Lecture 09

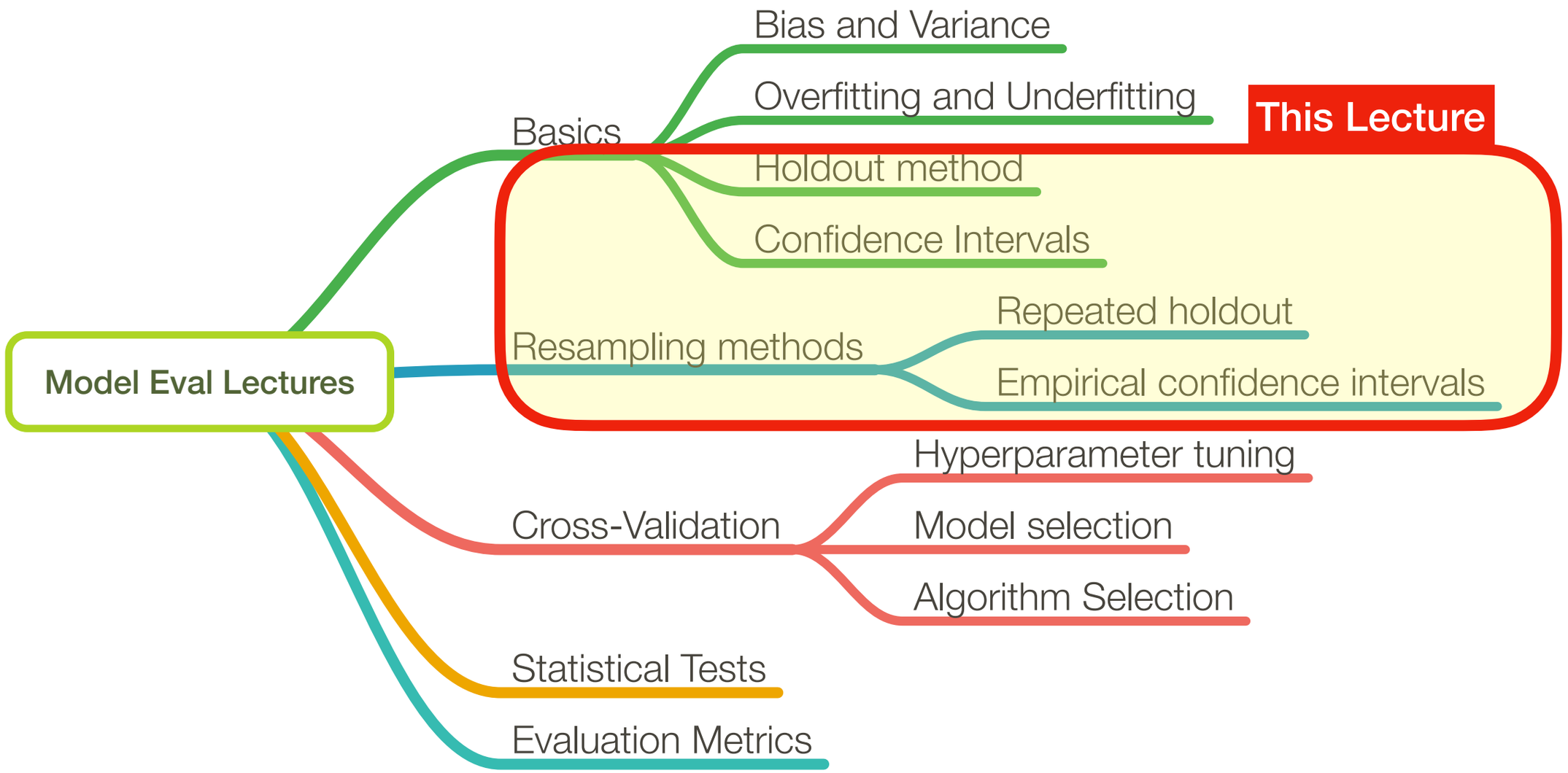
Model Evaluation 2:
Confidence Intervals

STAT 479: Machine Learning, Fall 2018

Sebastian Raschka

http://stat.wisc.edu/~sraschka/teaching/stat479-fs2018/

Overview


Model Eval Lectures
Basics
Bias and Variance
Overfitting and Underfitting
Holdout method
Confidence Intervals
This Lecture
Resampling methods
Repeated holdout
Empirical confidence intervals
Cross-Validation
Hyperparameter tuning
Model selection
Algorithm Selection
Statistical Tests
Evaluation Metrics

Main points why we evaluate the predictive performance of a model:

1. Want to estimate the generalization performance, the predictive performance of our model on future (unseen) data.

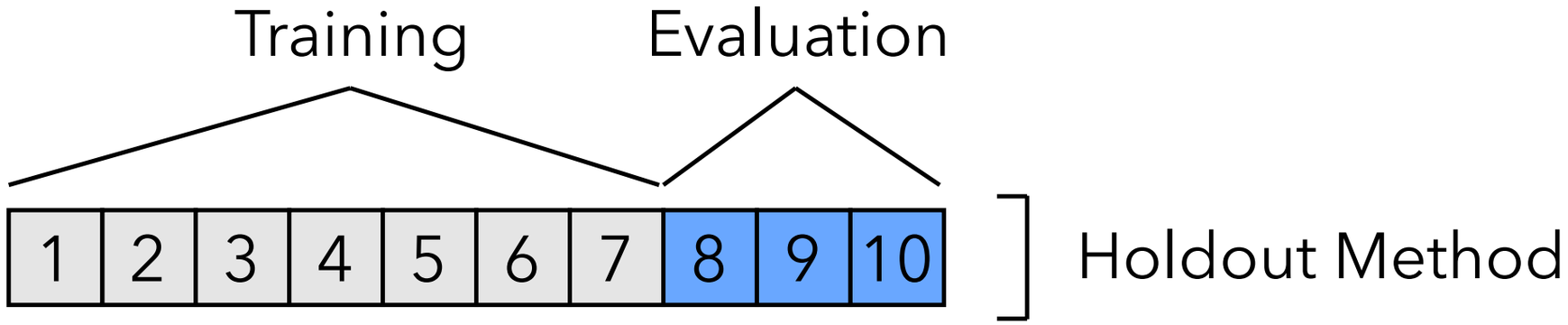
2. Want to increase the predictive performance by tweaking the learning algorithm and selecting the best performing model from a given hypothesis space.

3. Want to identify the ML algorithm that is best-suited for the problem at hand; thus, we want to compare different algorithms, selecting the best-performing one as well as the best performing model from the algorithm's hypothesis space.

- Training set error is an optimistically biased estimator of the generalization error

- Test set error is an unbiased estimator of the generalization error (test sample and hypothesis chosen independently)

- (in practice, it is actually pessimistically biased; why?)


Training
Evaluation
1
2
3
4
5
6
7
8
9
10
Holdout Method

Often using the holdout method is not a good idea ...

Often using the holdout method is not a good idea ...

Test set error as generalization error estimator is pessimistically biased (not so bad)

But it does not account for variance in the training data (bad)

Why is pessimistic bias not "so bad"?

Suppose we have the following ranking based on accuracy:

h_2: 75% > h_1: 70% > h_3: 65%,

we would still rank them the same way if we add a 10% pessimistic bias:

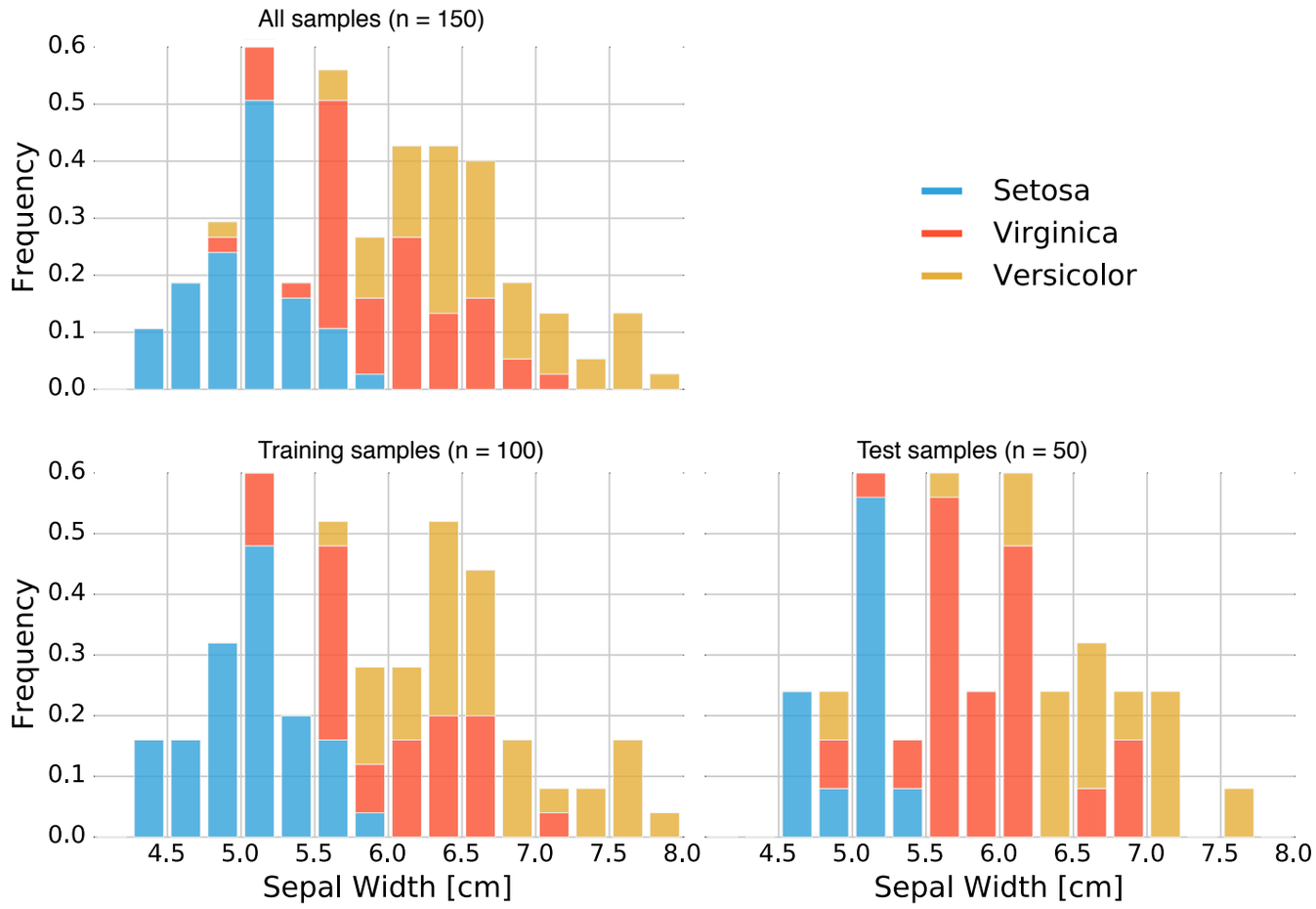
h_2: 65% > h_1: 60% > h_3: 55%.

Often using the holdout method is not a good idea ...

- Test set error as generalization error estimator is pessimistically biased (not so bad)

- Does not account for variance in the training data (bad)

Issues with Subsampling (Independence violation)



The Iris dataset consists of 50 Setosa, 50 Versicolor, and 50 Virginica flowers; the flower species are distributed uniformly:

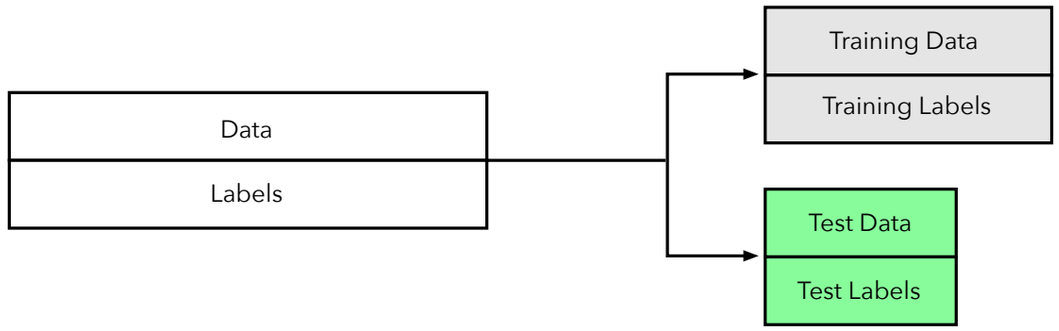
- 33.3% Setosa
- 33.3% Versicolor
- 33.3% Virginia

If our random function assigns 2/3 of the flowers (100) to the training set and 1/3 of the flowers (50) to the test set, it may yield the following:

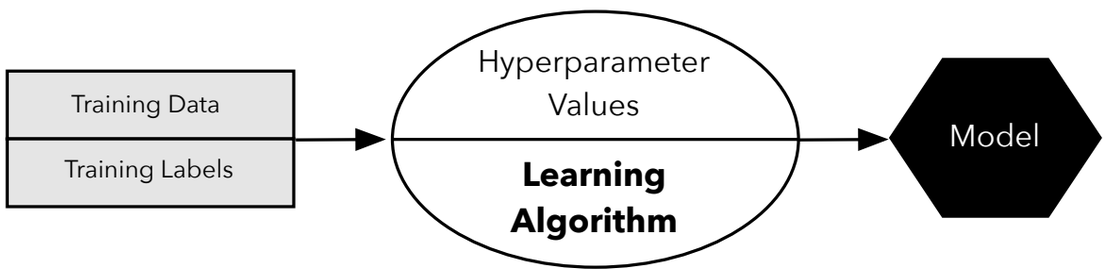
- training set → 38 x Setosa, 28 x Versicolor, 34 x Virginica
- test set → 12 x Setosa, 22 x Versicolor, 16 x Virginica

Holdout evaluation

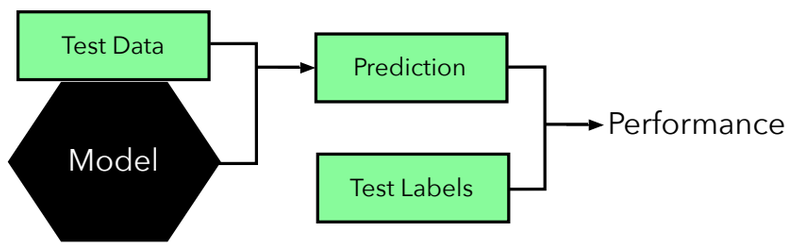
1



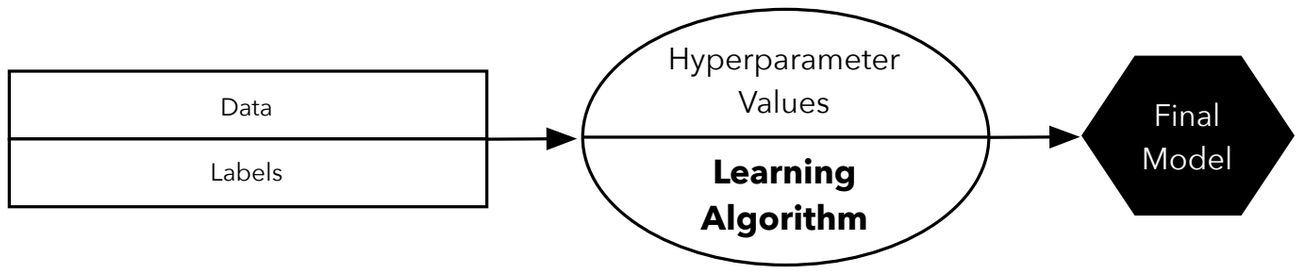
2



3

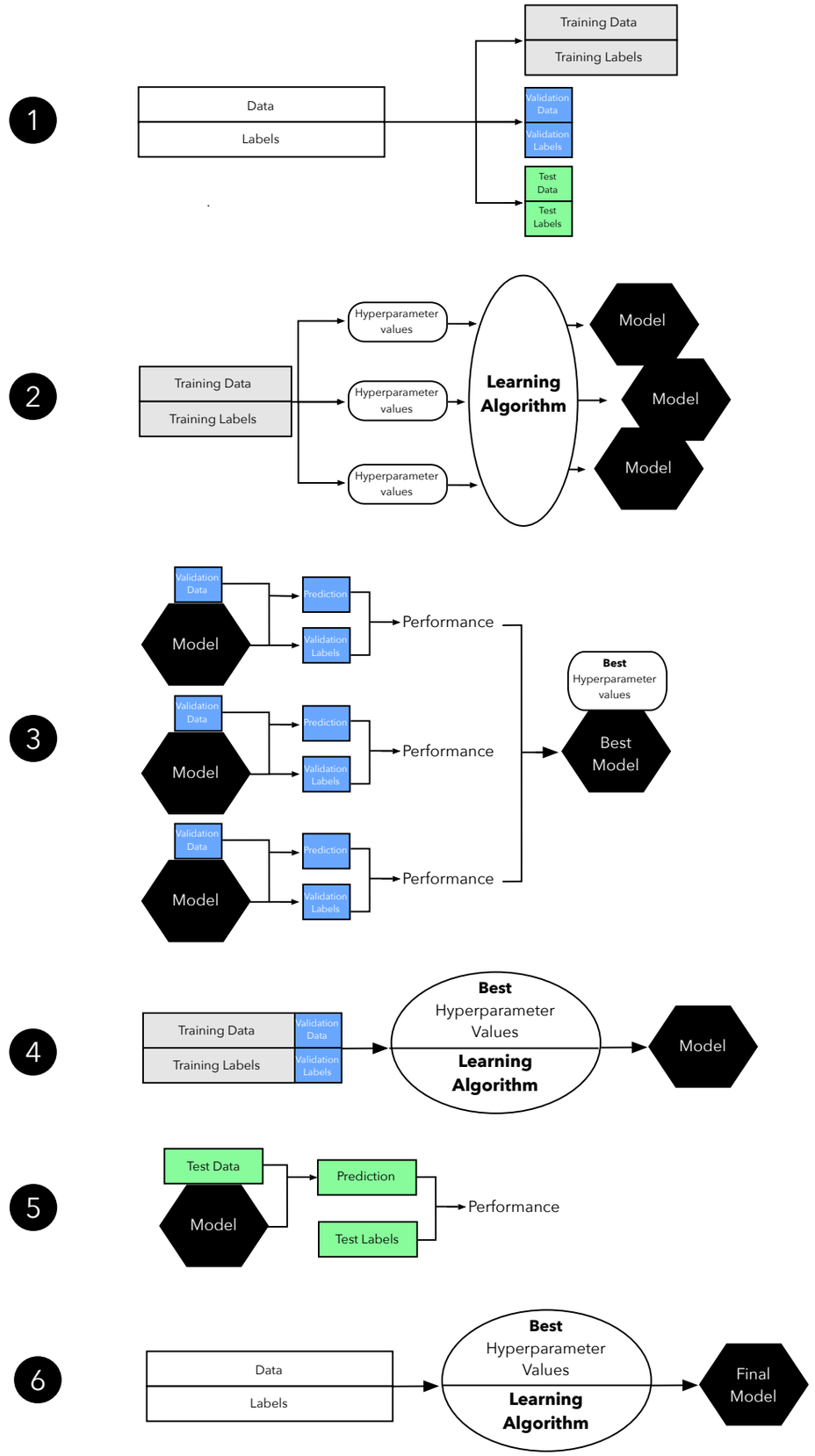


4



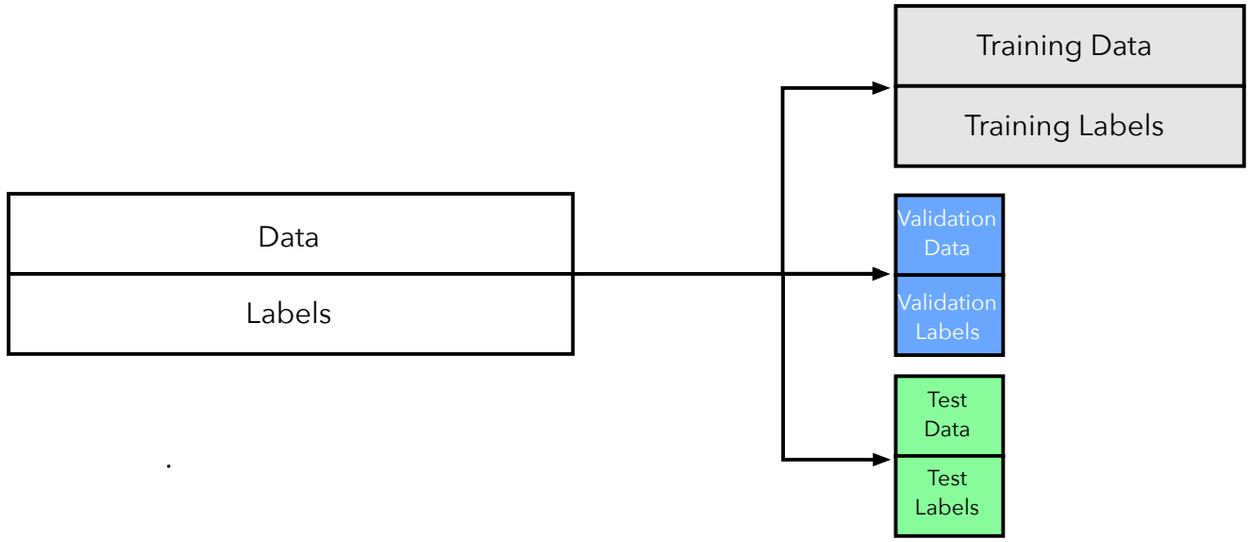
Can we use the holdout method for model selection?

Holdout validation (hyperparam. tuning)

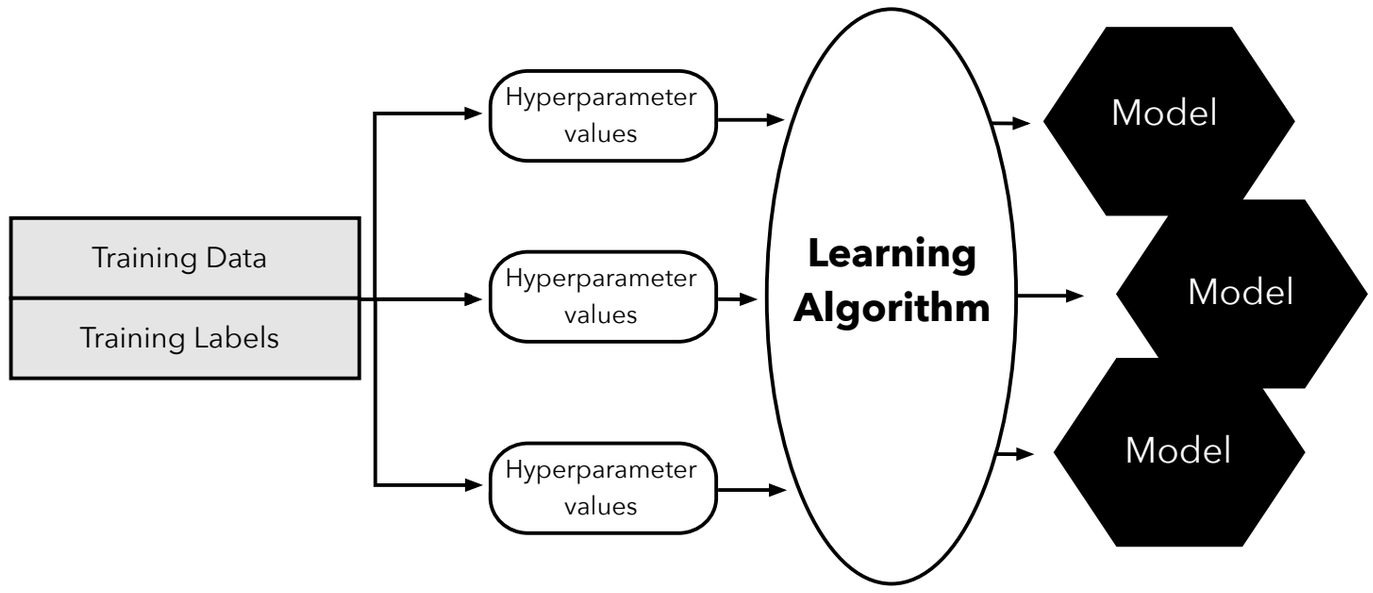


Holdout validation (hyperparam. tuning)

1

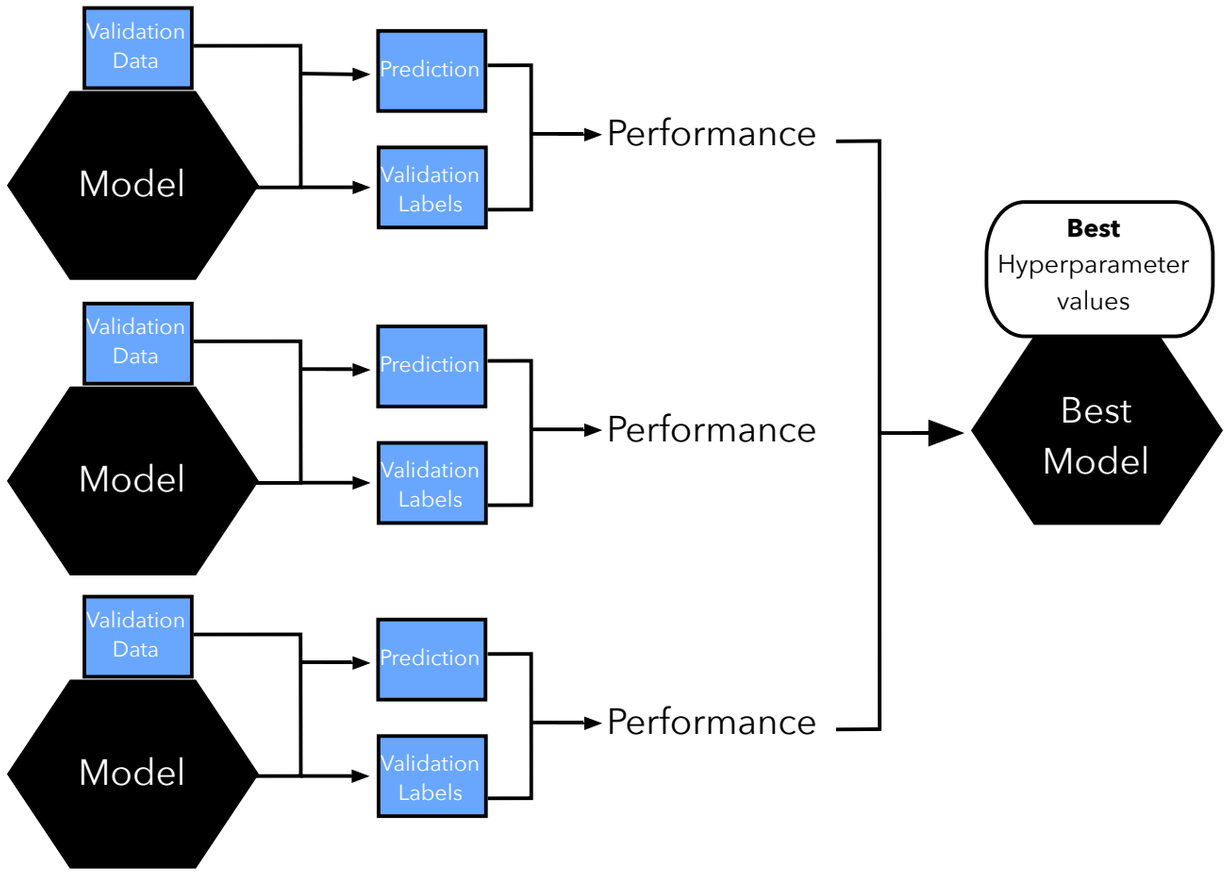


2

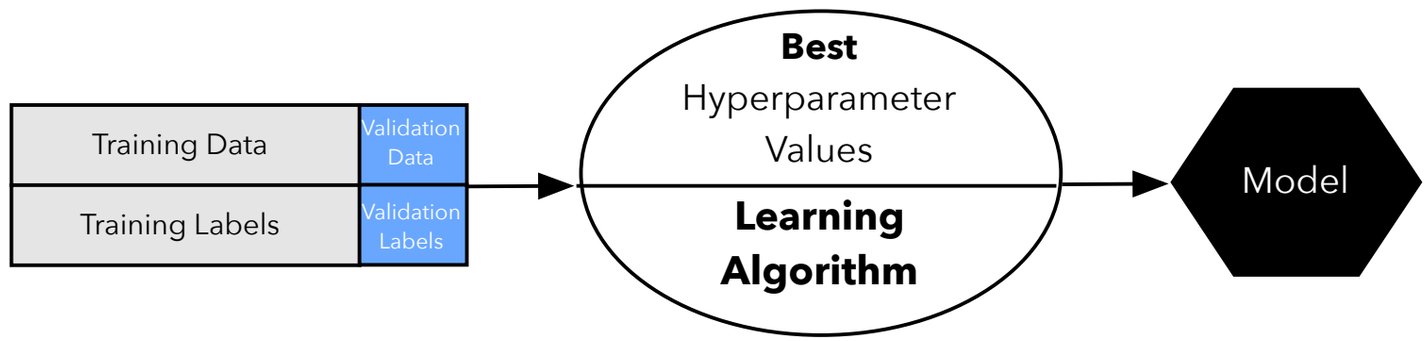


Holdout validation (hyperparam. tuning)

3

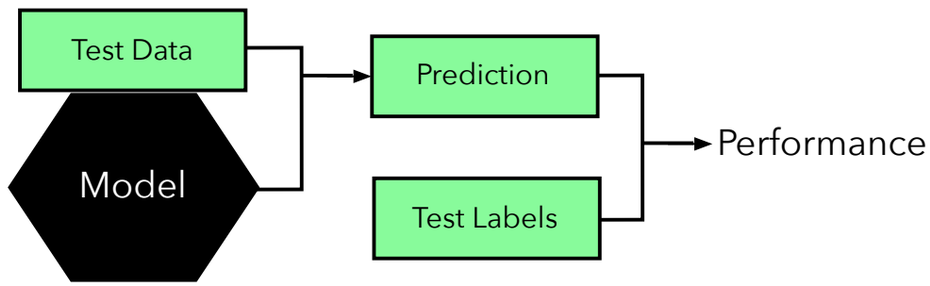


4

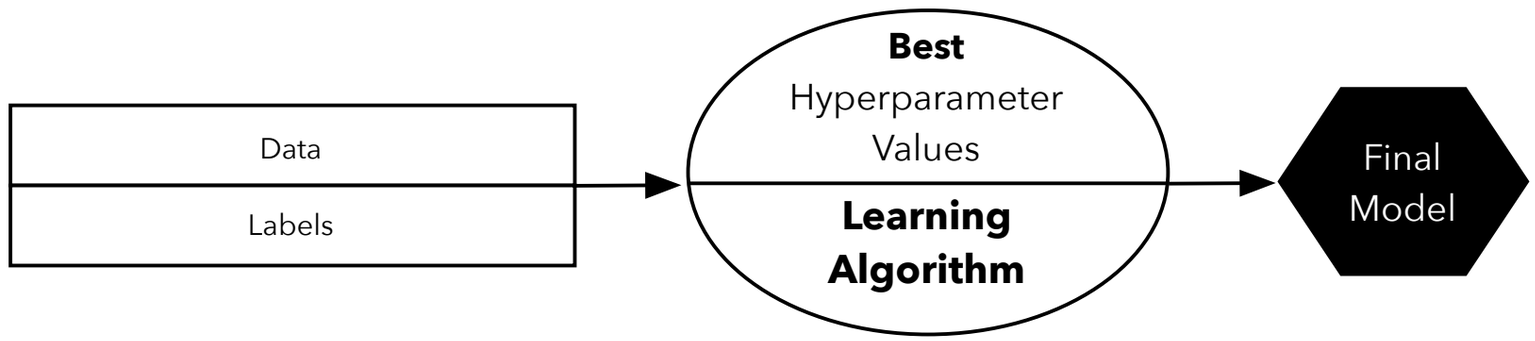


Holdout validation (hyperparam. tuning)

5



6



Cross-Validation is generally better

... but ...

Bengio, Y., & Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. *Journal of machine learning research*, *5*(Sep), 1089-1105.

Bias of Estimators Example

Normal Distribution: $\mathcal{N}(\mu, \sigma^2)$

Probability density function: $$f(x^{[i]}; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{(x^{[i]} - \mu)^2}{\sigma^2} \right)$$

Is the sample mean an unbiased estimator of the mean of the Gaussian?

$$\hat{\mu} = \frac{1}{n} \sum_i x^{[i]}$$

Bias of Estimators Example

Is the sample mean an unbiased estimator of the mean of the Gaussian?

$$\hat{\mu} = \frac{1}{n}\sum_i x^{[i]}$$

$$\begin{aligned} Bias(\hat{\mu}) &= E[\hat{\mu}] - \mu \\ &= E[\frac{1}{n}\sum_i x^{[i]}] - \mu \\ &= \frac{1}{n}\sum_i E[x^{[i]}] - \mu \\ &= \frac{1}{n}\sum_i \mu - \mu \\ &= \mu - \mu = 0 \end{aligned}$$

Bias of Estimators Example

Is the sample variance an unbiased estimator of the mean of the Gaussian

$$\hat{\sigma}^2 = \frac{1}{n}\sum_i (x^{[i]} - \hat{\mu})^2$$

$$\begin{aligned} Bias(\hat{\sigma}^2) &= E[\hat{\sigma}^2] - \sigma^2 \\ &= E\left[\frac{1}{n}\sum_i \left(x^{[i]} - \hat{\mu}\right)^2\right] - \sigma^2 \\ &= \ldots \\ &= \frac{m-1}{m}\sigma^2 - \sigma^2 \end{aligned}$$

Bias of Estimators Example

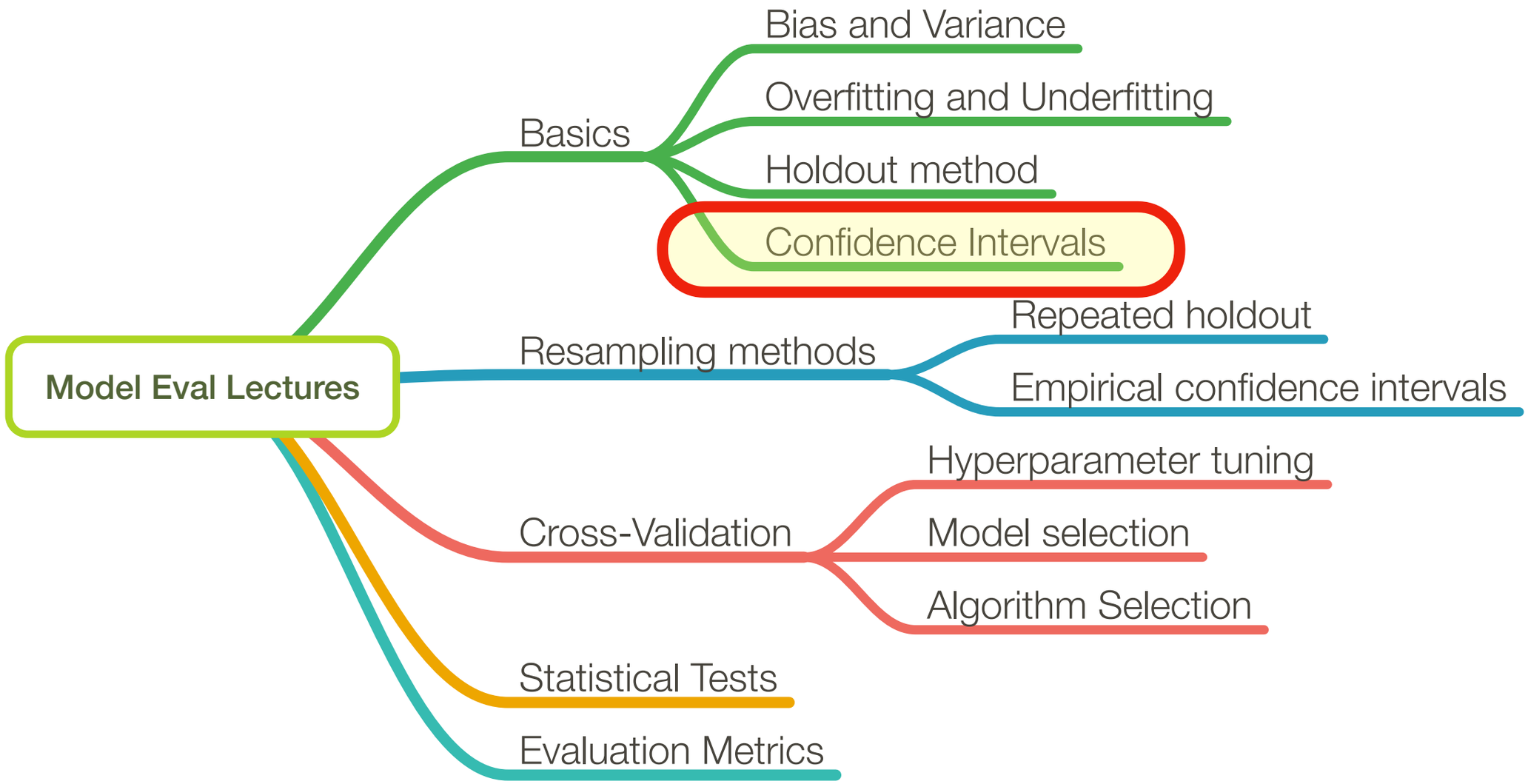
Is the sample variance an unbiased estimator of the mean of the Gaussian?

$$\hat{\sigma}^2 = \frac{1}{n}\sum_i (x^{[i]} - \hat{\mu})^2$$

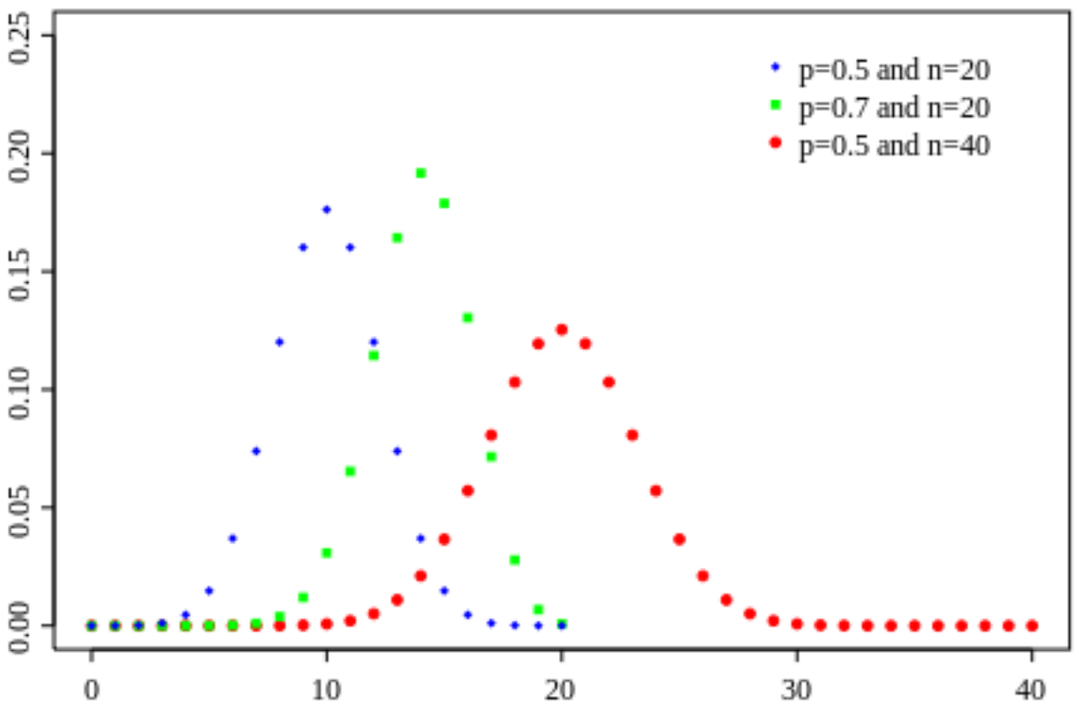
$$\begin{aligned} Bias(\hat{\sigma}^2) &= E[\hat{\sigma}^2] - \sigma^2 \\ &= E\left[\frac{1}{n}\sum_i \left(x^{[i]} - \hat{\mu}\right)^2\right] - \sigma^2 \\ &= \ldots \\ &= \frac{m-1}{m}\sigma^2 - \sigma^2 \end{aligned}$$

The unbiased estimator is actually

$$\hat{\sigma}'^2 = \frac{1}{n-1}\sum_i (x^{[i]} - \hat{\mu})^2$$


Model Eval Lectures
Basics
Bias and Variance
Overfitting and Underfitting
Holdout method
Confidence Intervals
Resampling methods
Repeated holdout
Empirical confidence intervals
Cross-Validation
Hyperparameter tuning
Model selection
Algorithm Selection
Statistical Tests
Evaluation Metrics

Binomial distribution



Notation	$B(n, p)$
Parameters	$n \in \mathbb{N}_0$ — number of trials $p \in [0,1]$ — success probability in each trial
Support	$k \in \{0, \ldots, n\}$ — number of successes
pmf	$\binom{n}{k} p^k (1-p)^{n-k}$
CDF	$I_{1-p}(n-k, 1+k)$
Mean	np
Median	$\lfloor np \rfloor$ or $\lceil np \rceil$
Mode	$\lfloor (n+1)p \rfloor$ or $\lceil (n+1)p \rceil - 1$
Variance	$np(1-p)$

(Image credit: Screenshot from https://en.wikipedia.org/wiki/Binomial_distribution_)

Binomial distribution $Pr(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}.$

Coin Flip (Bernoulli Trial)

- coin lands on head ("success")
- probability of success p
- $\frac{k}{n}$, estimator of p
- mean, number of successes

  $k = np$

0-1 Loss

- example misclassified (0-1 loss)
- true error $ERR_{\mathcal{D}}(h) = \underset{x \in \mathcal{D}}{Pr} [f(x) \neq h(x)]$
- sample (test set) error

  $ERR_S(h) = \frac{1}{n} \sum_{x \in S} \delta(f(x), h(x))$

Binomial distribution $Pr(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}.$

Coin Flip (Bernoulli Trial)

- mean, number of successes
  $\mu_k = np$

- variance $\sigma_k^2 = np(1-p)$

- standard deviation $\sigma_k = \sqrt{np(1-p)}$

0-1 Loss

We are interested in proportions!

$$ERR_S(h) = \frac{1}{n} \sum_{x \in S} \delta(f(x), h(x))$$

$$\sigma_{ERR_S(h)} = \frac{\sigma_k}{n} = \frac{\sqrt{np(1-p)}}{n} = \sqrt{\frac{p(1-p)}{n}}$$

$$\sigma_{ERR_S(h)} \approx \sqrt{\frac{ERR_S(h)(1-ERR_S(h))}{n}}$$

Confidence Intervals

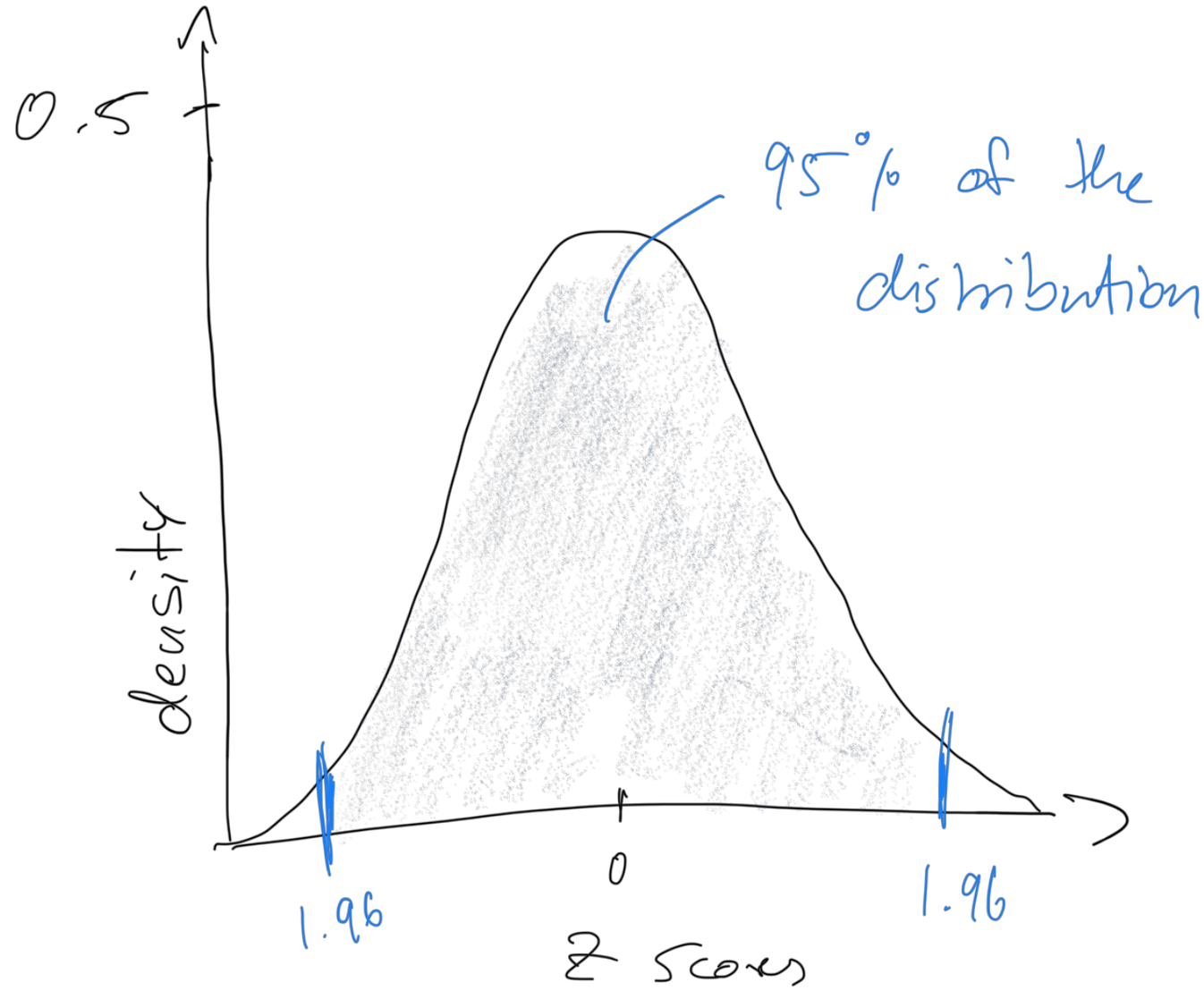
By definition, a XX% confidence interval of some parameter *p* is an interval that is expected to contain *p* with probability XX%

Normal Approximation

- Less tedious than confidence interval for Binomial distribution and hence often used in (ML) practice for large *n*

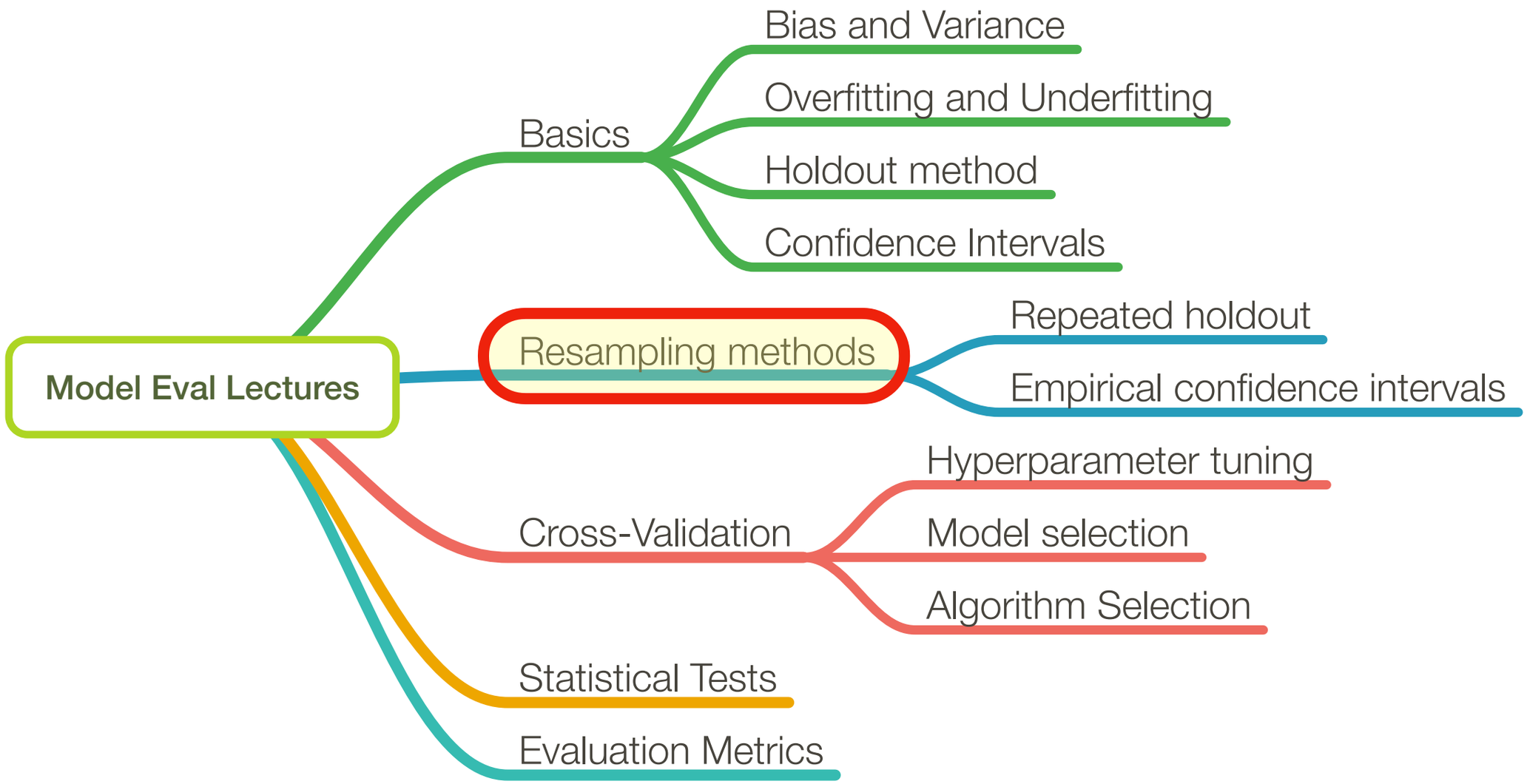
- Rule of thumb: if *n* larger than 40, the Binomial distribution can be reasonably approximated by a Normal distribution; and *np* and *n*(1 - *p*) should be greater than 0

$$CI = ERR_S(h) \pm z\sqrt{\frac{ERR_S(h)(1 - ERR_S(h))}{n}}$$

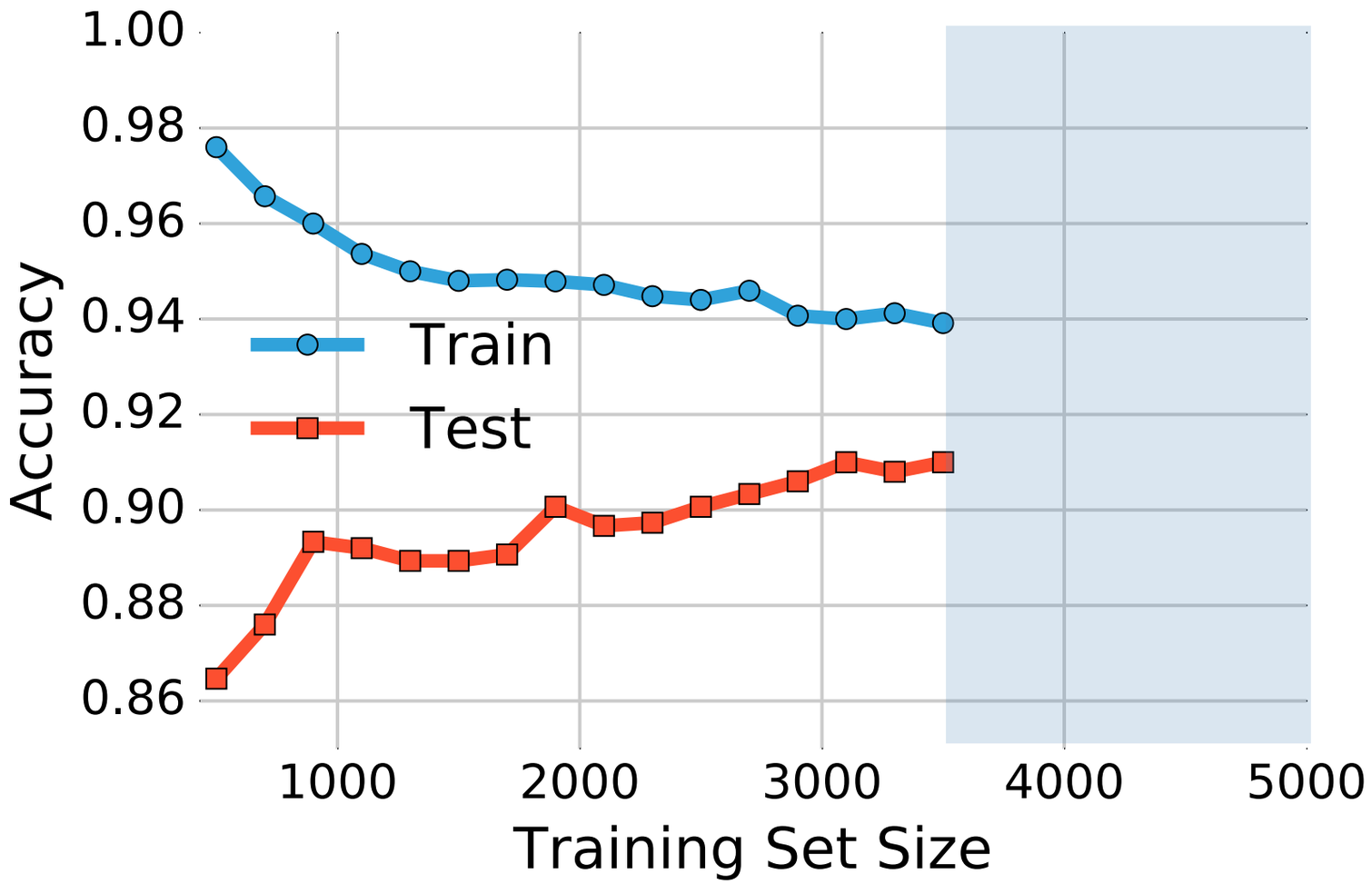


The z constant for different confidence intervals:

- 99%: z=2.58
- 95%: z=1.96
- 90%: z=1.64

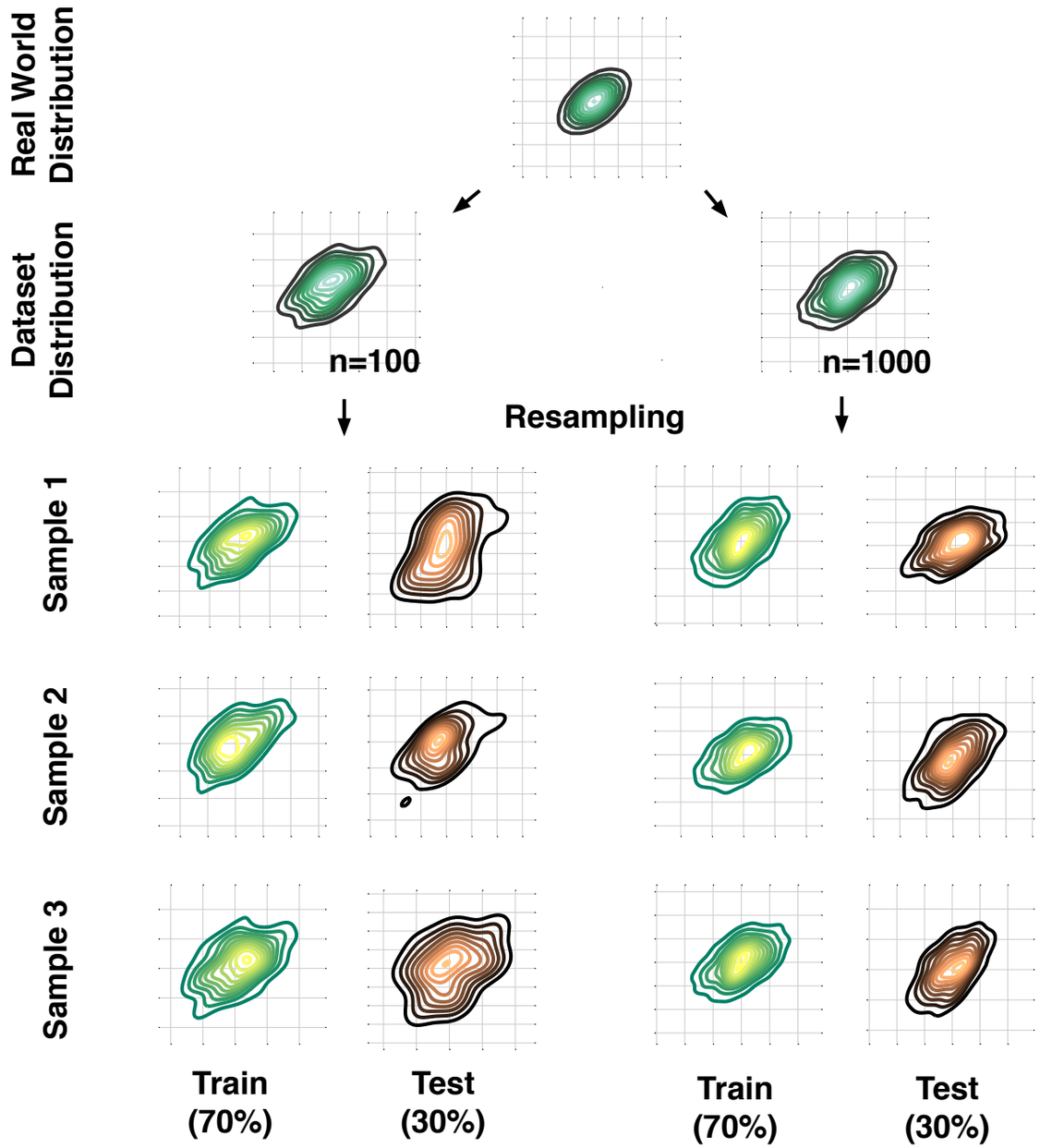

Model Eval Lectures
Basics
Bias and Variance
Overfitting and Underfitting
Holdout method
Confidence Intervals
Resampling methods
Repeated holdout
Empirical confidence intervals
Cross-Validation
Hyperparameter tuning
Model selection
Algorithm Selection
Statistical Tests
Evaluation Metrics

Proportionally large test sets increase the pessimistic bias if a model has not reached its full capacity, yet.



- To produce the plot above, I took 500 random samples of each of the ten classes from MNIST
- The sample was then randomly divided into a 3500-example training subset and a test set (1500 examples) via stratification.
- Even smaller subsets of the 3500-sample training set were produced via randomized, stratified splits, and I used these subsets to fit softmax classifiers and used the same 1500-sample test set to evaluate their performances; samples may overlap between these training subsets.

Decreasing the size of the test set brings up another problem:
It may result in a substantial variance increase of our model's performance estimate.



Here, I repeatedly subsampled a two-dimensional Gaussian

The reason is that it depends on which instances end up in training set, and which particular instances end up in test set. Keeping in mind that each time we resample our data, we alter the statistics of the distribution of the sample.

Repeated Holdout: Estimate Model Stability

(also called Monte Carlo Cross-Validation)

Average performance over *k* repetitions

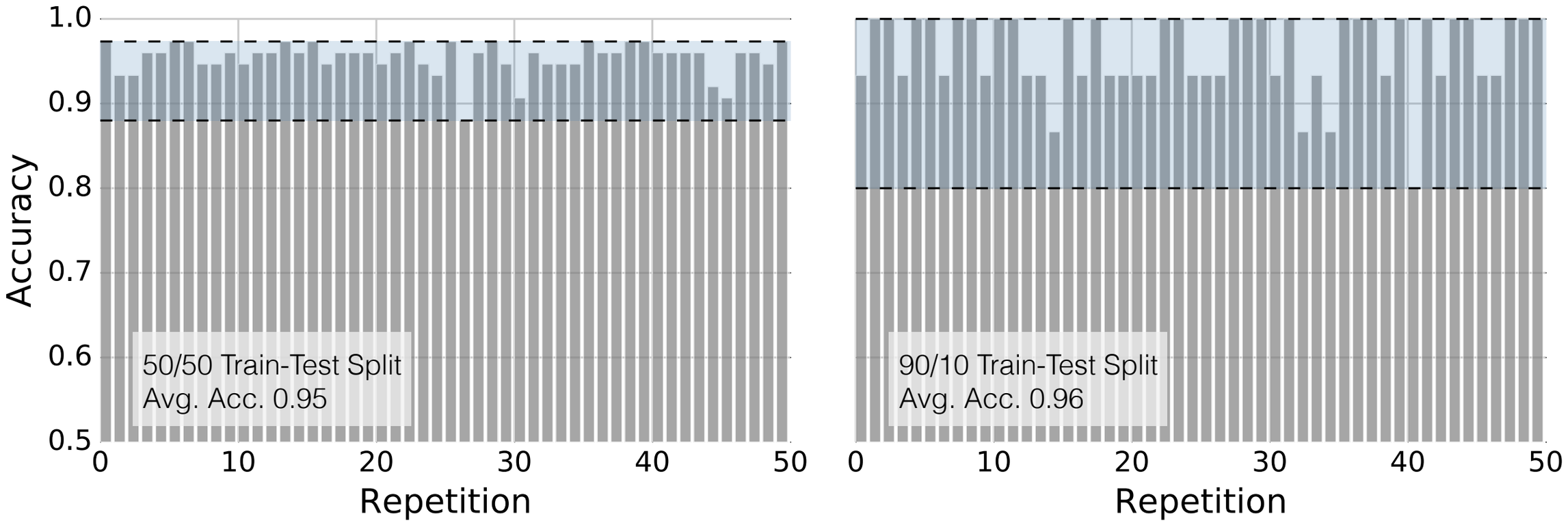
$$\textbf{ACC}_{avg} = \frac{1}{k} \sum_{j=1}^{k} ACC_j,$$

where ACC_j is the accuracy estimate of the *j*th test set of size *m*,

$$\textbf{ACC}_j = 1 - \frac{1}{n} \sum_{i=1}^{n} L\left(h(x^{[i]}), f(x^{[i]})\right) .$$

Repeated Holdout: Estimate Model Stability

How repeated holdout validation may look like for different training-test split using the Iris dataset to fit to 3-nearest neighbors classifiers:



Left: I performed 50 stratified training/test splits with 75 samples in the test and training set each; a K-nearest neighbors model was fit to the training set and evaluated on the test set in each repetition.

Right: Here, I repeatedly performed 90/10 splits, though, so that the test set consisted of only 15 samples.

The Bootstrap Method and Empirical Confidence Intervals

Circa 1900, to pull (oneself) up by (one's) bootstraps was used figuratively of an impossible task (Among the "practical questions" at the end of chapter one of Steele's "Popular Physics" schoolbook (1888) is, "30. Why can not a man lift himself by pulling up on his boot-straps?"). By 1916 its meaning expanded to include "better oneself by rigorous, unaided effort." The meaning "fixed sequence of instructions to load the operating system of a computer" (1953) is from the notion of the first-loaded program pulling itself, and the rest, up by the bootstrap.

(Source: Online Etymology Dictionary)

The Bootstrap Method and Empirical Confidence Intervals

- The bootstrap method is a resampling technique for estimating a sampling distribution
- Here, we are particularly interested in estimating the uncertainty of our performance estimate
- The bootstrap method was introduced by Bradley Efron in 1979 [1]
- About 15 years later, Bradley Efron and Robert Tibshirani even devoted a whole book to the bootstrap, “An Introduction to the Bootstrap” [2]
- In brief, the idea of the bootstrap method is to generate *new* data from a population by repeated sampling from the original dataset *with replacement* — in contrast, the repeated holdout method can be understood as sampling *without* replacement.

[1] Efron, Bradley. 1979. “Bootstrap Methods: Another Look at the Jackknife.” *The Annals of Statistics* 7 (1). Institute of Mathematical Statistics: 1–26. doi:10.1214/aos/1176344552.

[2] Efron, Bradley, and Robert Tibshirani. 1994. *An Introduction to the Bootstrap*. Chapman & Hall.

The Bootstrap Method and Empirical Confidence Intervals

1. We are given a dataset of size *n*.

2. For *b* bootstrap rounds:

    1. We draw one single instance from this dataset and assign it to our *j*th bootstrap sample. We repeat this step until our bootstrap sample has size *n* (the size of the original dataset). Each time, we draw samples from the same original dataset so that certain samples may appear more than once in our bootstrap sample and some not at all.

3. We fit a model to each of the *b* bootstrap samples and compute the resubstitution accuracy.

4. We compute the model accuracy as the average over the *b* accuracy estimates

$$\textbf{ACC}_{boot} = \frac{1}{b}\sum_{j=1}^{b}\frac{1}{n}\sum_{i=1}^{n}\left(1 - L\left(h(x^{[i]}), f(x^{[i]})\right)\right.\text{.}$$

The Bootstrap Method and Empirical Confidence Intervals

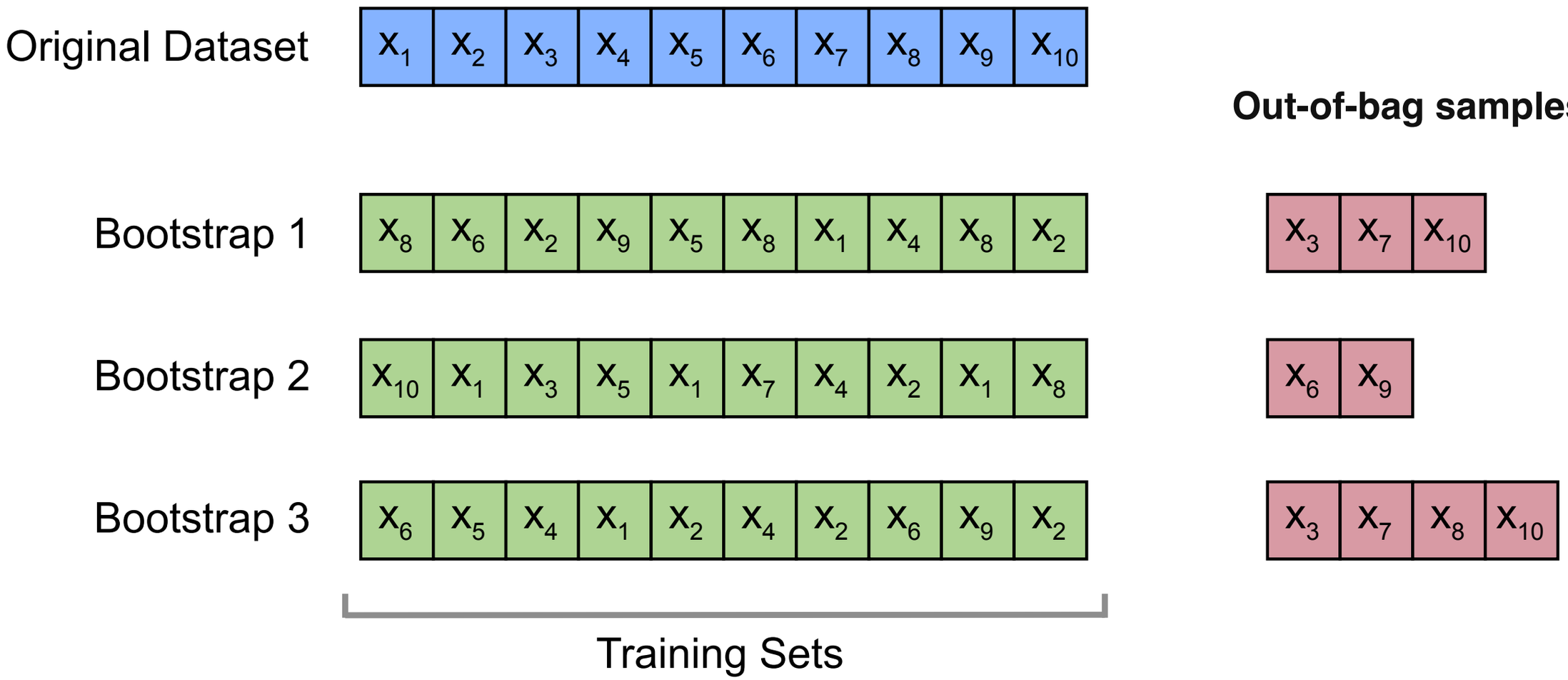
- As we discussed previously, the resubstitution accuracy usually leads to an extremely optimistic bias, since a model can be overly sensible to noise in a dataset.

- Originally, the bootstrap method aims to determine the statistical properties of an estimator when the underlying distribution was unknown and additional samples are not available.

- So, in order to exploit this method for the evaluation of predictive models, such as hypotheses for classification and regression, we may prefer a slightly different approach to bootstrapping using the so-called *Leave-One-Out Bootstrap* (LOOB) technique.

- Here, we use *out-of-bag* samples as test sets for evaluation instead of evaluating the model on the training data. Out-of-bag samples are the unique sets of instances that are not used for model fitting

Bootstrap Sampling



The Bootstrap Method and Empirical Confidence Intervals

We can compute the 95% confidence interval of the bootstrap estimate as

$$\mathbf{ACC}_{boot} = \frac{1}{b}\sum_{i=1}^{b}\mathbf{ACC}_i$$

and use it to compute the standard error

$$\mathbf{SE}_{boot} = \sqrt{\frac{1}{b-1}\sum_{i=1}^{b}(\mathbf{ACC}_i - \mathbf{ACC}_{boot})^2}\ .$$

Finally, we can then compute the confidence interval around the mean estimate as

$$\mathbf{ACC}_{boot} \pm t \times \mathbf{SE}_{boot}\,.$$

For instance, given a sample with n=100, we find that $t_{95} = 1.984$

(In practice, at least 200 bootstrap rounds are recommended)

The Bootstrap Method and Empirical Confidence Intervals

And if our samples do *not* follow a normal distribution? A more robust, yet computationally straight-forward approach is the **percentile method** as described by B. Efron (Efron, 1981). Here, we pick our lower and upper confidence bounds as follows:

$$\textbf{ACC}_{lower} = \alpha_1 th \text{ percentile of the } \textbf{ACC}_{boot} \text{ distribution}$$

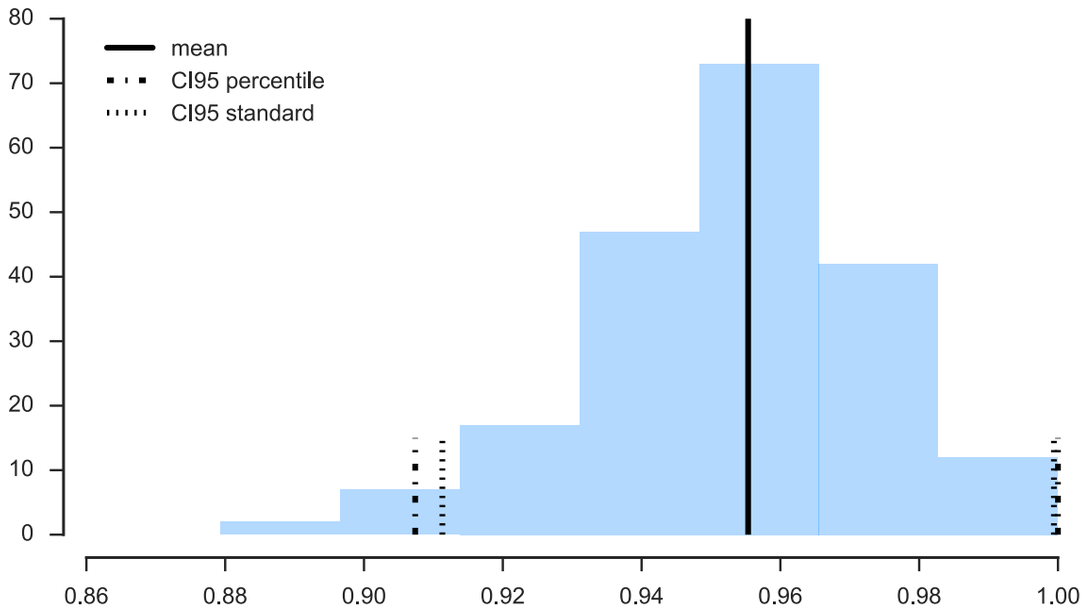
$$\textbf{ACC}_{upper} = \alpha_1 th \text{ percentile of the } \textbf{ACC}_{boot} \text{ distribution}$$

where $\alpha_1 = \alpha$ and $\alpha_2 = 1 - \alpha$ and α is our degree of confidence to compute the $100 \times (1 - 2 \times \alpha)$ confidence interval.

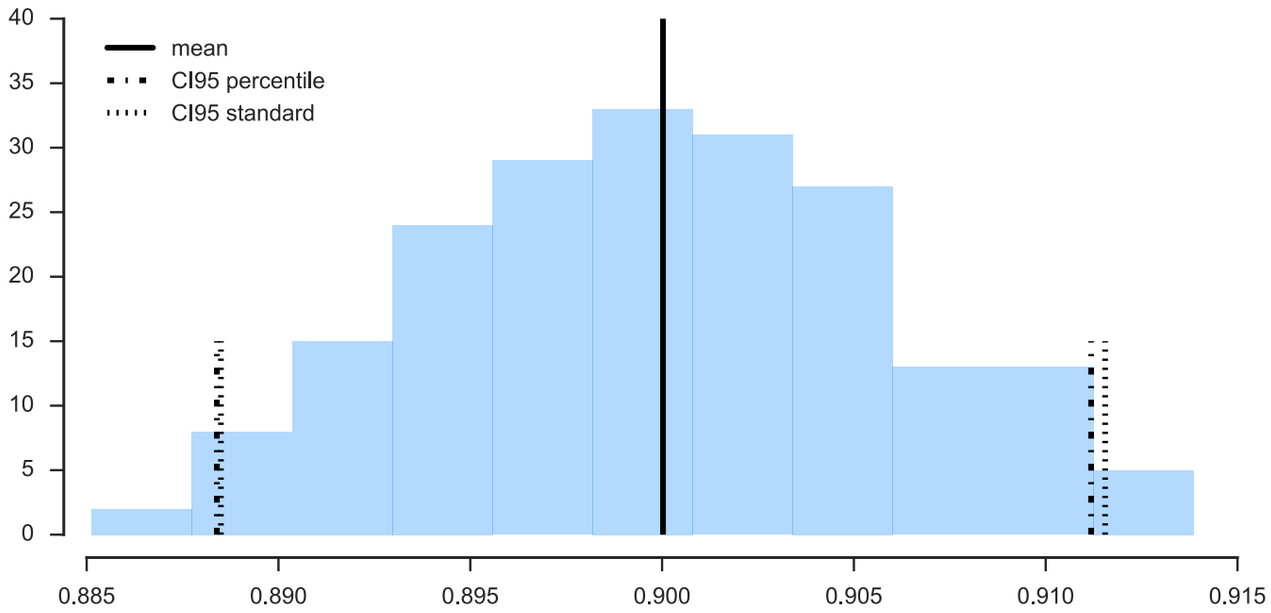
For instance, to compute a 95% confidence interval, we pick $\alpha = 0.025$ to obtain the 2.5th and 97.5th percentiles of the *b* bootstrap samples distribution as our upper and lower confidence bounds.

The Bootstrap Method and Empirical Confidence Intervals

A



B



In the left subplot, I applied the *Leave-One-Out Bootstrap* technique to evaluate 3-nearest neighbors models on Iris, and the right subplot shows the results of the same model evaluation approach on MNIST, using the same softmax algorithm as mentioned earlier.

The 0.632 Bootstrap Method

- In 1983, Bradley Efron described the *.632 Estimate*, a further improvement to address the pessimistic bias of the bootstrap [1].

- The pessimistic bias in the "classic" bootstrap method can be attributed to the fact that the bootstrap samples only contain approximately 63.2% of the unique samples from the original dataset.

[1] Efron, Bradley. 1983. "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation." *Journal of the American Statistical Association* 78 (382): 316. doi:10.2307/2288636.

Bootstrap Sampling

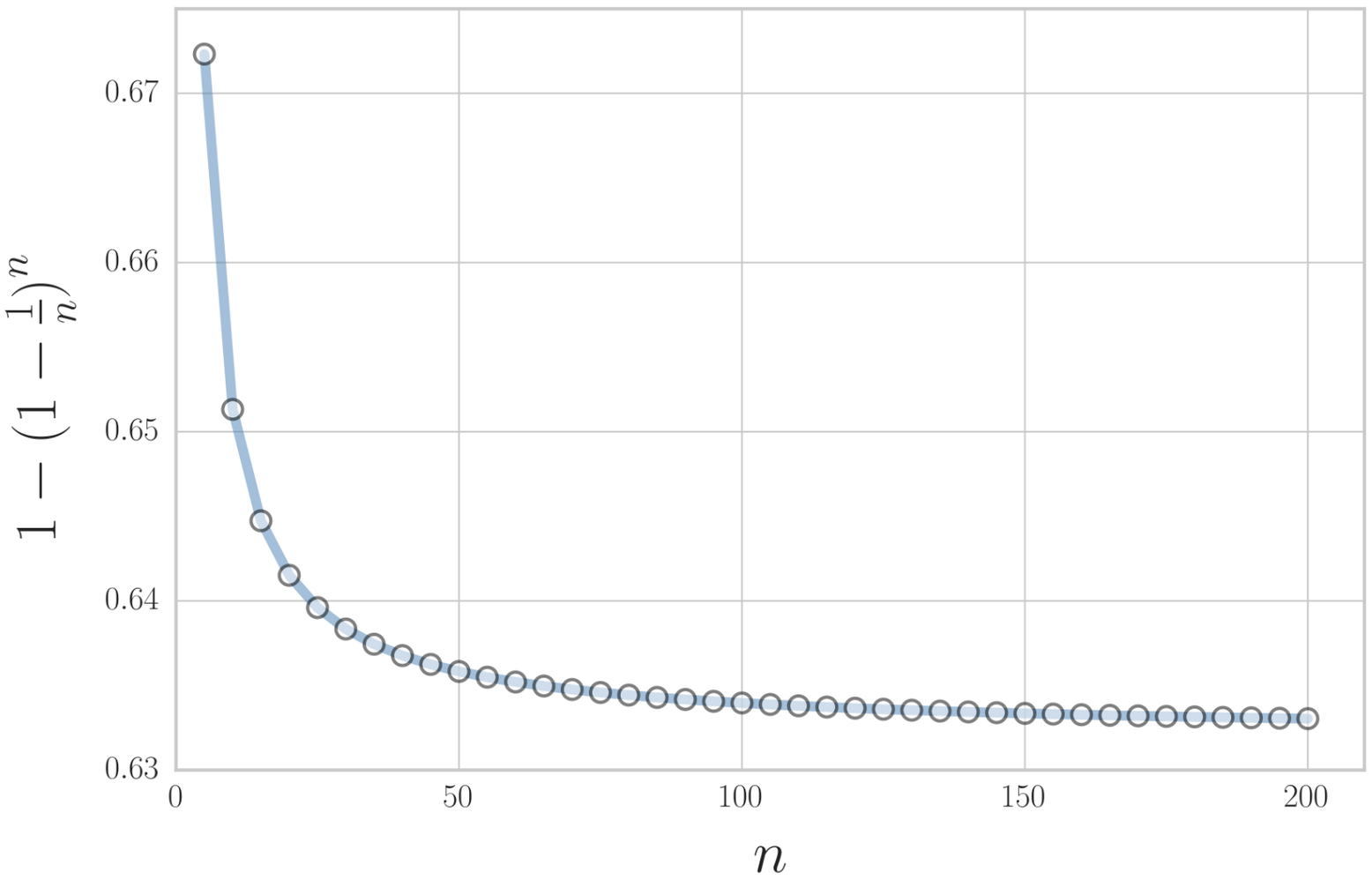
$$P(\textbf{not chosen}) = \left(1 - \frac{1}{n}\right)^n,$$

$$\frac{1}{e} \approx 0.368, \quad n \to \infty \, .$$

$$P(\textbf{not chosen}) = \left(1 - \frac{1}{n}\right)^n,$$

$$\frac{1}{e} \approx 0.368, \quad n \to \infty\,.$$

$$P(\textbf{chosen}) = 1 - \left(1 - \frac{1}{n}\right)^n \approx 0.632$$



The .632 Bootstrap Method

The *.632 Estimate,* is computed via the following equation:

$$\textbf{ACC}_{boot} = \frac{1}{b} \sum_{i=1}^{b} \left(0.632 \cdot \textbf{ACC}_{h,i} + 0.368 \cdot \textbf{ACC}_{r,i}\right),$$

where

$\textbf{ACC}_{r,i}$ is the resubstitution accuracy

$\textbf{ACC}_{h,i}$ is the accuracy on the out-of-bag sample.

[1] Efron, Bradley. 1983. “Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation.” *Journal of the American Statistical Association* 78 (382): 316. doi:10.2307/2288636.

The .632+ Bootstrap Method

Now, while the *.632 Boostrap* attempts to address the pessimistic bias of the estimate, an optimistic bias may occur with models that tend to overfit so that Bradley Efron and Robert Tibshirani proposed the *The .632+ Bootstrap Method* [1].

Instead of using a fixed "weight" $\omega = 0.632$ in

$$ACC_{\textbf{boot}} = \frac{1}{b} \sum_{i=1}^{b} \left(\omega \cdot \textbf{ACC}_{h,i} + (1 - \omega) \cdot \textbf{ACC}_{r,i} \right),$$

we compute the weight as $$\omega = \frac{0.632}{1 - 0.368 \times R},$$

where R is the *relative overfitting rate*

$$R = \frac{(-1) \times (\textbf{ACC}_{h,i} - \textbf{ACC}_{r,i})}{\gamma - (1 - \textbf{ACC}_{h,i})}.$$

[1] Efron, Bradley, and Robert Tibshirani. 1997. "Improvements on Cross-Validation: The .632+ Bootstrap Method." *Journal of the American Statistical Association* 92 (438): 548. doi:10.2307/2965703.

The .632+ Bootstrap Method

R is the *relative overfitting rate*

$$R = \frac{(-1) \times (\mathbf{ACC}_{h,i} - \mathbf{ACC}_{r,i})}{\gamma - (1 - \mathbf{ACC}_{h,i})} \, .$$

Now, we need to determine the *no-information rate* γ
in order to compute *R*.

For instance, we can compute γ
by fitting a model to a dataset that contains all possible combinations between the examples and target class labels:

$$\gamma = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \left(1 - L\left(h(x^{[i]}), f(x^{[i]})\right) \, .$$

[1] Efron, Bradley, and Robert Tibshirani. 1997. “Improvements on Cross-Validation: The .632+ Bootstrap Method.” *Journal of the American Statistical Association* 92 (438): 548. doi:10.2307/2965703.