

Kernels

Machine Learning – CSE446
Kevin Jamieson
University of Washington

May 13, 2019

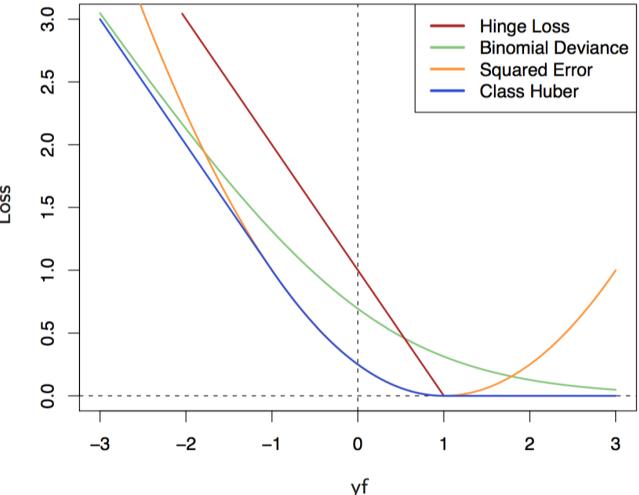
Machine Learning Problems



- 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.



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

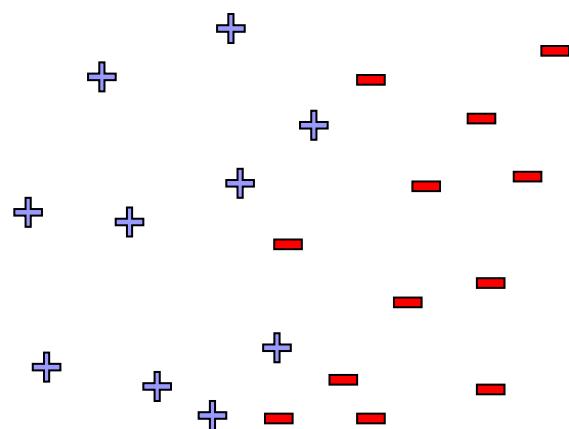
Hinge Loss: $\ell_i(w) = \max\{0, 1 - y_i x_i^T w\}$

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....**

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

Project into some higher dimensional space (or nonlinear transform into some better coordinate system)

Write least squares in terms of Φ :

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

Kernel Trick

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

Phi is a matrix who's ith column is phi(x_i)

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

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

n x n matrix

(i, j) entry of Phi $\Phi^\top = \phi(x_i)^\top \phi(x_j)$

we can write \hat{w} as a linear combination of the $\phi(x_i)$ with weights α_i

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

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

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

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

Kernel Trick

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

$$= \arg \min_w \|y - \Phi^\top w\|_2^2 + \lambda \|w\|_2^2$$

We can write \hat{w} as a linear combination of x_i in the original unkernelized problem

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

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

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

where \mathbf{K} is some function that outputs a single real number

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

General Solution

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

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

solve for alpha by taking the gradient with respect to it and setting it equal to zero.

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

Note \mathbf{K} is symmetric so $\mathbf{K} = \mathbf{K}^\top$

why can I ignore the \mathbf{K} ? How do I know \mathbf{K} does not have a nullspace? Well $\mathbf{K} = \Phi\Phi^\top$ is at least positive semidefinite.

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

solve the linear system
rather than taking the inverse

Predict new point x as:

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

$\Phi \underset{\text{phi}(x)}{\Phi} = [\phi(x_1) \phi(x_2) \dots \phi(x_n)]^\top \underset{\text{phi}(x)}{\Phi}$

General Solution

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

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

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

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

Thus, we have that: $\hat{w} = (\Phi^\top\Phi + \lambda I_d)^{-1}\Phi^\top y$ give exactly the same value
but also that: $\hat{w} = \Phi^\top\hat{\alpha} = \Phi^\top(\Phi\Phi^\top + \lambda I_p)^{-1}y$

Why regularization?

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

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

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

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

What if $\lambda = 0$?

$\text{hat}\{\alpha\} = \mathbf{K}^{-1}y$

You will have zero training error as $w^\top x$ just outputs y
plug in $\mathbf{K}^{-1}y$ here

Common kernels

Maps vectors into another space and performs the inner product in that space.

- Polynomials of degree exactly d

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

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

- Polynomials of degree up to d

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

- Gaussian (squared exponential) kernel

Radial Basis Functions

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

- Sigmoid

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

Each one of these have a corresponding phi function

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 $\mathbf{K}_{i,j} = K(x_i, x_j)$.

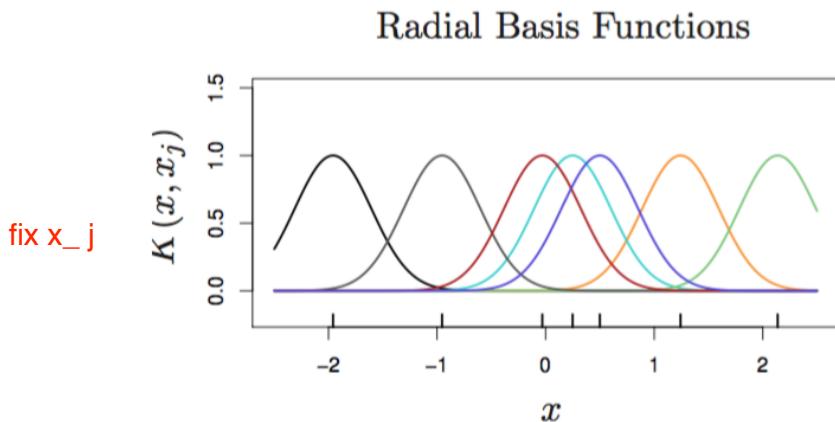
Can write down any kernel function and its valid iff K is symmetric and positive semidefinite (dont need to know for class)

RBF Kernel

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

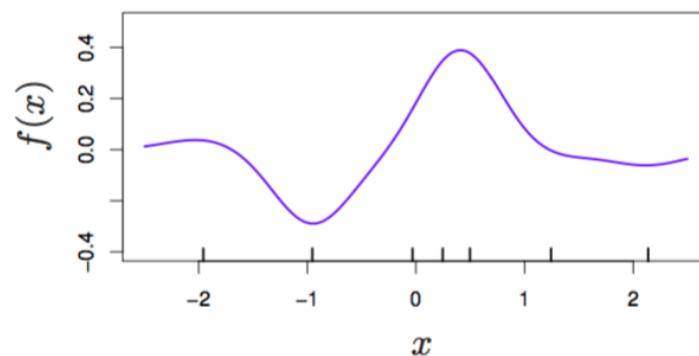
controls the width of these bumps...

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



choose weights alpha so that these “bumps” sum to line up with data.

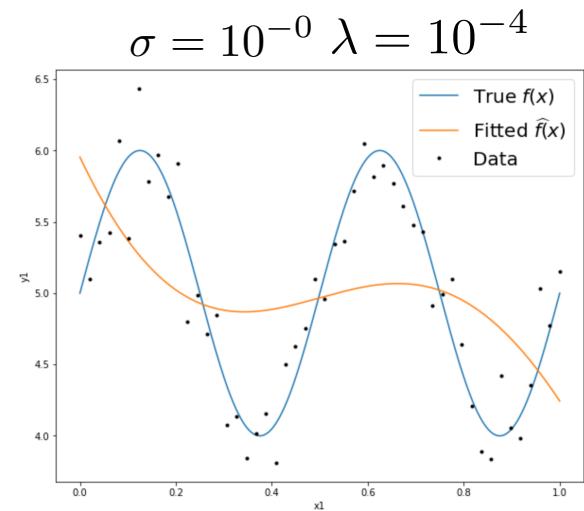
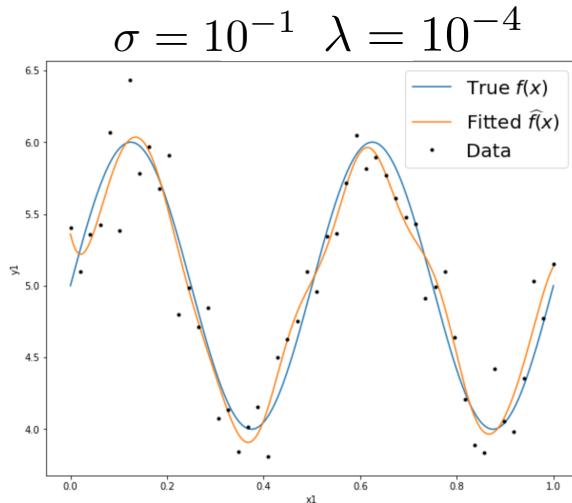
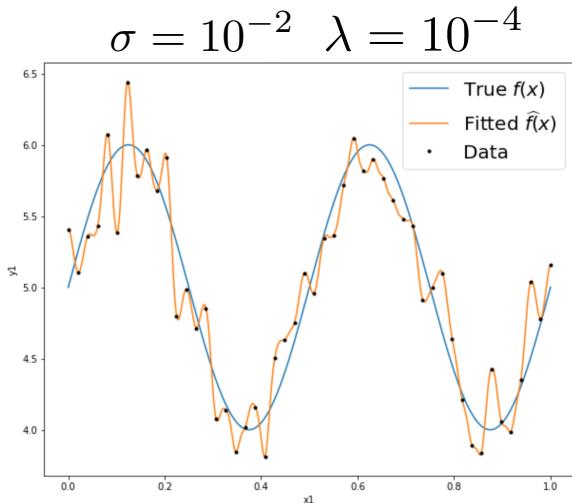
$$f(x) = \alpha_0 + \sum_j \alpha_j K(x, x_j)$$



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:



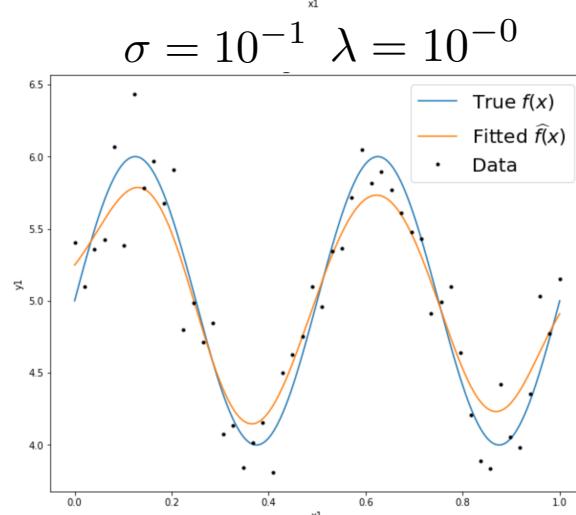
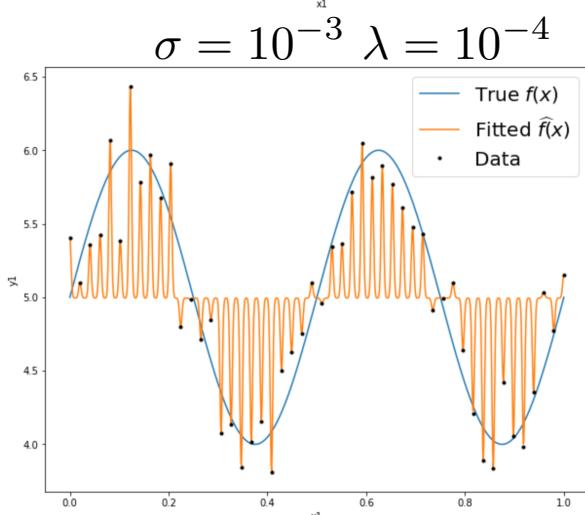
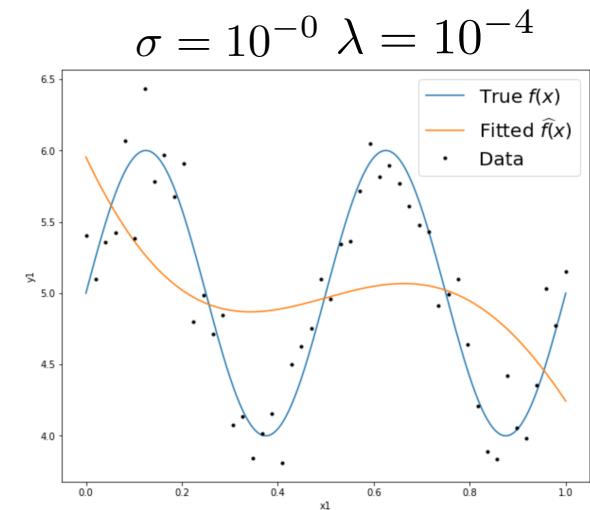
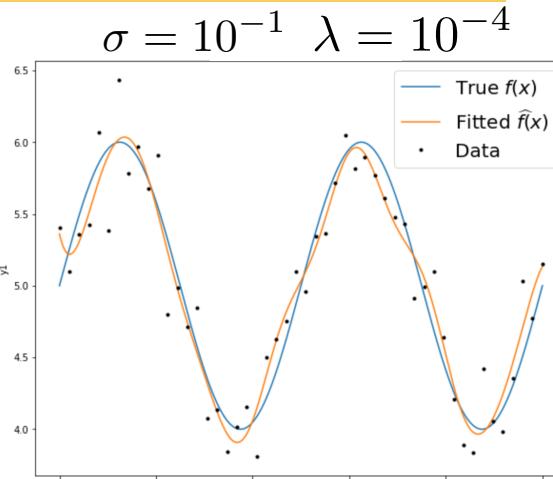
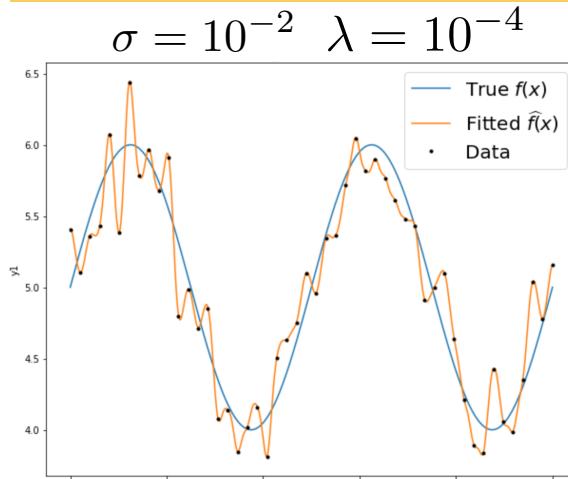
lambda controls amount of regularization (on the alphas)

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

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: another hyperparameter! Tune with cross-validation



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

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:

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

$$\begin{aligned} \mathbb{E}\left[\frac{1}{p} \phi(x)^T \phi(y)\right] &= \frac{1}{p} \sum_{k=1}^p \mathbb{E}[2 \cos(w_k^T x + b_k) \cos(w_k^T y + b_k)] \\ &= \mathbb{E}_{w,b}[2 \cos(w^T x + b) \cos(w^T y + b)] = \mathbb{E}[\cos(w^T(x-y)) + \cos(w^T(x+y) + 2b)] \\ &= e^{-\gamma \|x-y\|_2^2} \end{aligned}$$

[Rahimi, Recht NIPS 2007]

“NIPS Test of Time Award, 2018”

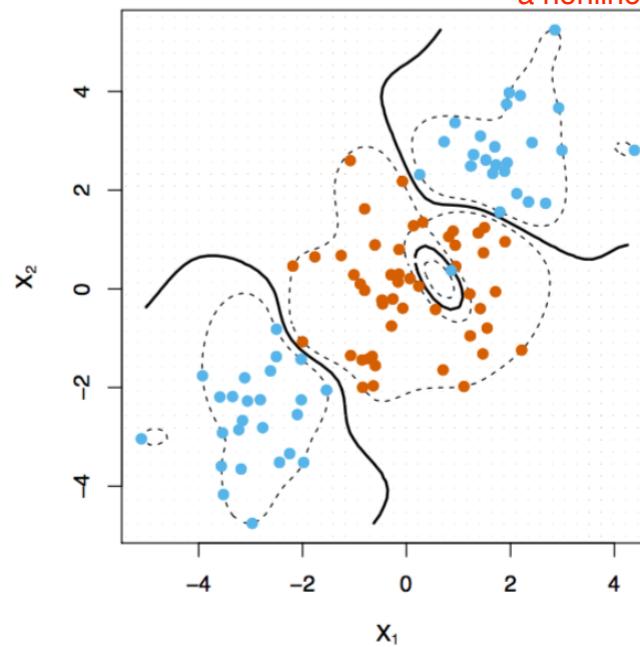
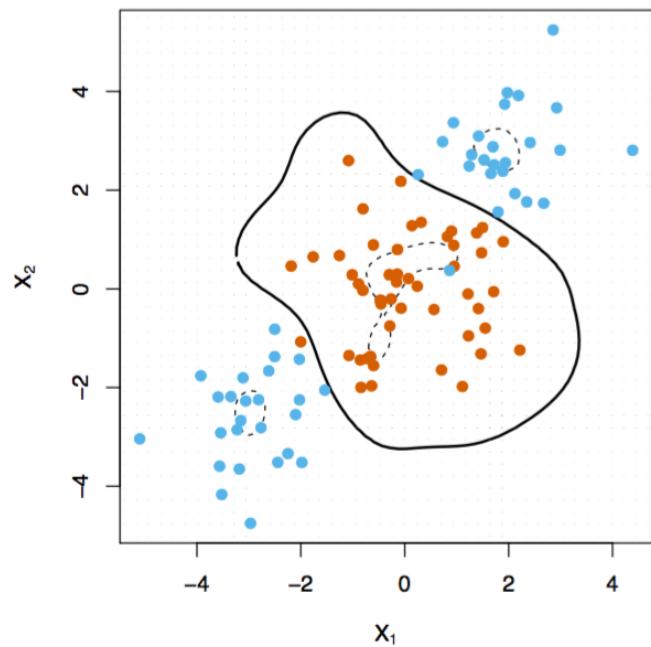
Random Features Trick: gets same result as RBF kernel without having to allocate nearly as much space?

as b is uniform $0, \pi$
then $2b$ is uniform $0, 2\pi$
so the expected value
of this is zero

RBF Classification

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

replace with
 $K(x_i, x_j)$ and generates
a nonlinear decision boundary



Wait, infinite dimensions?

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

String Kernels

Example from Efron and Hastie, 2016

how do I convert this to some space where we can perform separation on them?

Amino acid sequences of different lengths:

x_1

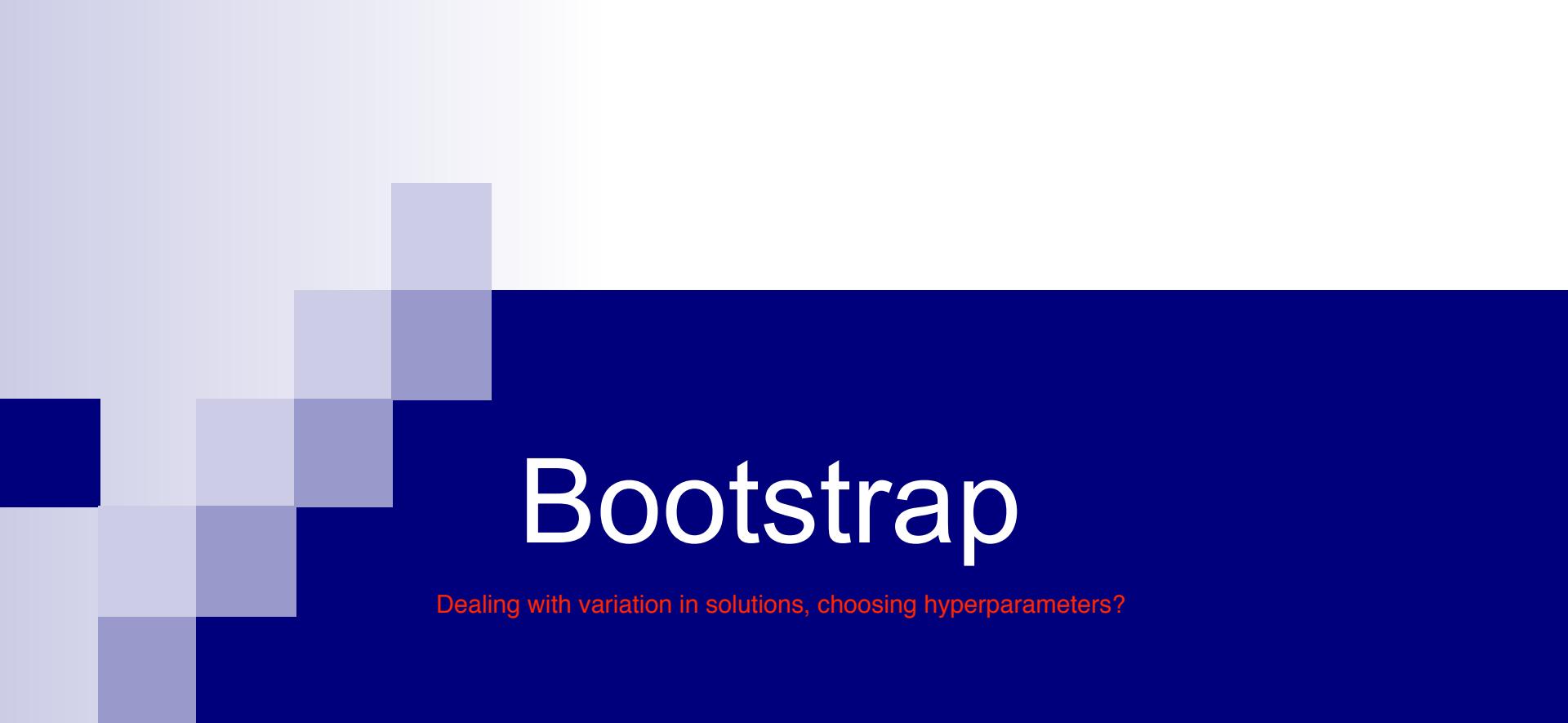
IPT SALV KET LALLS THRT LLI ANET LRI PVP VHK NHQL CTEE IFQ GIG TLE SQT VQGG TV
ERLF KNLS LIKKY IDG QKKC GEERR RVN QFL DY **LQE** FLG VMN TEWI

x_2

PHRR DLCS RSRI WLARK IRSDL TALTES YVKH QGLW SELTEA ER **LQE** NLQAY RTFH VLLA
RLLED QQV HFT PTEG DFHQ AIHT LLQVA AFAY QIEEL MILLEY KIPR NEAD GML FEKK
LWGL KV **LQE** LSQWT VRSI HDLRF ISSH QTGIP

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

Dealing with variation in solutions, choosing hyperparameters?

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*?

on average, we can determine how much error we are going to make, but what about the error a particular data point will have in its prediction?

The Bootstrap: Developed by Efron in 1979.

Bootstrap: basic idea

Given dataset drawn iid samples with CDF F_Z :

$$\mathcal{D} = \{z_1, \dots, z_n\} \stackrel{i.i.d.}{\sim} F_Z \quad \text{drawn from some CDF}$$

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

takes in the dataset D and
outputs some statistic
(i.e. the median of the dataset)

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

drawing from our DATASET
not the overall distribution

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

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

compute an estimate for each one of these

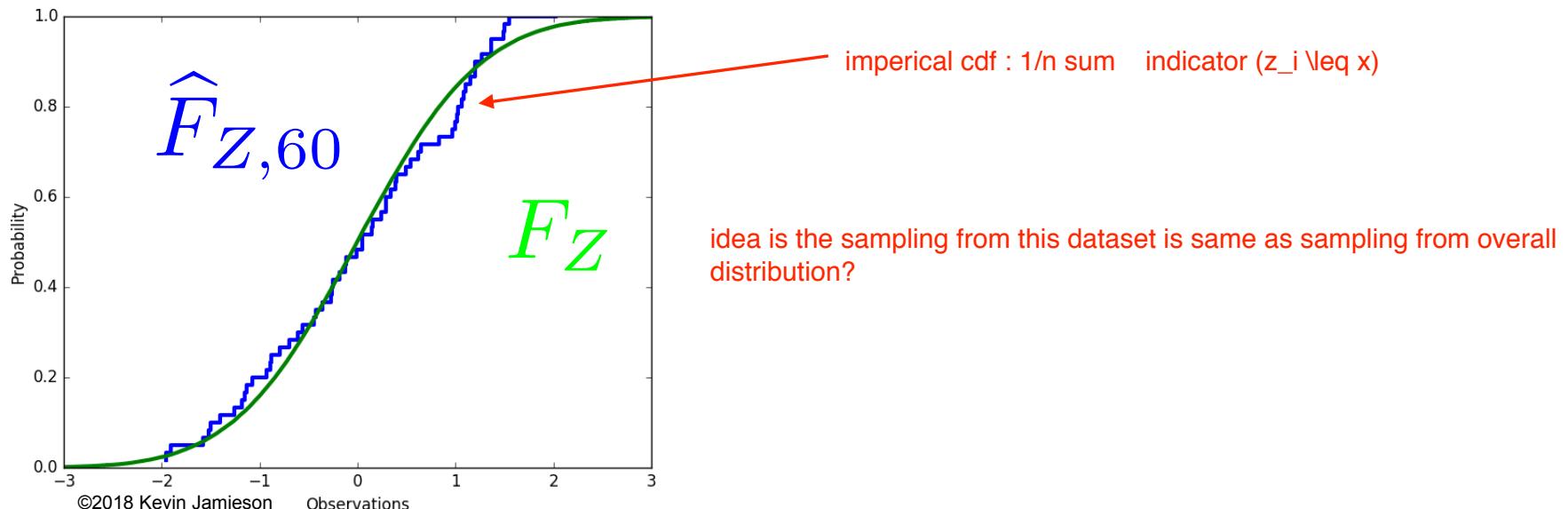
Bootstrap: basic idea

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

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



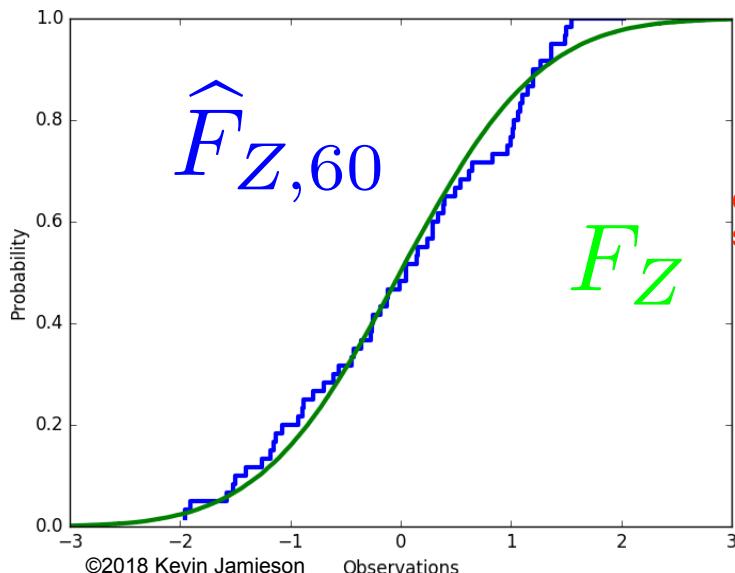
Bootstrap: basic idea

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

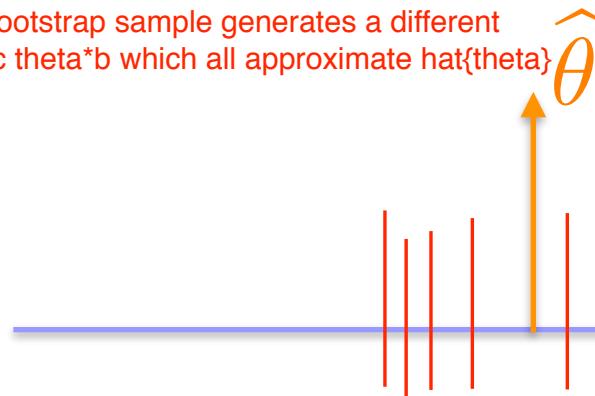
$$\mathcal{D}^{*b} = \{z_1^{*b}, \dots, z_n^{*b}\} \stackrel{i.i.d.}{\sim} \hat{F}_{Z,n} \quad \theta^{*b} = t(\mathcal{D}^{*b})$$



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

error rate goes to zero...

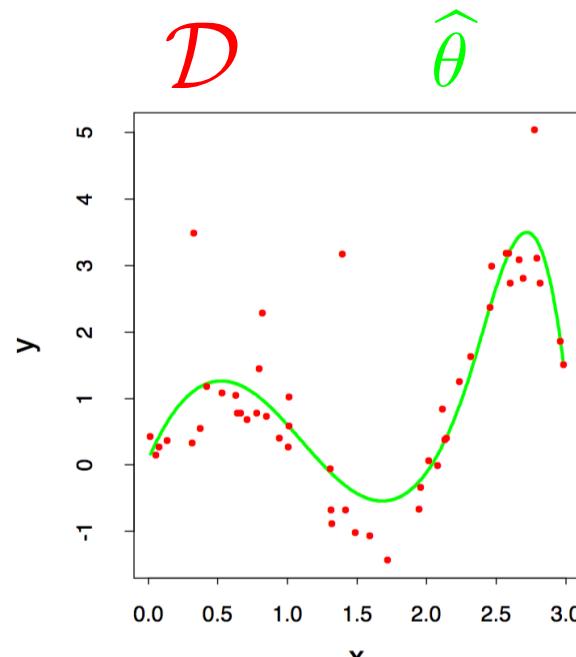
each bootstrap sample generates a different statistic θ^{*b} which all approximate $\hat{\theta}$



Applications

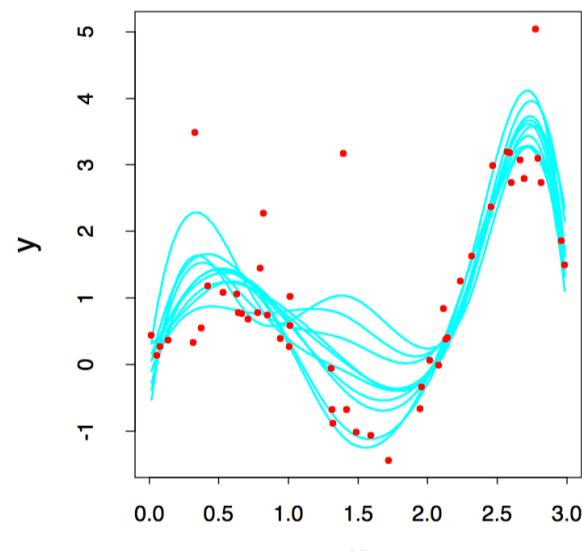
Common applications of the bootstrap:

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

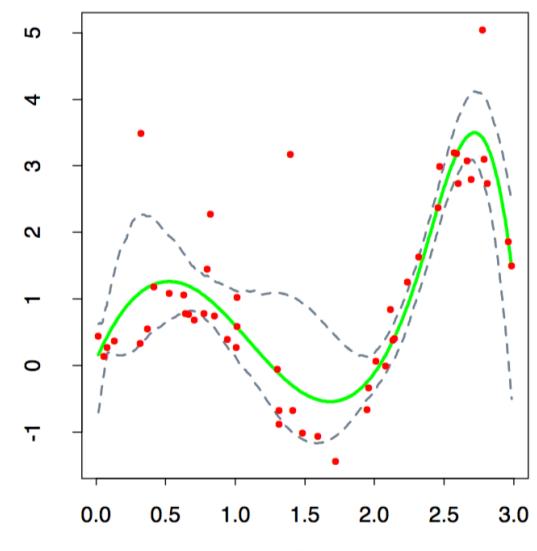


$\hat{\theta}$

θ^{*b} for $b = 1, \dots, 10$



95% confidence interval



dataset generates a single prediction

bootstrap allows us to build these confidence intervals

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):

Give loan to x 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?

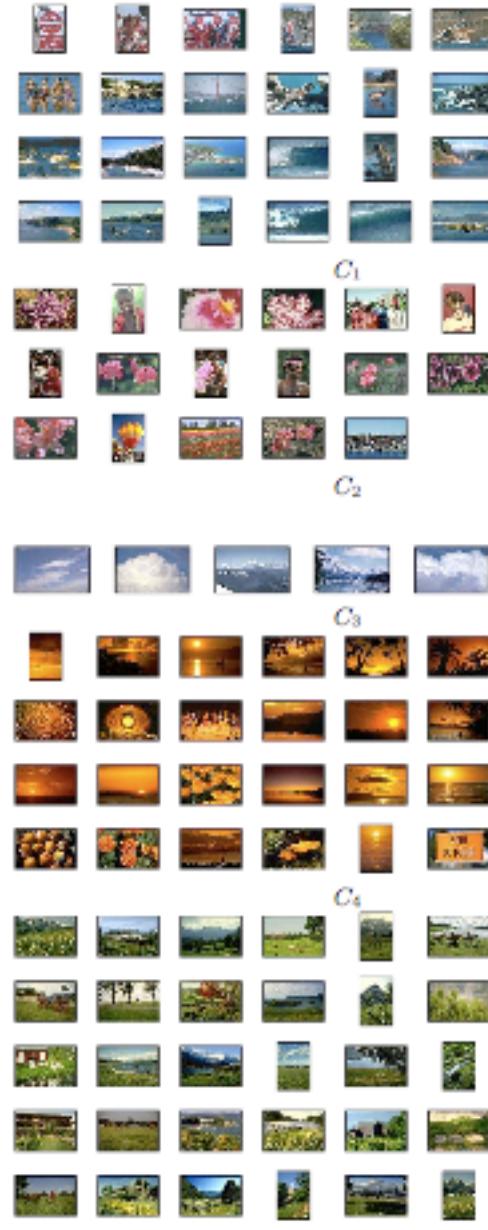
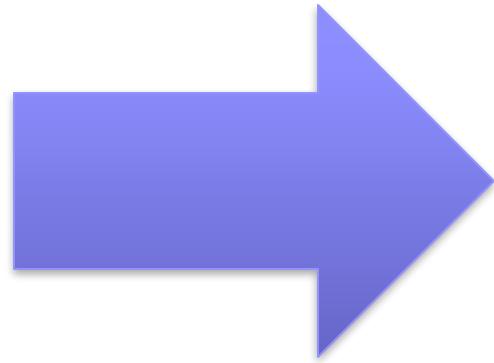


Clustering K-means

Machine Learning – CSE 446
Kevin Jamieson
University of Washington

May 13, 2019

Clustering images



Clustering web search results

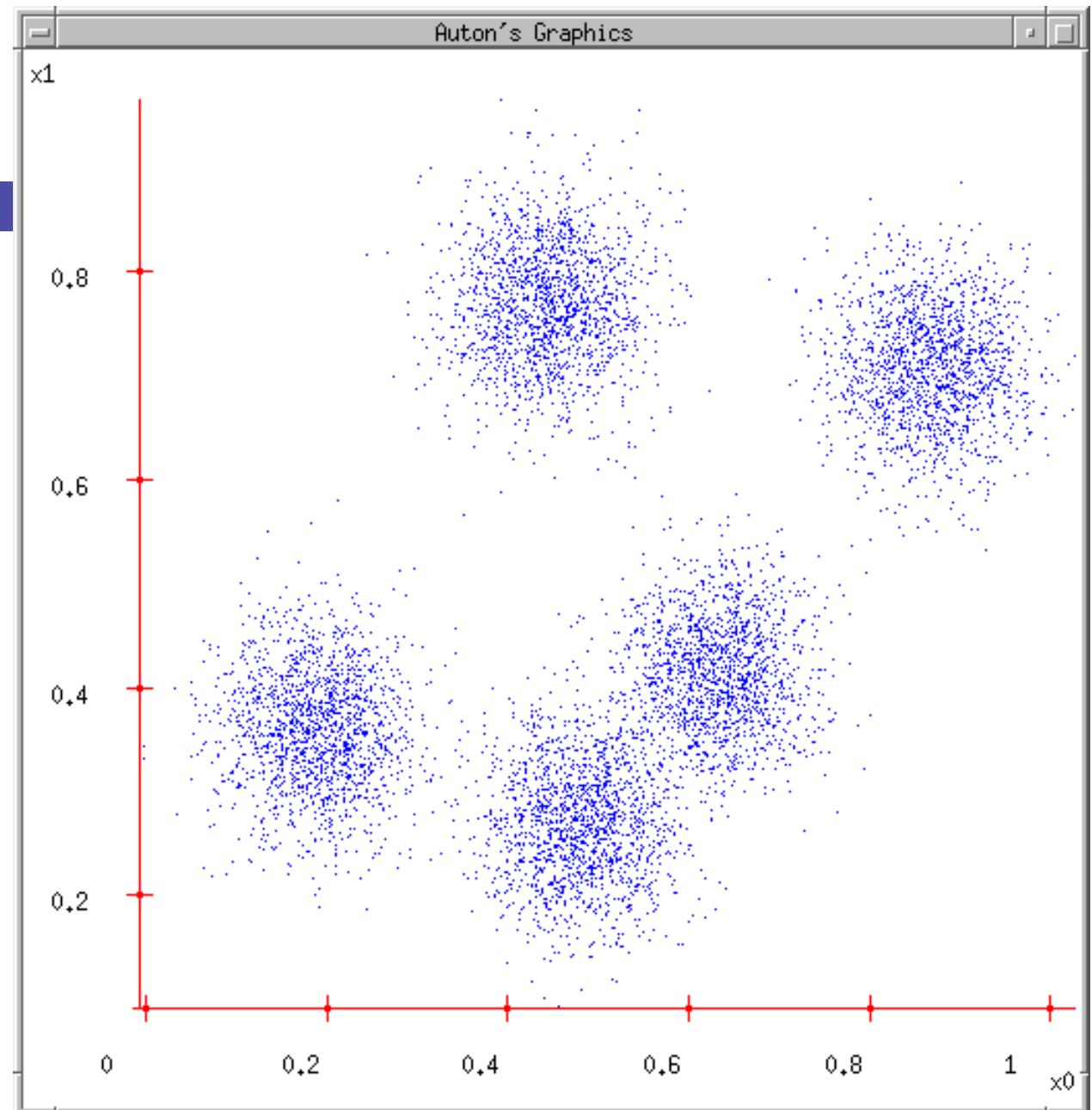
The screenshot shows the Clusty search interface. At the top, there's a navigation bar with links for 'web', 'news', 'images', 'wikipedia', 'blogs', 'jobs', and 'more ». Below this is a search bar containing the word 'race'. To the right of the search bar are 'Search' and 'advanced preferences' buttons.

On the left, there's a sidebar with tabs for 'clusters', 'sources', and 'sites'. Under the 'clusters' tab, there's a list of categories: 'All Results (238)', followed by 'Car (28)', 'Race cars (7)', 'Photos, Races Scheduled (5)', 'Game (4)', 'Track (3)', 'Nascar (2)', 'Equipment And Safety (2)', 'Other Topics (7)', '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)', and 'more | all clusters'. A 'Find' button is located at the bottom of this sidebar.

The main content area displays the results for the cluster 'Human', which contains 8 documents. The results are listed as follows:

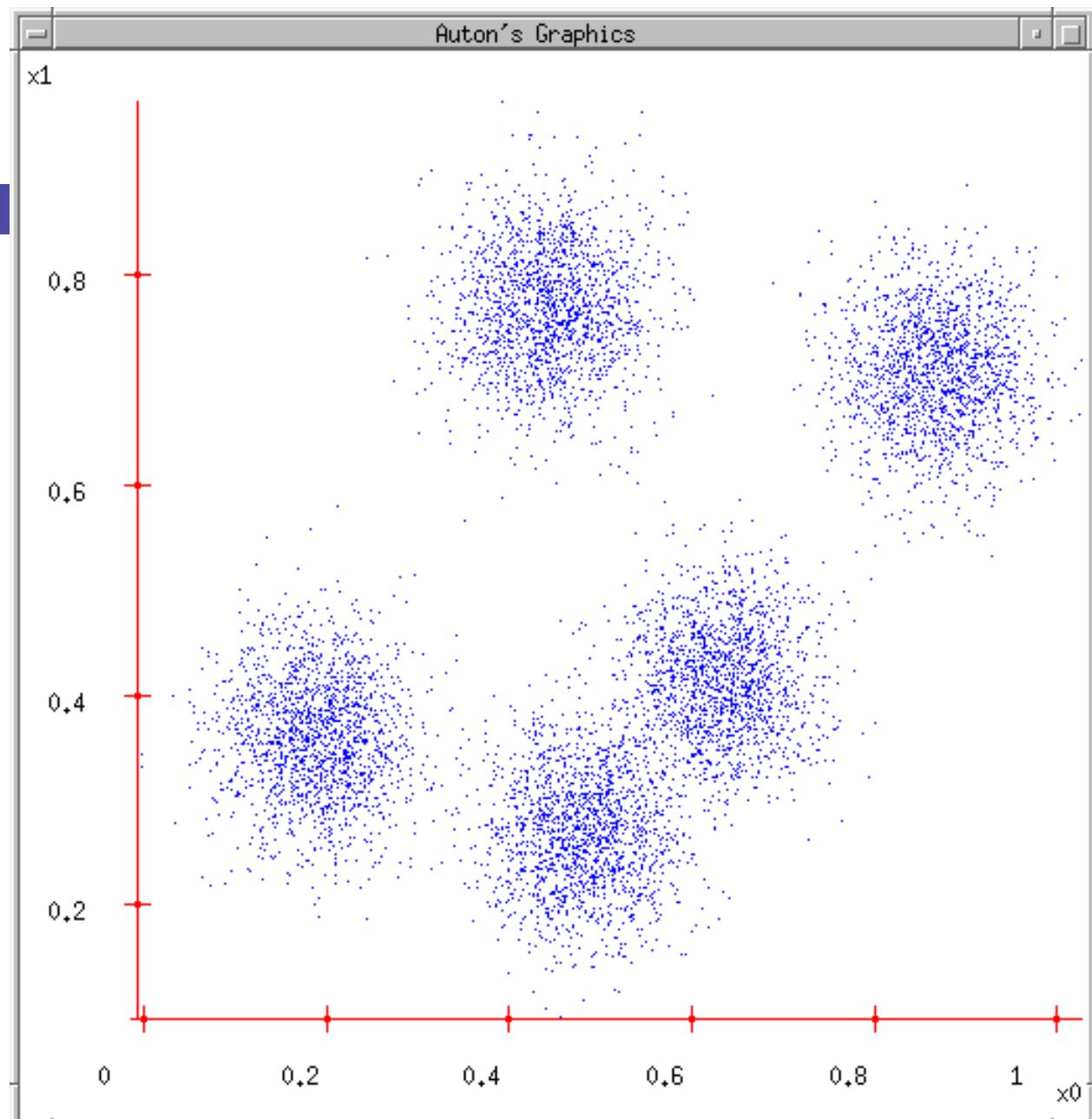
- Race (classification of human beings) - Wikipedia, the free ...**
The term **race** or racial group usually refers to the concept of dividing **humans** into populations or groups on the basis of various sets of characteristics. The most widely used **human** racial categories are based on visible 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.
[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 - [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 - [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 - [cache] - Live
- AAPA Statement on Biological Aspects of Race**
AAPA 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 - [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 - [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 - [cache] - Open Directory

Some Data



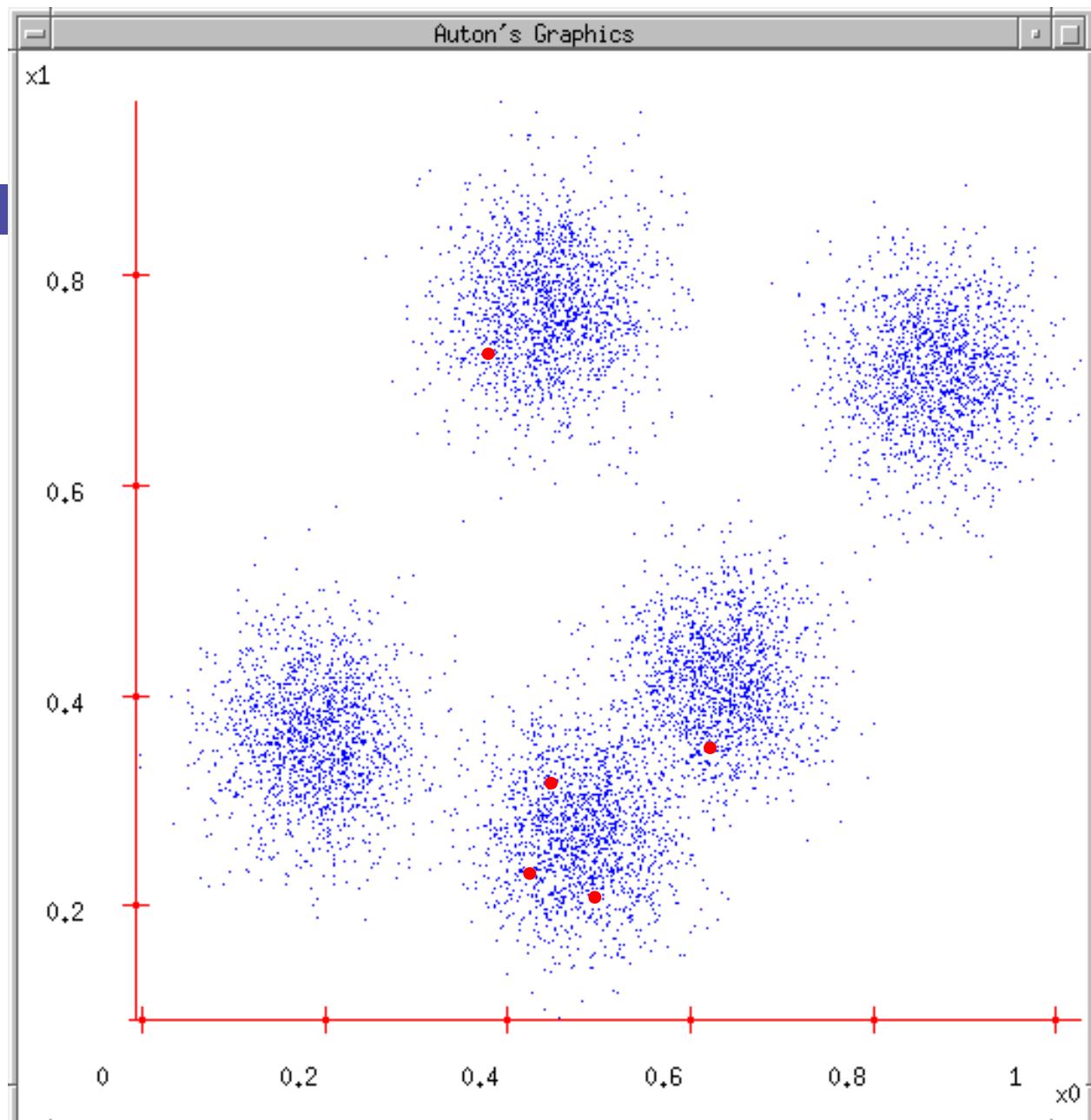
K-means

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



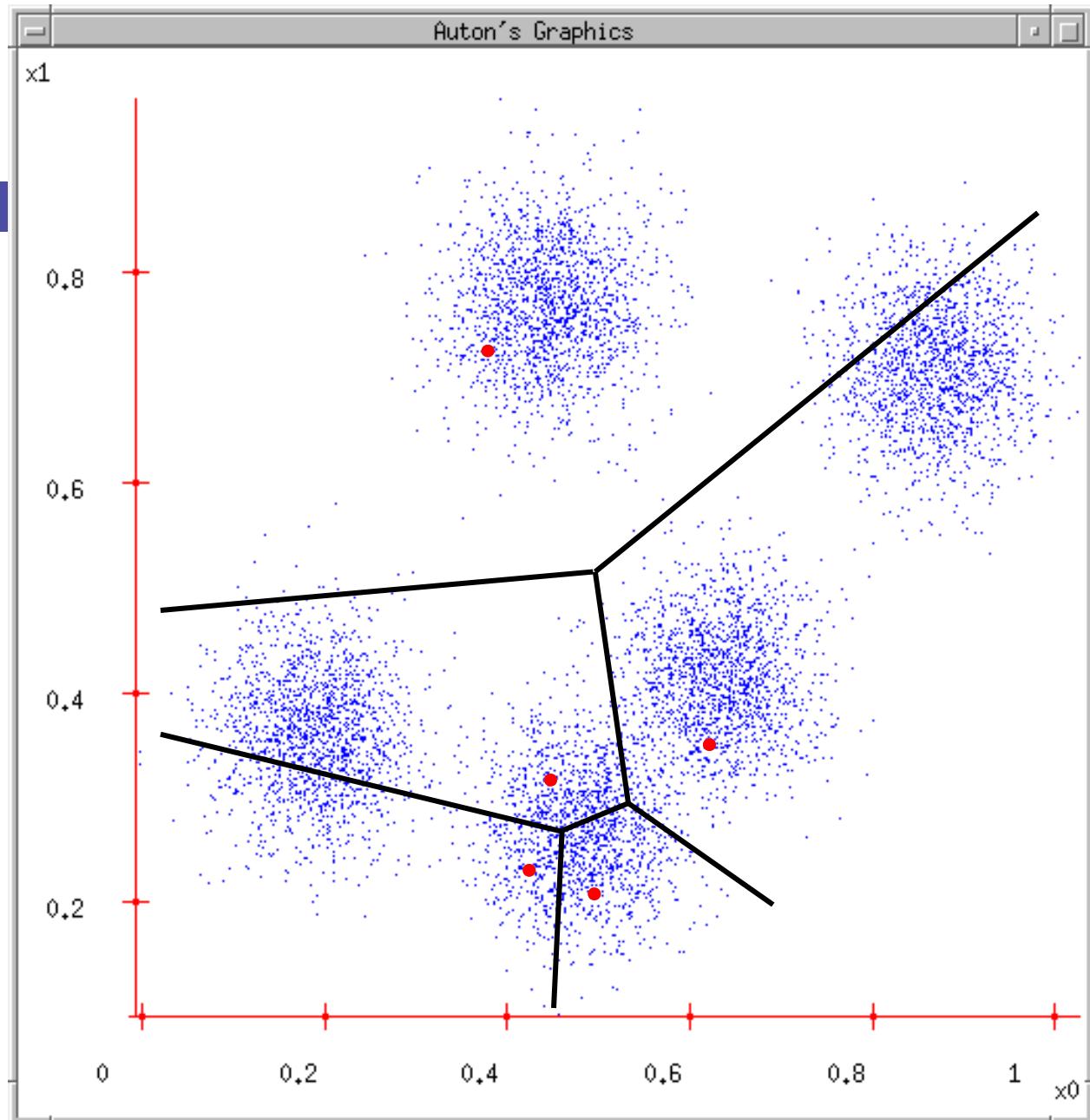
K-means

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



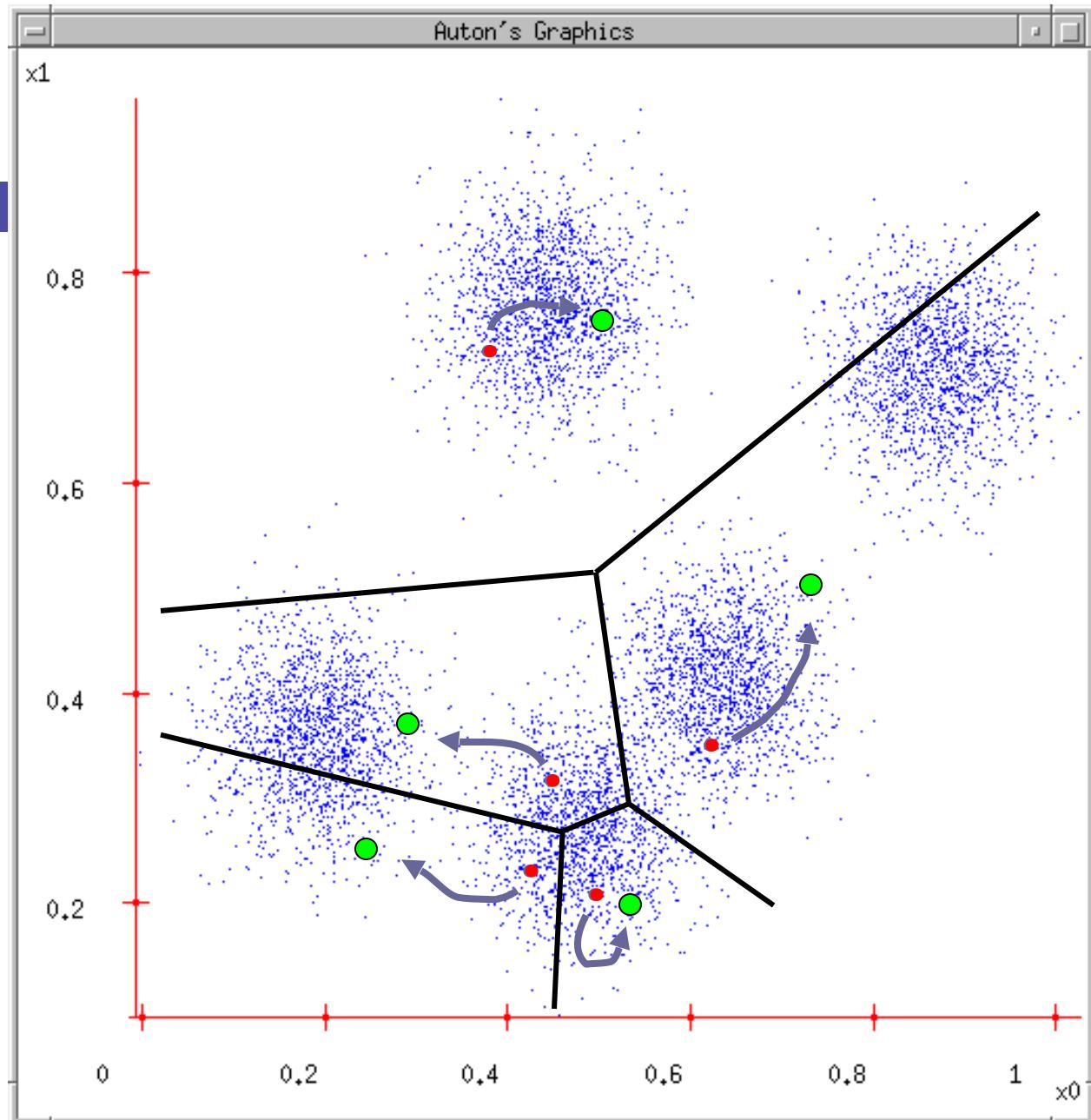
K-means

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)



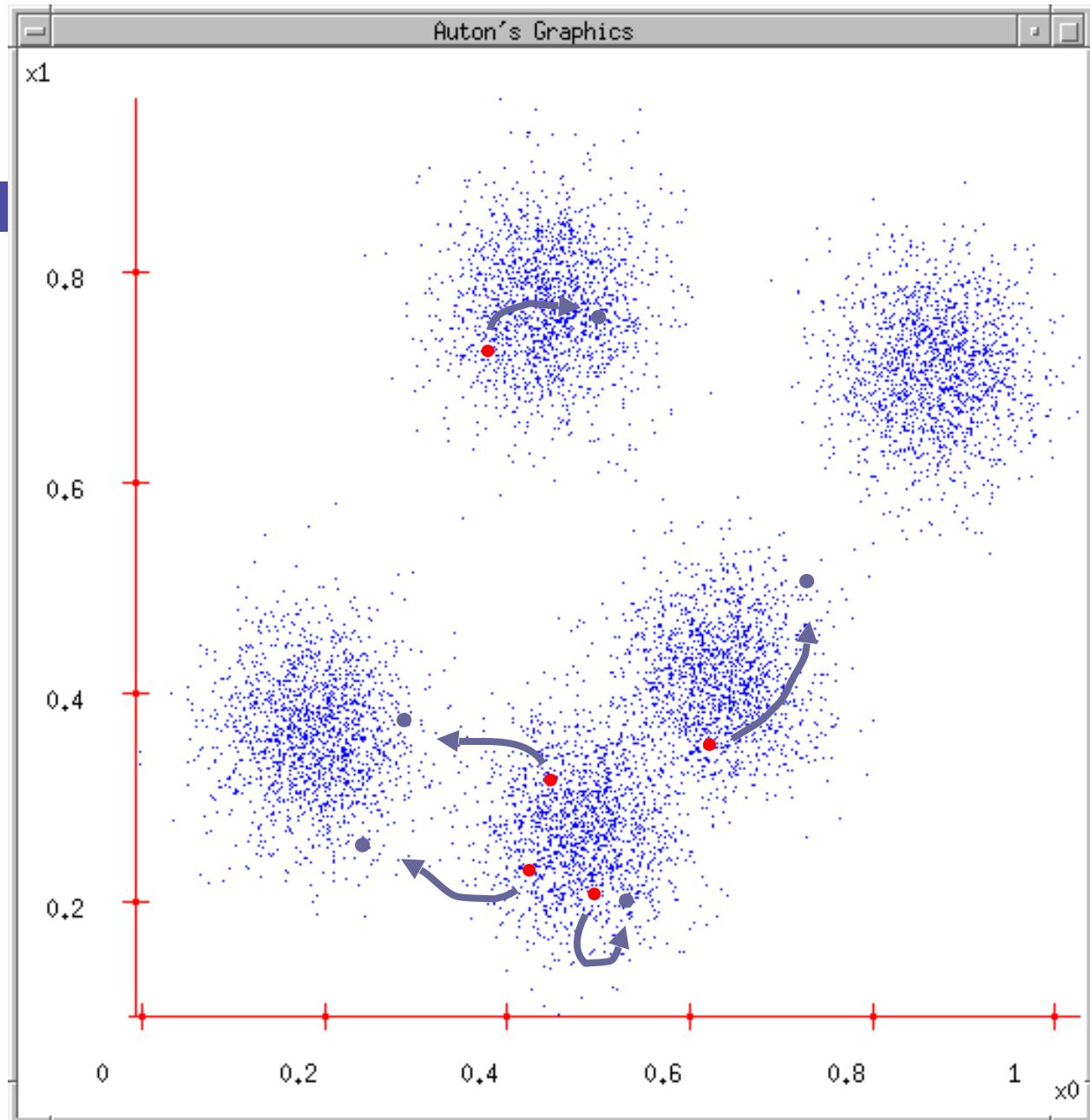
K-means

- 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:** 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 $\mu_i \leftarrow \text{average of its points!}$

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 μ , optimize C This function can only decrease after each iteration

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

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

Objective function decreases at every step \implies No configuration repeated
Only $\binom{n}{k} \approx n^k$ unique configurations \implies convergence in finite # iterations
always converges in a finite amount of time!

Vector Quantization, Fisher Vectors

Vector Quantization (for compression)

1. Represent image as grid of patches 10 x 10 patches of the image
2. Run k-means on the patches to build code book
3. Represent each patch as a code word.

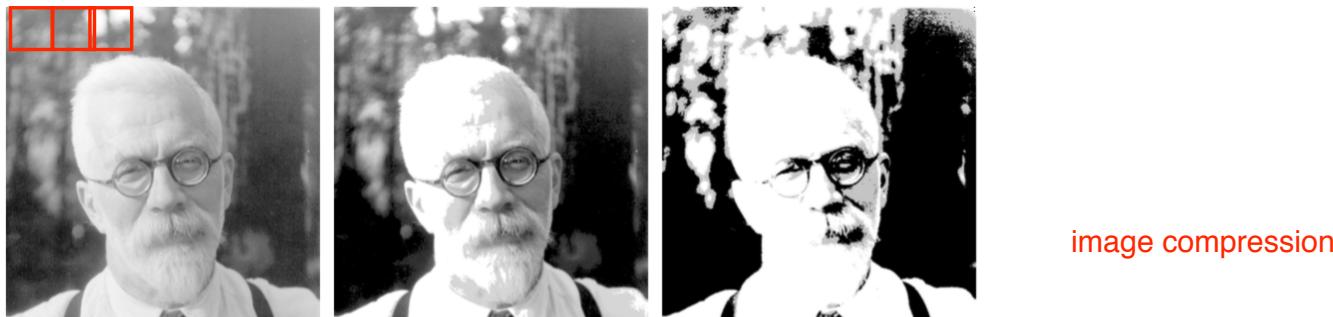


FIGURE 14.9. Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a 1024×1024 grayscale image at 8 bits per pixel. The center image is the result of 2×2 block 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.

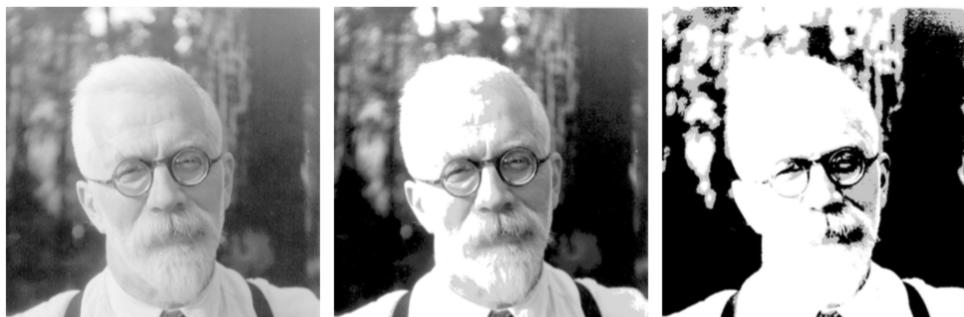
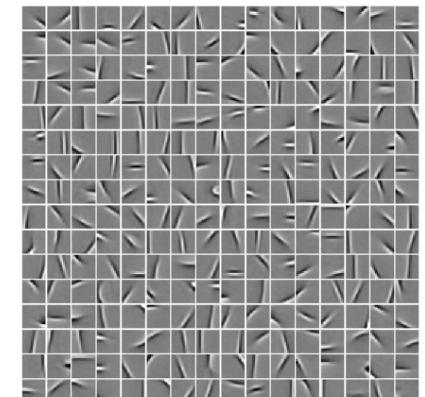


FIGURE 14.9. Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a 1024×1024 grayscale image at 8 bits per pixel. The center image is the result of 2×2 block 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

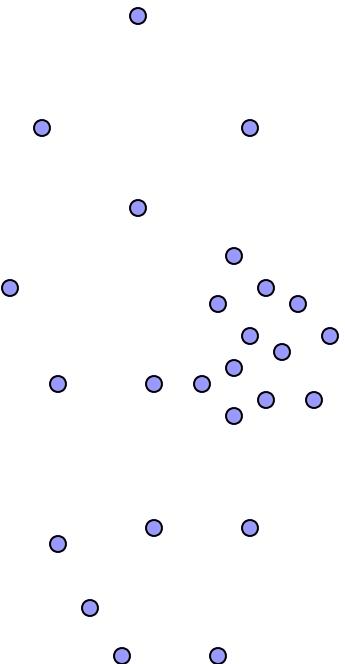
Typical output of k-means on patches



Similar reduced representation can be used as a feature vector

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

(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)