Machine and Statistical Learning Fundamentals

Dorit Hammerling

Department of Applied Mathematics and Statistics
Colorado School of Mines(CSM)
Visiting Appointment: National Center for Atmospheric Research(NCAR)

Joint work with William Daniels (CSM)
. . . and many other students and collaborators

June 22, 2020

Outline

1. General framework, inference vs prediction
2. Forms for f, cross-validation and model selection
3. Practical application combining concepts

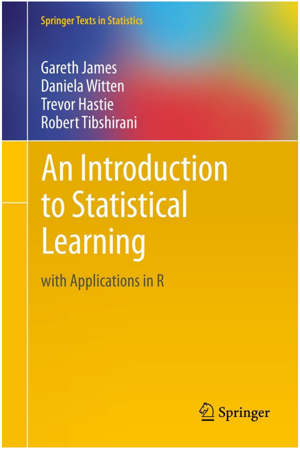
Outline

1. General framework, inference vs prediction

2. Forms for f, cross-validation and model selection

3. Practical application combining concepts

Useful reference books:



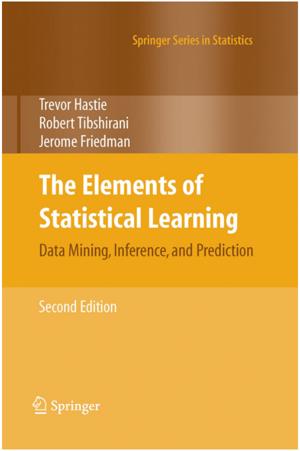
Springer Texts in Statistics

Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani

An Introduction to Statistical Learning

with Applications in R

Springer



Springer Series in Statistics

Trevor Hastie
Robert Tibshirani
Jerome Friedman

The Elements of Statistical Learning

Data Mining, Inference, and Prediction

Second Edition

Springer

- free and well-written
- worked-out code examples

Some definitions for starters

Statistical learning: large set of tools to gain insights from data

Supervised versus unsupervised:

- supervised: output and one or more inputs
  - classification
  - regression
  - ...
- unsupervised: only inputs, the structure of these inputs is of interest
  - clustering
  - association analysis
  - dimension reduction, e.g. principal components analysis
  - ...

⇒ *We will focus on the supervised setting.*

Some definitions for starters

Statistical learning: large set of tools to gain insights from data

Supervised versus unsupervised:

- supervised: output and one or more inputs
  - classification
  - regression
  - ...
- unsupervised: only inputs, the structure of these inputs is of interest
  - clustering
  - association analysis
  - dimension reduction, e.g. principal components analysis
  - ...

$\Rightarrow$ *We will focus on the supervised setting.*

Some definitions for starters

Statistical learning: large set of tools to gain insights from data

Supervised versus unsupervised:

- supervised: output and one or more inputs
    - classification
    - regression
    - ...
- unsupervised: only inputs, the structure of these inputs is of interest
    - clustering
    - association analysis
    - dimension reduction, e.g. principal components analysis
    - ...

⇒ *We will focus on the supervised setting.*

Some definitions for starters

Statistical learning: large set of tools to gain insights from data

Supervised versus unsupervised:

- supervised: output and one or more inputs
  - classification
  - regression
  - ...
- unsupervised: only inputs, the structure of these inputs is of interest
  - clustering
  - association analysis
  - dimension reduction, e.g. principal components analysis
  - ...

$\Rightarrow$ *We will focus on the supervised setting.*

Basic model formulation

The supervised model in its simplest form:

$$Y = f(X) + \epsilon$$

Model components:

Y: some variable we are interested in, output

f: some fixed but unknown function of X

X: variables $X_1, \ldots, X_p$ we believe might have a relationship to Y, inputs

ϵ : random error term

Main goal:

estimate f

Basic model formulation

The supervised model in its simplest form:

$$Y = f(X) + \epsilon$$

Model components:

Y: some variable we are interested in, output

f: some fixed but unknown function of X

X: variables $X_1, \ldots, X_p$ we believe might have a relationship to Y, inputs

ϵ : random error term

Main goal:

estimate f

Regression versus classification

The supervised model in its simplest form:

$$Y = f(X) + \epsilon$$

Supervised scenarios can be further categorized as *regression* versus *classification* problems:

- if the output Y is a quantitative variable ⇒ regression
- if the output Y is a qualitative (categorical) variable ⇒ classification

The categorization does not depend on the input variables, which can be either quantitative or qualitative. There is also a grey area, e.g. in the case of logistic or multinomial regression, where the outputs are categorical.

Regression versus classification

The supervised model in its simplest form:

$$Y = f(X) + \epsilon$$

Supervised scenarios can be further categorized as *regression* versus *classification* problems:

- if the output Y is a quantitative variable ⇒ regression
- if the output Y is a qualitative (categorical) variable ⇒ classification

The categorization does not depend on the input variables, which can be either quantitative or qualitative. There is also a grey area, e.g. in the case of logistic or multinomial regression, where the outputs are categorical.

Regression versus classification

The supervised model in its simplest form:

$$Y = f(X) + \epsilon$$

Supervised scenarios can be further categorized as *regression* versus *classification* problems:

- if the output Y is a quantitative variable ⇒ regression
- if the output Y is a qualitative (categorical) variable ⇒ classification

The categorization does not depend on the input variables, which can be either quantitative or qualitative. There is also a grey area, e.g. in the case of logistic or multinomial regression, where the outputs are categorical.

Why do want to estimate f?

Two main reasons:

- Prediction: if we get a new set X, what will Y be.
- Inference: what is the relationship between X and Y

$\Rightarrow$ *Our motivation influences the approaches we choose to model f!*

Trade-off between prediction accuracy and model interpretability:

A simpler, less flexible, model is generally easier to interpret, but might not be as accurate as a more flexible model.

Why do want to estimate f?

Two main reasons:

- Prediction: if we get a new set X, what will Y be.
- Inference: what is the relationship between X and Y

$\Rightarrow$ *Our motivation influences the approaches we choose to model f!*

Trade-off between prediction accuracy and model interpretability:

A simpler, less flexible, model is generally easier to interpret, but might not be as accurate as a more flexible model.

Why do want to estimate f?

Two main reasons:

- Prediction: if we get a new set X, what will Y be.
- Inference: what is the relationship between X and Y

$\Rightarrow$ *Our motivation influences the approaches we choose to model f!*

Trade-off between prediction accuracy and model interpretability:

A simpler, less flexible, model is generally easier to interpret, but might not be as accurate as a more flexible model.

Prediction

The prediction equation in its simplest form:

$$\hat{Y} = \hat{f}(X)$$

Meaning of terms:

$\hat{Y}$: prediction of Y

$\hat{f}$: estimate of f

X : input variables $X_1, \ldots, X_p$

If prediction is our only goal, than $\hat{f}$ can be treated as a black box, meaning we are not concerned with the exact form of $\hat{f}$ and how the Xs are related to the Ys. What we care about are *accurate* predictions.

Prediction

The prediction equation in its simplest form:

$$\hat{Y} = \hat{f}(X)$$

Meaning of terms:

$\hat{Y}$: prediction of Y
$\hat{f}$: estimate of f
X : input variables $X_1, \ldots, X_p$

If prediction is our only goal, than $\hat{f}$ can be treated as a black box, meaning we are not concerned with the exact form of $\hat{f}$ and how the Xs are related to the Ys. What we care about are *accurate* predictions.

Prediction Accuracy

How close is our estimated $\hat{Y}$ to the true Y?

Usually expressed as the squared difference between predicted and true value of Y, which depends on two error components.

Decomposition in reducible and irreducible error:

$$E(Y-\hat{Y})^2 = E[f(X)+\epsilon-\hat{f}(X)]^2$$
$$= \underbrace{[f(X)-\hat{f}(X)]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}$$

Prediction Accuracy

How close is our estimated $\hat{Y}$ to the true Y?

Usually expressed as the squared difference between predicted and true value of Y, which depends on two error components.

Decomposition in reducible and irreducible error:

$$
\begin{aligned}
E(Y-\hat{Y})^2 &= E[f(X)+\epsilon-\hat{f}(X)]^2 \\
&= \underbrace{[f(X)-\hat{f}(X)]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}
\end{aligned}
$$

Illustration of irreducible error

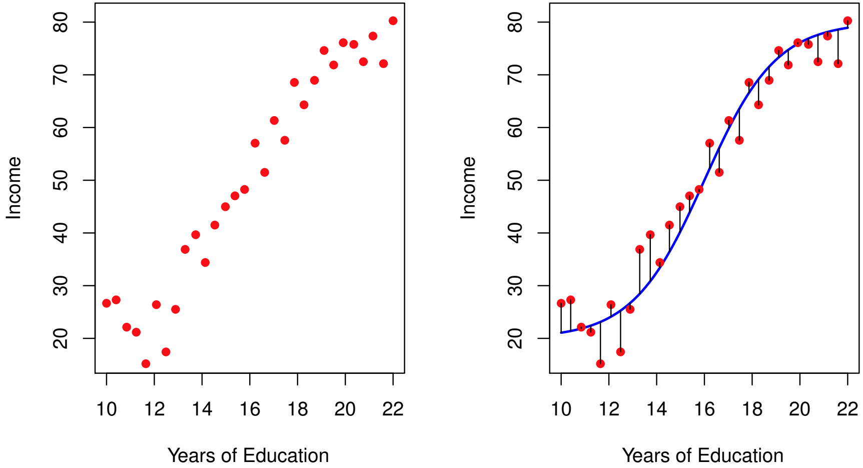


Figure credit: Introduction to Statistical Learning, Figure 2.2

Prediction Accuracy cont.

Decomposition in reducible and irreducible error:

$$
\begin{aligned}
E(Y - \hat{Y})^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\
&= \underbrace{[f(X) - \hat{f}(X)]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}
\end{aligned}
$$

Focus of statistical learning is on minimizing the *reducible* error. By definition, this can not be done for the *irreducible* error, which provides a bound for the prediction accuracy, which is unfortunately almost always unknown in practice.

Why is there irreducible error?

- variables that might be useful in predicting Y are not measured or part of X
- there is inherent variability in the system modeled

Prediction Accuracy cont.

Decomposition in reducible and irreducible error:

$$E(Y-\hat{Y})^2 = E[f(X)+\epsilon-\hat{f}(X)]^2$$
$$= \underbrace{[f(X)-\hat{f}(X)]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}$$

Focus of statistical learning is on minimizing the *reducible* error. By definition, this can not be done for the *irreducible* error, which provides a bound for the prediction accuracy, which is unfortunately almost always unknown in practice.

Why is there irreducible error?

- variables that might be useful in predicting Y are not measured or part of X
- there is inherent variability in the system modeled

Inference

We want to understand the relationship between X and Y, specifically how Y changes as a function of $X_1, \ldots, X_p$.

In this case we can NOT treat $\hat{f}$ as a black box, but are interested in its exact form.

Typical questions that arise in the inference context:

- Which of the predictors are related to the response? ⇒ Variable selection.
- What is the nature of the relationship between the predictors and the response? ⇒ Model selection.

There are scenarios where we are interested in both prediction and inference.

Inference

We want to understand the relationship between X and Y, specifically how Y changes as a function of $X_1, \ldots, X_p$.

In this case we can NOT treat $\hat{f}$ as a black box, but are interested in its exact form.

Typical questions that arise in the inference context:

- Which of the predictors are related to the response? $\Rightarrow$ Variable selection.
- What is the nature of the relationship between the predictors and the response? $\Rightarrow$ Model selection.

There are scenarios where we are interested in both prediction and inference.

Inference

We want to understand the relationship between X and Y, specifically how Y changes as a function of $X_1, \dots, X_p$.

In this case we can NOT treat $\hat{f}$ as a black box, but are interested in its exact form.

Typical questions that arise in the inference context:

- Which of the predictors are related to the response? $\Rightarrow$ Variable selection.
- What is the nature of the relationship between the predictors and the response? $\Rightarrow$ Model selection.

There are scenarios where we are interested in both prediction and inference.

Inference

We want to understand the relationship between X and Y, specifically how Y changes as a function of $X_1, \ldots, X_p$.

In this case we can NOT treat $\hat{f}$ as a black box, but are interested in its exact form.

Typical questions that arise in the inference context:

- Which of the predictors are related to the response? $\Rightarrow$ Variable selection.
- What is the nature of the relationship between the predictors and the response? $\Rightarrow$ Model selection.

There are scenarios where we are interested in both prediction and inference.

Question Break 1

Time for questions!

Outline

1 General framework, inference vs prediction

2 Forms for f, cross-validation and model selection

3 Practical application combining concepts

How do we estimate f?

Reminder of the model in its simplest form:

$$Y = f(X) + \epsilon$$

Goal in estimating $\hat{f}$:

Find a function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for all (X,Y).

While details depend on the specific methods, there are some common characteristics we can discuss. In doing so, it helps to classify a method as either *parametric*, i.e. assuming a functional form, or *non-parametric*.

How do we estimate f?

Reminder of the model in its simplest form:

$$Y = f(X) + \epsilon$$

Goal in estimating $\hat{f}$:

Find a function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for all (X,Y).

While details depend on the specific methods, there are some common characteristics we can discuss. In doing so, it helps to classify a method as either *parametric*, i.e. assuming a functional form, or *non-parametric*.

Parametric methods

Parametric modeling involves a two-step approach:

1. Assumption about the functional form. As simple example is a linear model:

   $$f(X) = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \ldots + \beta_p * X_p$$

   The problem of estimating f is now reduced to estimating the parameters $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$.

2. Actual estimation of the parameters using *training data* to *fit* or *train* the model. Depending on the functional form and number of parameters, this step can be numerically challenging. In simple cases, e.g. linear models, there are explicit solutions such as *least squares*.

Parametric methods

Parametric modeling involves a two-step approach:

1. Assumption about the functional form. As simple example is a linear model:

$$f(X) = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \ldots + \beta_p * X_p$$

   The problem of estimating f is now reduced to estimating the parameters $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$.

2. Actual estimation of the parameters using *training data* to *fit* or *train* the model. Depending on the functional form and number of parameters, this step can be numerically challenging. In simple cases, e.g. linear models, there are explicit solutions such as *least squares*.

Parametric methods

Parametric modeling involves a two-step approach:

1. Assumption about the functional form. As simple example is a linear model:

   $$f(X) = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \ldots + \beta_p * X_p$$

   The problem of estimating f is now reduced to estimating the parameters $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$.

2. Actual estimation of the parameters using *training data* to *fit* or *train* the model. Depending on the functional form and number of parameters, this step can be numerically challenging. In simple cases, e.g. linear models, there are explicit solutions such as *least squares*.

Non-parametric methods

Non-parametric methods do not make explicit assumptions about the functional form of f. Rather they target an estimate of f that is close to the data while conforming to smoothness constraints.

Highlights:

- The main advantage of non-parametric methods is that they do not impose a specific functional form, which might be far from the true f.
- Their main disadvantage is that the number of observations required is large, as they do not reduce the problem of estimating f to a small number of parameters. They are also not very informative in inferential settings.
- The parameter that balances the fit to the data with the smoothness constraint needs to be determined.

Non-parametric methods

Non-parametric methods do not make explicit assumptions about the functional form of f. Rather they target an estimate of f that is close to the data while conforming to smoothness constraints.

Highlights:

- The main advantage of non-parametric methods is that they do not impose a specific functional form, which might be far from the true f.
- Their main disadvantage is that the number of observations required is large, as they do not reduce the problem of estimating f to a small number of parameters. They are also not very informative in inferential settings.
- The parameter that balances the fit to the data with the smoothness constraint needs to be determined.

Non-parametric methods

Non-parametric methods do not make explicit assumptions about the functional form of f. Rather they target an estimate of f that is close to the data while conforming to smoothness constraints.

Highlights:

- The main advantage of non-parametric methods is that they do not impose a specific functional form, which might be far from the true f.
- Their main disadvantage is that the number of observations required is large, as they do not reduce the problem of estimating f to a small number of parameters. They are also not very informative in inferential settings.
- The parameter that balances the fit to the data with the smoothness constraint needs to be determined.

Illustration of a parametric (linear) model:

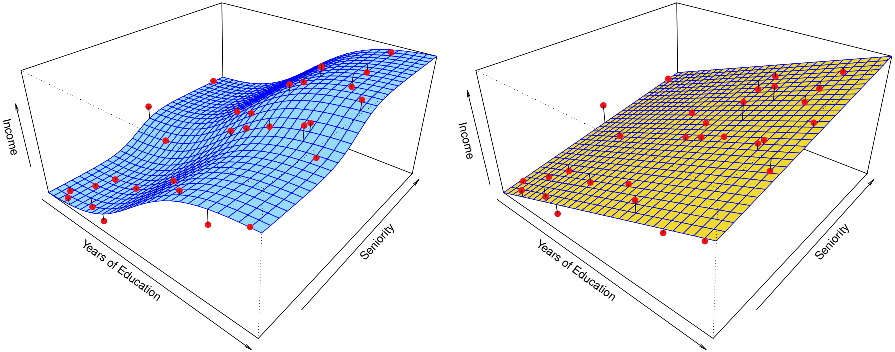


Figure credit: Introduction to Statistical Learning, Figures 2.3 and 2.4

Illustration of a non-parametric model (thin-plate spline):

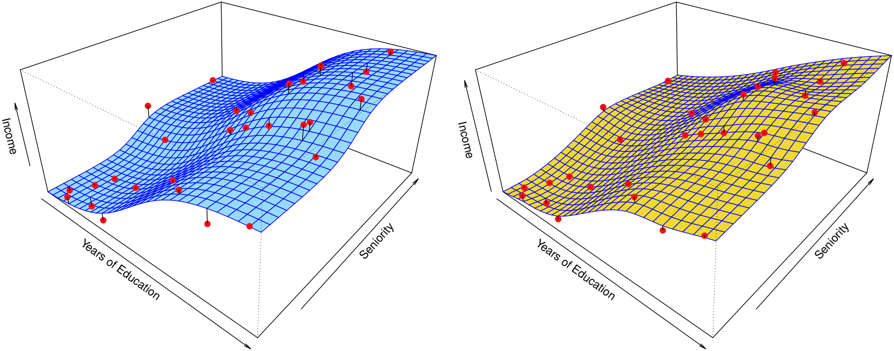


Figure credit: Introduction to Statistical Learning, Figures 2.3 and 2.5

Comparison of different smoothness assumptions:

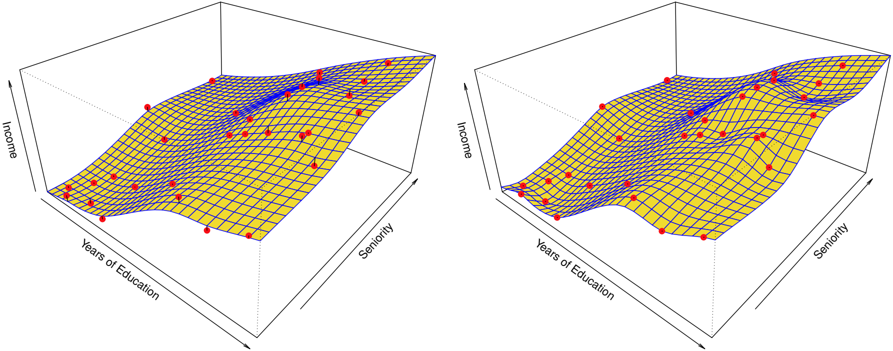


Figure credit: Introduction to Statistical Learning, Figures 2.5 and 2.6

How do we go about estimating tuning parameters?

Cross-validation is often the answers!

The main idea of cross-validation is to split the data into *training data* and *test data*. As the names imply, we use the *training data* to *train* our model and the *test data* to *evaluate* its performance on new data.

We don't want to use the training data exclusively to evaluate our model as such an approach would automatically favor more flexible models.

One can think of a reasonable fit to the training data as a necessary but not sufficient condition, while the performance on new data is the litmus test.

Cross-validation is very versatile and can be used in a wide variety of settings to evaluate models or find parameters!

How do we go about estimating tuning parameters?

Cross-validation is often the answers!

The main idea of cross-validation is to split the data into *training data* and *test data*. As the names imply, we use the *training data* to *train* our model and the *test data* to *evaluate* its performance on new data.

We don't want to use the training data exclusively to evaluate our model as such an approach would automatically favor more flexible models.

One can think of a reasonable fit to the training data as a necessary but not sufficient condition, while the performance on new data is the litmus test.

Cross-validation is very versatile and can be used in a wide variety of settings to evaluate models or find parameters!

How do we go about estimating tuning parameters?

Cross-validation is often the answers!

The main idea of cross-validation is to split the data into *training data* and *test data*. As the names imply, we use the *training data* to *train* our model and the *test data* to *evaluate* its performance on new data.

We don't want to use the training data exclusively to evaluate our model as such an approach would automatically favor more flexible models.

One can think of a reasonable fit to the training data as a necessary but not sufficient condition, while the performance on new data is the litmus test.

Cross-validation is very versatile and can be used in a wide variety of settings to evaluate models or find parameters!

How do we go about estimating tuning parameters?

Cross-validation is often the answers!

The main idea of cross-validation is to split the data into *training data* and *test data*. As the names imply, we use the *training data* to *train* our model and the *test data* to *evaluate* its performance on new data.

We don't want to use the training data exclusively to evaluate our model as such an approach would automatically favor more flexible models.

One can think of a reasonable fit to the training data as a necessary but not sufficient condition, while the performance on new data is the litmus test.

Cross-validation is very versatile and can be used in a wide variety of settings to evaluate models or find parameters!

How do we go about estimating tuning parameters?

Cross-validation is often the answers!

The main idea of cross-validation is to split the data into *training data* and *test data*. As the names imply, we use the *training data* to *train* our model and the *test data* to *evaluate* its performance on new data.

We don't want to use the training data exclusively to evaluate our model as such an approach would automatically favor more flexible models.

One can think of a reasonable fit to the training data as a necessary but not sufficient condition, while the performance on new data is the litmus test.

Cross-validation is very versatile and can be used in a wide variety of settings to evaluate models or find parameters!

Illustration of training versus test error

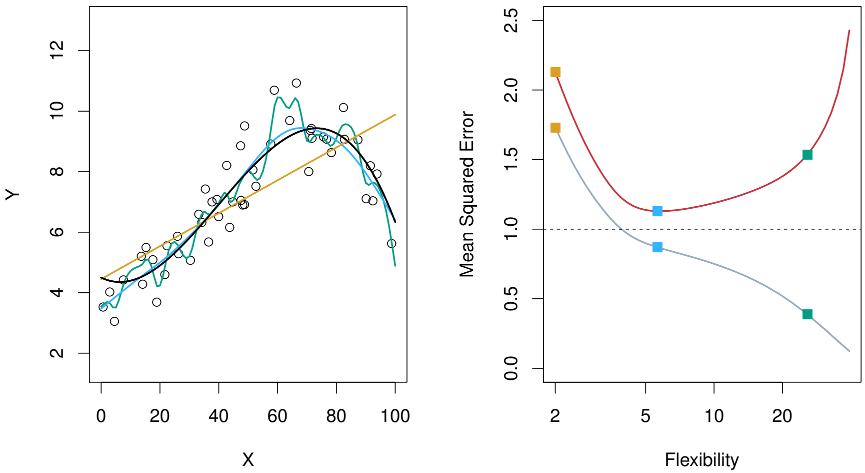


Figure credit: Introduction to Statistical Learning, Figure 2.9

The main flavors of cross-validation:

- Leave-one-out cross-validation (LOOCV): A single observation comprises the validation set and the remainder of the data is used for training.

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

Can be computationally expensive if model fitting is expensive.

- k-fold cross-validation: data is randomly divided into k-groups, or folds. The first group is treated as validation data, the remaining groups as training data. This is repeated k times switching the validation sets.

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

LOOCV is a special case of k-fold cross-validation with $k = n$.

The main flavors of cross-validation:

- Leave-one-out cross-validation (LOOCV): A single observation comprises the validation set and the remainder of the data is used for training.

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

Can be computationally expensive if model fitting is expensive.

- k-fold cross-validation: data is randomly divided into k-groups, or folds. The first group is treated as validation data, the remaining groups as training data. This is repeated k times switching the validation sets.

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

LOOCV is a special case of k-fold cross-validation with $k = n$.

Illustration of cross-validation

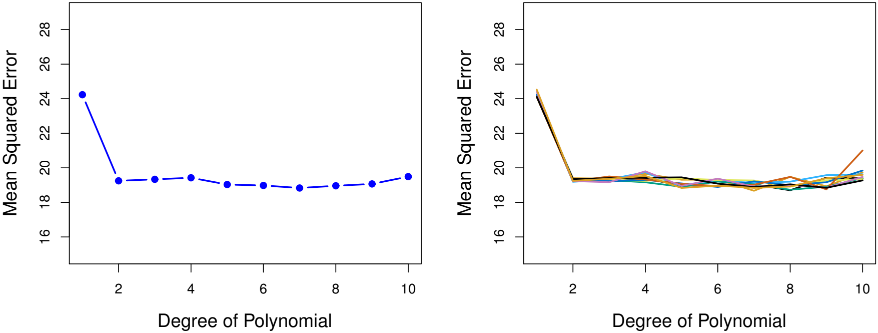


Figure credit: Introduction to Statistical Learning, Figure 5.4

Question Break 2

Time for questions!

Outline

1 General framework, inference vs prediction

2 Forms for f, cross-validation and model selection

3 Practical application combining concepts

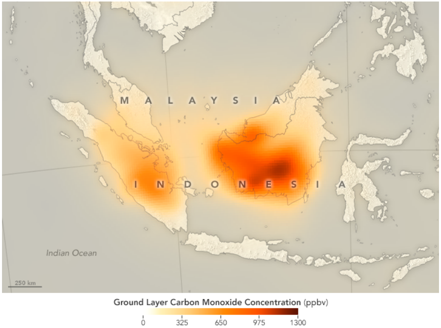
Motivation for study

Big picture: We are using natural variability in the climate to model atmospheric carbon monoxide (CO) concentrations.

Motivation: Why bother modeling CO?

1. Fires are the primary source of CO in the Southern Hemisphere.
2. CO can be used as a proxy for fires.
3. Predictive CO models can help countries prepare for large burn events.

2015 Indonesia Fires



Response Variable

- CO measurements from MOPITT instrument on board the Terra satellite.
- CO is aggregated into seven biomass burning regions.
- A separate model is created for each region.

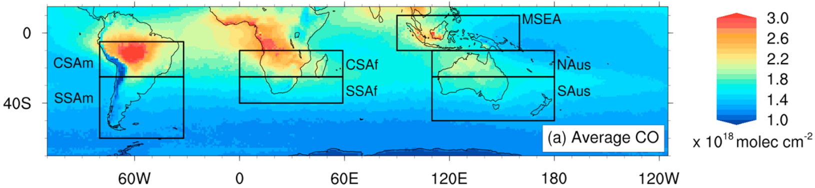
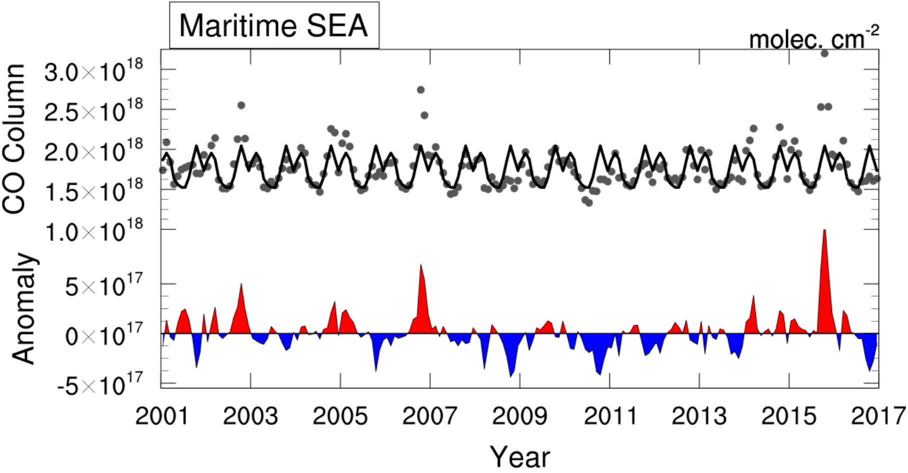


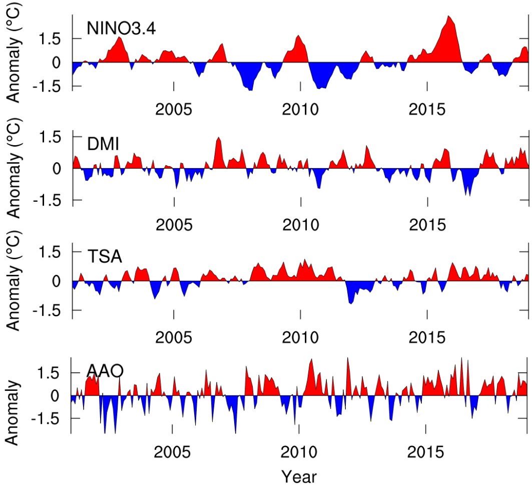
Figure: Regions of interest plotted over average total column CO.

Response Variable

Response variable: De-seasonalized CO anomaly at a given time, t.



Predictor Variables



- Burn events are related to climate through availability and dryness of fuel.
- Climate indices are metrics that summarize aperiodic changes in climate.

Predictor variables: Climate indices, lagged at time t-τ.

Statistical Model

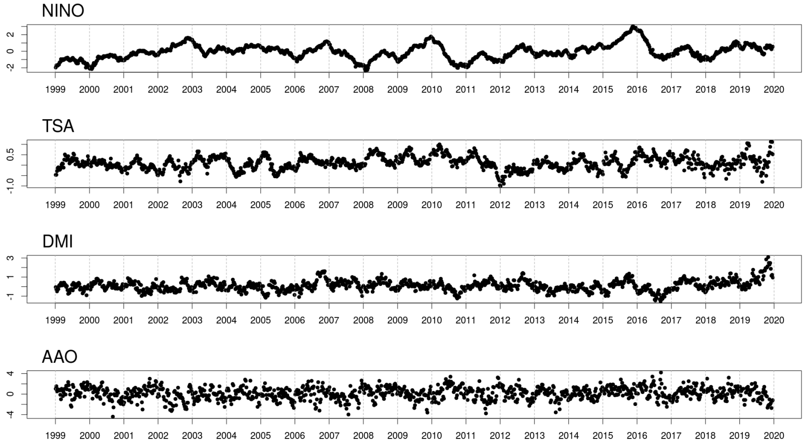
We use a lagged multiple linear regression model with first order interaction terms to explain the relationship between atmospheric CO and the climate indices.

$$CO(t) = \mu + \sum_{k} a_k \cdot \chi_k(t - \tau_k) + \sum_{i,j} b_{ij} \cdot \chi_i(t - \tau_i) \cdot \chi_j(t - \tau_j)$$

- $CO(t)$ is the CO anomaly in a given response region at time t
- μ is a constant mean displacement
- a_k and b_{ij} are coefficients
- χ are the climate indices
- τ is the lag value for each index

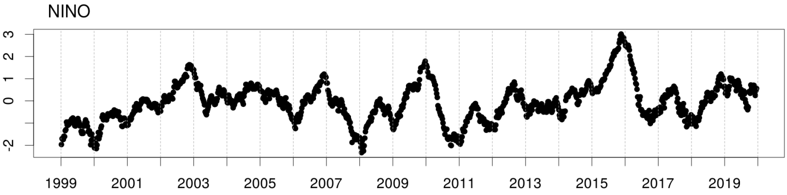
Variability in Climate Indices

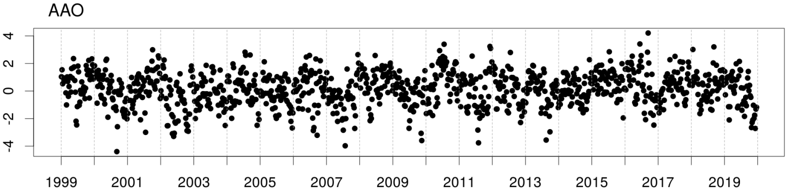
- Some climate indices are smoother than others.



Variability in Climate Indices

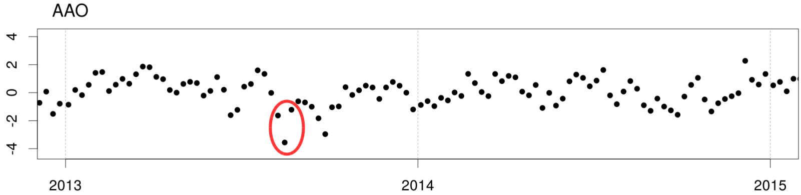
- Difference between NINO and AAO is very apparent.





Variability in Climate Indices

- Lots of variability makes choice of lag values important.
- Potentially large differences from one week to the next.



- Circled points are close in time but have very different values.

Are features like this noise or signal?

Smoothing Climate Indices

Smoothing climate indices can protect against these noisy jumps.

Smoothing kernel:

- Move an averaging “window” across the data.
- Apply weights to the average so that the current data point has the most influence.

Gaussian kernel:

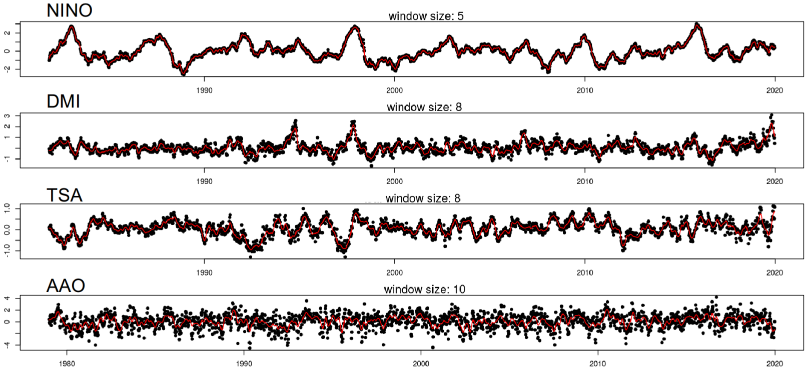
$$K(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-t^2/2\right),$$

where the parameter t controls the size of the smoothing “window”

⇒ The window size is typically selected with cross-validation.

Smoothing Climate Indices

- Gaussian kernel:



Smoothing Climate Indices

Smoothing climate indices can protect against these noisy jumps.

Smoothing splines:

- Optimize a loss function of the "Loss + Penalty" form.
- Loss term encourages smoothing spline to fit data well.
- Penalty term prevents smoothing spline from overfitting.

Find the function f that minimizes

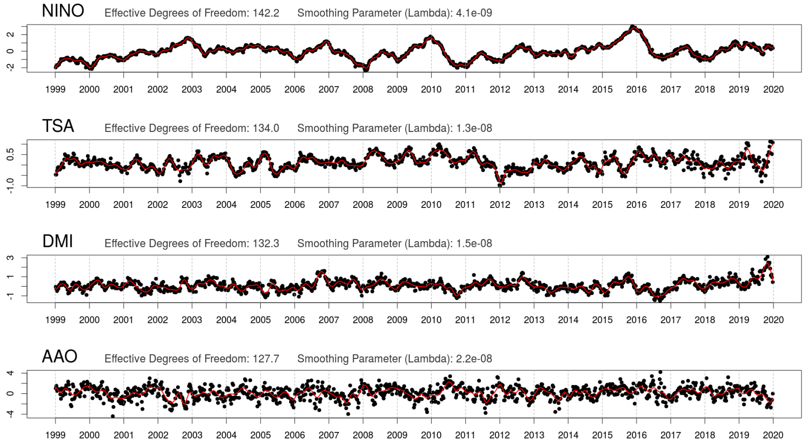
$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int f''(t)^2 dt$$

where the tuning parameter λ balances the loss and penalty terms.

$\Rightarrow$ The tuning parameter is typically selected with cross-validation.

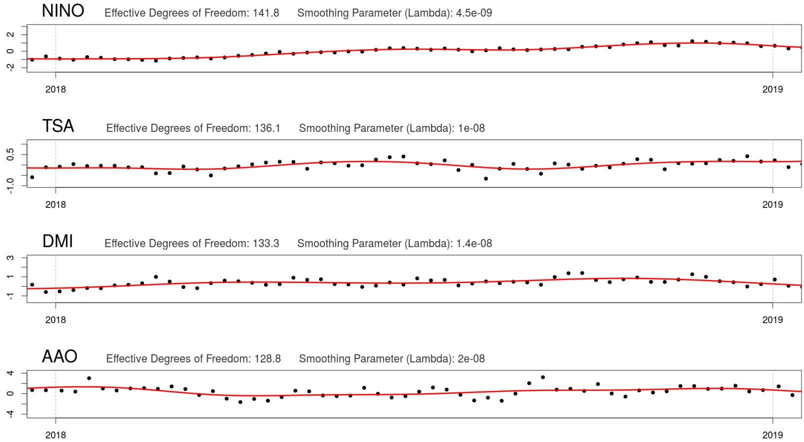
Smoothing Climate Indices

- Smoothing splines:



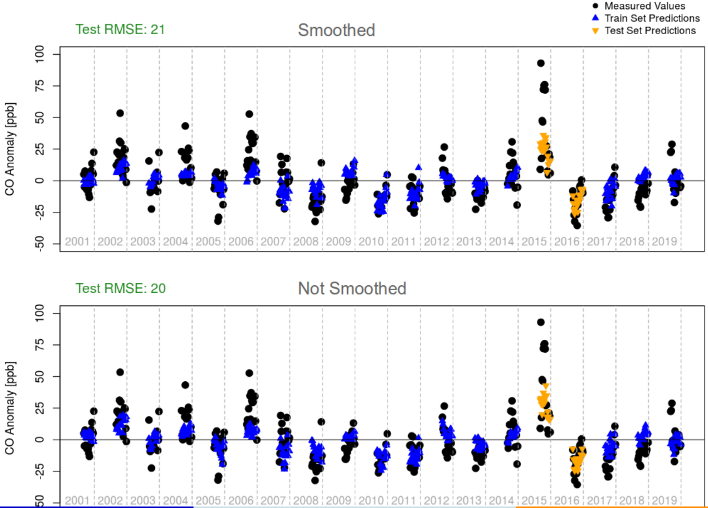
Smoothing Climate Indices

- Smoothing reduces noise, but potentially eliminates signal as well.



Model Performance

In this case, smoothing actually increases test RMSE! Perhaps the variability is signal after all...



Final time for Questions

Time for questions!
Thanks!