{\alpha}$  with

$$\widehat{\alpha} = \arg \min_{\alpha} \|y - \underbrace{\underline{\Phi} \underline{\Phi}^T}_{n \times n} \alpha\|_2^2 + \lambda \alpha^T \underbrace{\underline{\Phi} \underline{\Phi}^T}_{n \times n} \alpha$$

$$[\underline{\Phi} \underline{\Phi}^T]_{i,j} = \phi(x_i)^T \phi(x_j)$$

$$\widehat{w}^T \phi(x)$$

# Kernel Trick

$$\begin{aligned}\widehat{w} &= \arg \min_w \sum_{i=1}^n (y_i - \phi(x_i)^T w)^2 + \lambda \|w\|_2^2 \\ &= \arg \min_w \|y - \Phi^\top w\|_2^2 + \lambda \|w\|_2^2\end{aligned}$$

**Claim:** We can write  $\widehat{w} = \Phi^\top \widehat{\alpha}$  with

$$\widehat{\alpha} = \arg \min_\alpha \|y - \Phi \Phi^\top \alpha\|_2^2 + \lambda \alpha^\top \Phi \Phi^\top \alpha$$

Proof: write  $\widehat{w} = P_\Phi \widehat{w} + (I - P_\Phi) \widehat{w}$  where  $P_\Phi$  is the projection onto  $\{\Phi^\top \beta : \beta \in \mathbb{R}^n\}$ . Use facts:

i)  $\Phi^\top \widehat{w} = \Phi^\top P_\Phi \widehat{w}$

ii)  $\|\widehat{w}\|_2^2 = \|P_\Phi \widehat{w}\|_2^2 + \|(I - P_\Phi) \widehat{w}\|_2^2$

to argue  $\|(I - P_\Phi) \widehat{w}\|_2^2 = 0 \implies \widehat{w} = P_\Phi \widehat{w} = \Phi^\top \widehat{\alpha}$

# Kernel Trick

$$\begin{aligned}\hat{w} &= \arg \min_w \sum_{i=1}^n (y_i - \phi(x_i)^T w)^2 + \lambda \|w\|_2^2 \\ &= \arg \min_w \|y - \Phi^\top w\|_2^2 + \lambda \|w\|_2^2\end{aligned}$$

**Claim:** We can write  $\hat{w} = \Phi^\top \hat{\alpha}$  with

$$\hat{\alpha} = \arg \min_\alpha \|y - \Phi \Phi^\top \alpha\|_2^2 + \lambda \alpha^\top \Phi \Phi^\top \alpha$$

$$= \arg \min_\alpha \|y - \mathbf{K} \alpha\|_2^2 + \lambda \alpha^\top \mathbf{K} \alpha$$

$$\mathbf{K}_{i,j} = [\Phi \Phi^\top]_{i,j} = \underbrace{\phi(x_i)^\top \phi(x_j)}_{\text{def}} = \underbrace{K(x_i, x_j)}$$

# General Solution

$$(\phi\phi^\top)^\top = (\phi^\top)^\top\phi^\top = \phi\phi^\top$$

$$(AB)^\top = B^\top A^\top$$

$$K^\top = K$$

$$\mathbf{K}_{i,j} = [\Phi\Phi^\top]_{i,j} = \phi(x_i)^\top \phi(x_j) = K(x_i, x_j)$$

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{K}\alpha - y\|_2^2 + \lambda \alpha^\top \mathbf{K}\alpha$$

$$\nabla_\alpha (-) = 0 = 2K^\top(K\alpha - y) + 2\lambda K\alpha$$

$$= 2K[K\alpha - y + \lambda I\alpha]$$

$$= 2K[(K + \lambda I)\alpha - y]$$

$$\hat{\alpha} = (K + \lambda I)^{-1}y$$

# General Solution

$$\mathbf{K}_{i,j} = [\Phi\Phi^\top]_{i,j} = \phi(x_i)^\top \phi(x_j) = K(x_i, x_j)$$

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{K}\alpha - y\|_2^2 + \lambda \alpha^\top \mathbf{K}\alpha$$
$$0 = \nabla_{\alpha}(\cdot) = 2\mathbf{K}^\top (\mathbf{K}\alpha - y) + 2\lambda \mathbf{K}\alpha$$
$$= 2\mathbf{K}[(\mathbf{K} + \lambda I)\alpha - y]$$
$$\Phi_{\phi(x)} = \begin{bmatrix} \phi(x_1) & \cdots & \phi(x_n) \end{bmatrix}^\top \phi(x)$$
$$= \begin{bmatrix} K(x_1, x) \\ \vdots \\ K(x_n, x) \end{bmatrix}$$

$$\hat{\alpha} = (\mathbf{K} + \lambda I)^{-1}y$$

Predict new point  $x$  as:

$$\hat{w}^\top \phi(x) = \hat{\alpha}^\top \Phi \phi(x) = \sum_{i=1}^n K(x_i, x) \hat{\alpha}_i$$

# General Solution

$$\mathbf{K}_{i,j} = [\Phi\Phi^\top]_{i,j} = \phi(x_i)^\top \phi(x_j) = K(x_i, x_j)$$

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{K}\alpha - y\|_2^2 + \lambda \alpha^\top \mathbf{K}\alpha$$

$$\begin{aligned} 0 &= \nabla_{\alpha}(\cdot) = 2\mathbf{K}^\top(\mathbf{K}\alpha - y) + 2\lambda\mathbf{K}\alpha \\ &= 2\mathbf{K}[(\mathbf{K} + \lambda I)\alpha - y] \end{aligned}$$

$$\boxed{\hat{\alpha} = (\mathbf{K} + \lambda I)^{-1}y}$$

Thus, we have that:  $\hat{w} = (\Phi^\top\Phi + \lambda I_p)^{-1}\Phi^\top y$

but also that:  $\hat{w} = \Phi^\top\hat{\alpha} = \Phi^\top(\Phi\Phi^\top + \lambda I_n)^{-1}y$

# Why regularization?

$$\mathbf{K}_{i,j} = [\Phi\Phi^\top]_{i,j} = \phi(x_i)^\top \phi(x_j) = K(x_i, x_j)$$

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{K}\alpha - y\|_2^2 + \lambda \alpha^\top \mathbf{K}\alpha$$

$$\hat{\alpha} = (\mathbf{K} + \lambda I)^{-1}y$$

$$\hat{w}^\top \phi(x) = \hat{\alpha}^\top \Phi(x) = \sum_{i=1}^n K(x_i, x) \hat{\alpha}_i$$

What if  $\lambda = 0$ ?

# Common kernels

Linear :  $K(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v}$

- Polynomials of degree exactly d

$$K(\mathbf{u}, \mathbf{v}) = (\underbrace{\mathbf{u} \cdot \mathbf{v}}_d)^d$$

- Polynomials of degree up to d

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v} + 1)^d$$

- Gaussian (squared exponential) kernel

$$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|_2^2}{2\sigma^2}\right)$$

(RBF)

- Sigmoid

$$K(\mathbf{u}, \mathbf{v}) = \tanh(\eta \mathbf{u} \cdot \mathbf{v} + \nu)$$

# Mercer's Theorem

- When do we have a valid Kernel  $K(x,x')$ ?

- Sufficient:

$K(x, x')$  is a valid kernel if there exists  $\phi(x)$  such that  $K(x, x') = \phi(x)^T \phi(x')$

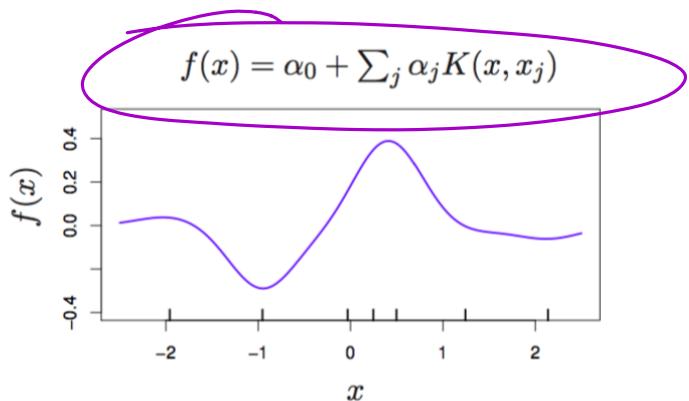
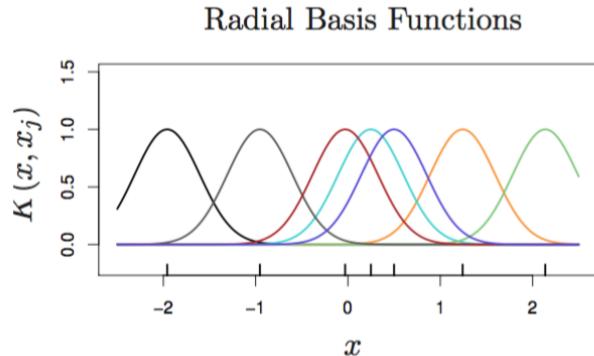
- Mercer's Theorem:

$K(x, x')$  is a valid kernel if and only if  $\mathbf{K}$  is symmetric and positive semi-definite for any pointset  $(x_1, \dots, x_n)$  where  $\mathbf{K}_{i,j} = K(x_i, x_j)$ .

# RBF Kernel

$$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|_2^2}{2\sigma^2}\right)$$

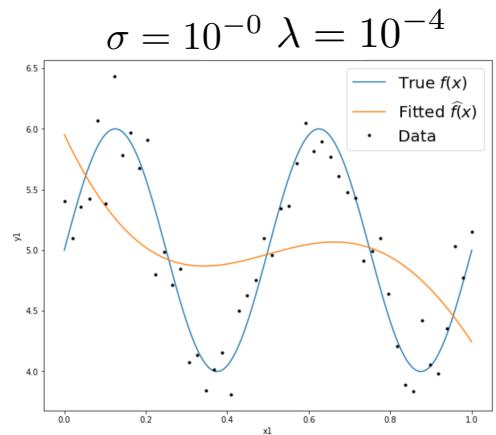
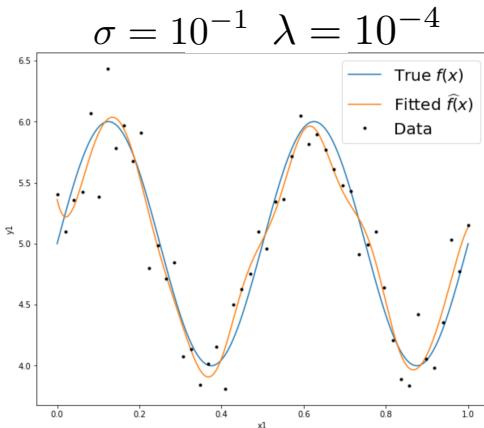
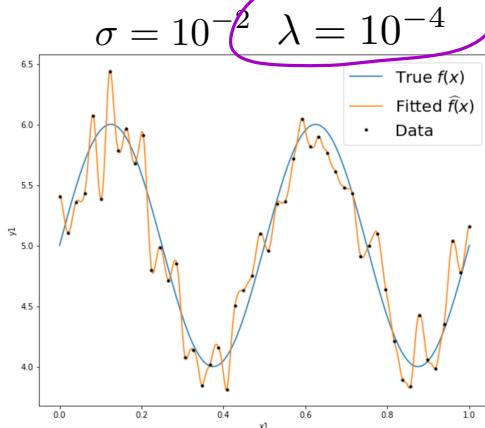
- Note that this is like weighting “bumps” on each point like kernel smoothing but now we **learn** the weights



# RBF Kernel

$$\underline{K(\mathbf{u}, \mathbf{v})} = \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|_2^2}{2\sigma^2}\right)$$

The bandwidth sigma has an enormous effect on fit:

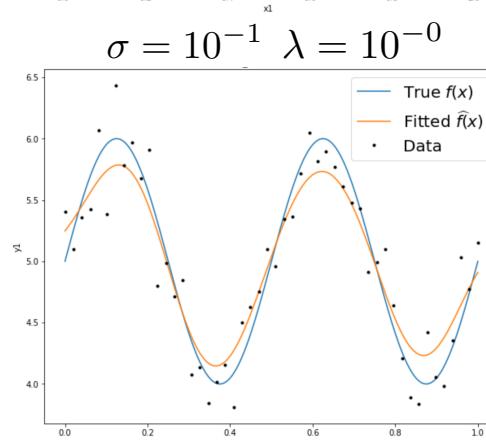
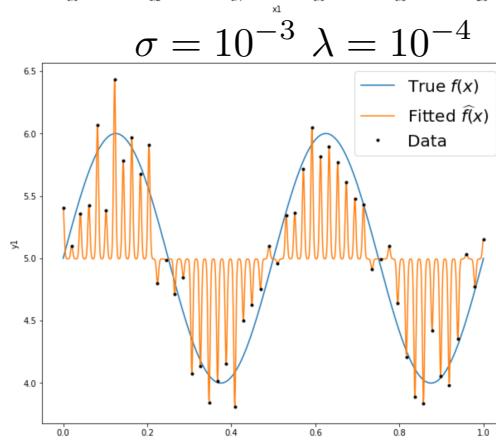
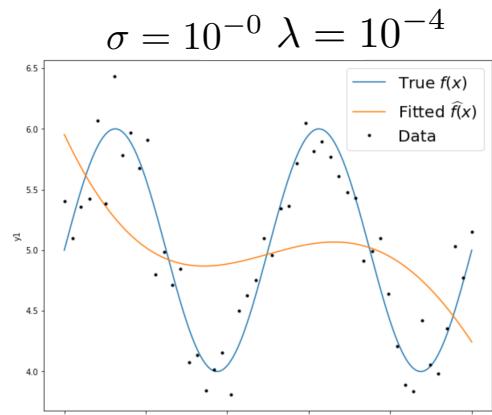
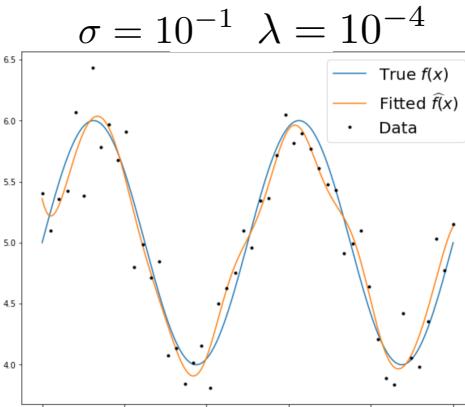
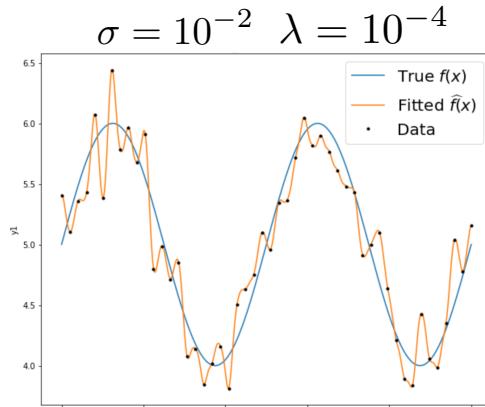


$$\widehat{f}(x) = \sum_{i=1}^n \widehat{\alpha}_i K(x_i, x)$$

# RBF Kernel

$$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|_2^2}{2\sigma^2}\right)$$

The bandwidth sigma has an enormous effect on fit:



$$\hat{f}(x) = \sum_{i=1}^n \hat{\alpha}_i K(x_i, x)$$

# RBF kernel and random features

$$2 \cos(\alpha) \cos(\beta) = \cos(\alpha + \beta) + \cos(\alpha - \beta)$$

$$e^{jz} = \cos(z) + j \sin(z)$$

Recall HW1 where we used the feature map:

$$\phi(x) = \begin{bmatrix} \sqrt{2} \cos(w_1^T x + b_1) \\ \vdots \\ \sqrt{2} \cos(w_p^T x + b_p) \end{bmatrix} \quad w_k \sim \mathcal{N}(0, 2\gamma I) \quad b_k \sim \text{uniform}(0, \pi)$$

$$\mathbb{E}\left[\underbrace{\frac{1}{p} \phi(x)^T \phi(y)}_{\text{RBF kernel}}\right] = \frac{1}{p} \sum_{k=1}^p \mathbb{E}[2 \cos(w_k^T x + b_k) \cos(w_k^T y + b_k)] \quad \leftarrow$$

$$= \mathbb{E}_{w,b}[2 \cos(\underline{w^T x + b}) \cos(\underline{w^T y + b})]$$

$$= \mathbb{E}_{w,b} [\underbrace{\cos(\omega^T (x-y))}_{0} + \cos(\omega^T (x+y) + 2b)]$$

# RBF kernel and random features

$$2 \cos(\alpha) \cos(\beta) = \cos(\alpha + \beta) + \cos(\alpha - \beta)$$

$$e^{jz} = \cos(z) + j \sin(z)$$

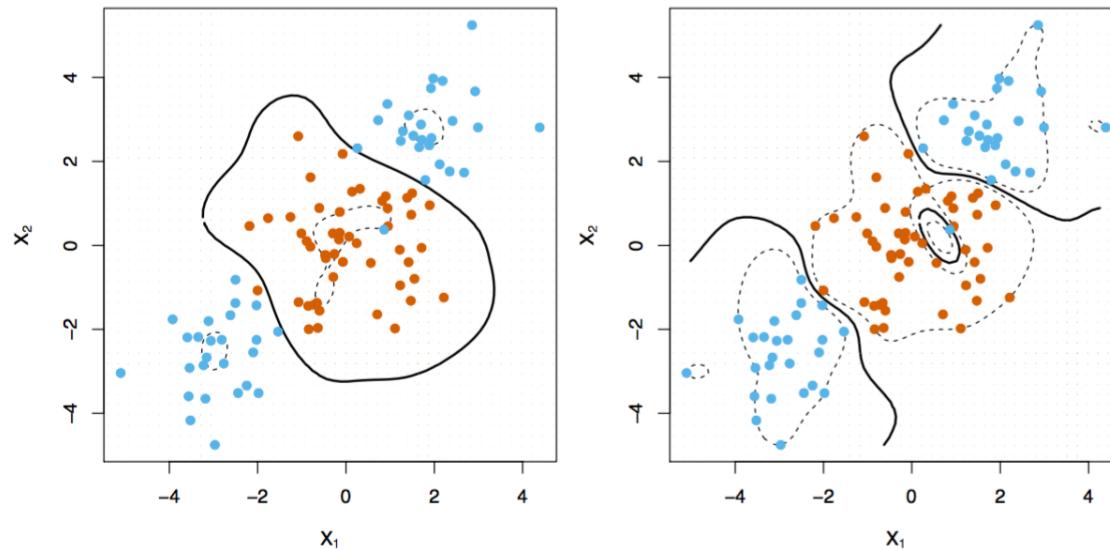
Recall HW1 where we used the feature map:

$$\phi(x) = \begin{bmatrix} \sqrt{2} \cos(w_1^T x + b_1) \\ \vdots \\ \sqrt{2} \cos(w_p^T x + b_p) \end{bmatrix} \quad \begin{aligned} w_k &\sim \mathcal{N}(0, 2\gamma I) \\ b_k &\sim \text{uniform}(0, \pi) \end{aligned}$$

$$\begin{aligned} \mathbb{E}\left[\frac{1}{p} \phi(x)^T \phi(y)\right] &= \frac{1}{p} \sum_{k=1}^p \mathbb{E}[2 \cos(w_k^T x + b_k) \cos(w_k^T y + b_k)] \\ &= \mathbb{E}_{w,b}[2 \cos(w^T x + b) \cos(w^T y + b)] \\ &= \underline{e^{-\gamma ||x-y||_2^2}} \end{aligned} \quad \begin{aligned} &[\text{Rahimi, Recht NIPS 2007}] \\ &\text{“NIPS Test of Time Award, 2018”} \end{aligned}$$

# RBF Classification

$$\widehat{w} = \sum_{i=1}^n \max\{0, 1 - y_i(b + x_i^T w)\} + \lambda \|w\|_2^2$$
$$\min_{\alpha, b} \sum_{i=1}^n \max\{0, 1 - y_i(b + \sum_{j=1}^n \alpha_j \langle x_i, x_j \rangle)\} + \lambda \sum_{i,j=1}^n \alpha_i \alpha_j \underbrace{\langle x_i, x_j \rangle}_{K(x_i, x_j)}$$



# Wait, infinite dimensions?

- Isn't everything separable there? How are we not overfitting?
- Regularization! Fat shattering  $(R/\text{margin})^2$

# String Kernels

Example from Efron and Hastie, 2016

Amino acid sequences of different lengths:

x1      IPTSALVKETLALLSTHRTLLIANETLRIPVPVHKNHQLCTEIFQGIGTLESQTVQGGTV  
ERLFKNLSLIKYYIDGQKKCGEERRRVNQFLDY**LQE**FLGVVMNTEWI

x2      PHRRDLCRSRSIWLARKIRSDLTALTESYVKHQGLWSELTEAER**LQE**NLQAYRTFHVLLA  
RLLEDQQVHFTPTEGDFHQAIHTLLLQVAAFAYQIEELMILLEYKIPRNEADGMLFEKK  
LWGLKV**LQE**LSQWTVRSIHDLRFISSHQTGIP

All subsequences of length 3 (of possible 20 amino acids)  $20^3 = 8,000$

$$h_{\text{LQE}}^3(x_1) = 1 \text{ and } h_{\text{LQE}}^3(x_2) = 2.$$



# Bootstrap

Machine Learning – CSE446  
Kevin Jamieson  
University of Washington

May 13, 2019

# Limitations of CV

- An 80/20 split throws out a relatively large amount of data if only have, say, 20 examples.
- Test error is informative, but how accurate is this number? (e.g., 3/5 heads vs. 30/50)
- How do I get confidence intervals on statistics like the median or variance of a distribution?
- Instead of the error for the entire dataset, what if I want to study the error for a *particular example*  $x$ ?

# Limitations of CV

- An 80/20 split throws out a relatively large amount of data if only have, say, 20 examples.
- Test error is informative, but how accurate is this number? (e.g., 3/5 heads vs. 30/50)
- How do I get confidence intervals on statistics like the median or variance of a distribution?
- Instead of the error for the entire dataset, what if I want to study the error for a *particular example*  $x$ ?

The Bootstrap: Developed by Efron in 1979.

# Bootstrap: basic idea

Given dataset drawn iid samples with CDF  $F_Z \stackrel{(x)}{=} P(Z \leq x)$

$$\mathcal{D} = \{z_1, \dots, z_n\} \stackrel{i.i.d.}{\sim} F_Z$$

We compute a *statistic* of the data to get:  $\hat{\theta} = \underline{t(\mathcal{D})}$

# Bootstrap: basic idea

Given dataset drawn iid samples with CDF  $F_Z$ :

$$\mathcal{D} = \{z_1, \dots, z_n\} \stackrel{i.i.d.}{\sim} F_Z$$

We compute a *statistic* of the data to get:  $\hat{\theta} = t(\mathcal{D})$

For  $b=1, \dots, B$  define the *bth bootstrapped* dataset as drawing  $n$  samples **with replacement** from  $D$

$$\mathcal{D}^{*b} = \{z_1^{*b}, \dots, z_n^{*b}\} \stackrel{i.i.d.}{\sim} \hat{F}_{Z,n}$$

and the *bth bootstrapped statistic* as:  $\underline{\theta}^{*b} = t(\mathcal{D}^{*b})$

# Bootstrap: basic idea

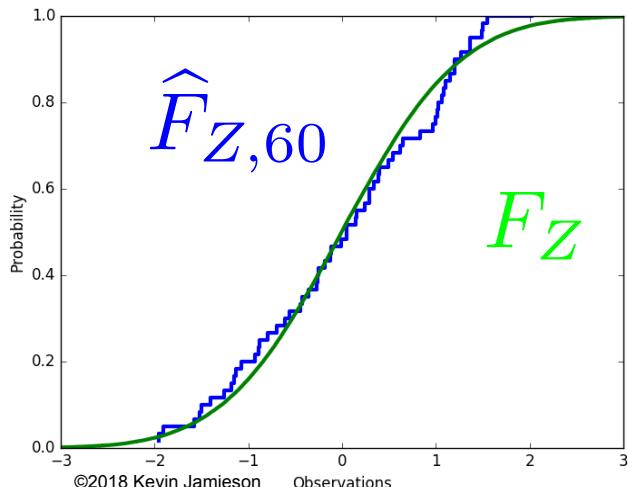
Given dataset drawn iid samples with CDF  $F_Z$ :

$$\mathcal{D} = \{z_1, \dots, z_n\} \stackrel{i.i.d.}{\sim} F_Z \quad \hat{\theta} = t(\mathcal{D})$$

For  $b=1, \dots, B$ , samples sampled **with replacement** from  $D$

$$\mathcal{D}^{*b} = \{z_1^{*b}, \dots, z_n^{*b}\} \stackrel{i.i.d.}{\sim} \hat{F}_{Z,n}(x) \quad \theta^{*b} = t(\mathcal{D}^{*b})$$

$= \frac{1}{n} \sum \mathbb{I}\{z_b \leq x\}$



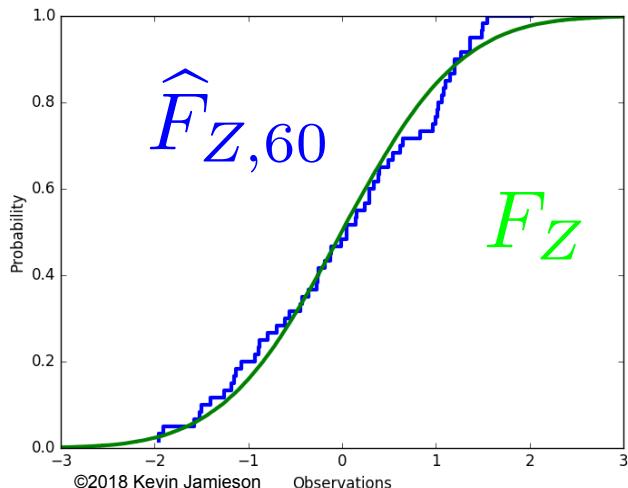
# Bootstrap: basic idea

Given dataset drawn iid samples with CDF  $F_Z$ :

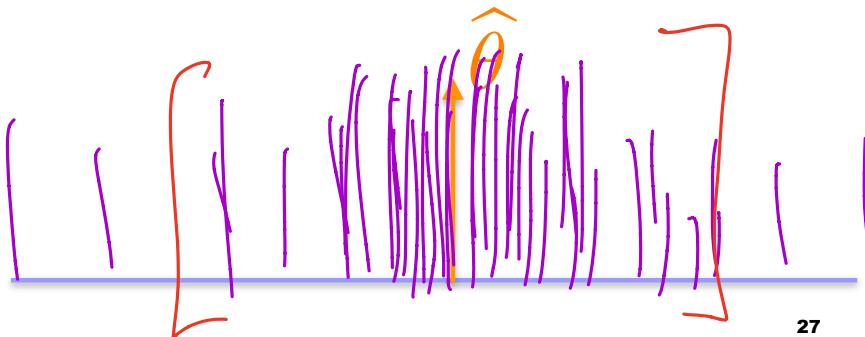
$$\mathcal{D} = \{z_1, \dots, z_n\} \stackrel{i.i.d.}{\sim} F_Z \quad \hat{\theta} = t(\mathcal{D})$$

For  $b=1, \dots, B$ , samples sampled **with replacement** from  $D$

$$\mathcal{D}^{*b} = \{z_1^{*b}, \dots, z_n^{*b}\} \stackrel{i.i.d.}{\sim} \hat{F}_{Z,n} \quad \theta^{*b} = t(\mathcal{D}^{*b})$$



$$\max_{\text{sup}} \sup_x |\hat{F}_n(x) - F(x)| \rightarrow 0 \quad \text{as } n \rightarrow \infty$$



# Applications

$$\theta_1, \theta_2$$

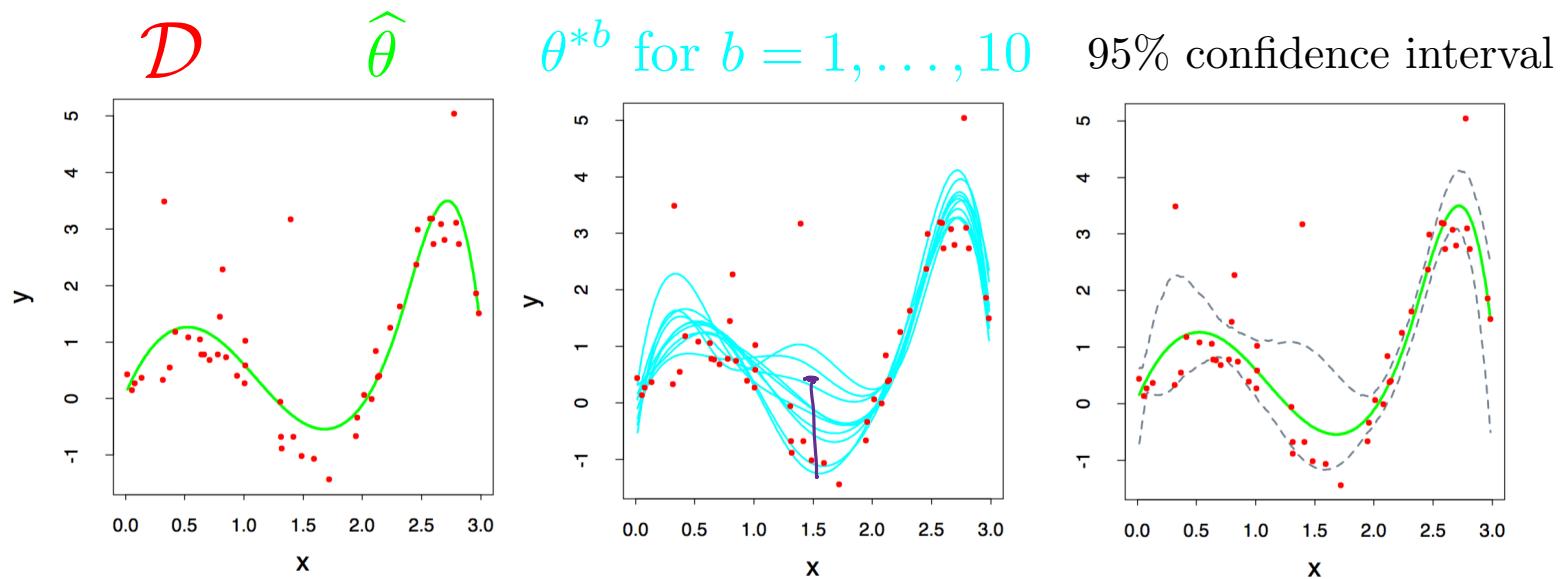
$$\theta_{B2}, \theta_{D1}, \theta_B$$

Common applications of the bootstrap:

- Estimate parameters that escape simple analysis like the variance or median of an estimate
- Confidence intervals
- Estimates of error for a particular example:

$$-5 \quad -4 \quad 7 \quad 9 \quad 10$$

$$1 \quad 2 \quad \dots \quad \dots \quad [0.5SB] \quad [0.95SB] \quad B$$



Figures from Hastie et al

# Takeaways

Advantages:

- Bootstrap is **very** generally applicable. Build a confidence interval around ***anything***
- **Very** simple to use
- Appears to give meaningful results even when the amount of data is very small
- Very strong **asymptotic theory** (as num. examples goes to infinity)

# Takeaways

## Advantages:

- Bootstrap is **very** generally applicable. Build a confidence interval around **anything**
- **Very** simple to use
- Appears to give meaningful results even when the amount of data is very small
- Very strong **asymptotic theory** (as num. examples goes to infinity)

## Disadvantages

- Very few meaningful finite-sample guarantees
- Potentially **computationally intensive**
- Reliability relies on test statistic and rate of convergence of empirical CDF to true CDF, which is unknown
- Poor performance on “extreme statistics” (e.g., the max)

Not perfect, but better than nothing.

# Warm up: risk prediction with logistic regression

- Boss gives you a bunch of data on loans defaulting or not:

$$\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$$

- You model the data as:  $P(Y = y|x, w) = \frac{1}{1 + \exp(-y w^T x)}$
- And compute the maximum likelihood estimator:

$$\hat{w}_{MLE} = \arg \max_w \prod_{i=1}^n P(y_i|x_i, w)$$

For a new loan application  $x$ , boss recommends to give loan if your model says they will repay it with probability at least .95 (i.e. low risk):

$$\text{Give loan to } x \text{ if } \frac{1}{1 + \exp(-\hat{w}_{MLE}^T x)} \geq .95$$

- One year later only half of loans are paid back and the bank folds. What might have happened?

How would you use the bootstrap to do this differently?

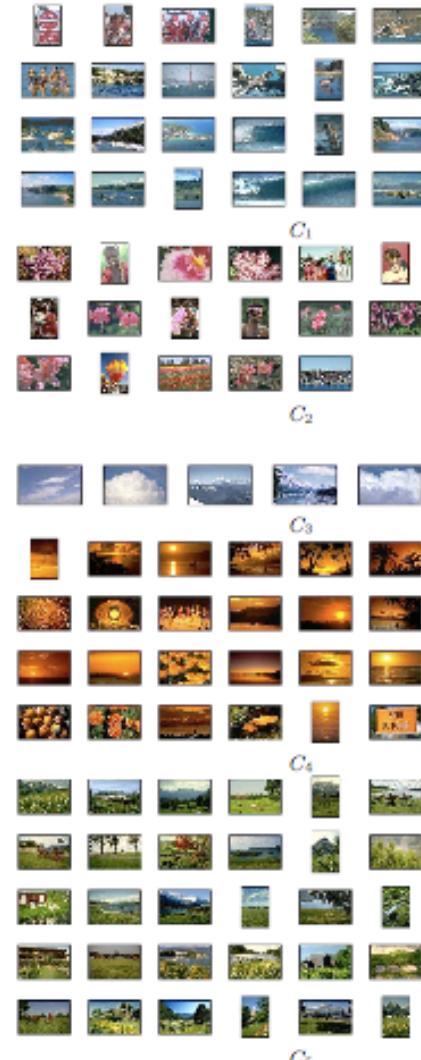
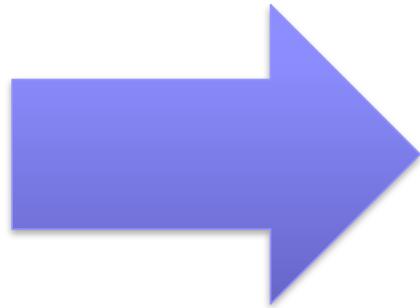


# Clustering K-means

Machine Learning – CSE 446  
Kevin Jamieson  
University of Washington

May 13, 2019

# Clustering images



# Clustering web search results

The screenshot shows the Clusty search interface. The top navigation bar includes links for web, news, images, wikipedia, blogs, jobs, and more. A search bar contains the query "race". Below the search bar are "advanced preferences" and a "Search Results" link.

The left sidebar features a navigation menu with categories like clusters, sources, sites, and a "remix" button. Under "clusters", there is a list of results:

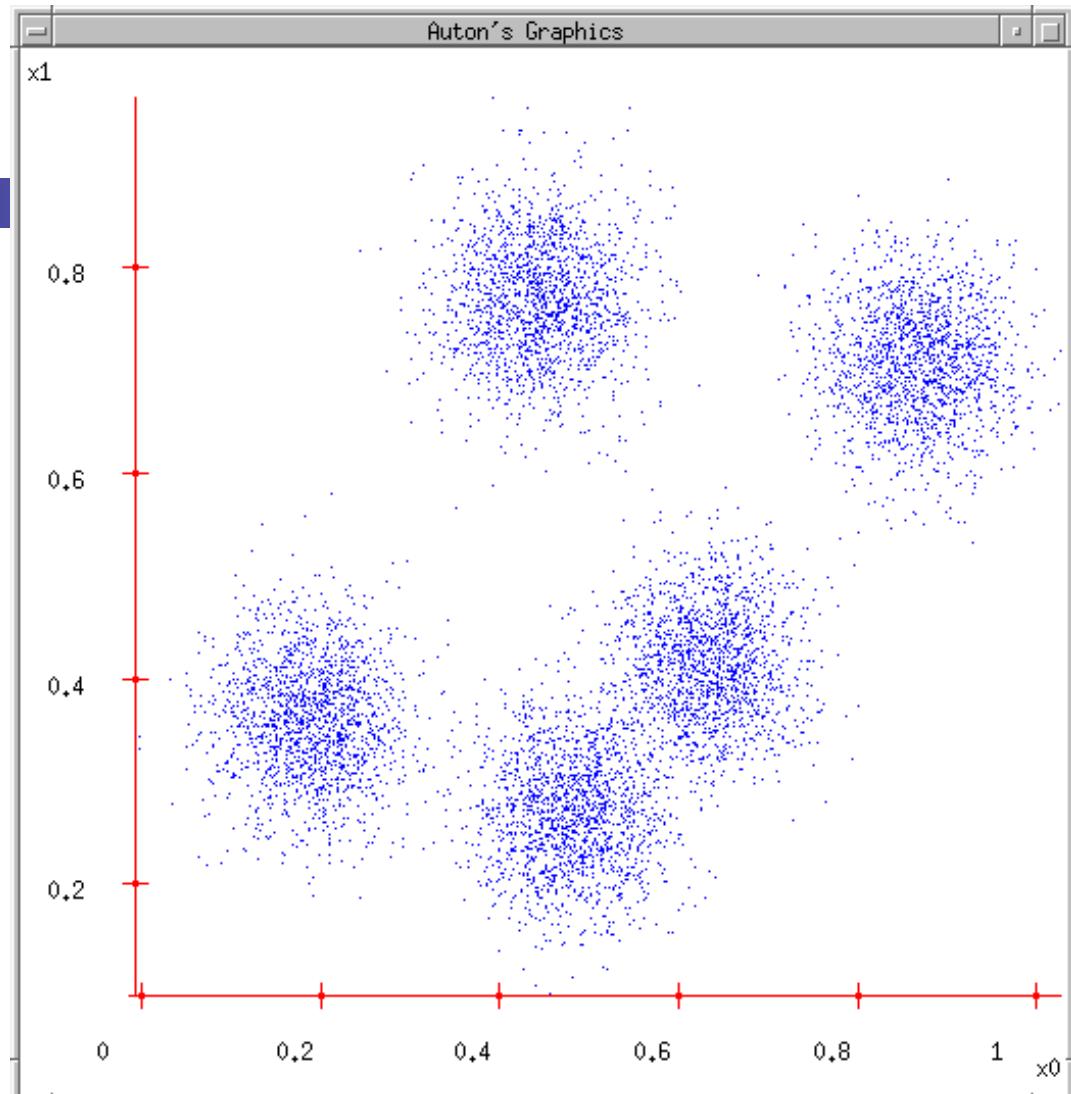
- All Results (238)
- Car (28)
- Race cars (?)
- Photos, Races Scheduled (5)
- Game (4)
- Track (3)
- Nascar (2)
- Equipment And Safety (2)
- Other Topics (?)
- Photos (22)
- Game (14)
- Definition (13)
- Team (18)
- Human (8)
  - Classification Of Human (2)
  - Statement, Evolved (2)
  - Other Topics (4)
- Weekend (8)
- Ethnicity And Race (7)
- Race for the Cure (8)
- Race Information (8)
- more | all clusters

Below the sidebar, a message states "Cluster Human contains 8 documents." The main content area displays 7 search results:

- Race (classification of human beings) - Wikipedia, the free ...**  
The term **race** or racial group usually refers to the concept of dividing **humans** into populations or groups on the basis of various sets of characteristics. The most widely used **human** racial categories are based on visible traits (especially skin color, cranial or facial features and hair texture), and self-identification. Conceptions of **race**, as well as specific ways of grouping **races**, vary by culture and over time, and are often controversial for scientific as well as social and political reasons. History · Modern debates · Political and ...  
[en.wikipedia.org/wiki/Race\\_\(classification\\_of\\_human\\_beings\)](http://en.wikipedia.org/wiki/Race_(classification_of_human_beings)) - [cache] - Live, Ask
- Race - Wikipedia, the free encyclopedia**  
General. **Racing** competitions The **Race** (yachting **race**), or La course du millénaire, a no-rules round-the-world sailing event; **Race** (biology), classification of flora and fauna; **Race** (classification of human beings) **Race** and ethnicity in the United States Census, official definitions of "**race**" used by the US Census Bureau; **Race** and genetics, notion of racial classifications based on genetics. Historical definitions of **race**; **Race** (bearing), the inner and outer rings of a rolling-element bearing. **RACE** in molecular biology "Rapid ... General · Surnames · Television · Music · Literature · Video games  
[en.wikipedia.org/wiki/Race](http://en.wikipedia.org/wiki/Race) - [cache] - Live, Ask
- Publications | Human Rights Watch**  
The use of torture, unlawful rendition, secret prisons, unfair trials, ... Risks to Migrants, Refugees, and Asylum Seekers in Egypt and Israel ... In the run-up to the Beijing Olympics in August 2008, ...  
[www.hrw.org/backgrounder/usa/race](http://www.hrw.org/backgrounder/usa/race) - [cache] - Ask
- Amazon.com: Race: The Reality Of Human Differences: Vincent Sarich ...**  
Amazon.com: **Race**: The Reality Of Human Differences: Vincent Sarich, Frank Miele: Books ... From Publishers Weekly Sarich, a Berkeley emeritus anthropologist, and Miele, an editor ...  
[www.amazon.com/Race-Reality-Differences-Vincent-Sarich/dp/0813340861](http://www.amazon.com/Race-Reality-Differences-Vincent-Sarich/dp/0813340861) - [cache] - Live
- APA Statement on Biological Aspects of Race**  
APA Statement on Biological Aspects of **Race** ... Published in the American Journal of Physical Anthropology, vol. 101, pp 569-570, 1996 ... PREAMBLE As scientists who study **human** evolution and variation, ...  
[www.physanth.org/positions/race.html](http://www.physanth.org/positions/race.html) - [cache] - Ask
- race: Definition from Answers.com**  
**race** n. A local geographic or global **human** population distinguished as a more or less distinct group by genetically transmitted physical  
[www.answers.com/topic/race-1](http://www.answers.com/topic/race-1) - [cache] - Live
- Dopefish.com**  
Site for newbies as well as experienced Dopefish followers, chronicling the birth of the Dopefish, its numerous appearances in several computer games, and its eventual take-over of the **human** **race**. Maintained by Mr. Dopefish himself, Joe Siegler of Apogee Software.  
[www.dopefish.com](http://www.dopefish.com) - [cache] - Open Directory

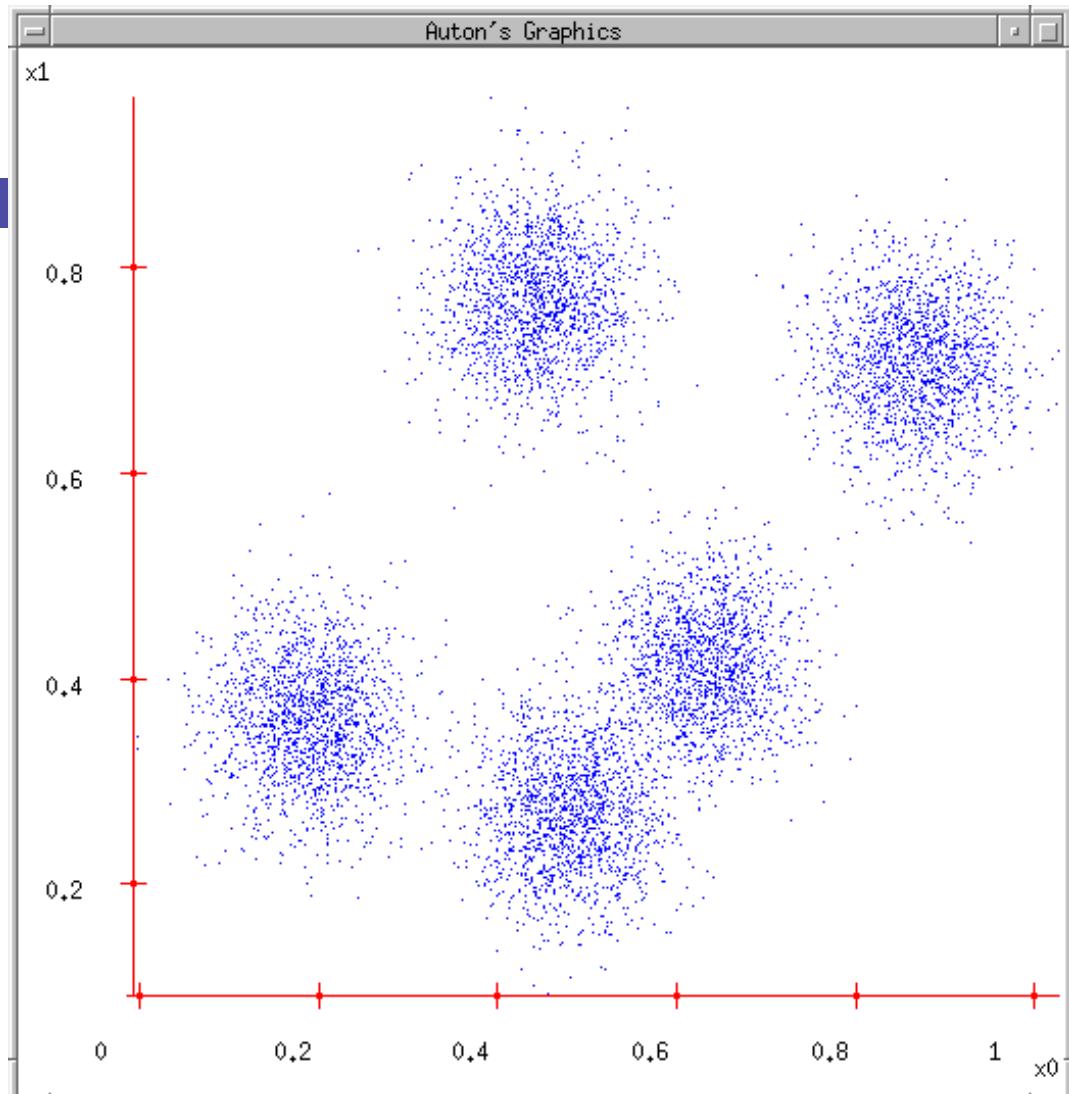
At the bottom, there is a "find in clusters:" input field and a "Find" button.

# Some Data



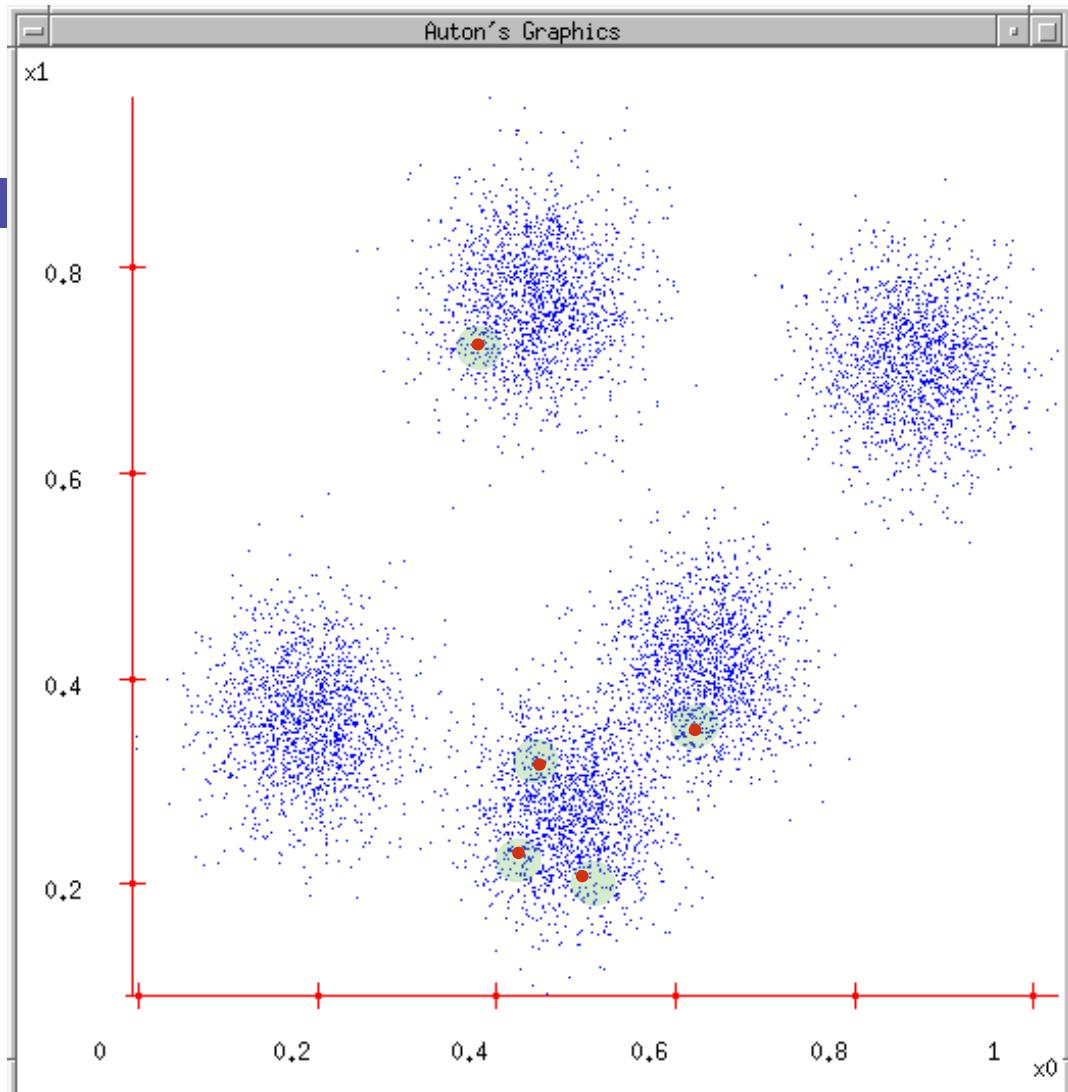
# K-means

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



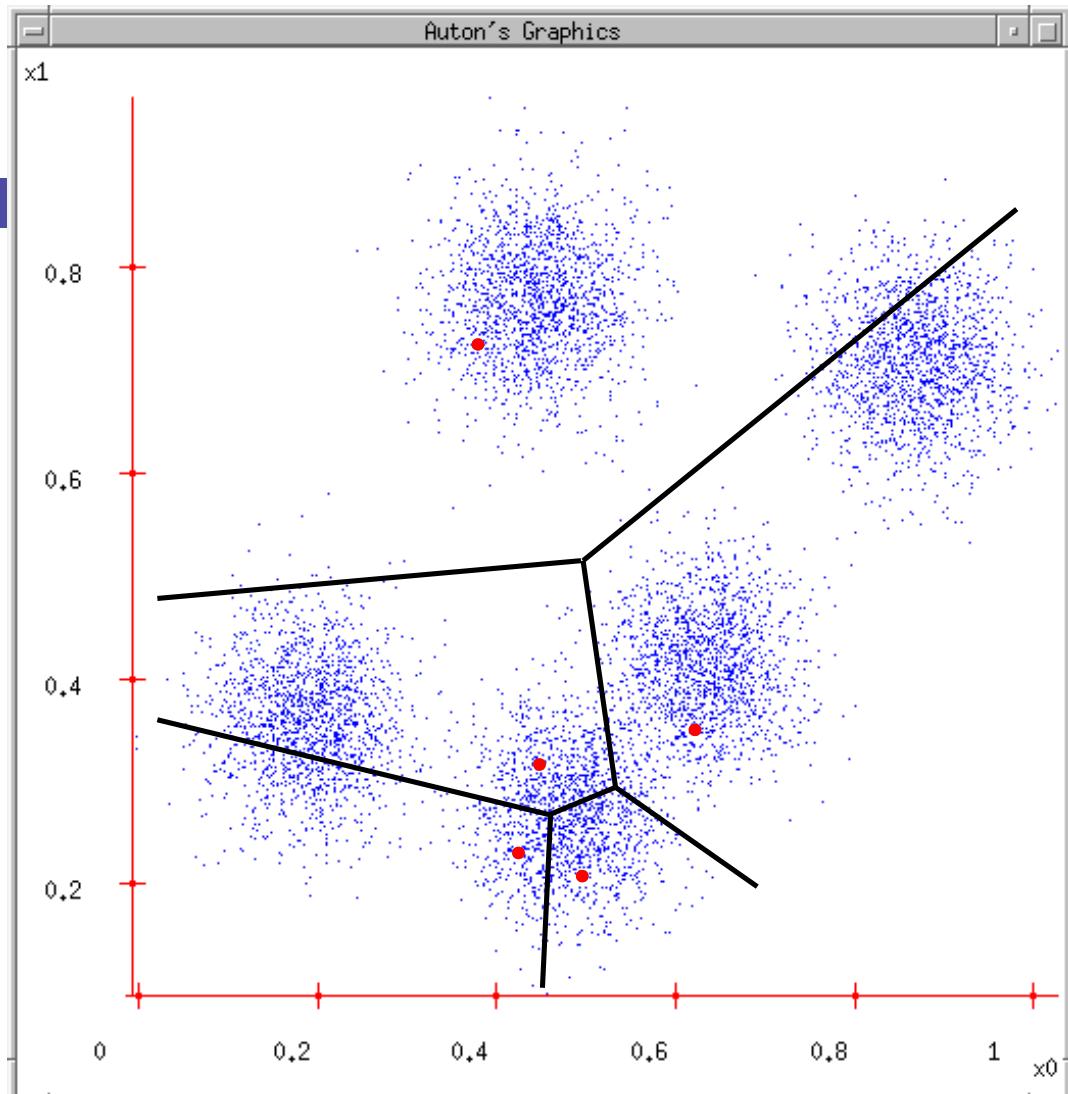
# K-means

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



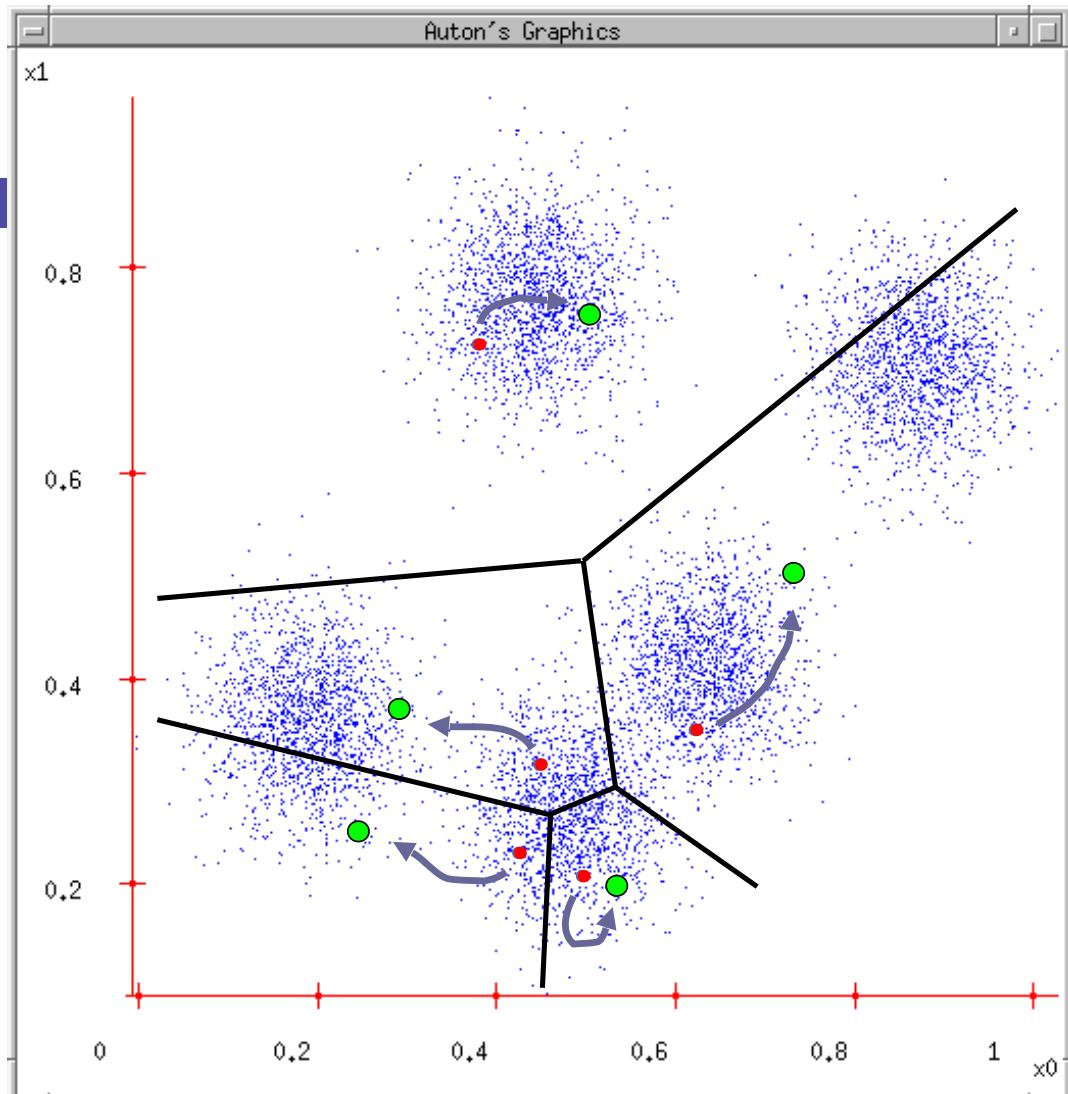
# K-means

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)



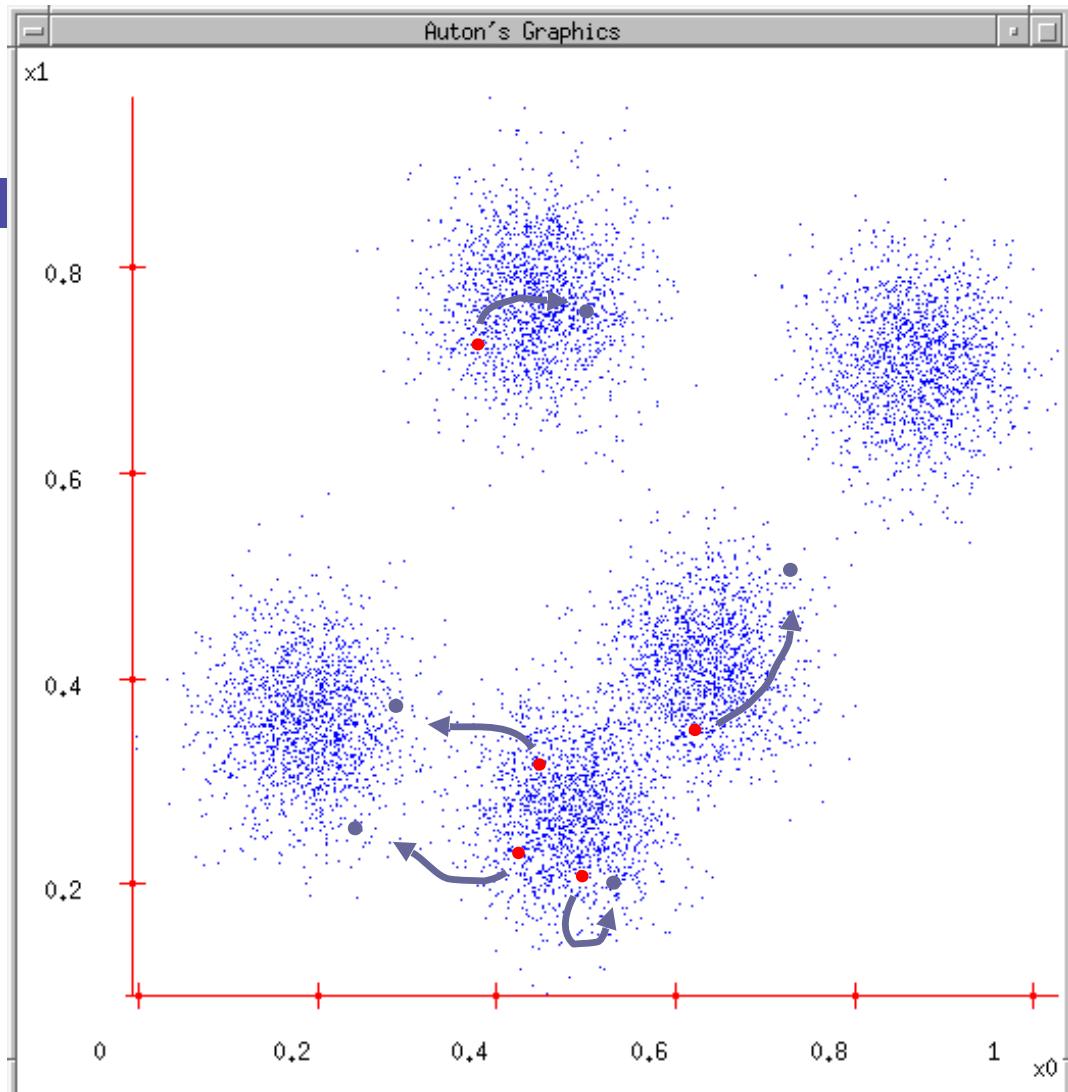
# K-means

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



# 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!



# K-means

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

# 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  $\mu$ , optimize  $C$

# 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  $\mu$

# 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  $\mu$

Objective function decreases at every step  $\Rightarrow$  No configuration repeated  
Only  $\binom{n}{k} \approx n^k$  unique configurations  $\Rightarrow$  convergence in finite # iterations

# 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  $1024 \times 1024$  grayscale image at 8 bits per pixel. The center image is the result of  $2 \times 2$  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  $1024 \times 1024$  grayscale image at 8 bits per pixel. The center image is the result of  $2 \times 2$  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

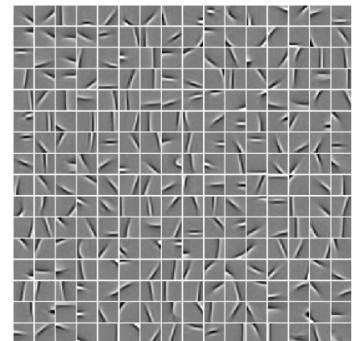
## Vector Quantization (for compression)

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  $1024 \times 1024$  grayscale image at 8 bits per pixel. The center image is the result of  $2 \times 2$  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

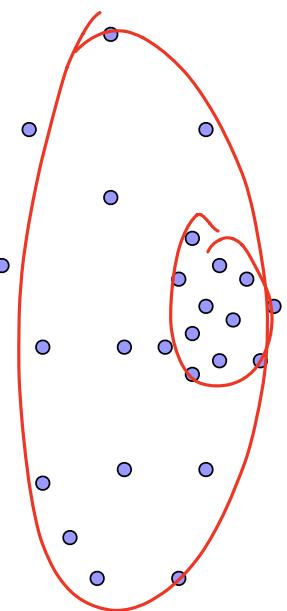
Typical output of k-means  
on patches



Similar reduced representation can be used as a feature vector

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

# (One) bad case for k-means



- Clusters may overlap
- Some clusters may be “wider” than others

# K-means summary

- Greedy algorithm that is sensitive to initialization
- Better initializations than “uniform at random”: Arthur and Vassilvitskii *k-means++: The Advantages of Careful Seeding*
- The centers can take values of data points or arbitrary vectors
- Extremely useful subroutine/pre-processing step for other algorithms
- If clusters vary widely in size, use more sophisticated method (mixture of Gaussians)