# Bootstrapping Handwritten Digits

Tim Healy

# Contents

# Objective

- Machine Learning (ML) models, particularly Neural Networks, often require a lot of training data, which can be intractable to collect.
- With so much computational power at our fingertips, we are also able to processing more data than ever before
- Essentially, a lot of data is used so that ML models can understand as much of the domain as possible.
- **This application will employ bootstrapping as a technique for generating additional training data**
- The hope is that bootstrapped data can be used as a proxy for real training data.

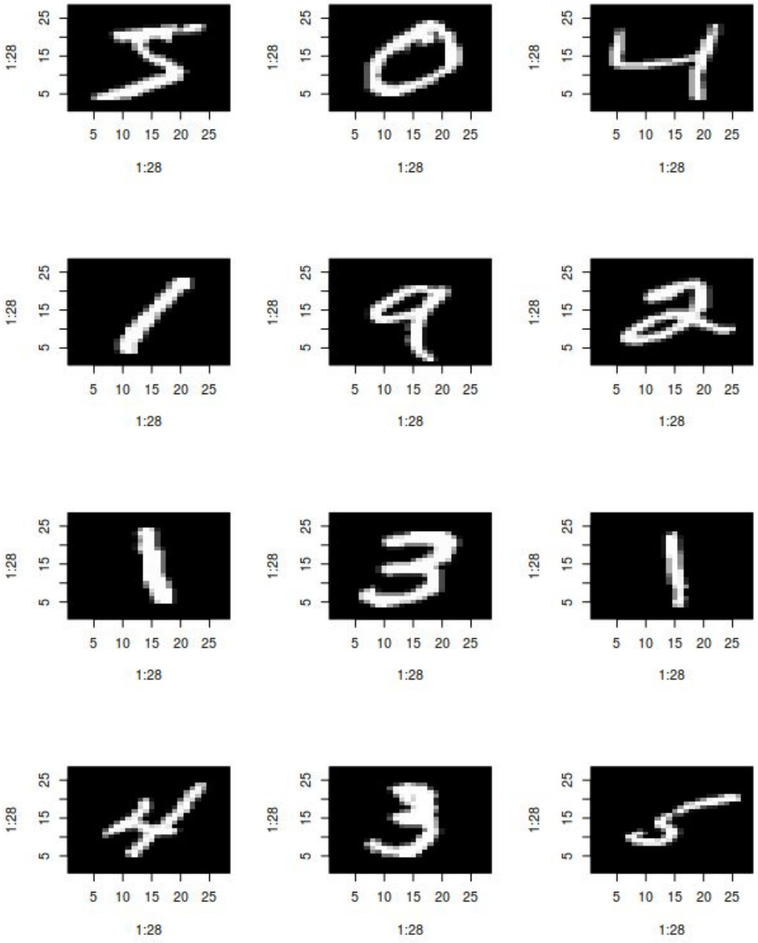
# Data

This application uses the MNIST dataset of handwritten digits.

- 60,000 training examples
- 10,000 test examples

Widely used to train machine learning models

Each image is encoded into a 28x28 array of grayscale pixel values.




http://yann.lecun.com/exdb/mnist/

# Bootstrapping - Notation

For each of the $n = 60,000$ digit images there are $m = 784$ pixels having grayscale values between 1 and 255.

Let,

- $K = \{0, 1, ..., 9\}$ be the set of 10 digit classes
- $N = \{1, 2, ..., 60,000\}$ be the set of training examples
- $M = \{1, 2, ..., 784\}$ be the set representing the pixels composing each digit.

For each digit $k \in K$, Let $X_{ij}^{(k)}$ be a random variable representing the $j$th pixel value for the $i$th image, where $i \in N$ and $j \in M$.

# Bootstrapping - Nonparametric

Let $X$ be the N x M matrix of training examples. For a given $k$, we have the matrix $X^{(k)} = \left[X_{ij}^{(k)}\right]_{k \in K}$ of $n^{(k)}$ training examples belonging to the $k$ digit class, where $X^{(k)} \subset X$.

For each pixel $j \in M$, we have the i.i.d sample $X_{*j}^{(k)} = X_{1j}^{(k)}, X_{2j}^{(k)}, ..., X_{n^{(k)}j}^{(k)}$ of pixel values across $X^{(k)}$, where $X_{*j}^{(k)} \sim \hat{F}_j^{(k)}$, the empirical distribution of the observed data for the $j$th pixel in the $k$th digit.

We therefore define the following, using the sample mean as our test statistic.

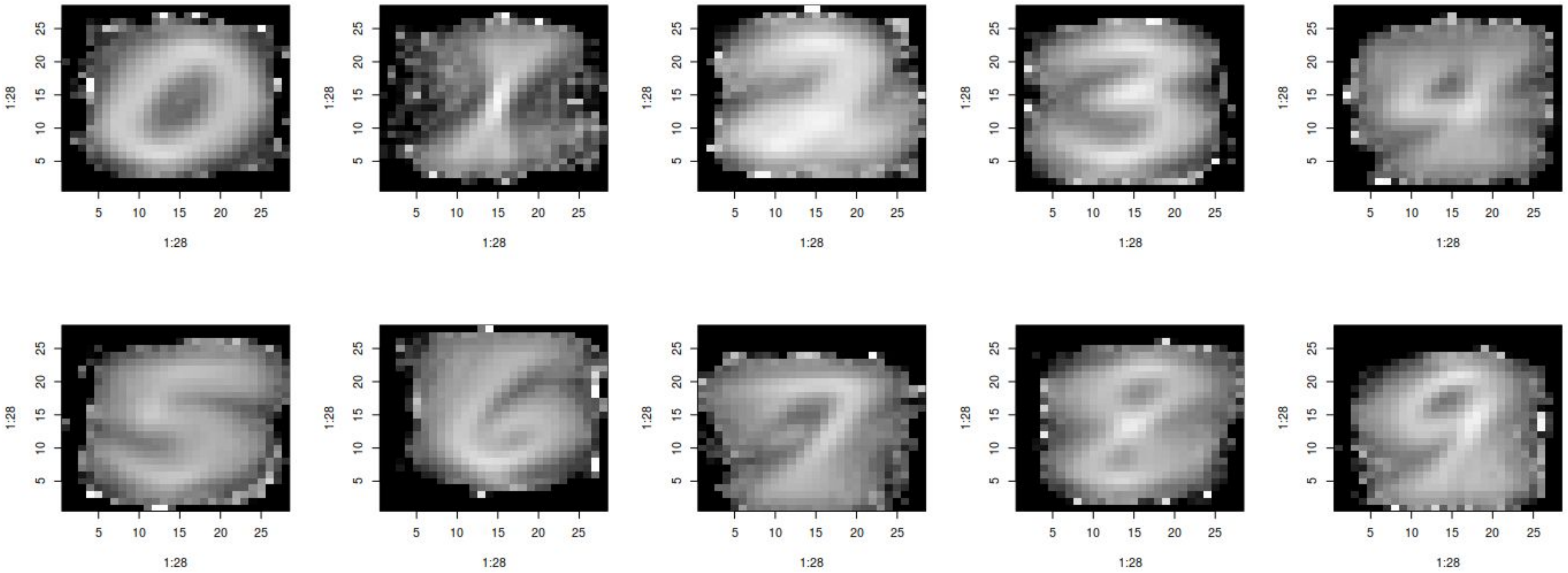
$$R(X_{*j}^{(k)}, \hat{F}_j^{(k)}) = \frac{1}{n^{(k)}} \sum_{i=1}^{n^{(k)}} X_{ij}^{(k)}$$

# Bootstrapping - Nonparametric

We therefore bootstrap additional training examples using the following algorithm.

1. Initialize random $k \in K$;
2. Initialize empty array, $A$, of size $M$;
3. Filter $X^{(k)}$ training examples from $X$;
4. For each $j \in M$,
   (a) Sample with replacement $X_j^{*(k)} = X_{1_j}^{*}{}^{(k)}, X_{2_j}^{*}{}^{(k)}, ..., X_{n^{(k)}{}_j}^{*}{}^{(k)}$;
   (b) Set pixel $j$ in $A$ equal to $R(X_j^{*(k)}, \hat{F}_j^{(k)})$;

# Results - Mean Digits



# Bootstrapping - Parametric

Now that we have a method of generating grayscale values, how do we decide which pixels to bootstrap?

Some considerations:

- There are many zero-valued pixels in the digit images, therefore we are only concerned with the nonzero pixel indices.
- The number of nonzero pixels varies across training examples within each digit class.

Therefore, the goal is to devise a sampling technique to generate varying length sets of pixel indices.

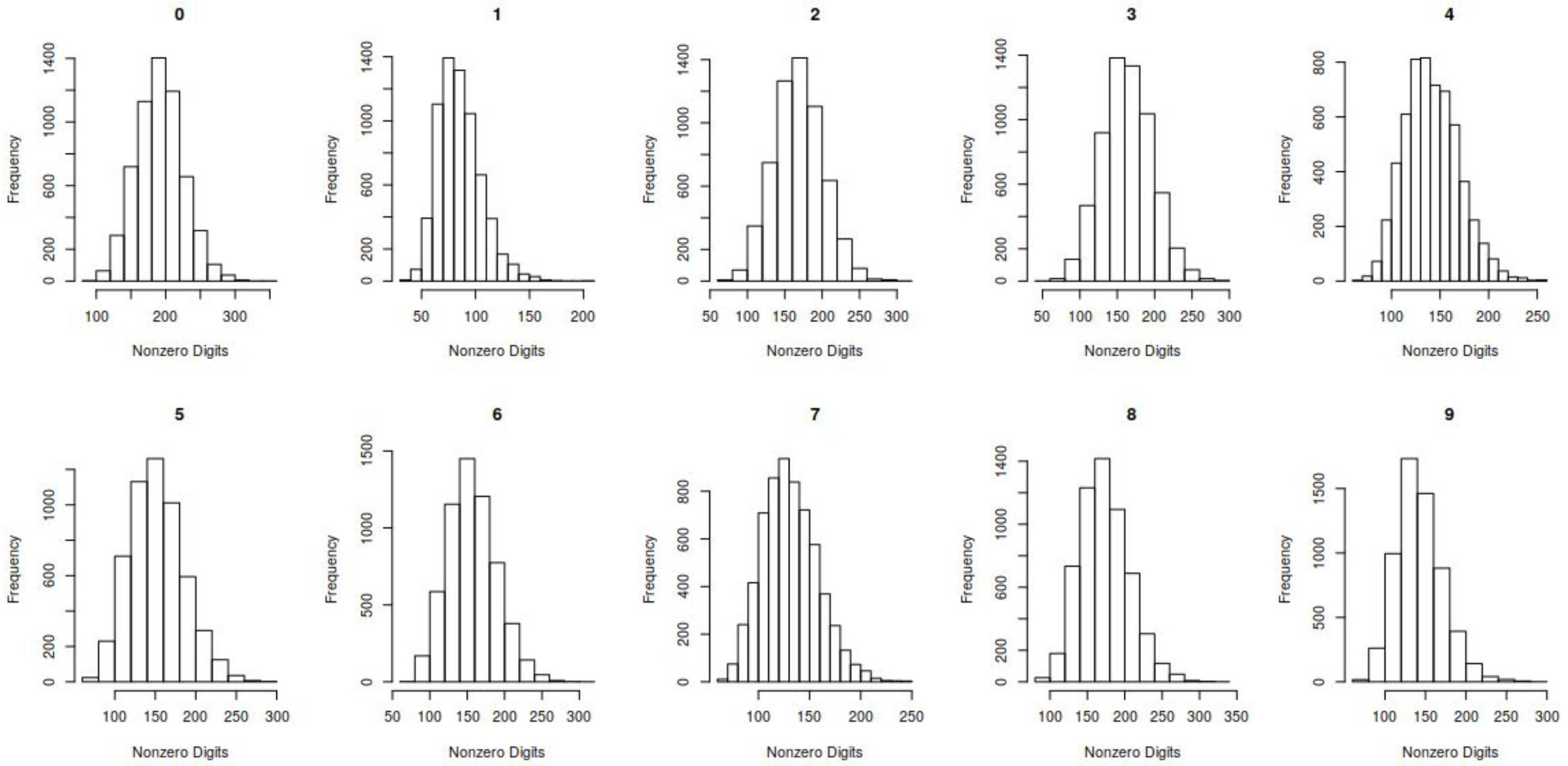
# Bootstrapping - Parametric

Let $H_i^{(k)} = \{h : h \in M | X_{ih}^{(k)} > 0\}$ be the set of nonzero pixel indices for the $i$th training example within the digit class $k$. For $h \in H_i^{(k)}$, $H_{ih}^{(k)}$ is the index of the $h$th nonzero pixel.

We thus have the set $H_i^{(k)} = \{H_{i1}^{(k)}, H_{i2}^{(k)}, ...H_{ih}^{(k)}\}_{h \in H_i^{(k)}}$, of nonzero pixel indices for the $i$th training example of digit $k$. Let the random variable $Y_i^{(k)} = \sum_{h \in H_i^{(k)}} 1$ be the number of nonzero pixels.

On the next slide we can see that the histogram plots of $Y^{(k)}$ are approximately normal for each digit class across all $Y_i^{(k)}$.

# Bootstrapping - Parametric



# Bootstrapping - Parametric

Therefore we can use a parametric bootstrapping technique, with $\hat{Y}^{(k)} \sim N\left(\hat{\mu}^{(k)}, \hat{\sigma}^{(k)2}\right)$ as the random variable for the length of the nonzero set.

Our bootstrapped set of pixel indices therefore becomes

$$H^{*(k)} = \{H_1^{*(k)}, H_2^{*(k)}, \ldots H^{*(k)}_{Y*(k)}\}$$

Where $Y^{*(k)}$ is a random value drawn from $N\left(\hat{\mu}^{(k)}, \hat{\sigma}^{(k)2}\right)$, and $H_1^{*(k)}, H_2^{*(k)}, \ldots H^{*(k)}_{Y*(k)}$ are sampled from $H^{(k)}$, across all $i$ training examples.

# Bootstrapping - Parametric

In this application, pixel indices are sampled according to their weighted empirical probabilities using the following,

$$f^{(k)}(j) = \frac{\sum_{j=1}^{n_j^{(k)}} X_{ij}^{(k)}}{\sum_{i=1}^{n^{(k)}} X_{ij}^{(k)}}$$

Conceptually this means that we are more likely to draw pixel indices that have a large amount of high grayscale values across the training examples for that digit.

# Bootstrapping - Parametric

We therefore bootstrap sets of pixel indices using the following algorithm.

1. Calculate $\hat{\mu}^{(k)}$ and $\hat{\sigma}^{(k)2}$ for each $k$ digit class;
2. Calculate weighted empirical probabilities $f^{(k)}(j)$ for each pixel $j$ in each digit class $k$;
3. Draw a random integer $k$ from $K = \{0, 1, ..., 9\}$;
4. Generate a random $Y^{*(k)} \sim N\left(\hat{\mu}^{(k)}, \hat{\sigma}^{(k)2}\right)$;
5. Sample $Y^{*(k)}$ pixel index values $H_i^{*(k)}$, from $H^{(k)}$, based on $f^{(k)}(j)$;

For each of these parametrically sampled bootstrapped pixel index values, we sample pixel values nonparametrically using the previous algorithm.

# Results - Example Bootstrapped Digits