Which of these do not belong?

Anomaly Detection In Images Using A Convolution Neural Network & Isolation Forest

Mark Wilber

Seattle Data Science Spring Lightning Talk, 29 Mar, 2016

Motivation

Interested in: rare, significant instances <u>something</u>

- fraud
- network intrusions
- sensor awareness of pending failures

Ideally, want to be able to find those things we don’t yet know about (unknown unknowns)

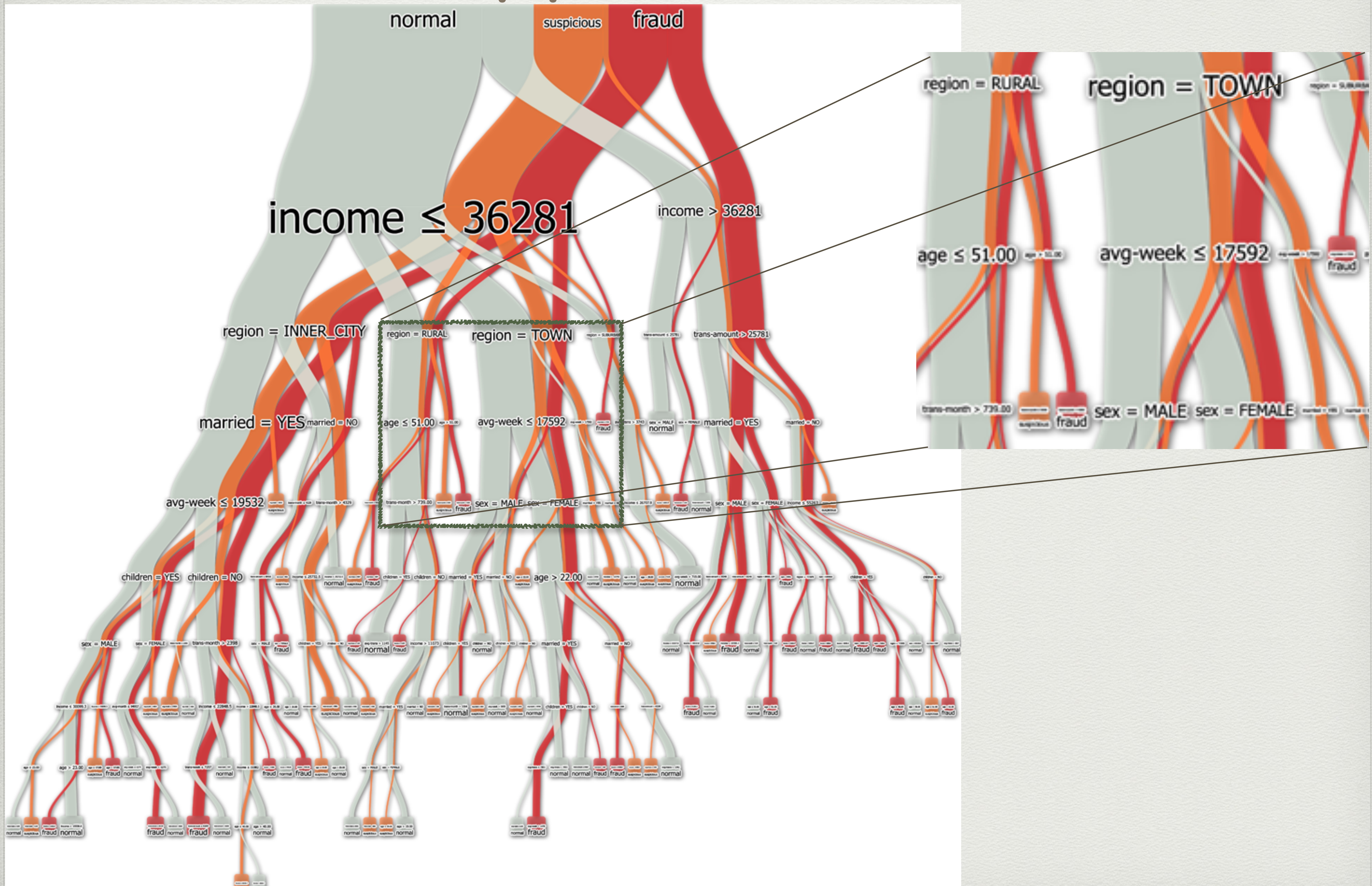
Motivation

Interested in: rare, significant instances <u>something</u>

- fraud
- network intrusions
- sensor awareness of pending failures

Ideally, want to be able to find those things we don’t yet know about (unknown unknowns)

Common Approach: classifier



Classifier-based

- Work with existing, labeled data set
- Build classifer (need not be decision tree-based)
- Useful for cases when we can expect consistent patterns for what we think of as anomalous.
- Not ideal for fraud detection, eg., since bad guys are trying to keep ahead of prevention methods

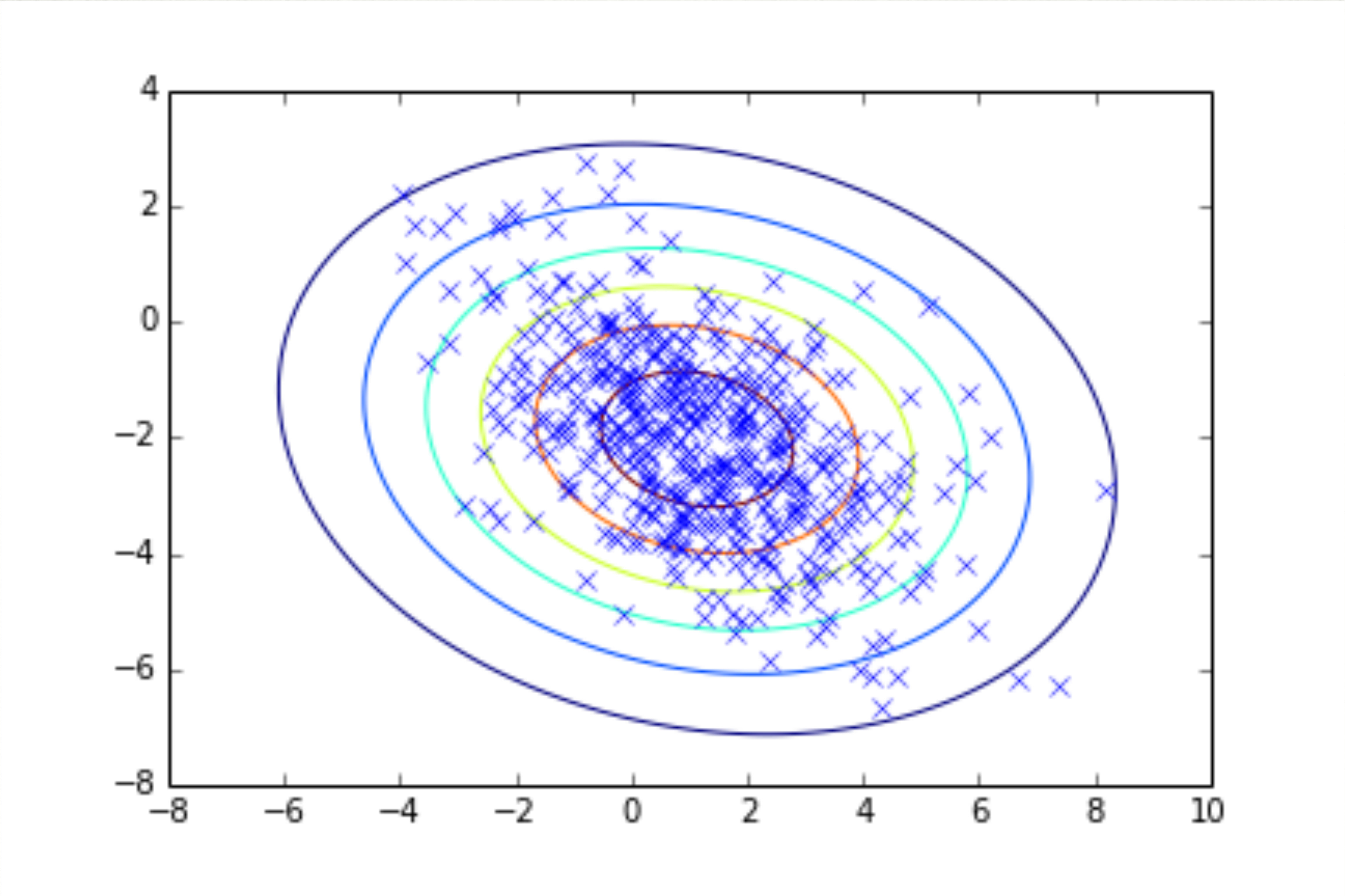
Classifier-based

- Work with existing, <u>labeled</u> data set
- Build classifer (need not be decision tree-based)
- Useful for cases when we can expect consistent patterns for what we think of as anomalous.
- Not ideal for fraud detection, eg., since bad guys are trying to keep ahead of prevention methods

Classifier-based

- Work with existing, <u>labeled</u> data set
- Build classifer (need not be decision tree-based)
- Useful for cases when we can expect consistent patterns for what we think of as anomalous.
- Not ideal for fraud detection, eg., since bad guys are trying to keep ahead of prevention methods

Another method: parameterized model



Parameterized model

- Assume data fit statistical distribution
- Treat points in tails of distribution as ‘anomalous’
- Many data sets do not have distributions that can be neatly parameterized
- Can be hard to guess form in high-dimension space
- Won’t find ‘embedded’ anomalies

Parameterized model

- Assume data fit statistical distribution
- Treat points in tails of distribution as ‘anomalous’
- Many data sets do not have distributions that can be neatly parameterized
- Can be hard to guess form in high-dimension space
- Won’t find ‘embedded’ anomalies

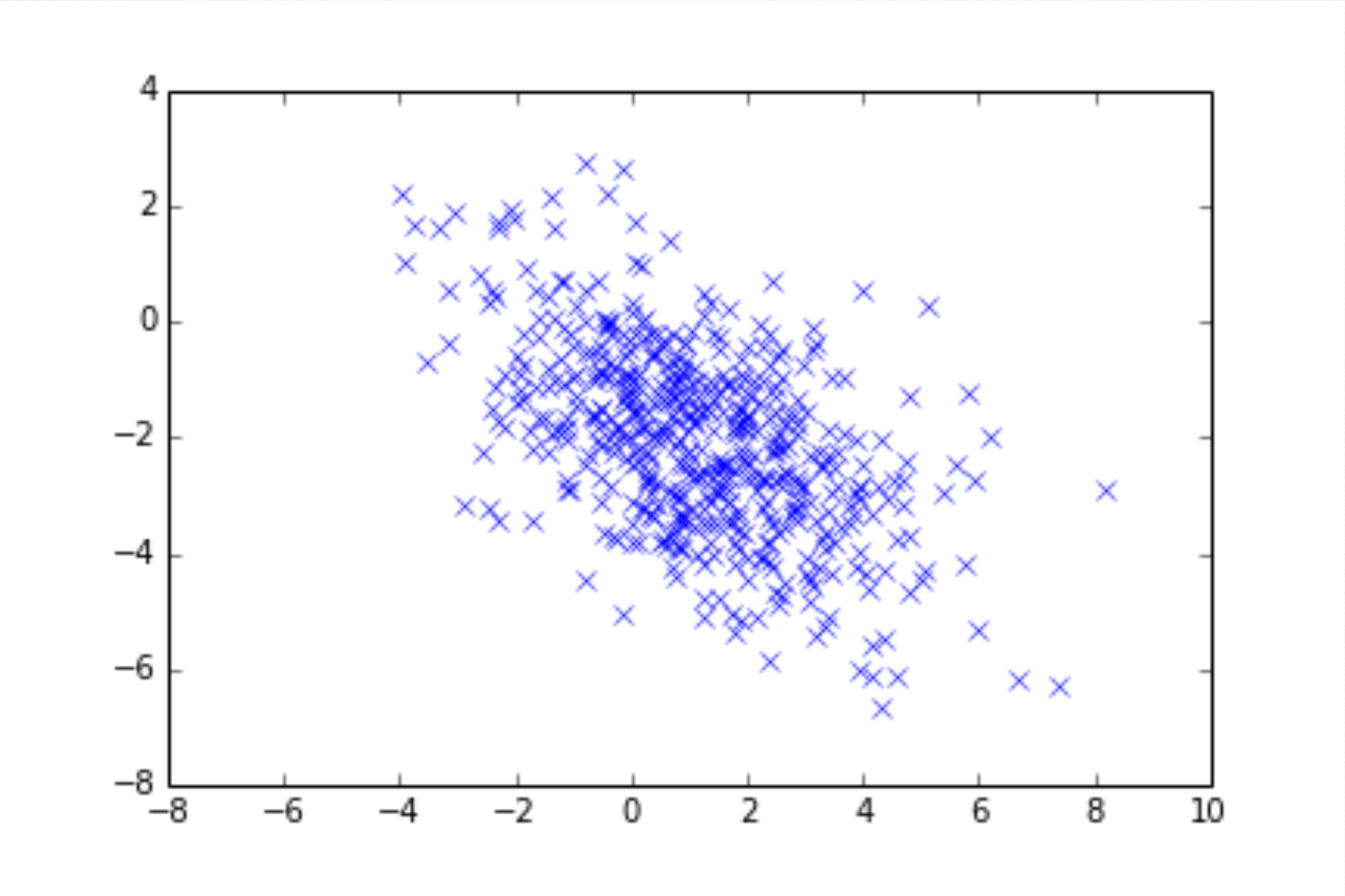
Parameterized model

- Assume data fit statistical distribution
- Treat points in tails of distribution as ‘anomalous’
- Many data sets do not have distributions that can be neatly parameterized
- Can be hard to guess form in high-dimension space
- Won’t find ‘embedded’ anomalies

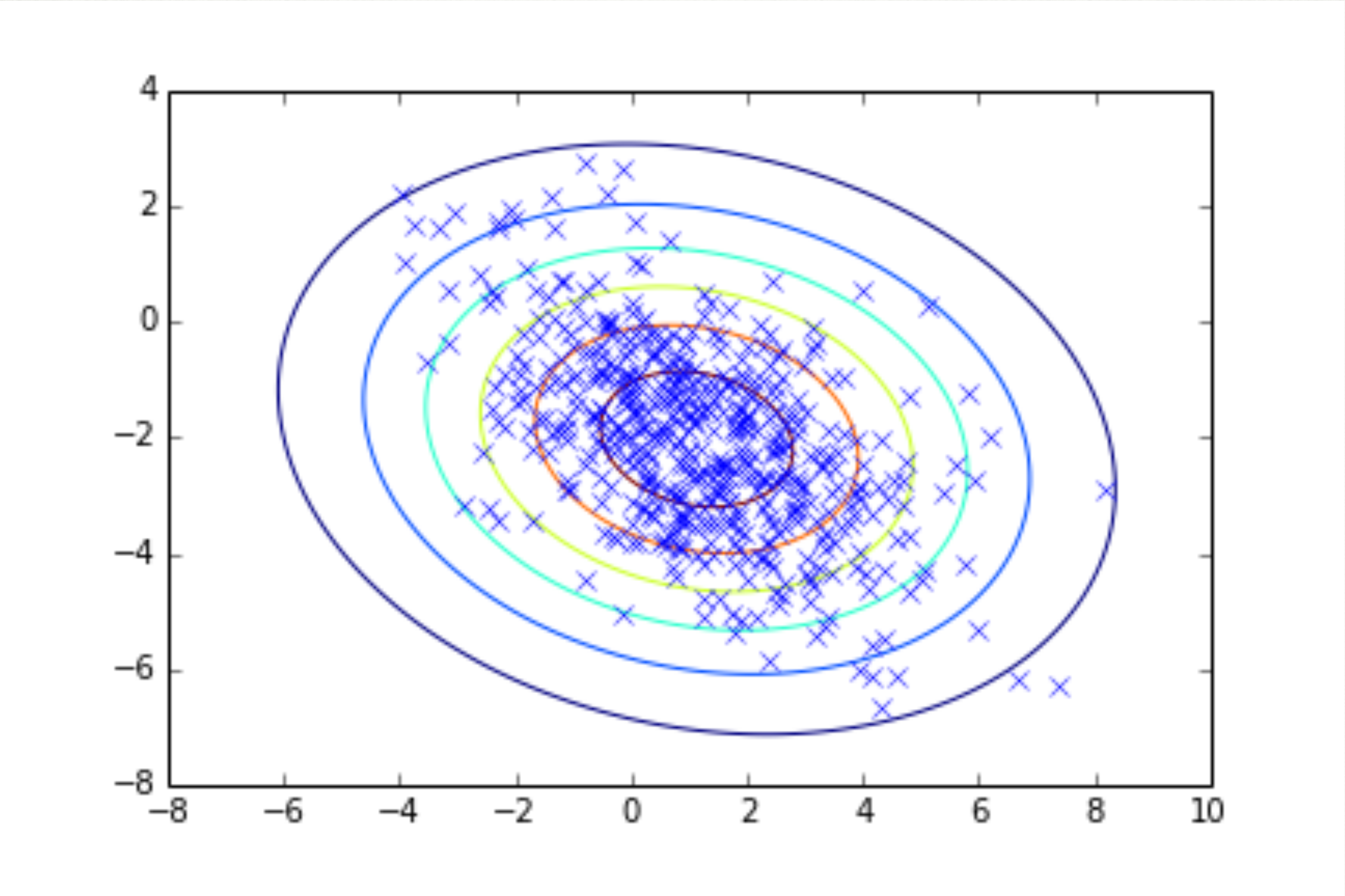
Parameterized model

- Assume data fit statistical distribution
- Treat points in tails of distribution as ‘anomalous’
- Many data sets do not have distributions that can be neatly parameterized
- Can be hard to guess form in high-dimension space
- Won’t find ‘embedded’ anomalies

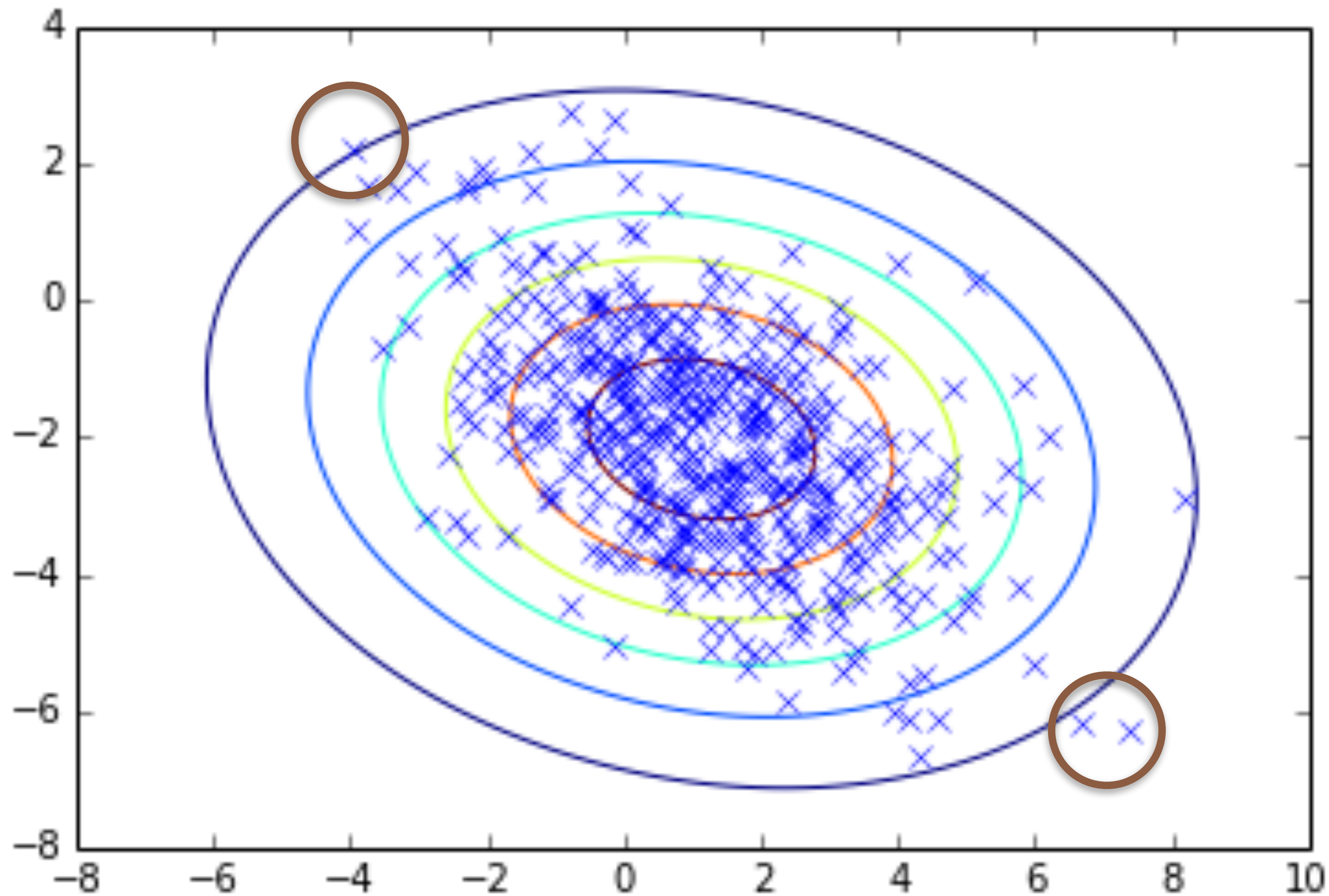
Parameterized model



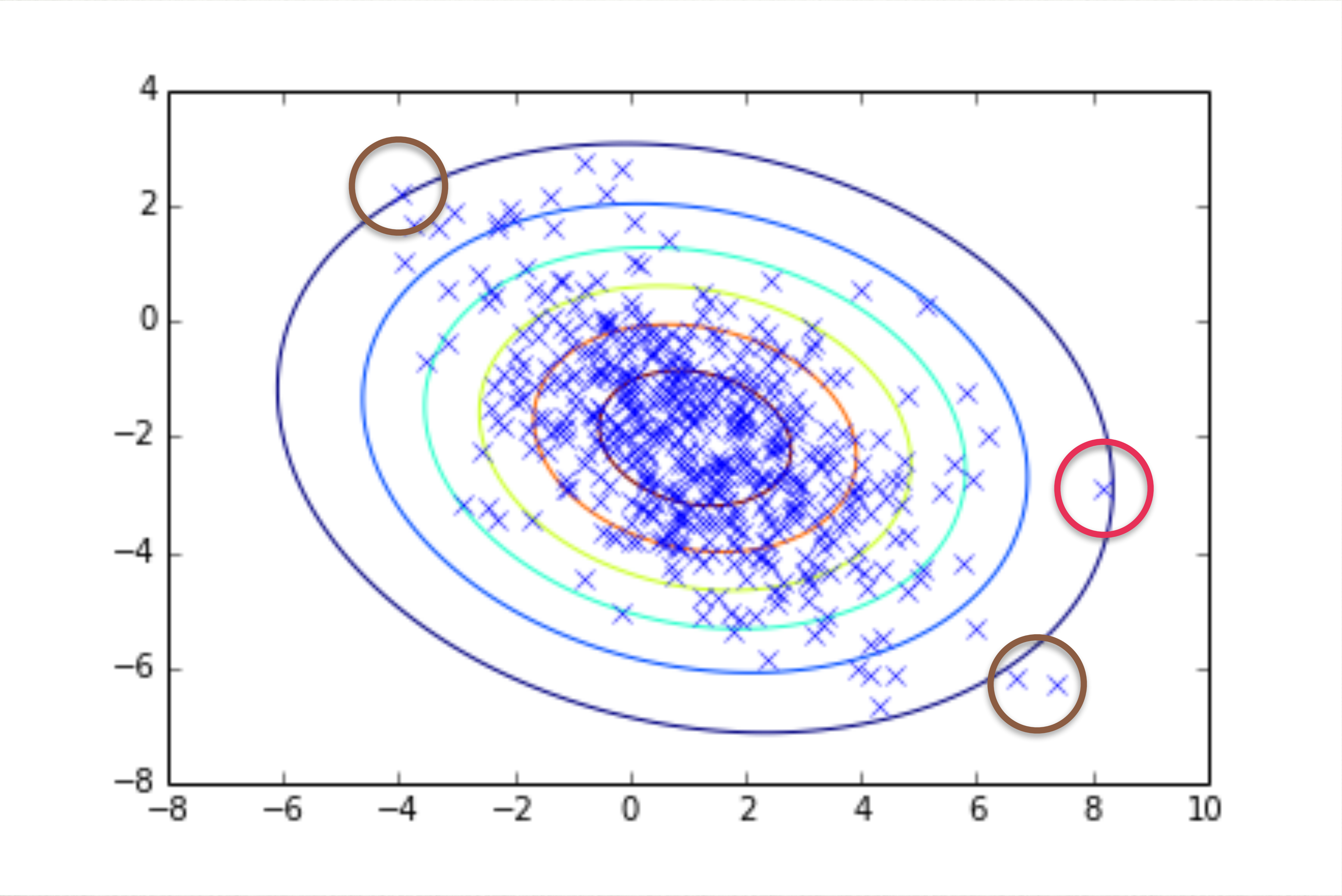
Parameterized model



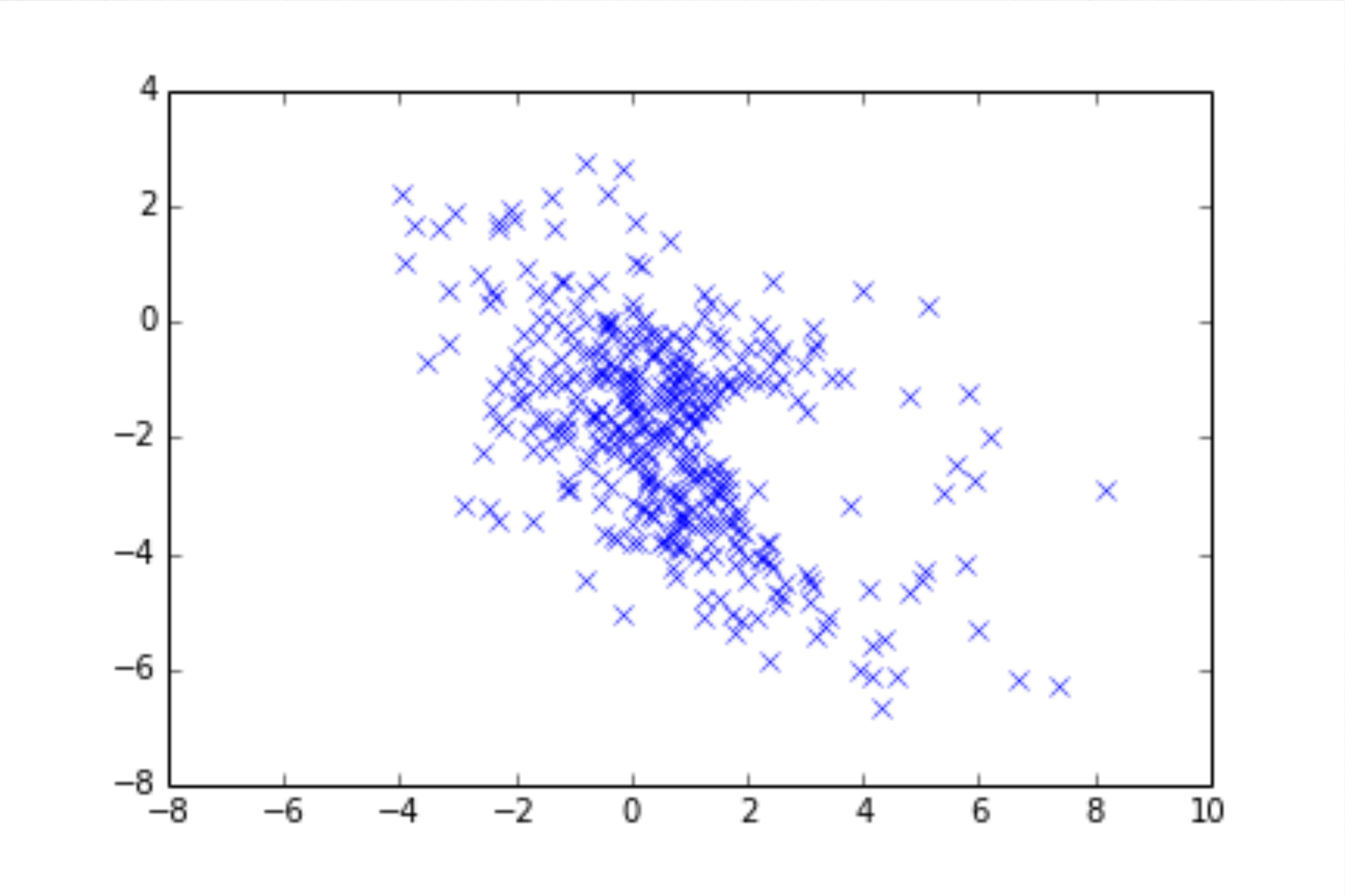
Parameterized model



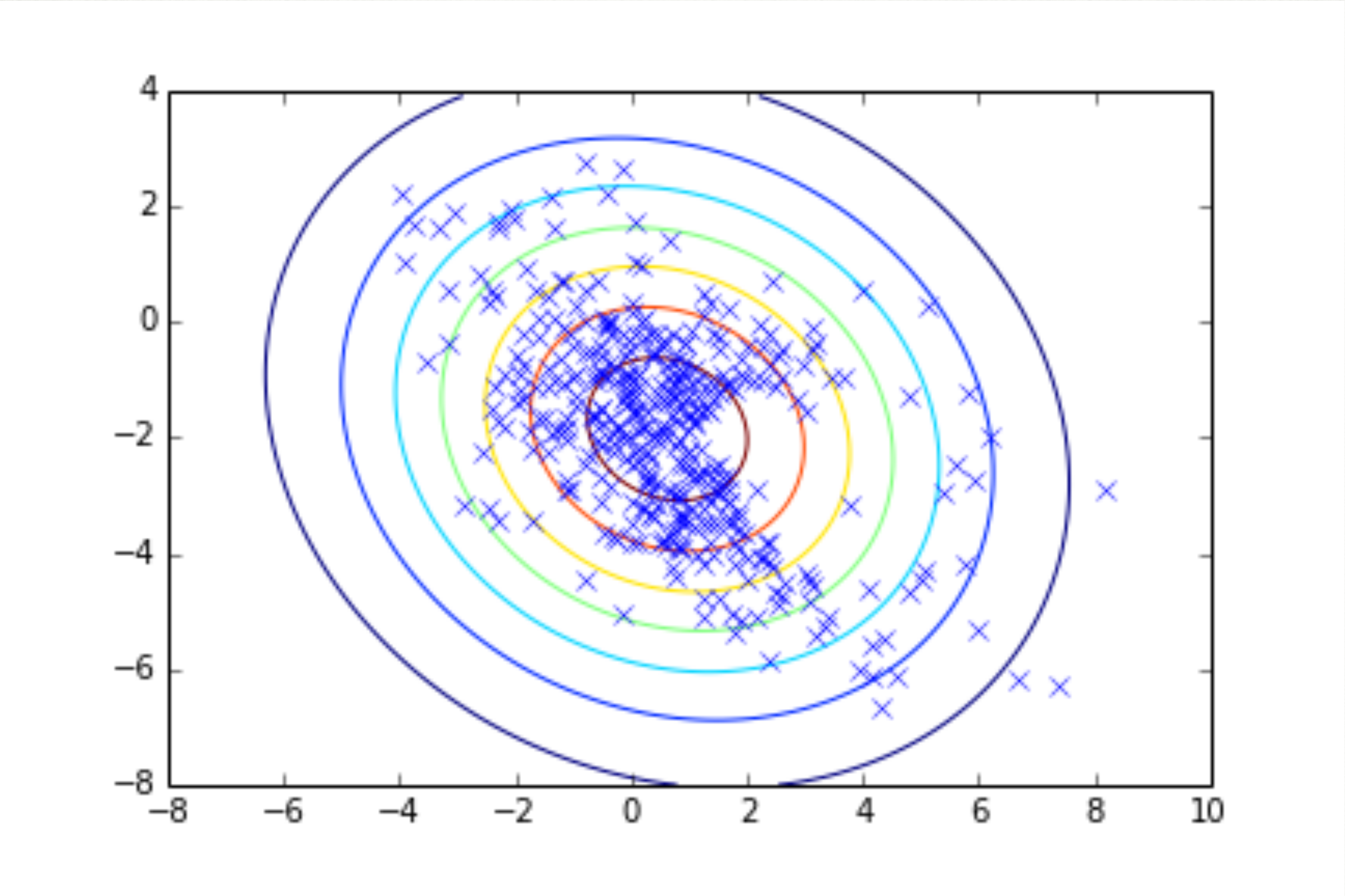
Parameterized model



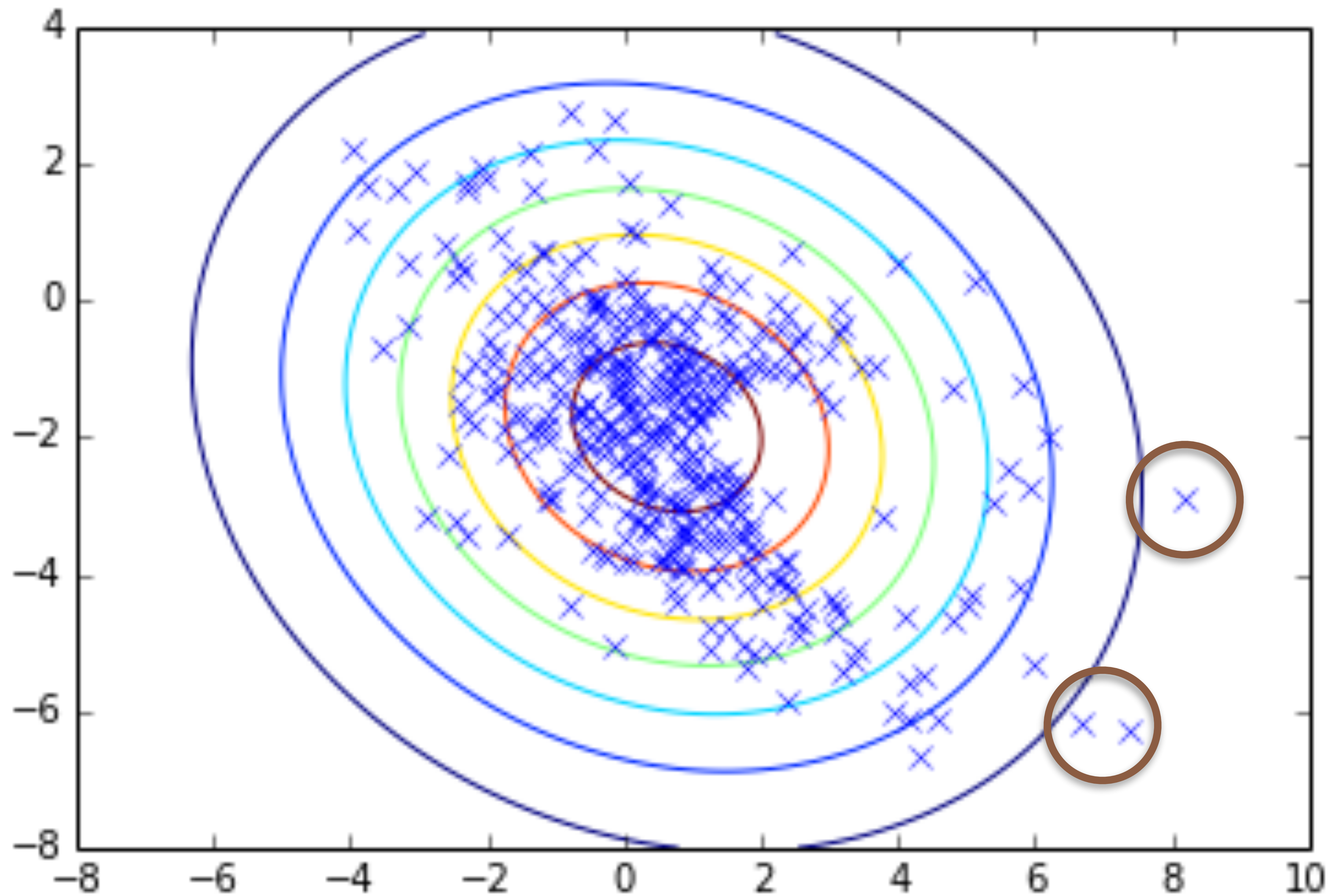
Parameterized model



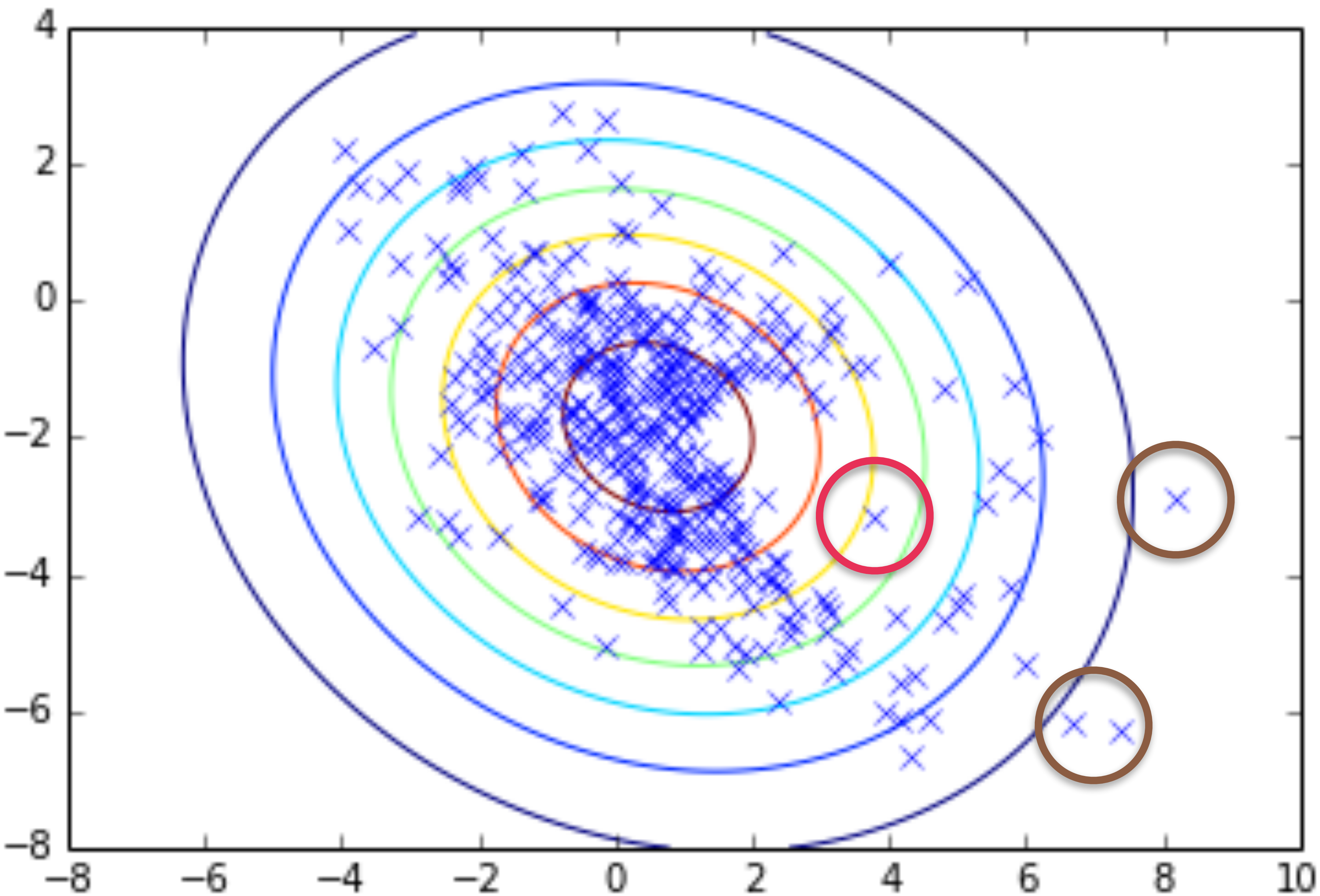
Parameterized model



Parameterized model



Parameterized model



Density-based

- Apply a method for computing density
- Treat low-density regions as ‘anomalous’
- Can find ‘embedded’ anomalies
- Does not require presumed distribution
- Can suffer (badly) from lack of scalability

Density-based

- Apply a method for computing density
- Treat low-density regions as ‘anomalous’
- Can find ‘embedded’ anomalies
- Does not require presumed distribution
- Can suffer (badly) from lack of scalability

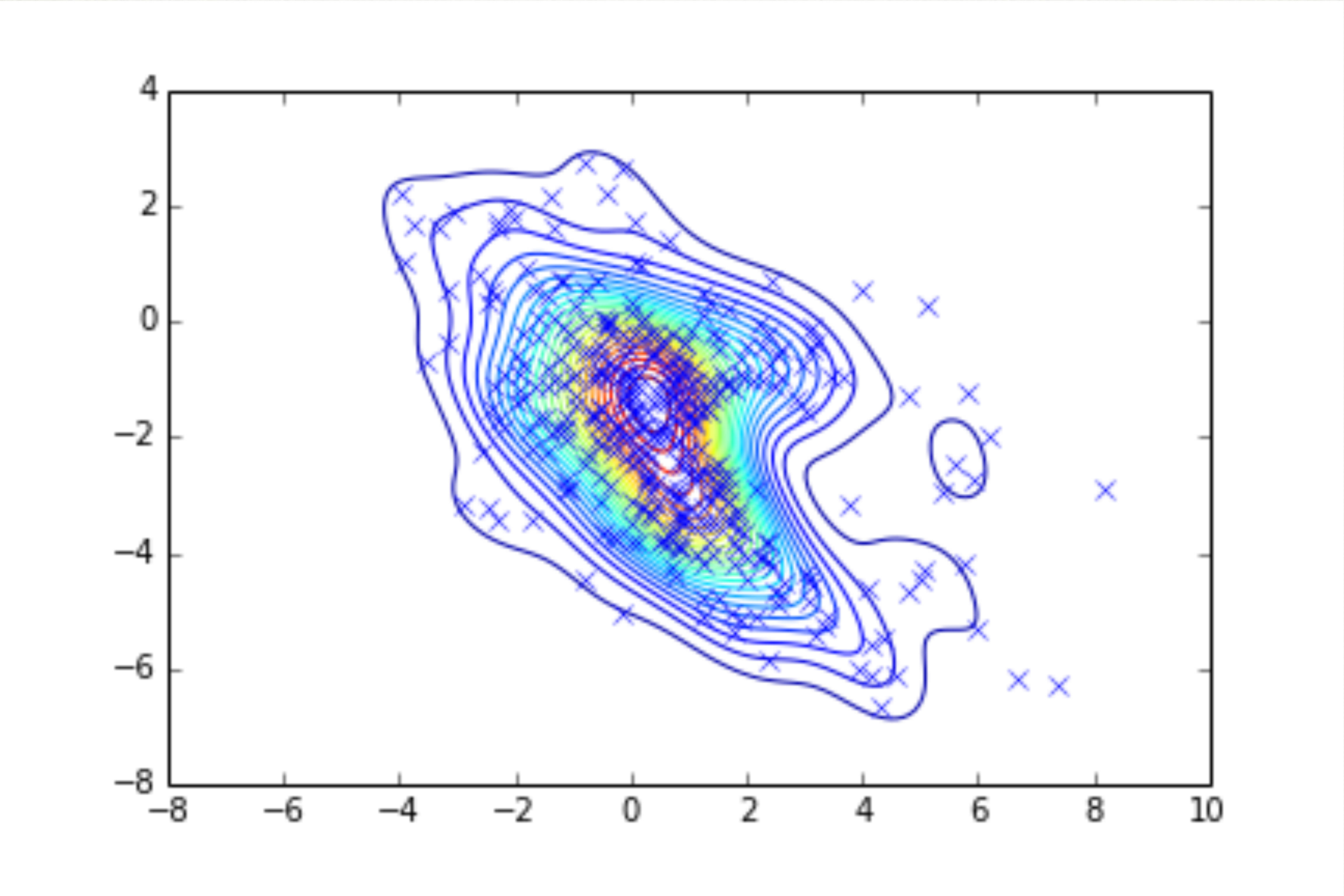
Density-based

- Apply a method for computing density
- Treat low-density regions as ‘anomalous’
- Can find ‘embedded’ anomalies
- Does not require presumed distribution
- Can suffer (badly) from lack of scalability

Density-based

- Apply a method for computing density
- Treat low-density regions as ‘anomalous’
- Can find ‘embedded’ anomalies
- Does not require presumed distribution
- Can suffer (badly) from lack of scalability

Density-based



Solution used here

Isolation Forest[1]

Un-supervised method of statistically isolating points that have no close neighbors in feature space

Ensembles of binary trees:

- branch on randomly-selected features
- and on randomly-selected feature values

- track depths at which points are isolated.

[1] Liu, F.T. et al, ICDM `08b, 2008

Solution used here

Isolation Forest[1]

Un-supervised method of statistically isolating points that have no close neighbors in feature space

Ensembles of binary trees:

- branch on randomly-selected features
- and on randomly-selected feature values

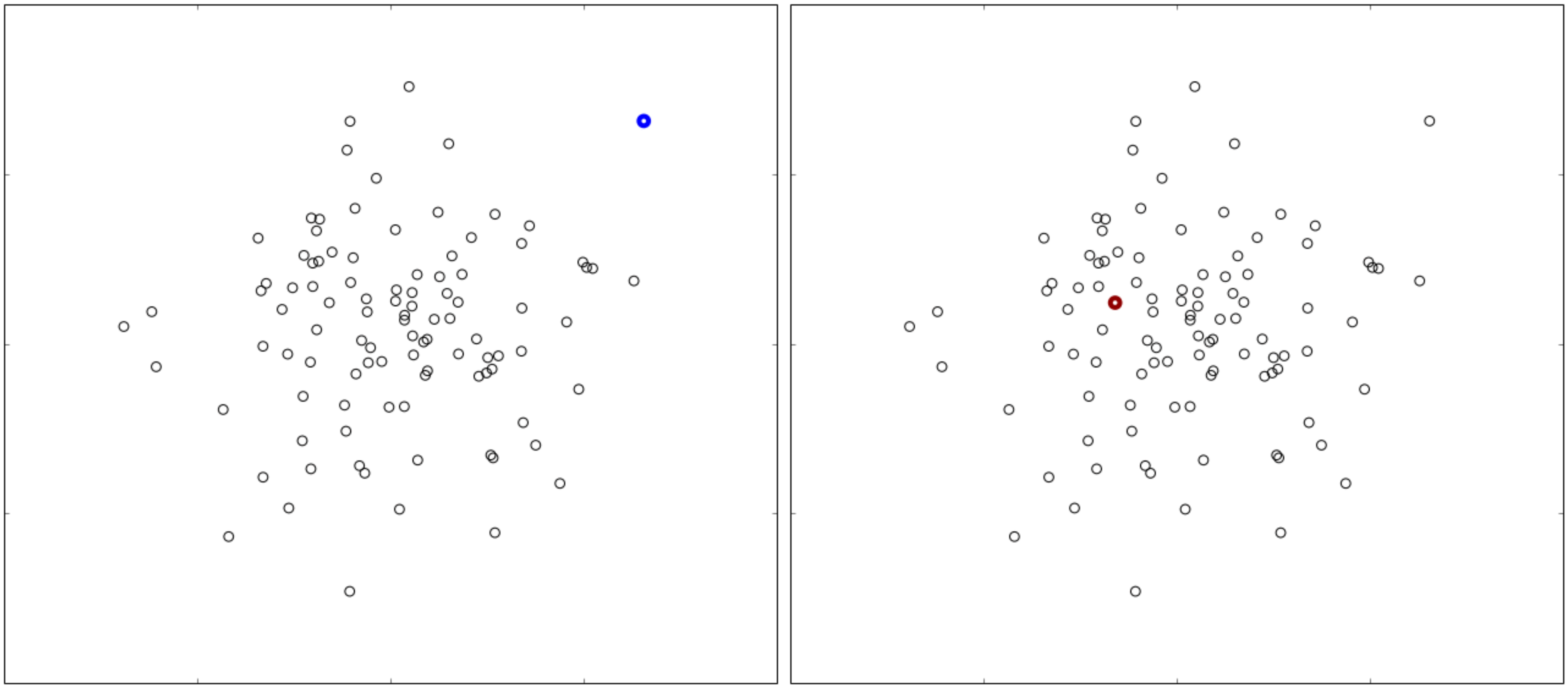
    no information gain/ Gini impurity used!

- track depths at which points are isolated.

[1] Liu, F.T. et al, ICDM `08b, 2008

Isolation Forest

- anomalous point on left (in blue) is isolated in a few partitions/tree branches
- point within main distribution (red) requires many



Solution

Isolation Forest

Anomaly scores derived from average tree length

- values < 0.5 uninteresting
- values > 0.6 usually are

Advantages:

- no need to compute many distances to determine proximity
- don't need prior knowledge of the number of clusters

Image Features

For image datasets, could work with raw pixels, but …

- Identical images translated by a few pixels can be located far apart in 6-dimensional space (x,y,r,g,b)
- Images identical in all but scale, similarly distinct

- Instead use modern image feature representation — convolution neural nets (CNNs)

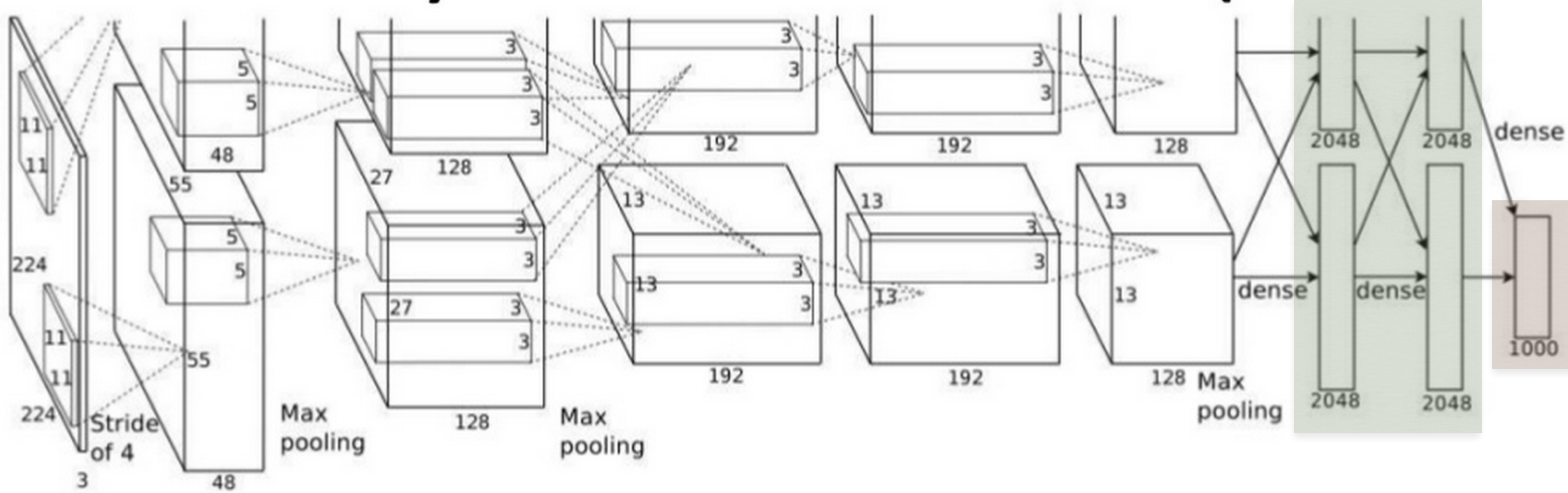
➡ Caffe: pre-trained CNN: alexNet[2]

[2] Krizhevsky, A, et al, Imagenet classification with deep convolutional neural networks, 2012

Image Features

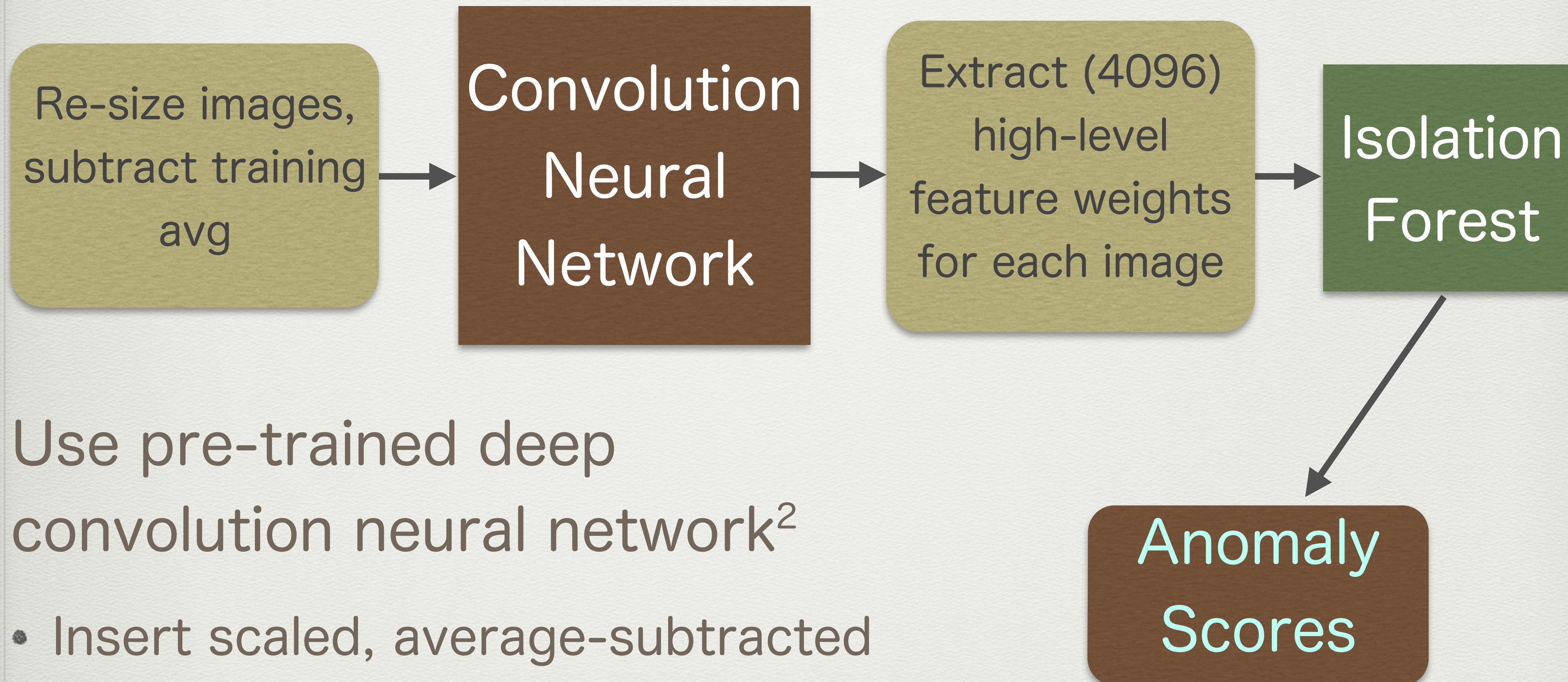
7 hidden layers, 650K units, 60M parameters

Krizhevsky et al model size (alexNet)



Layer:	Model Size(MB)	
Conv1:	float*(48+48)*(3*11^2)	= 0.1
Conv2:	float*(128+128)*(48*5^2)	= 1.2
Conv3:	float*(192+192)*(256*3^2	= 3.4
Conv4:	float*(192+192)*(192*3^2)	= 2.5
Conv5:	float*(128+128)*(192*3^2)	= 1.7
FC6:	float*((128+128)*6^2)*4096	= 144(66%)
FC7:	float*4096*4096	= 64(29%)

Featurizing Images



Use pre-trained deep convolution neural network[2]

- Insert scaled, average-subtracted images in one end
- Extract feature weights at the other

[2][Krizhevsky, A. et al, Imagenet classification with deep convolutional neural networks, 2012]

Results

61 images: tigers + 3 leopards + 3 house cats + 1 house

Results

c7 rank	image	fc6 score	fc7 score	fc8 score		fc7 rank	image	fc6 score	fc7 score	fc8 score
0		0.53	0.55	0.61		1		0.52	0.53	0.61
2		0.50	0.52	0.61		3		0.50	0.52	0.60
4		0.49	0.51	0.59		5		0.49	0.49	0.57
6		0.47	0.49	0.57		7		0.47	0.48	0.57
8		0.46	0.48	0.57		9		0.46	0.46	0.55

Results

Profiling

	initialize Caffe	validate images	pre-process images	push through network	iForest on fc6	iForest on fc7	iForest on fc8	iForest combined
time	0.005 s	00 m, 1.44 s	00 m, 15.90 s	00 m, 0.24 s	1.036 s	0.899 s	0.589 s	00 m, 2.52 s
fraction	0.000	0.072	0.791	0.012	0.052	0.045	0.029	0.126

- typically, > 70% of time is spent pre-processing images (GPUs not used)
- push through network ~ 1% (GPUs)
- Isolation Forest on different feature weights 3-5%

Results

61 images: houses + 1 hospital + 1 house boat + 2 “boats”

Results

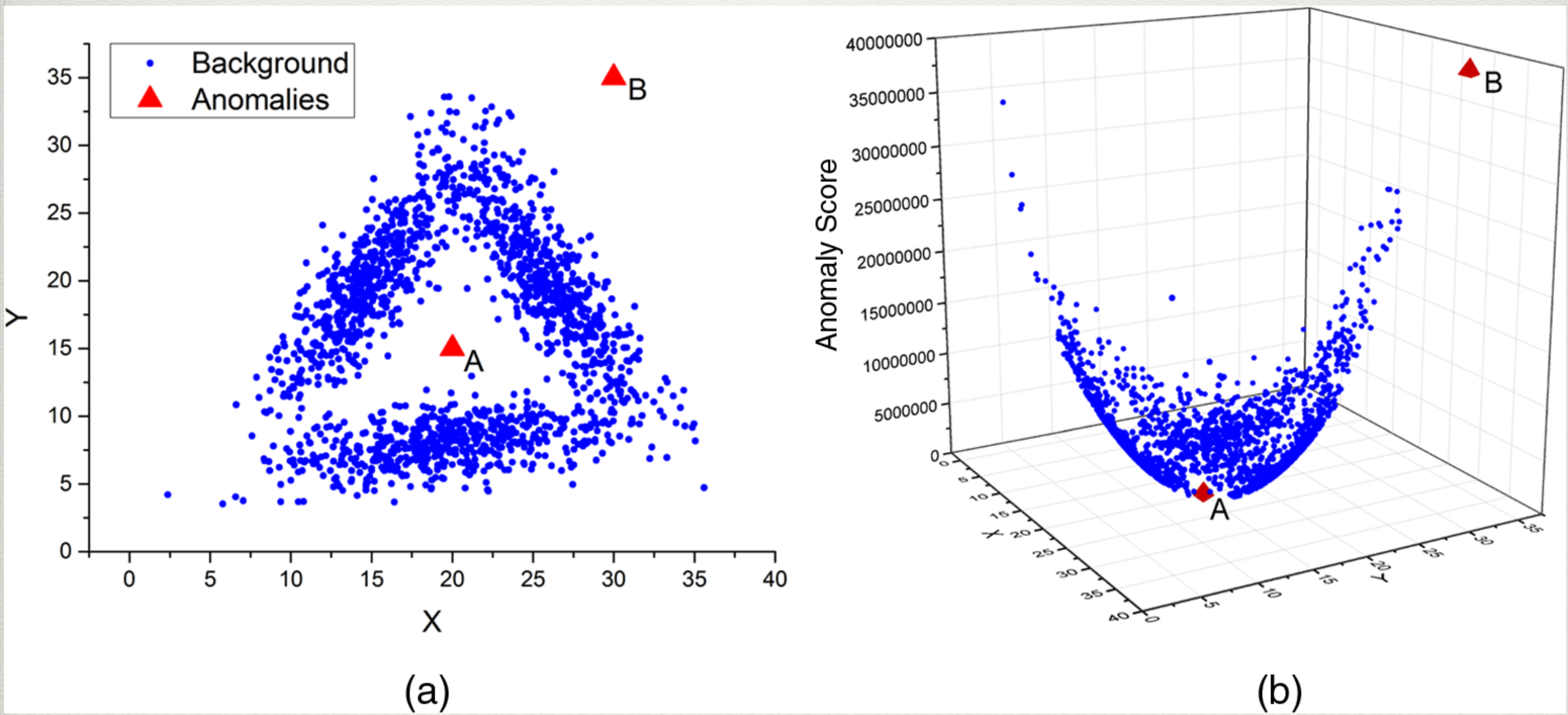
fc7 rank	image	fc6 score	fc7 score	fc8 score	fc7 rank	image	fc6 score	fc7 score	fc8 score
0		0.55	0.65	0.71	1		0.51	0.64	0.62
2		0.50	0.61	0.61	3		0.49	0.59	0.61
4		0.49	0.55	0.59	5		0.49	0.53	0.56
6		0.48	0.53	0.55	7		0.48	0.52	0.54
8		0.48	0.51	0.54	9		0.48	0.51	0.54

Questions?

LinkedIn: https://www.linkedin.com/in/markwilber

Try out the app! http://tinyurl.com/MWilberAnomImage

Density-based



Lou, C and H. Zhao, “Local density-based anomaly detection in hyper spectral image”, J. Appl.. Remote Sens., 9(1), 095070, 2015.