Types of Machine Learning and Model Selection

Machine Learning
CS5824/ECE5424
Bert Huang
Virginia Tech

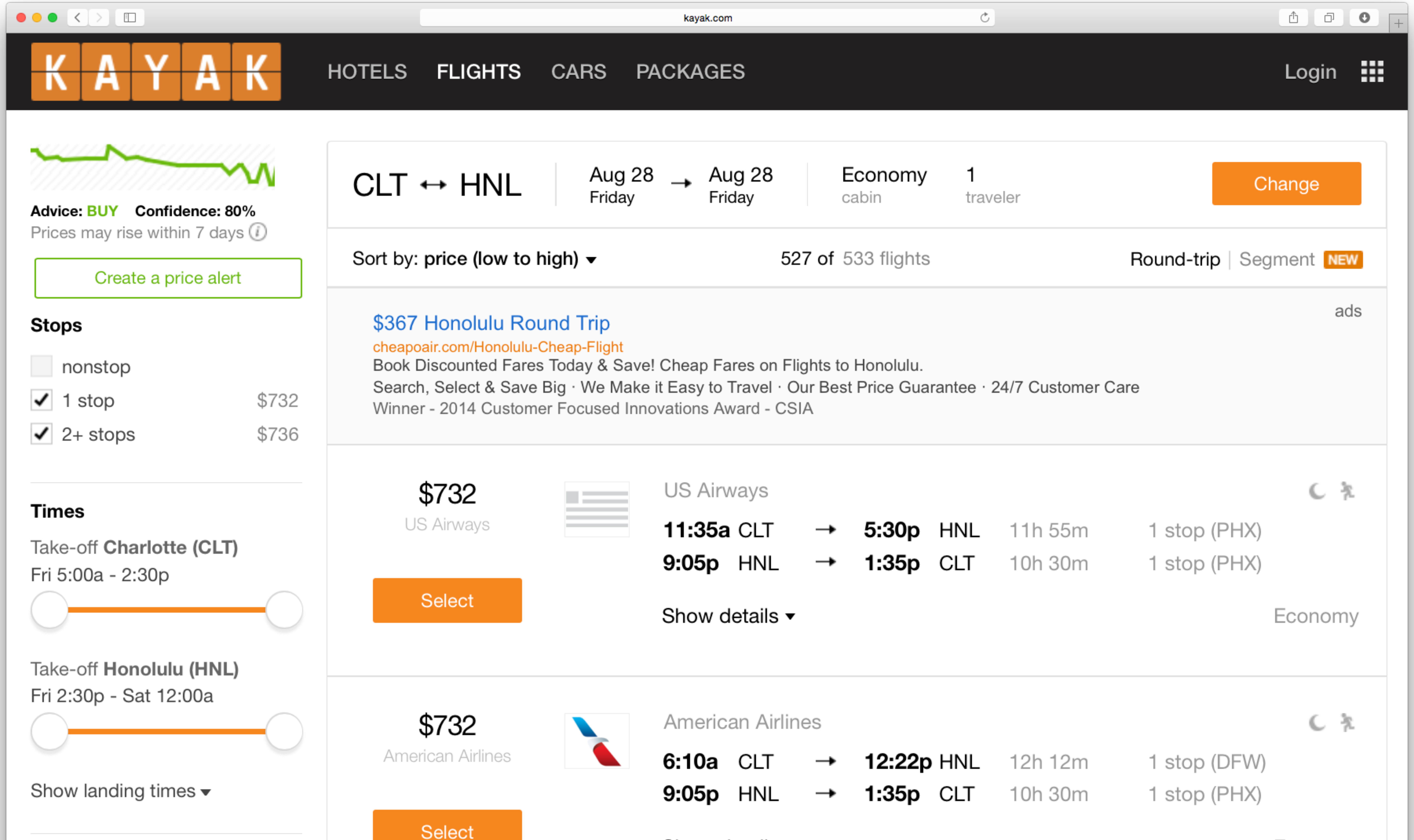
1st Learning Setting

- Draw data set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ from distribution $\mathbb{D}$
- Algorithm A learns hypothesis $h \in H$ from set H of possible hypotheses $A(D) = h$
- We measure the quality of h as the expected **loss**: $\underset{(x,y)\in\mathbb{D}}{E}[\ell(y, h(x))]$
  - This quantity is known as the **risk**
  - E.g., loss could be the Hamming loss $\ell_{\text{Hamming}}(a, b) = \begin{cases} 0 & \text{if } a = b \\ 1 & \text{otherwise} \end{cases}$

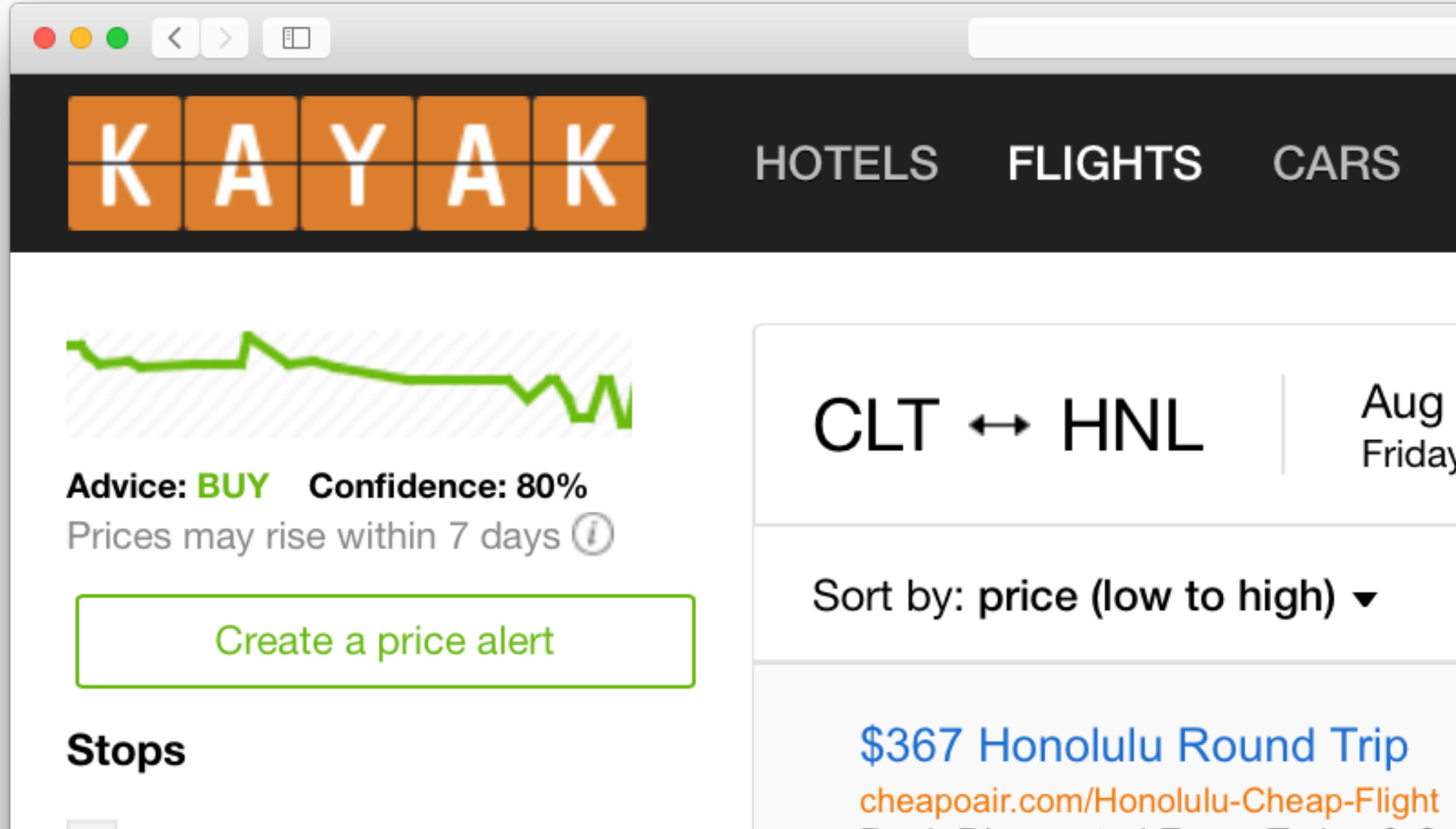
Example: Digit Classification

http://ufldl.stanford.edu/housenumbers/

Example: Airline Price Prediction



Example: Airline Price Prediction



Batch Supervised Learning

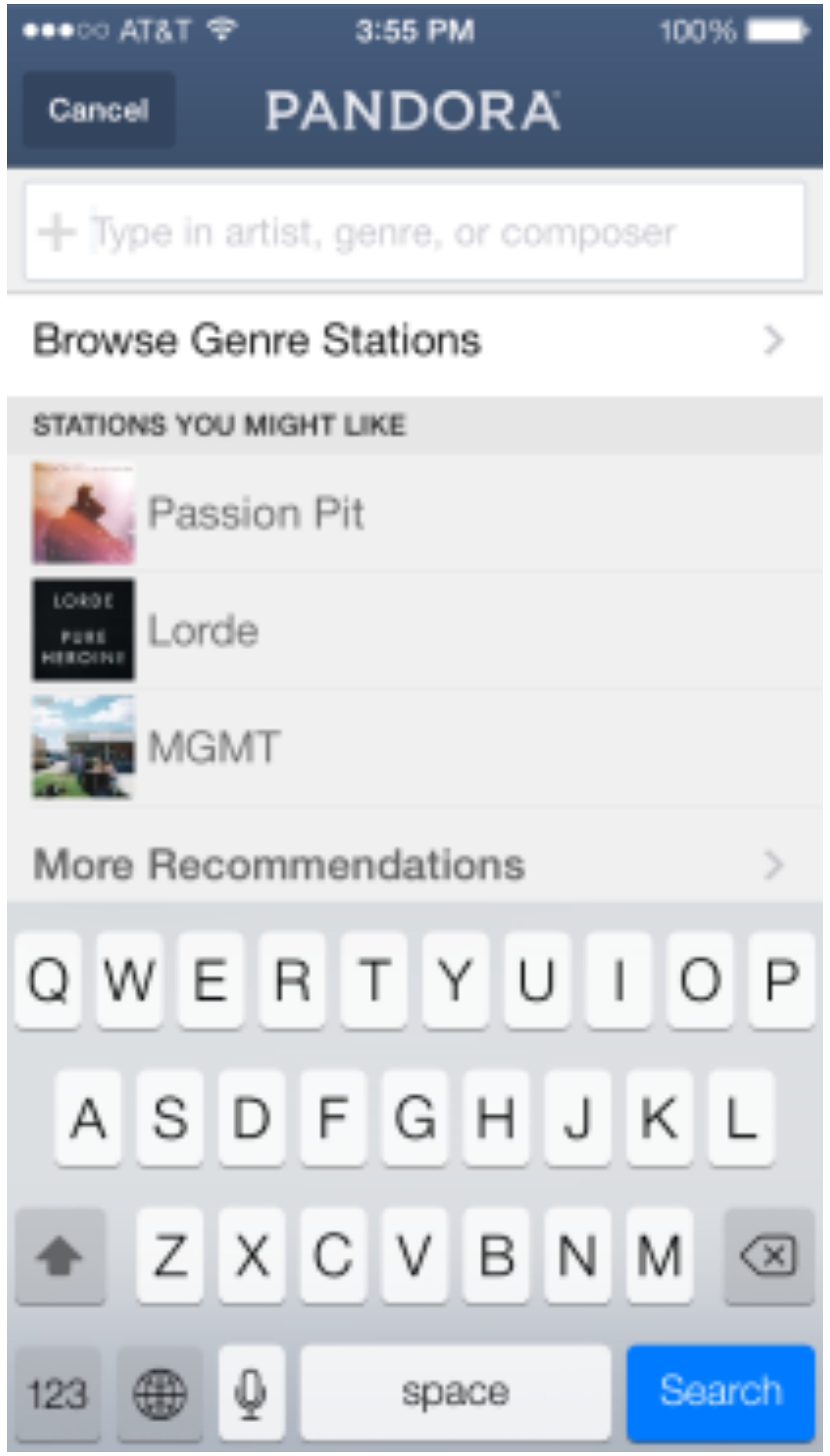
- Draw data set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ from distribution $\mathbb{D}$
- Algorithm A learns hypothesis $h \in H$ from set H of possible hypotheses $A(D) = h$
- We measure the quality of h as the expected **loss**: $\underset{(x,y) \in \mathbb{D}}{E} [\ell(y, h(x))]$
    - This quantity is known as the **risk**
    - E.g., loss could be the Hamming loss $\ell_{\mathrm{Hamming}}(a, b) = \begin{cases} 0 & \text{if } a = b \\ 1 & \text{otherwise} \end{cases}$

classification

Online Supervised Learning

- In step ***t***, draw data point ***x*** from distribution $\mathbb{D}$
- Current hypothesis ***h*** guesses the label of ***x***
- Get true label from oracle ***O***
- Pay penalty if ***h*(*x*)** is wrong (or earn reward if correct)
- Learning algorithm updates to new hypothesis based on this experience
  - Does not store history

Example: Recommendation



Learning Settings

- Supervised or unsupervised (or semi-supervised, weakly supervised, transductive…)
- Online or batch (or reinforcement…)
- Classification, regression
  - (or structured output, clustering, dimensionality reduction…)
- Parametric or non-parameteric

Functional Perspective

Input	Learning Setting
Batch of Data Points with Labels	Batch Supervised Learning
Batch of Data Points	Batch Unsupervised Learning
Data Point(s) and Previous Model	Online Supervised Learning

Concepts

- Supervised and unsupervised learning
- Online and batch learning
- Discriminative and generative
- Output of models: classification and regression

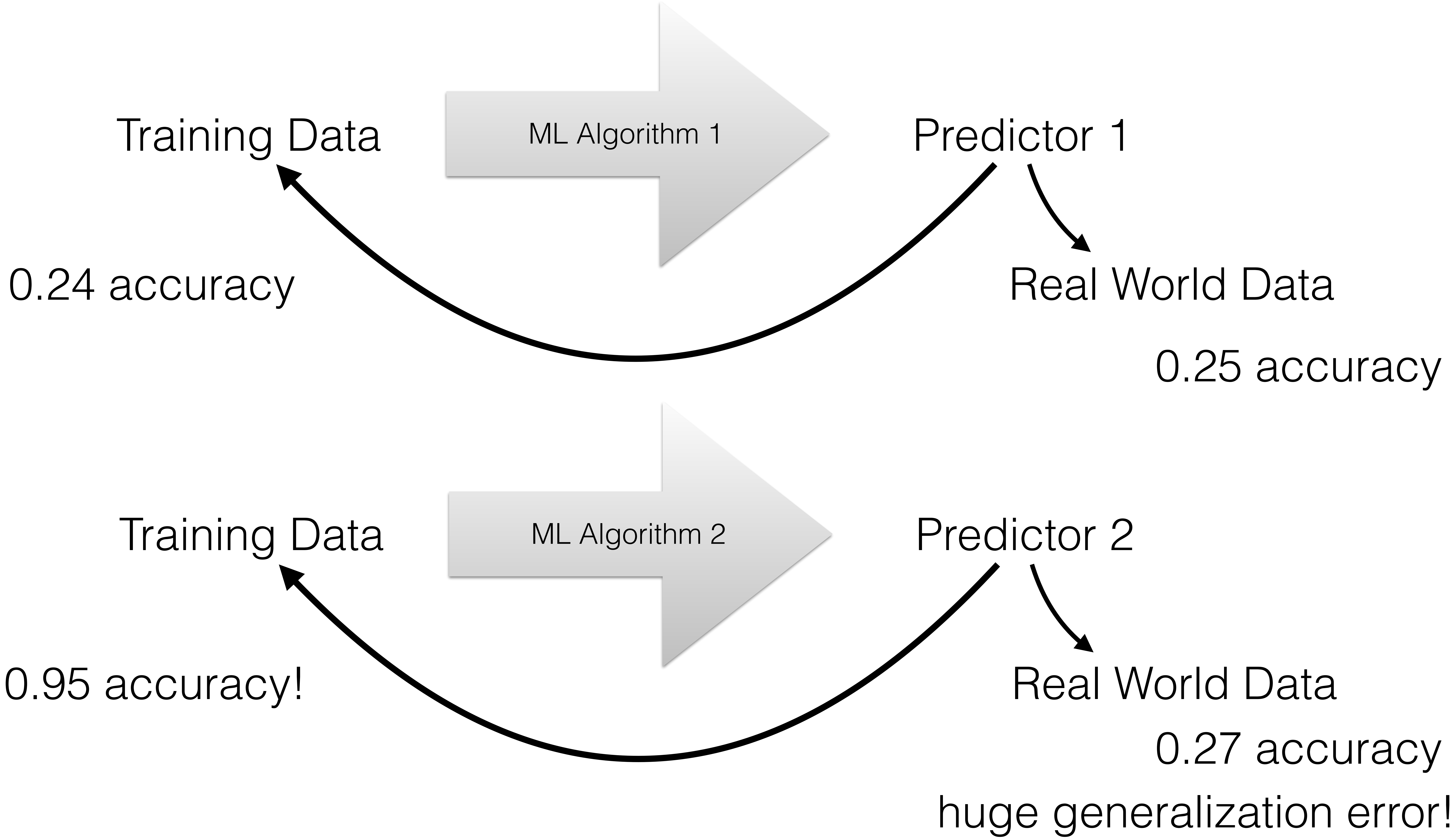
Model Selection

Outline

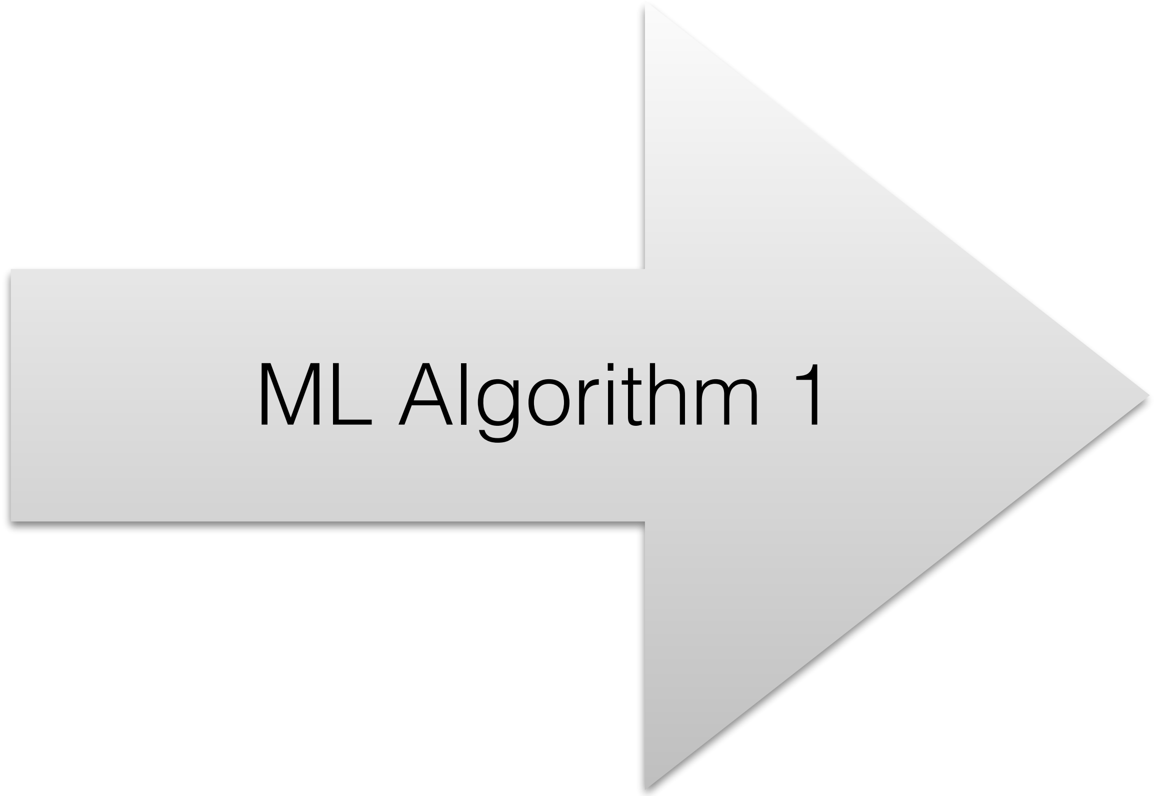
- Overfitting and underfitting
- Bias and variance
- Validation for model selection

Outline

- Overfitting and underfitting
- Bias and variance
- Validation for model selection

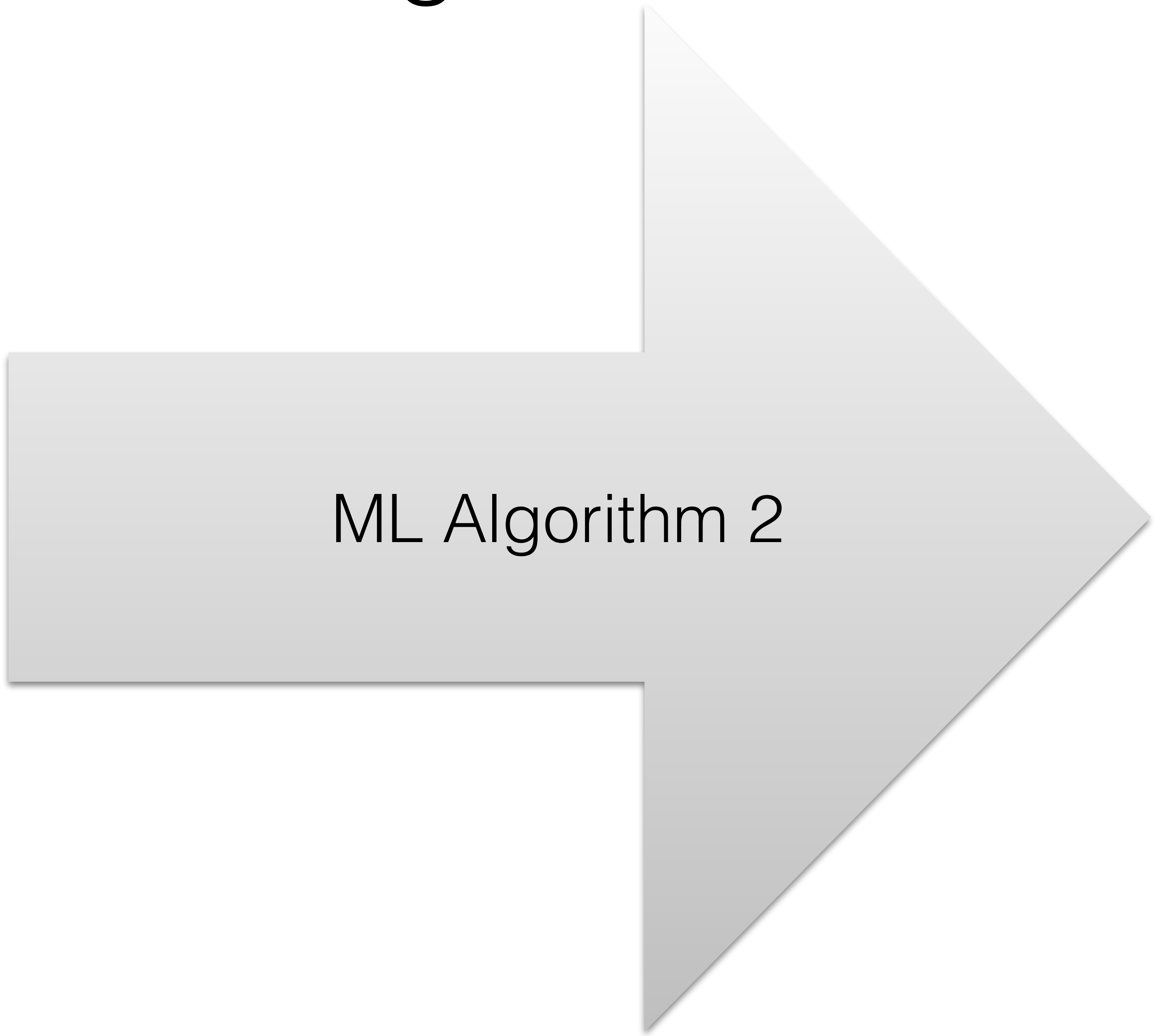

Training Data
ML Algorithm 1
Predictor 1
0.24 accuracy
Real World Data
0.25 accuracy
Training Data
ML Algorithm 2
Predictor 2
0.95 accuracy!
Real World Data
0.27 accuracy
huge generalization error!

Underfitting



- Low dimensional
- Heavily regularized
- Bad modeling assumptions

Overfitting



- High dimensional or non-parametric
- Weakly regularized
- Not enough modeling assumptions
- Not enough data


data


data
shape-preserving


data
shape-preserving
linear


data
shape-preserving
linear
quadratic

Overfitting and Underfitting

- Training models too complex can cause overfitting
- Training models too simple (or wrong) can cause underfitting

Outline

- Overfitting and underfitting
- Bias and variance
- Validation for model selection

Bias and Variance

- Both contribute to **error**
- Bias: error from incorrect modeling assumptions
- Variance: error from random noise

http://scott.fortmann-roe.com/docs/BiasVariance.html

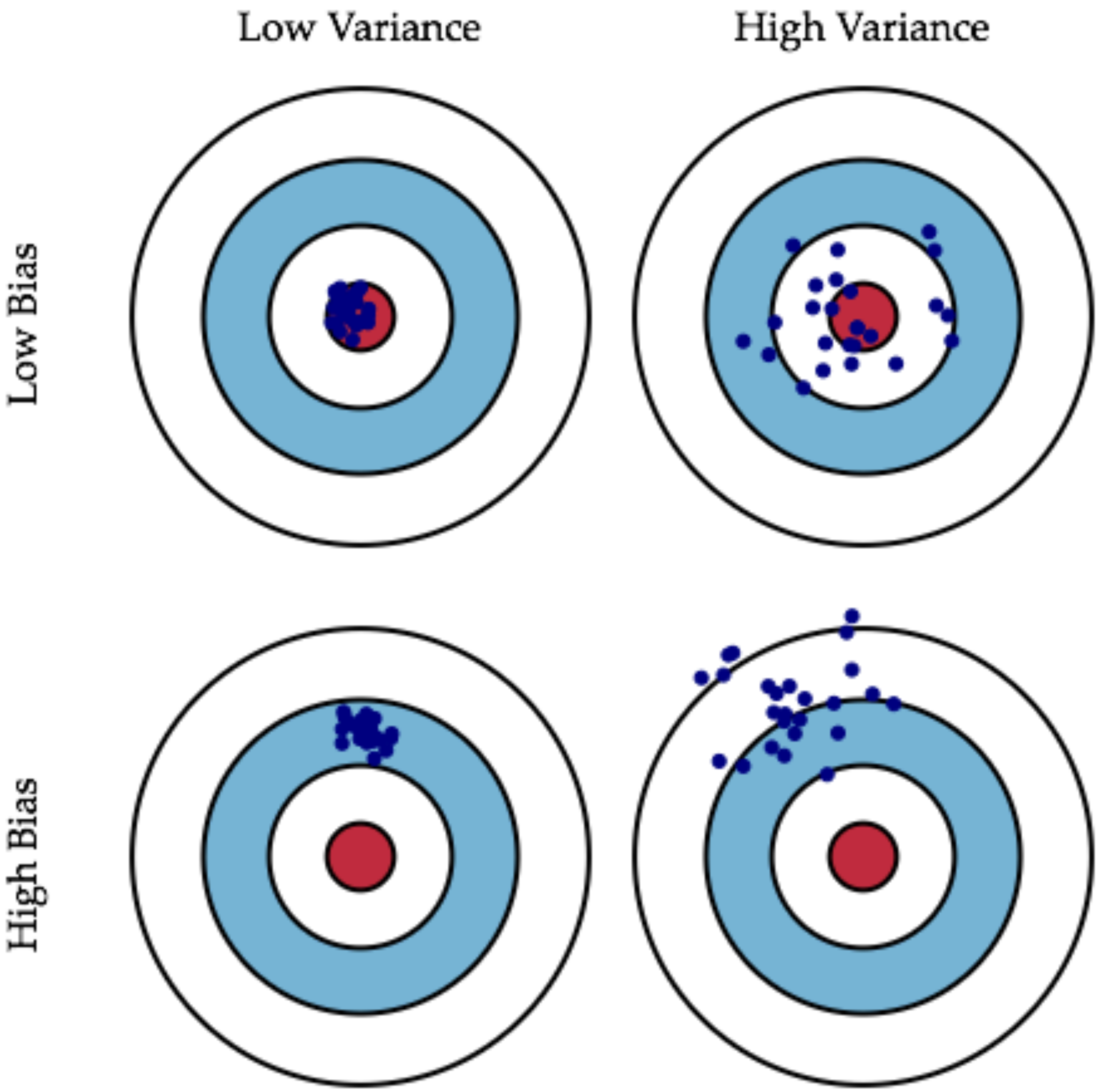


Fig. 1 Graphical illustration of bias and variance.

Mathematical Definition

after Hastie, et al. 2009 [1]

If we denote the variable we are trying to predict as Y and our covariates as X, we may assume that there is a relationship relating one to the other such as $Y = f(X) + \epsilon$ where the error term ϵ is normally distributed with a mean of zero like so $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$.

We may estimate a model $f\hat{(X)}$ of $f(X)$ using linear regressions or another modeling technique. In this case, the expected squared prediction error at a point x is:

$$Err(x) = E\left[(Y - f\hat{(x)})^2\right]$$

This error may then be decomposed into bias and variance components:

$$Err(x) = \left(E[f\hat{(x)}] - f(x)\right)^2 + E\left[\left(f\hat{(x)} - E[f\hat{(x)}]\right)^2\right] + \sigma_e^2$$

$$Err(x) = \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{Irreducible\ Error}$$

That third term, irreducible error, is the noise term in the true relationship that cannot fundamentally be reduced by any model. Given the true model and infinite data to calibrate it, we should be able to reduce both the bias and variance terms to 0. However, in a world with imperfect models and finite data, there is a tradeoff between minimizing the bias and minimizing the variance.

$$Err(x) = E\left[(Y - \hat{f}(x))^2\right]$$

be decomposed into bias and variance components:

$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \sigma_e^2$$

expected learned function

true function

learned function

expected learned function

$$Err(x) = \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{Irreducible\ Error}$$

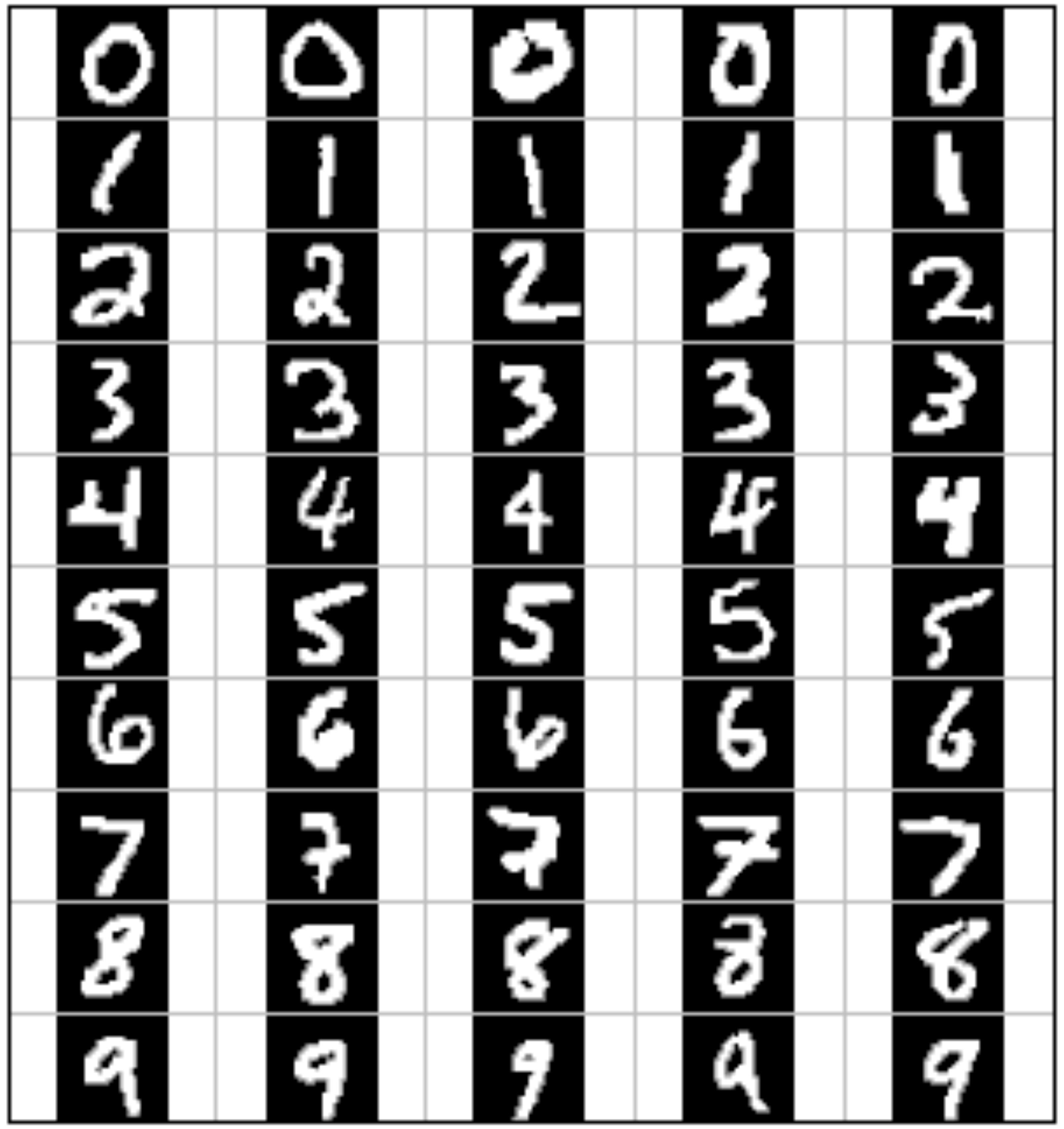
ducible error, is the noise term in the true relationship that ca

nodel. Given the true model and infinite data to calibrate it, w

Outline

- Overfitting and underfitting
- Bias and variance
- Validation for model selection

Nearest-Neighbor Classifiers



classifier = {

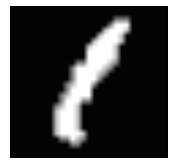
 : 0,

 : 0,

 : 0,

 : 0,

 : 0,

 : 1,

 : 1,

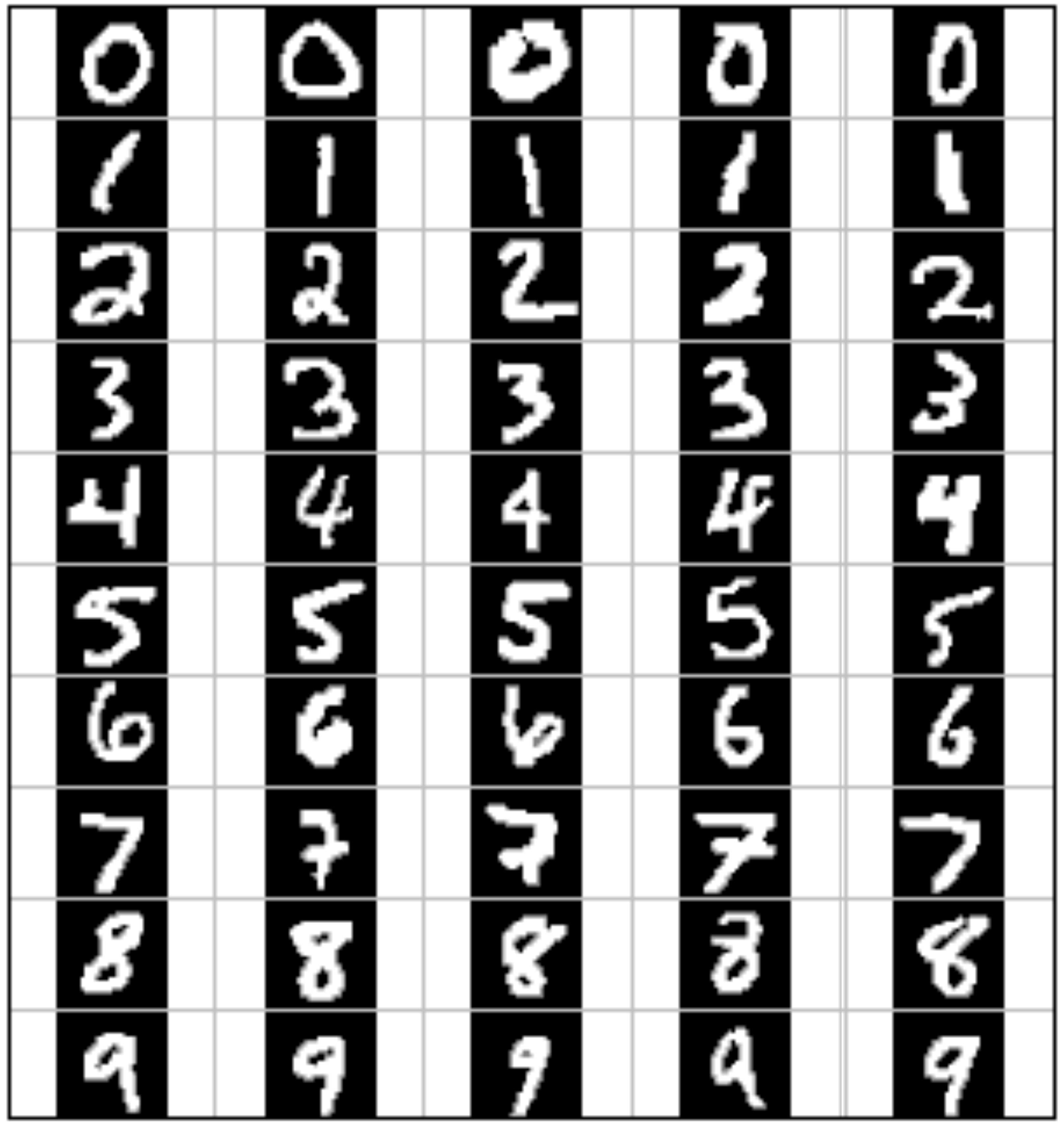
…

}

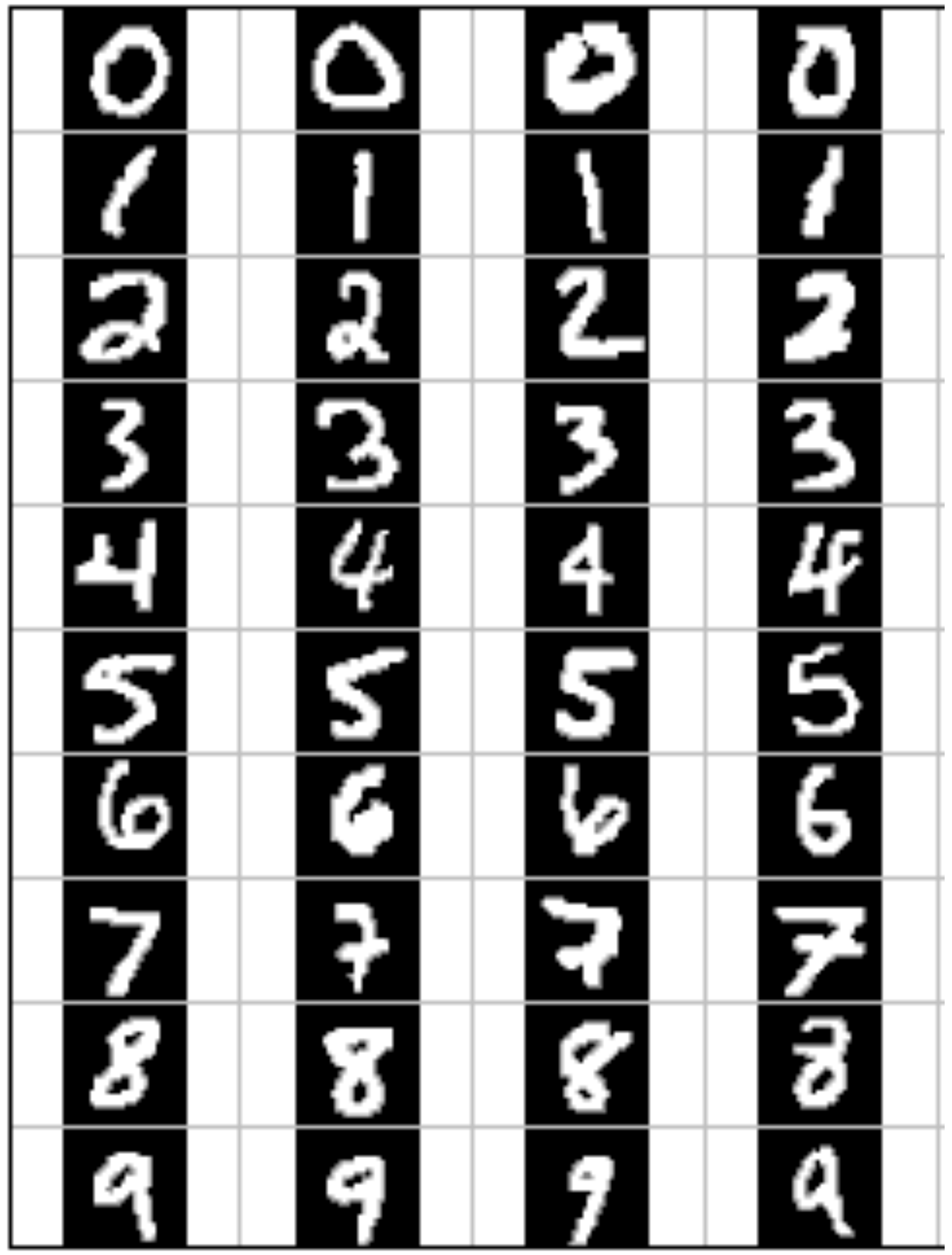
100% training accuracy!

53% testing accuracy…

Held-out Validation

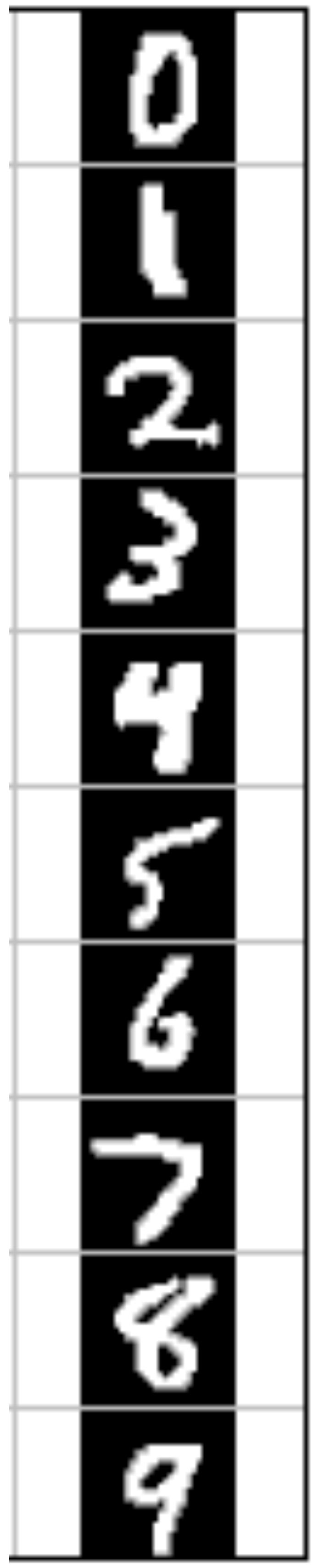


Held-out Validation



training data

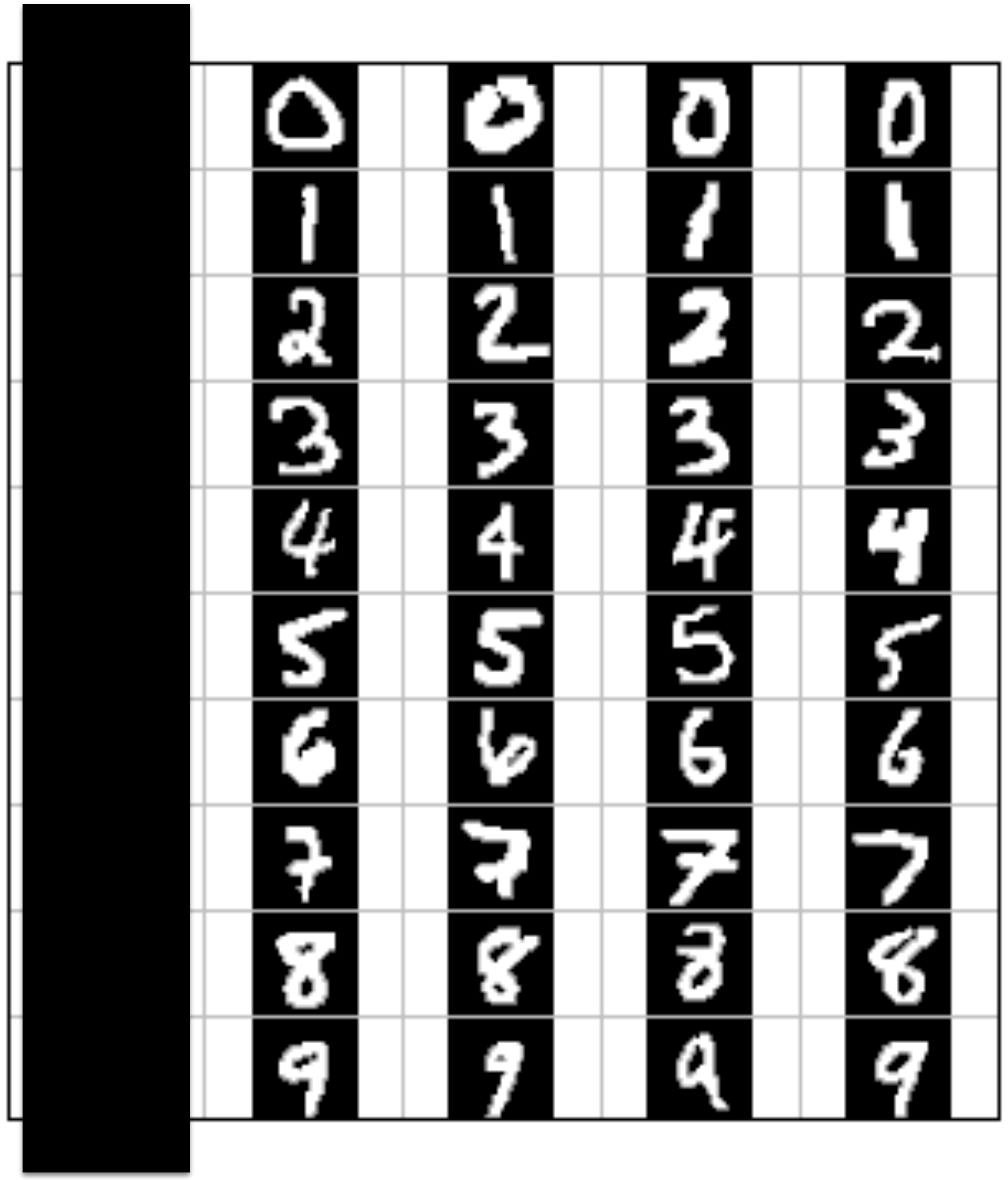
	Accuracy on training data	Accuracy on validation data
Simple	0.91	0.83
Medium	0.95	0.88
Complex	0.99	0.79
Super Complex	1.0	0.54



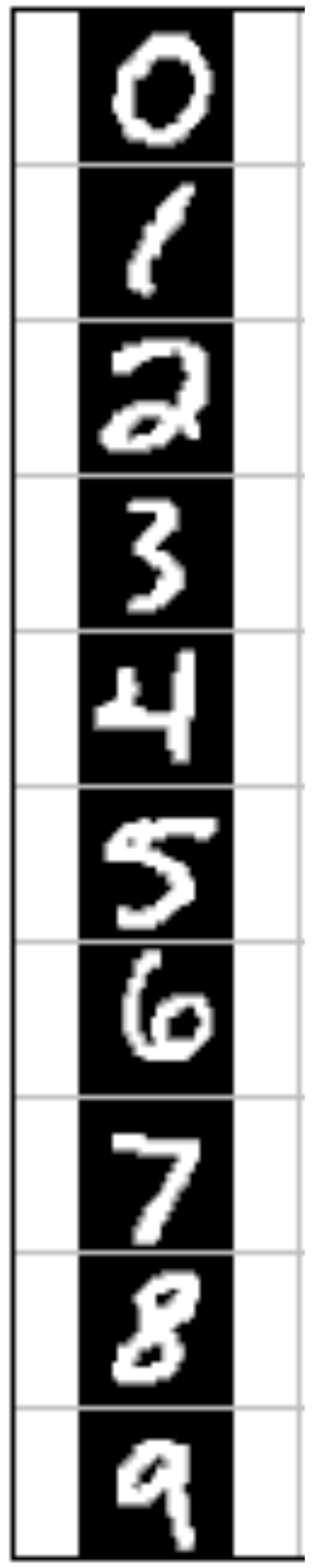
validation data

Cross Validation

Fold 1



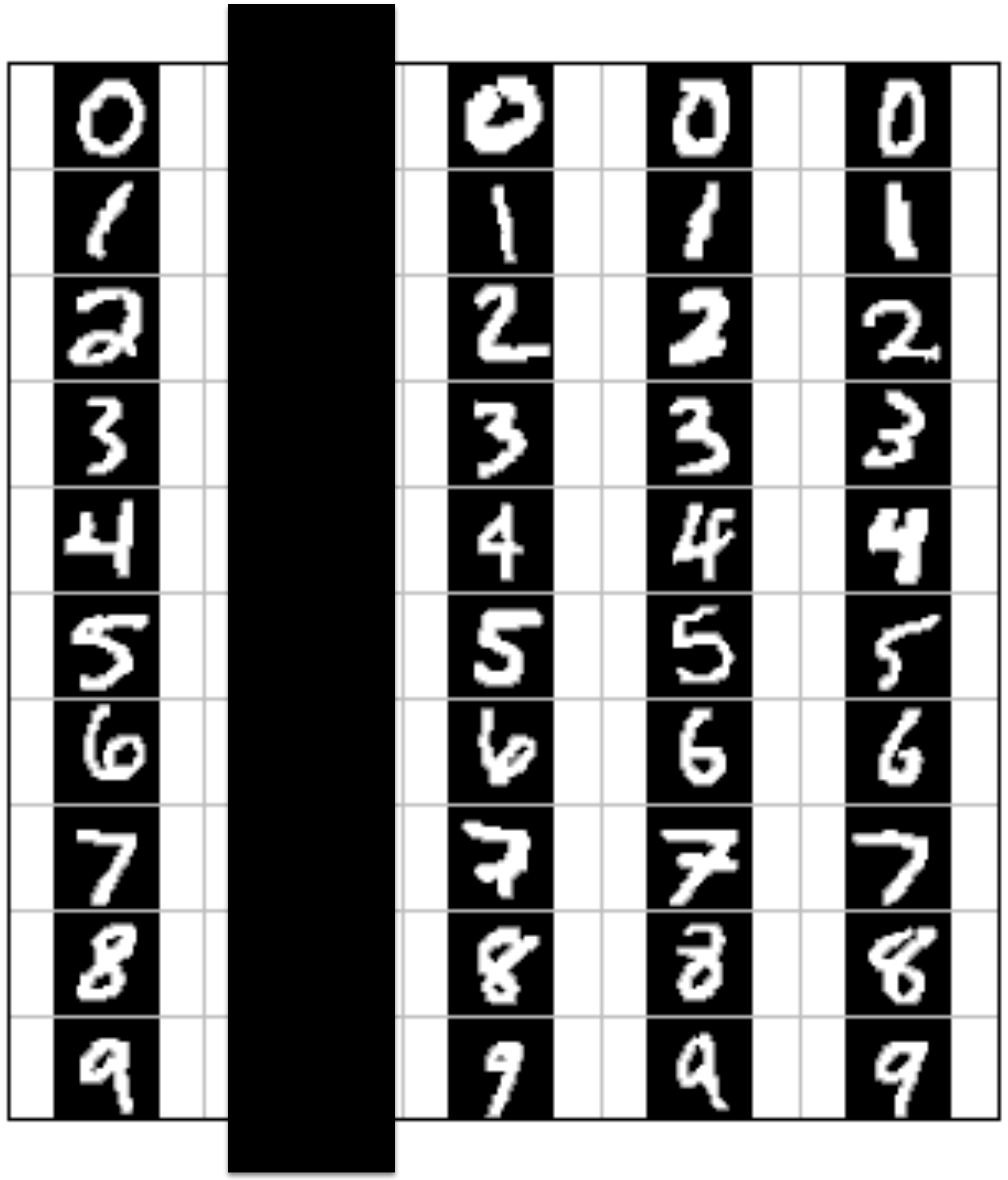
training data



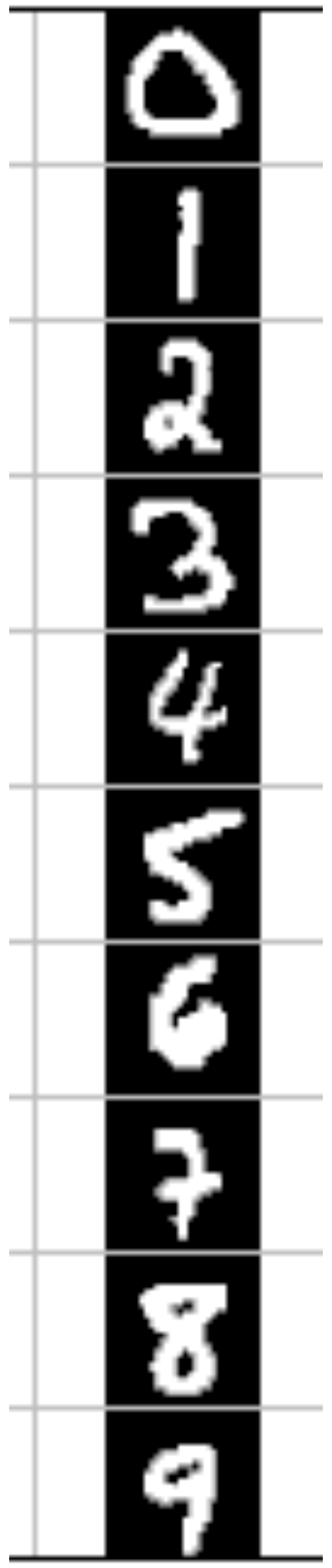
validation data

Cross Validation

Fold 2



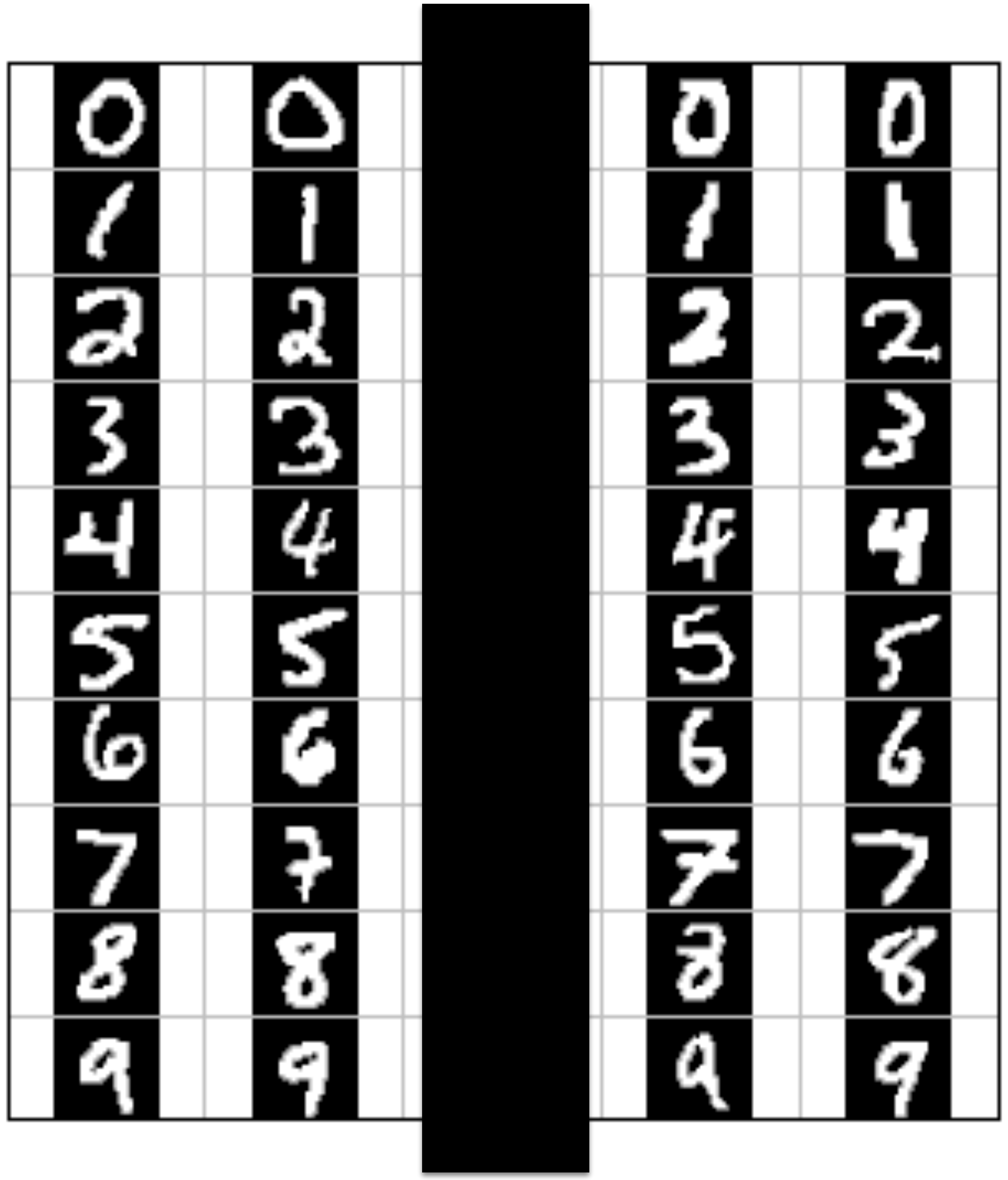
training data



validation data

Cross Validation

Fold 3

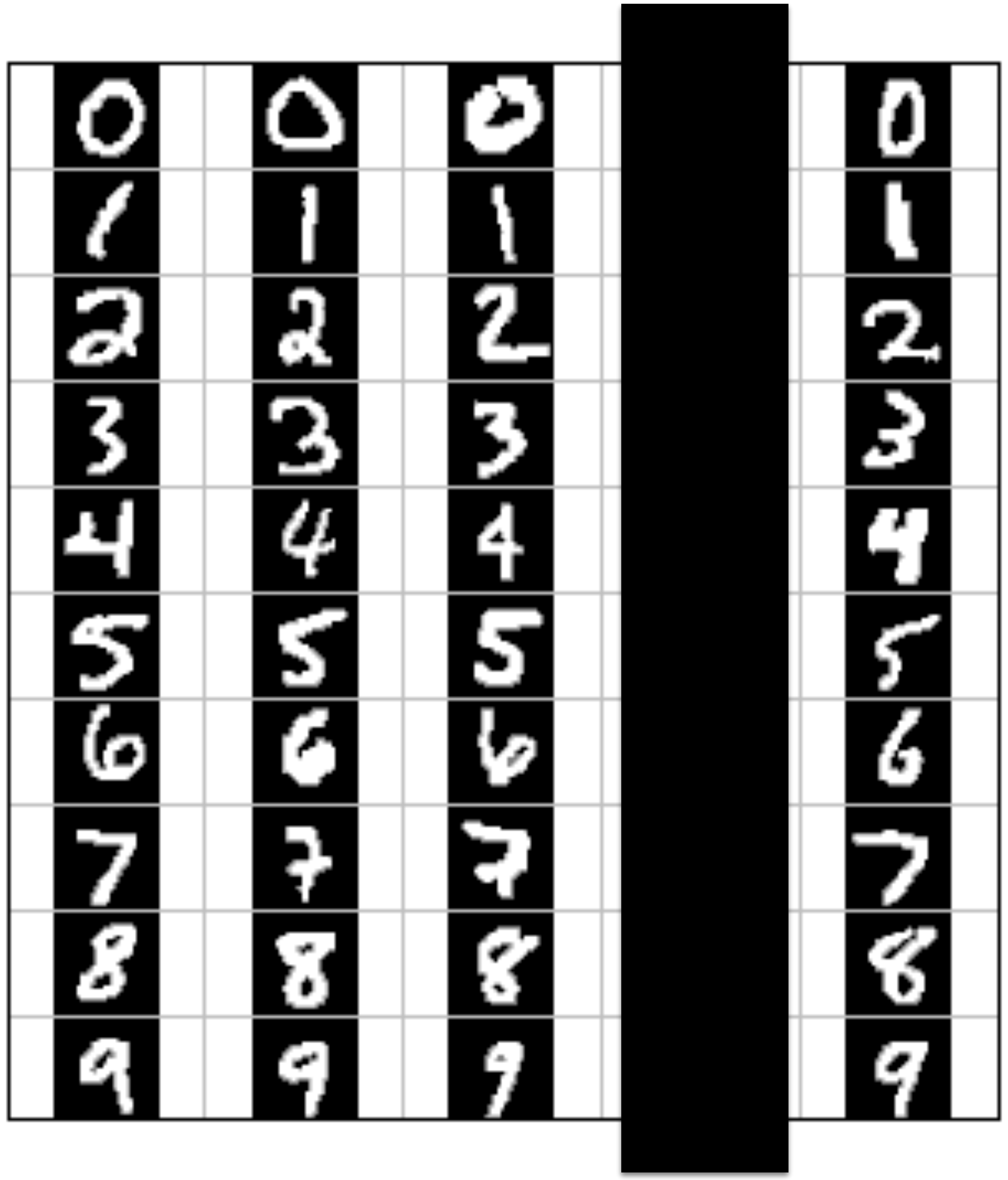


training data

validation data

Cross Validation

Fold 4

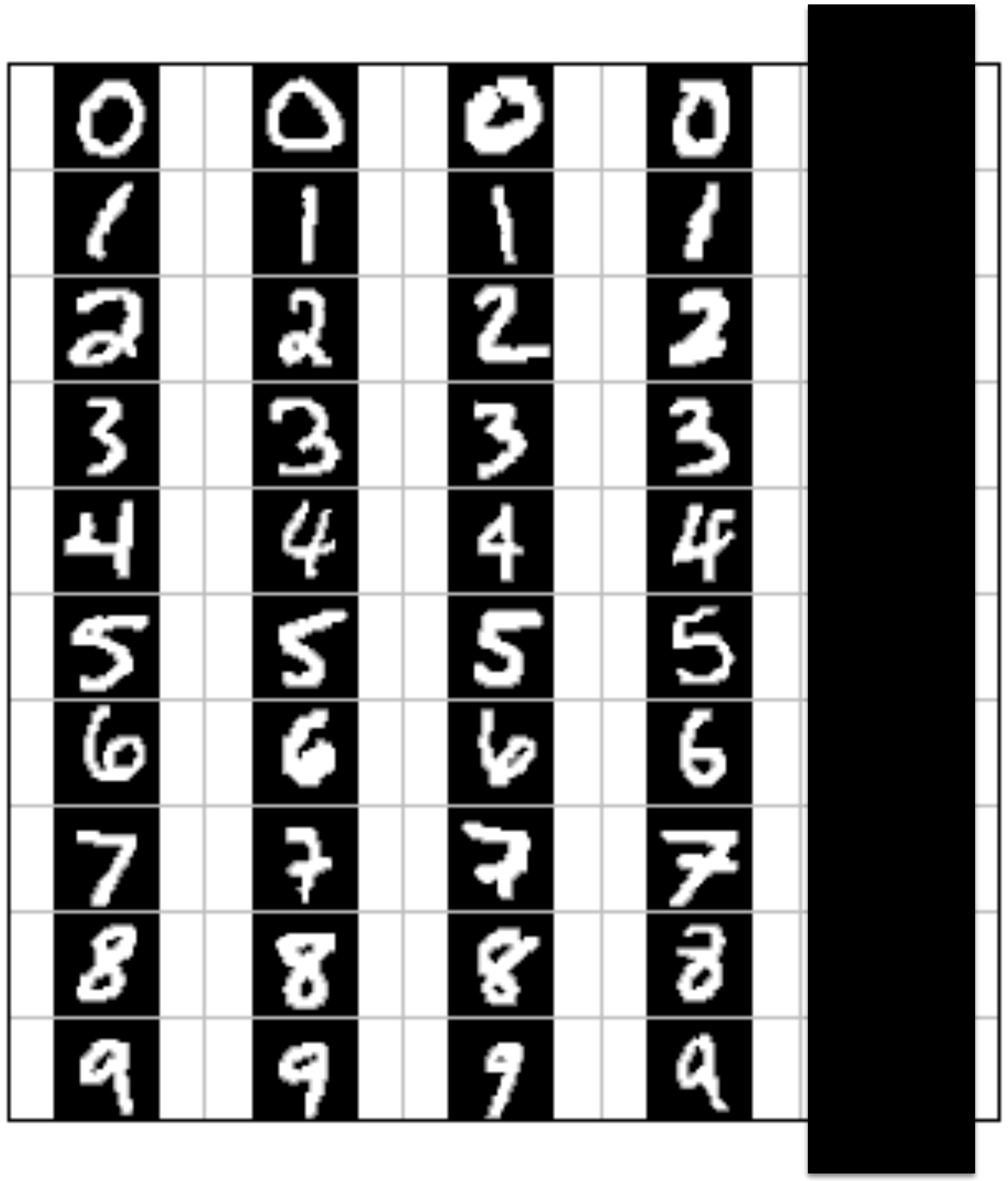


training data

validation data

Cross Validation

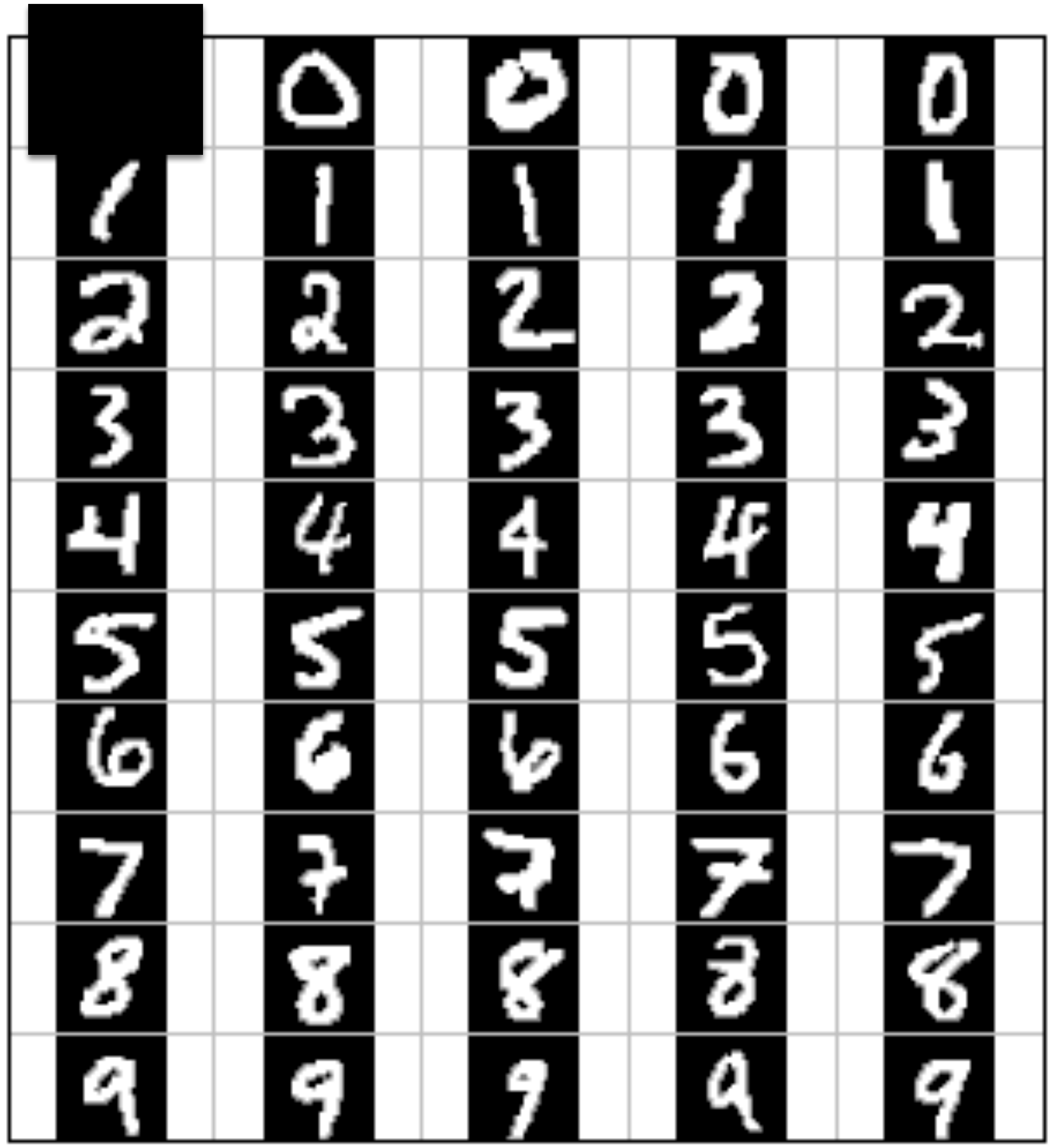
Fold 5



training data

validation data

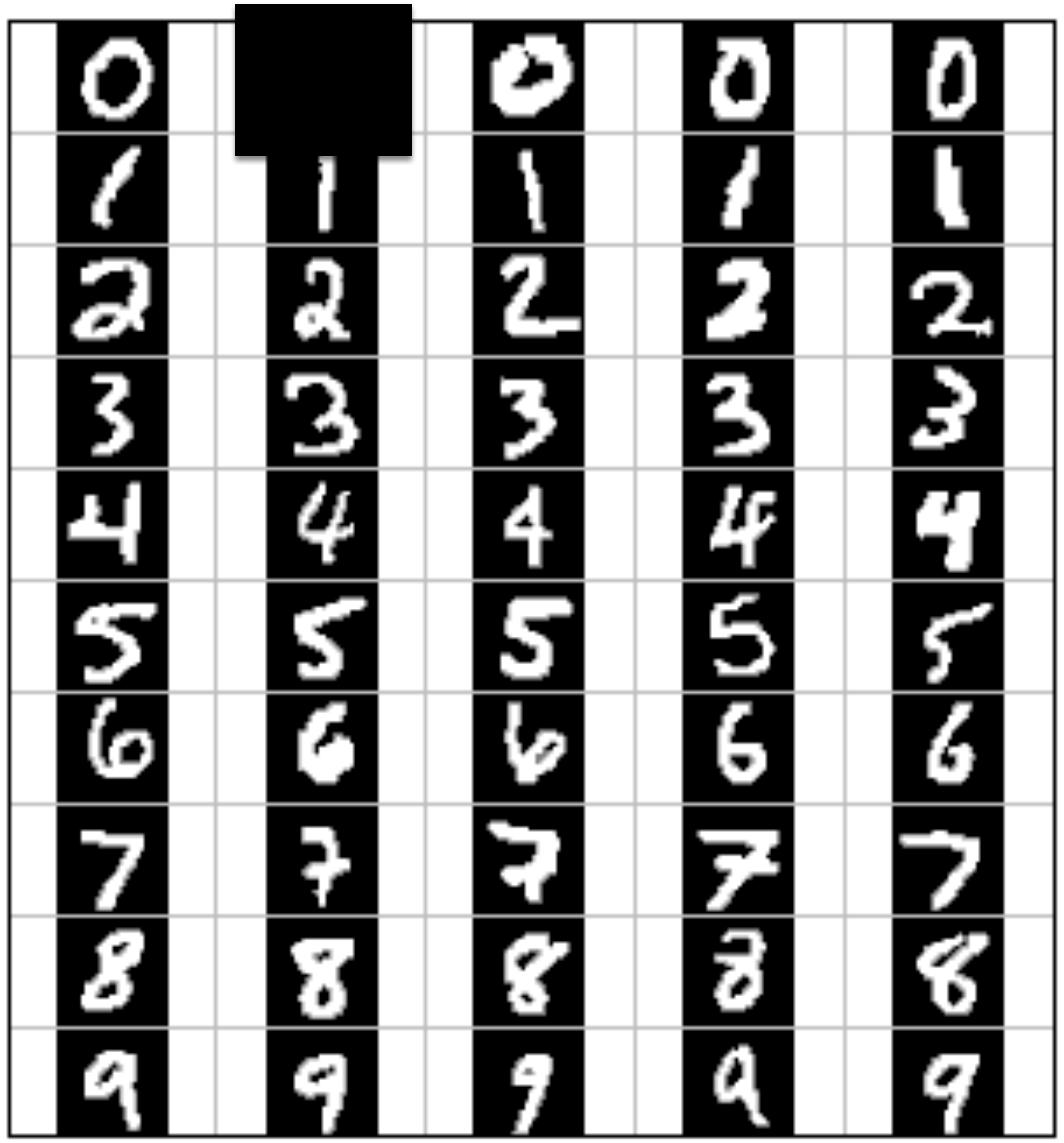
Leave-one-out Cross Validation



training data

validation data

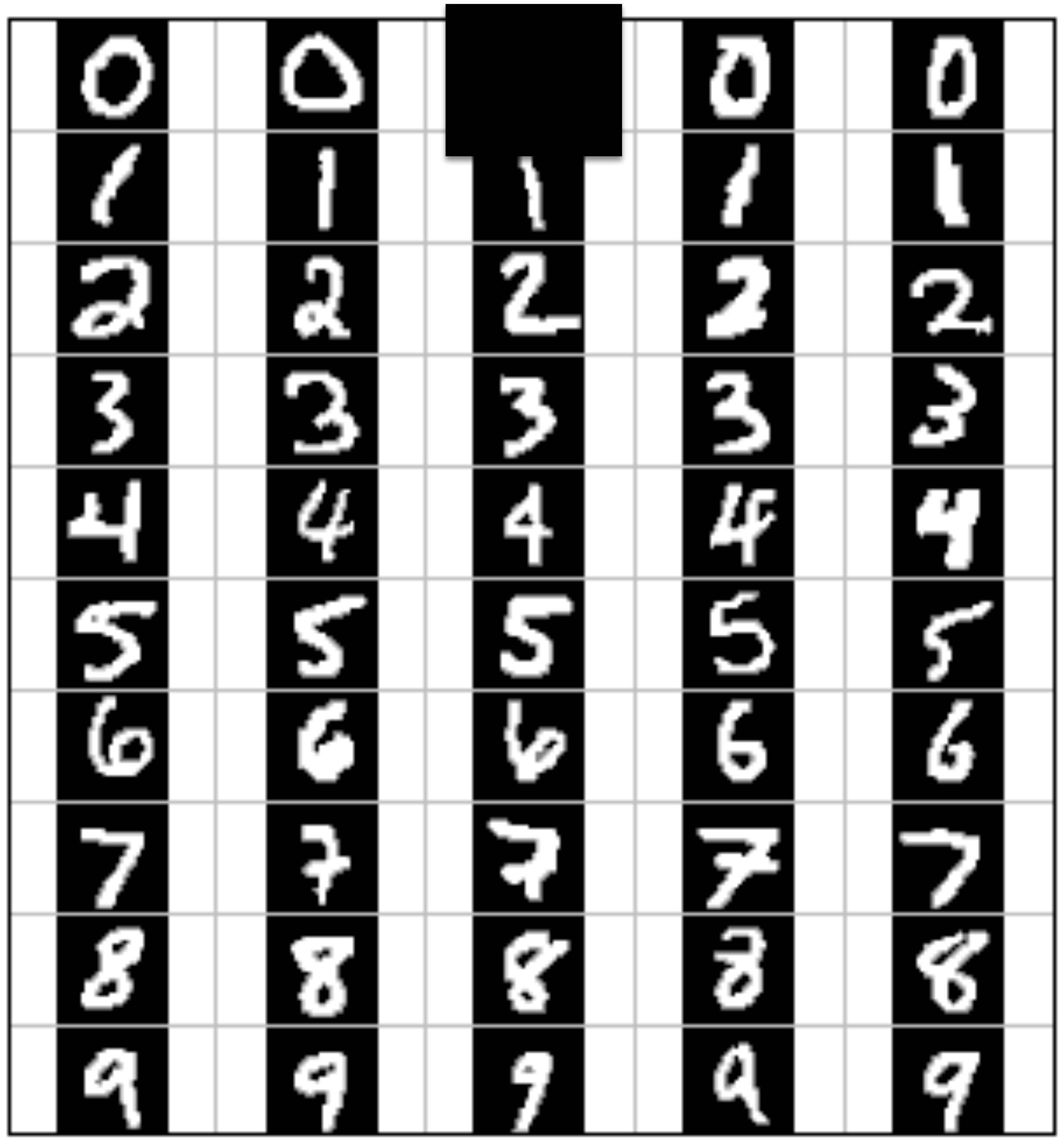
Leave-one-out Cross Validation



training data

validation data

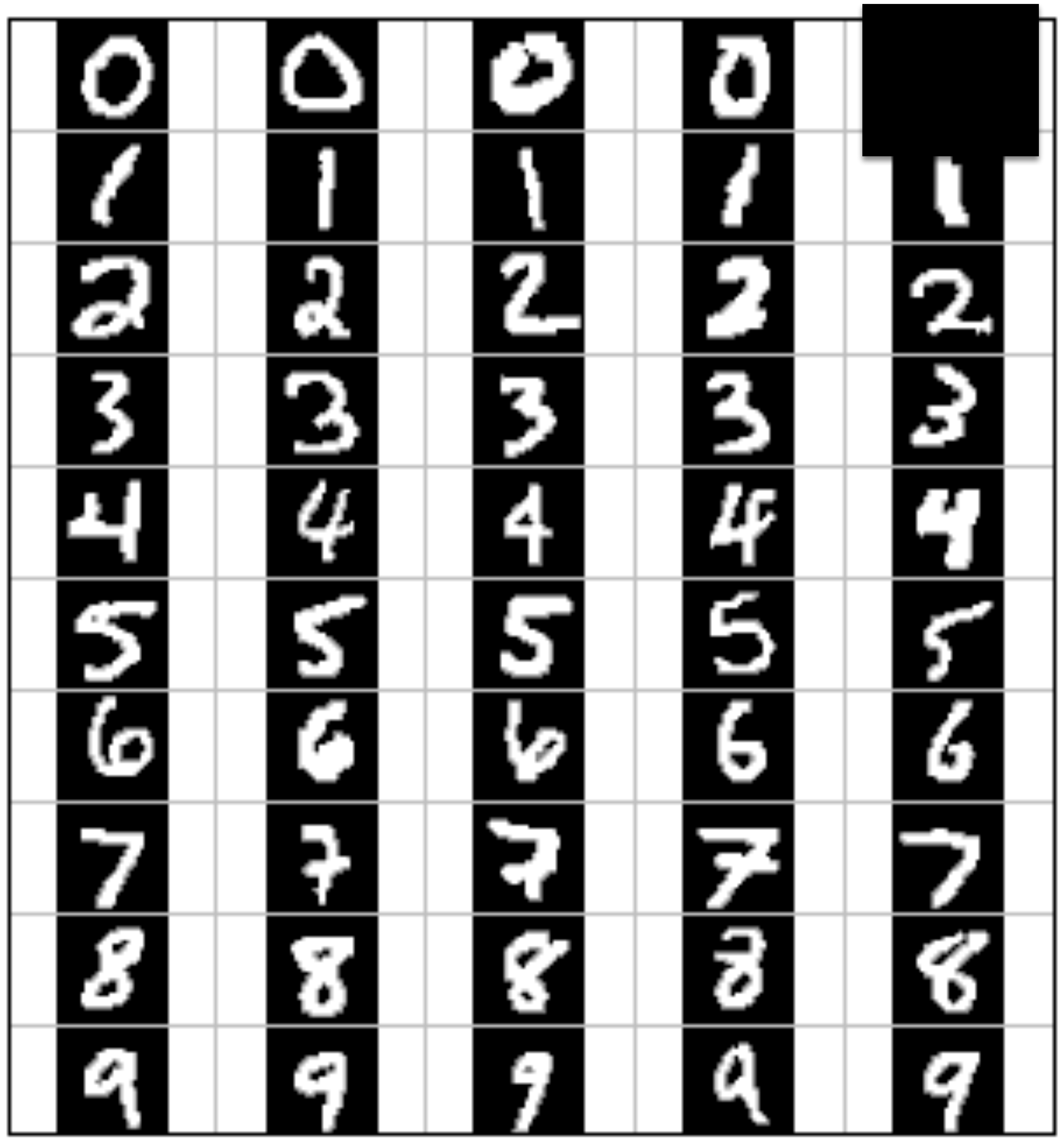
Leave-one-out Cross Validation



training data

validation data

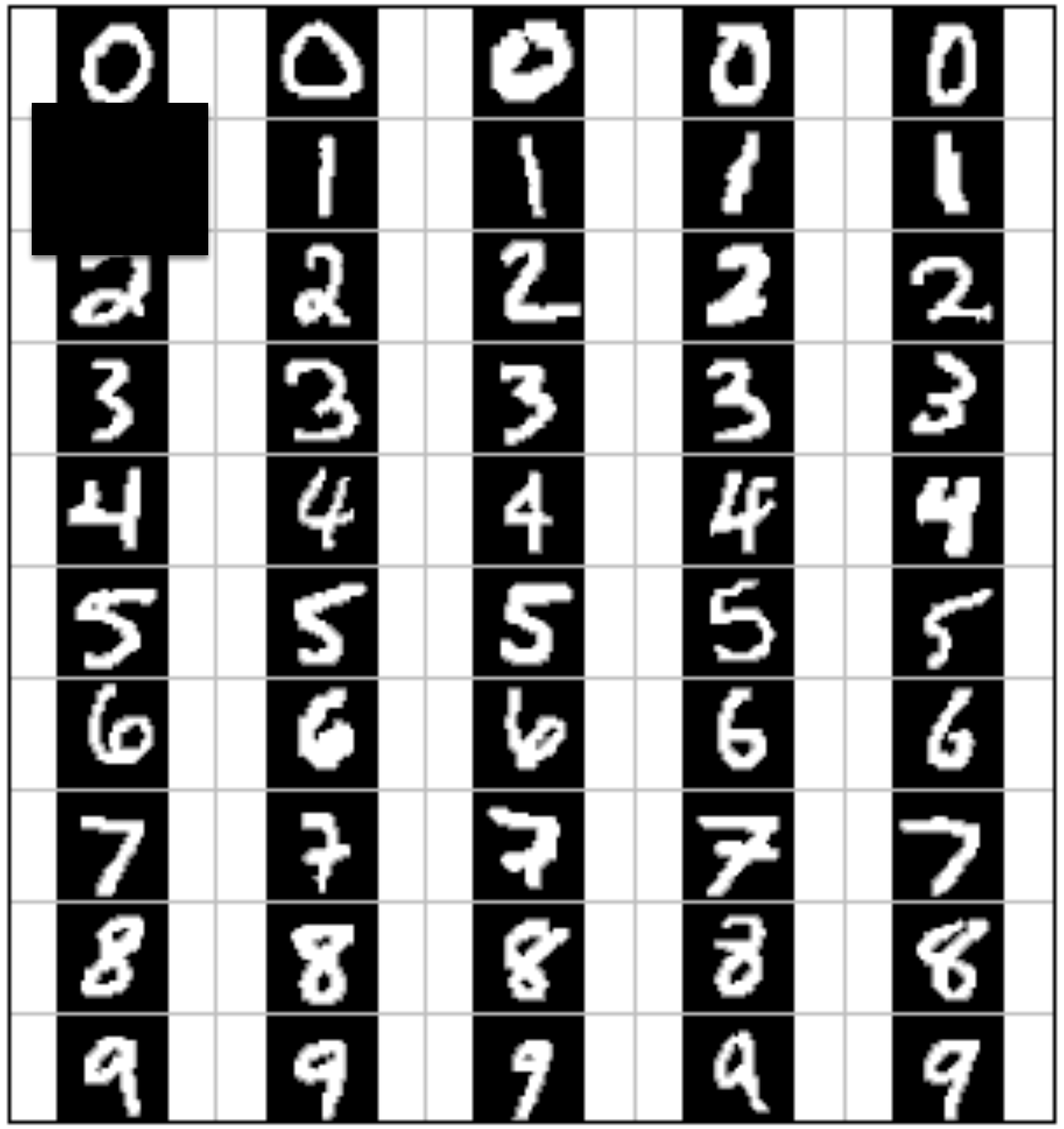
Leave-one-out Cross Validation



training data

validation data

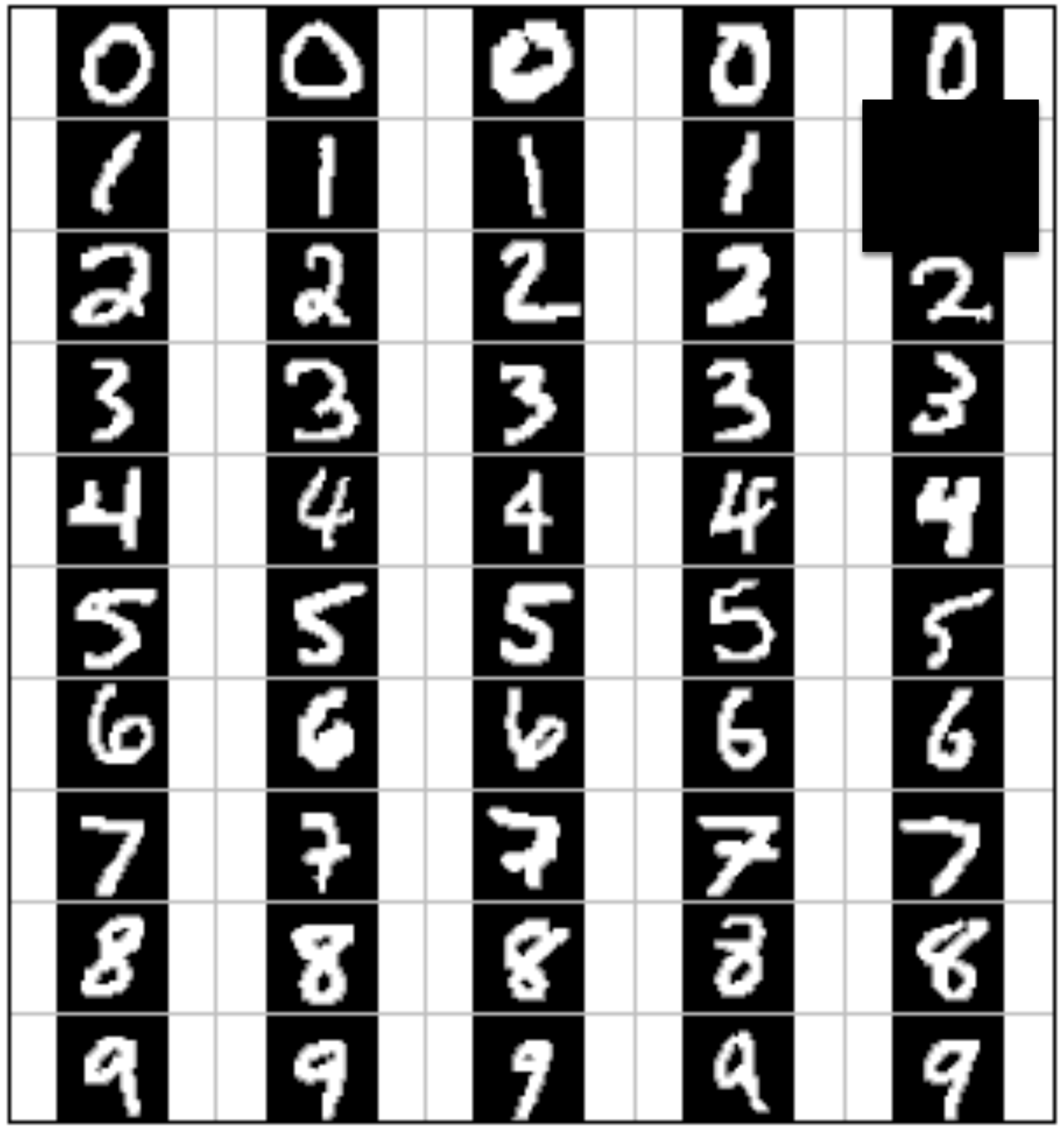
Leave-one-out Cross Validation



training data

validation data

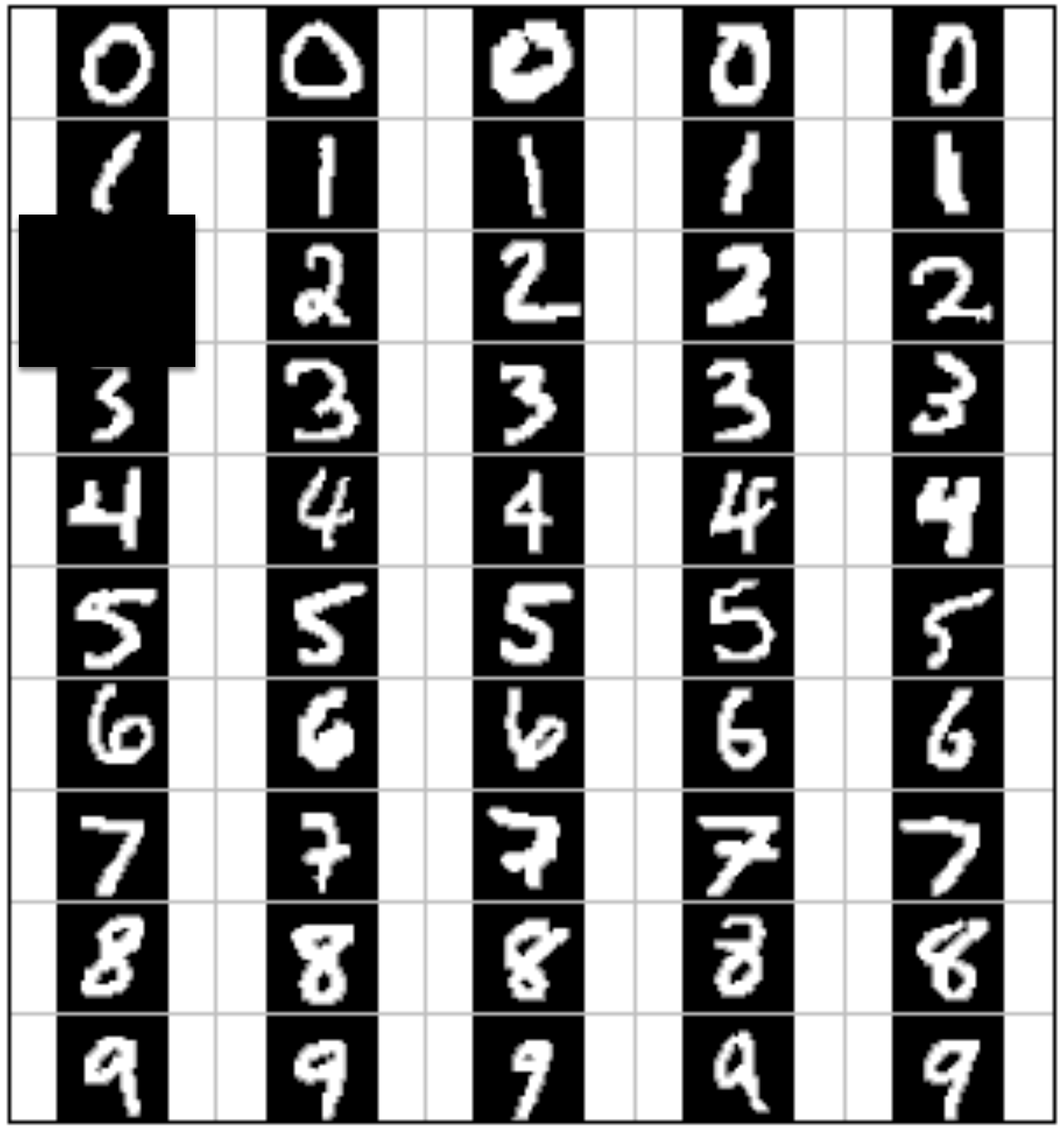
Leave-one-out Cross Validation



training data

validation data

Leave-one-out Cross Validation

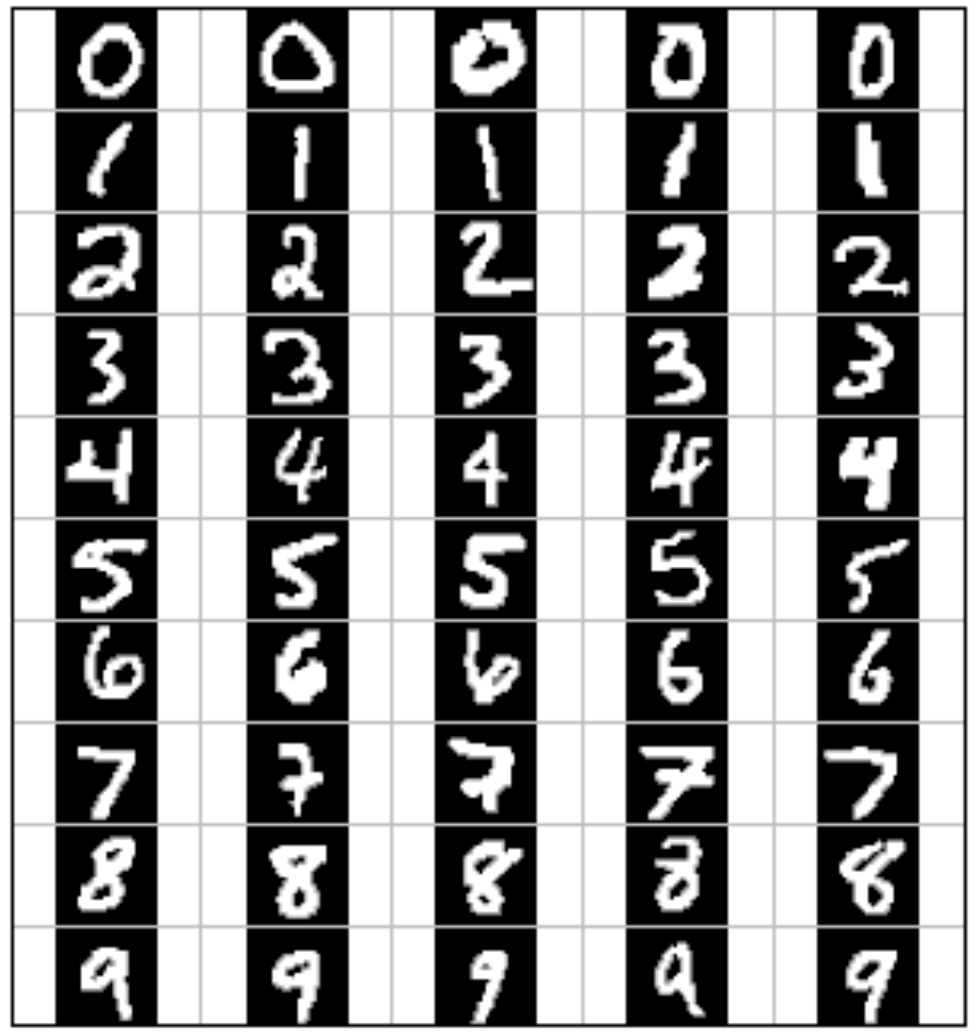


training data

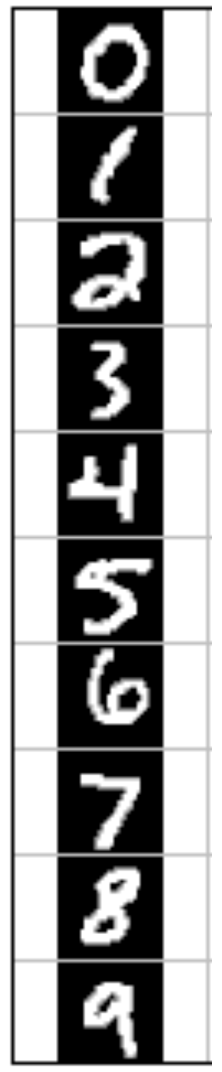
validation data

How Many Folds?

- What are the pros and cons of leave-one-out cross-validation?
- We usually train on N-1 folds and test on 1 fold. What are pros and cons of doing the inverse: train on 1 fold and test on N-1 folds?



Training



Testing

Testing versus Validation

- Best practice for experiments:
    - Hold out test set completely hidden from training
    - Use validation on training data for model (or parameter) selection
    - Evaluate on held-out test data

Model Selection via Validation

- Measure performance on **held-out** training data
  - Simulate testing environment
- Rotate **folds** of held-out subsets
- Can even hold out one at a time: **leave-one-out** validation
- Use (cross) validation performance to tune extra parameters

Summary

- Types of machine learning
- Complexity, overfitting, bias
- Validation, cross-validation