

Overview of Supervised Learning II

Statistical Data Mining I

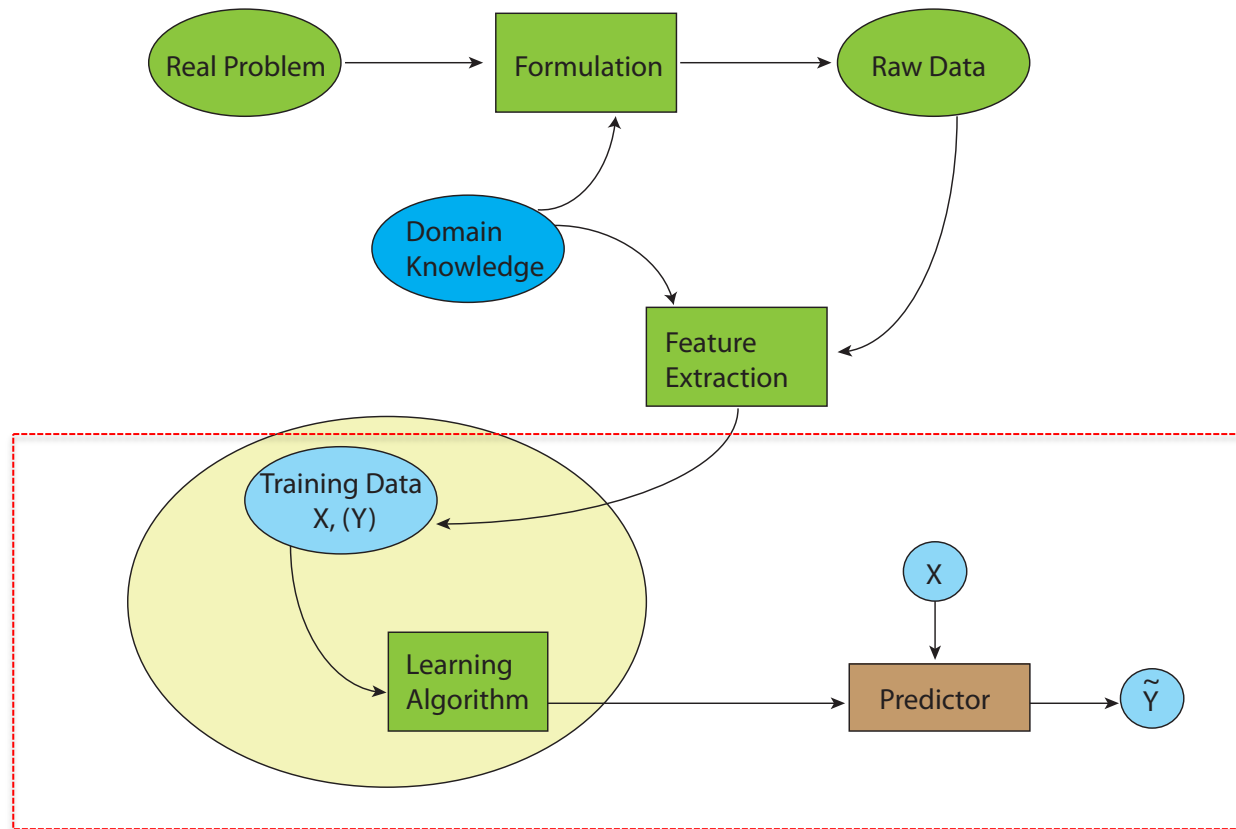
Rachael Hageman Blair

Our goal

Find a good approximation $f(X)$ of $\hat{f}(X)$ that accurately describes the relationship between input and output variables.

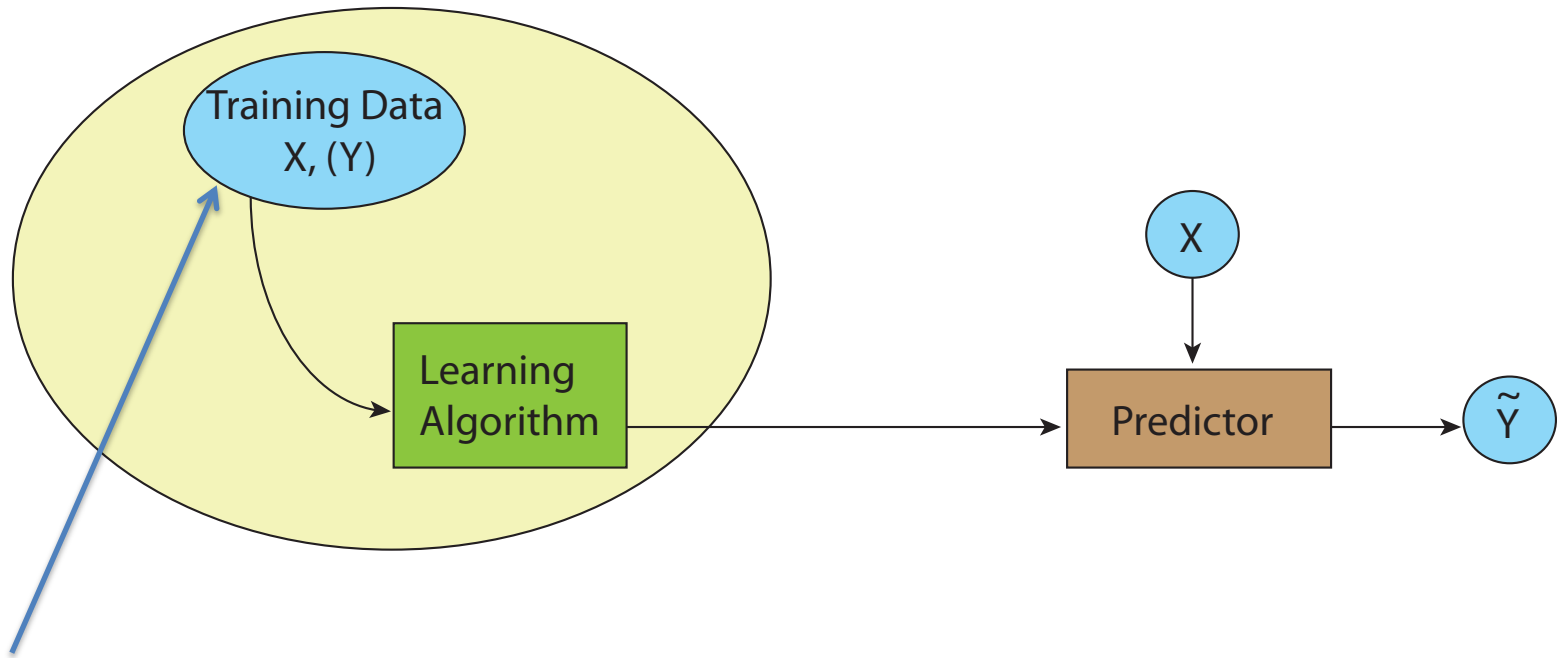
Functional Approximation

Supervised Learning with the additive error model $Y = f(x) + \varepsilon$



Functional Approximation

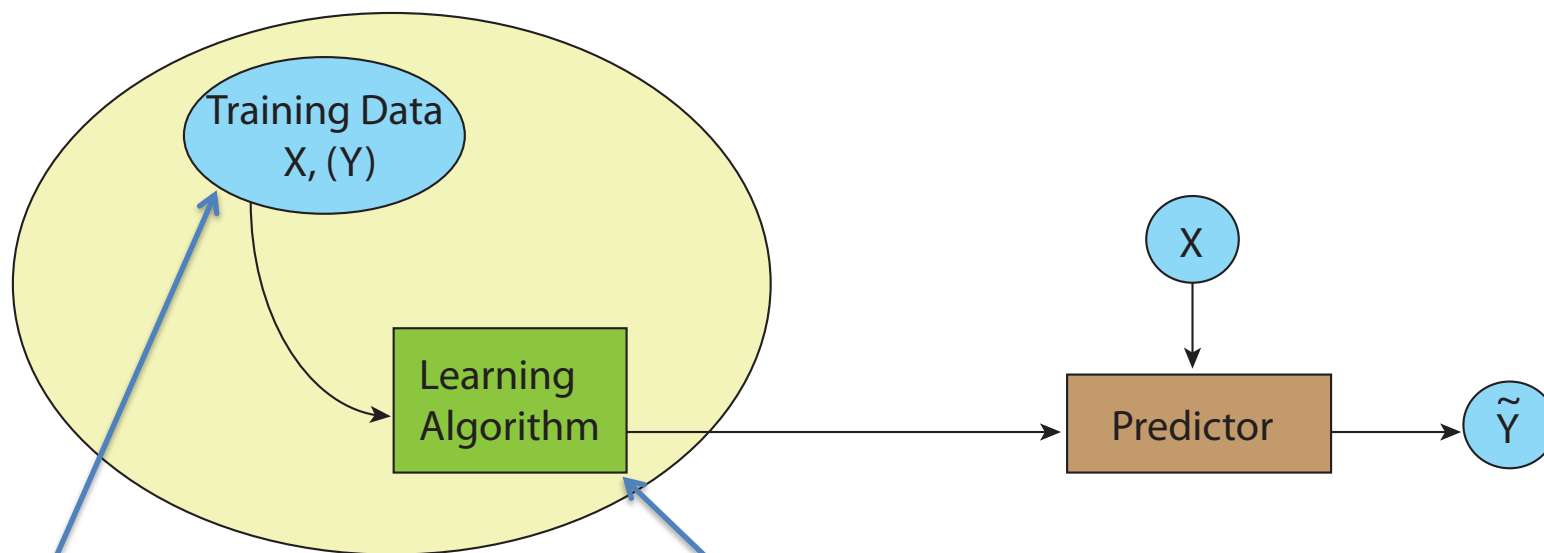
Supervised Learning with the additive error model $Y = f(x) + \varepsilon$



$f(x)$ may or may not be deterministic,
we only observe X and Y . Assume we
have no prior knowledge

Functional Approximation

Supervised Learning with the additive error model $Y = f(x) + \varepsilon$

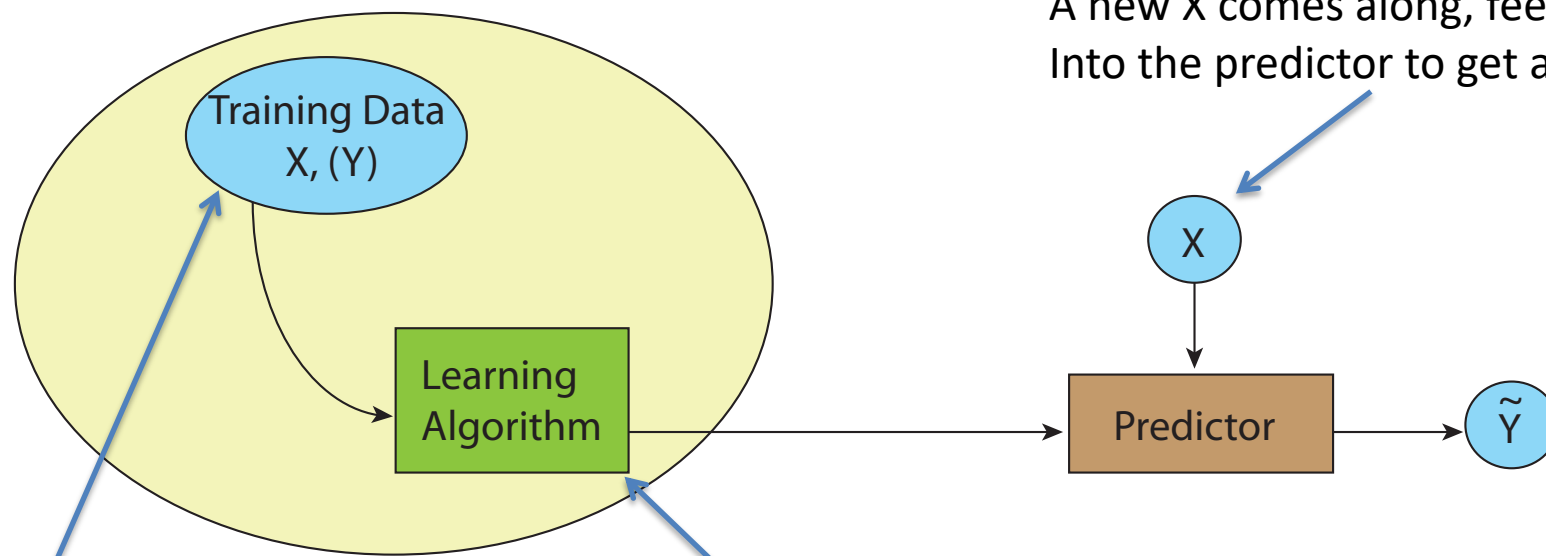


$f(x)$ may or may not be deterministic, we only observe X and Y . Assume we have no prior knowledge

Feed X and Y to the machine, it will tell you the $\hat{f}(x)$ that minimizes the residual sum of squares. We "learn" the optimal predictor.

Functional Approximation

Supervised Learning with the additive error model $Y = f(x) + \varepsilon$



$f(x)$ may or may not be deterministic, we only observe X and Y . Assume we have no prior knowledge

Feed X and Y to the machine, it will tell you the $\hat{f}(x)$ that minimizes the residual sum of squares. We "learn" the optimal predictor.

A new X comes along, feed it Into the predictor to get an output.

Functional Approximation

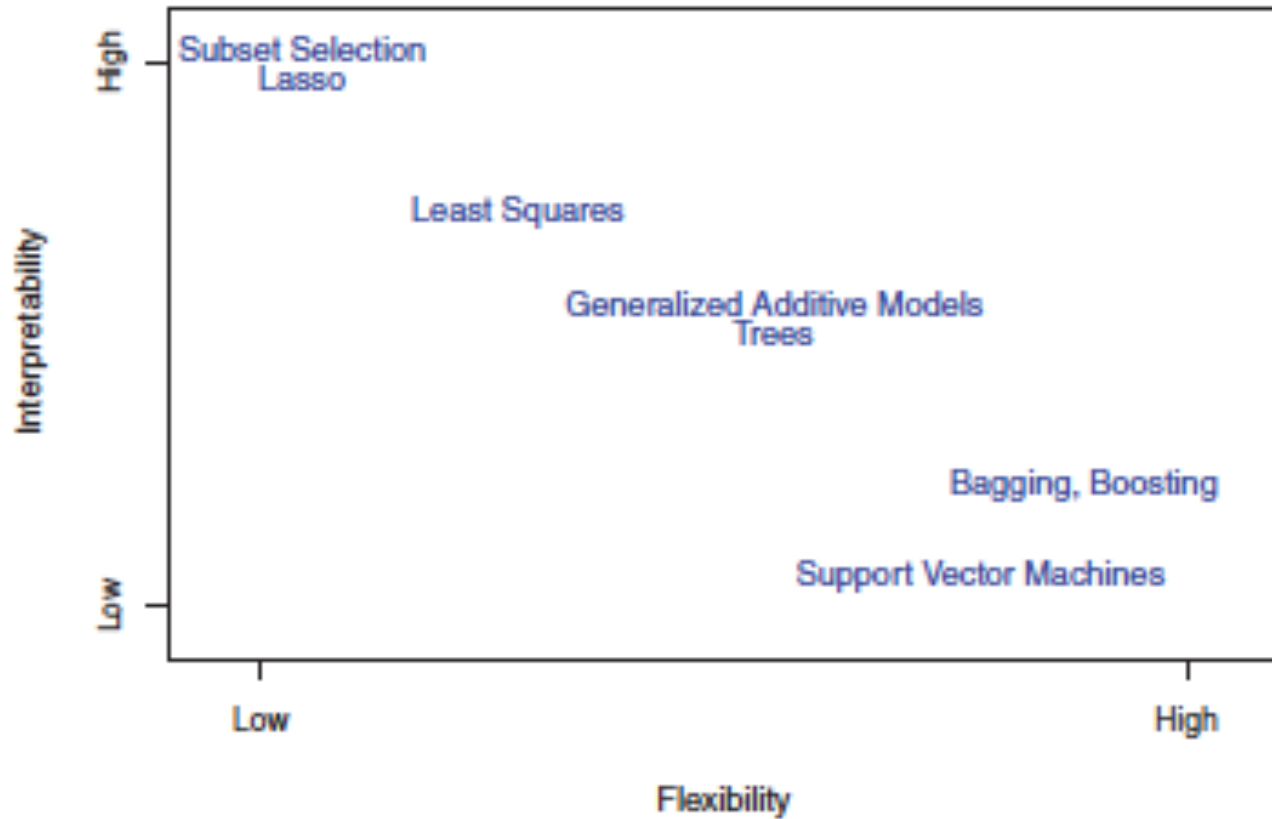
There are two reasons to estimate $f(x)$:

(1) Prediction – minimize “reducible error”.

(2) Inference

$$\hat{f}(x)$$

Functional Approximation



Functional Approximation

Goal: Obtain a ‘useful approximation’ to $f(x)$ for all $x \in \mathfrak{R}^p$ in a Euclidean space, given the training data T .

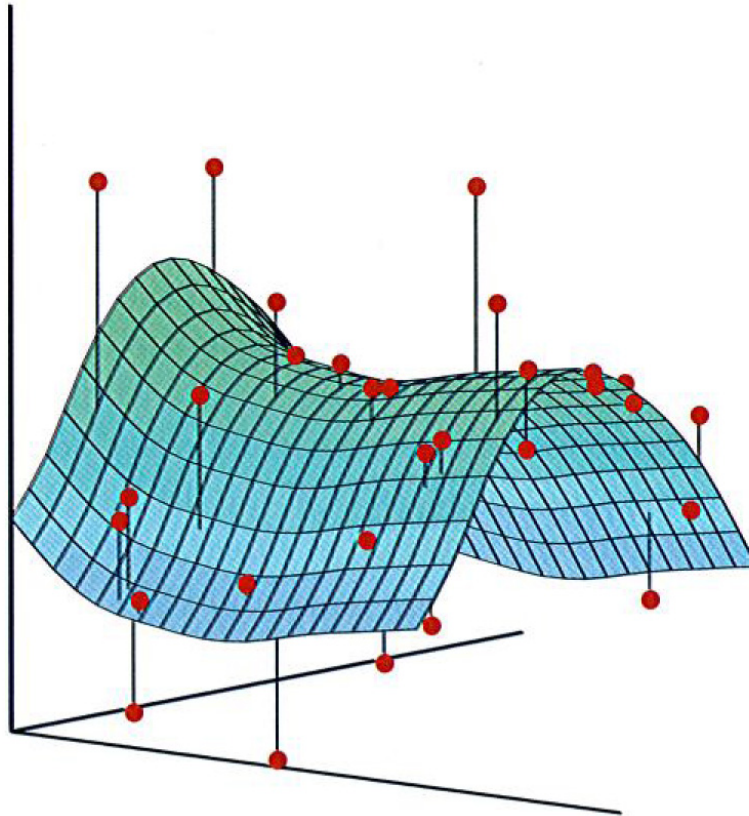
Parameters: Approximations depend on parameters θ , e.g.,

- Linear Model: $f(x) = x^T \beta$, the parameters $\theta = \beta$.
- Linear Basis Expansions: $f_\theta(x) = \sum_{k=1}^K h_k(x) \theta_k$,
 h_k where can be polynomial, trigonometric, sigmodial, etc.

Functional Approximation

We can take a least squares approach and find the parameters that minimize the RSS:

$$RSS(\theta) = \sum_{i=1}^N (y_i - f_{\theta}(x_i))^2.$$



Functional Approximation

A **more general approach** is maximum likelihood.

Suppose we have a random sample y_i , $i = 1 \dots N$ from a density $\Pr_\theta(y)$ indexed by some parameters θ .

The log-probability of the observed sample is:

$$L(\theta) = \sum_{i=1}^N \log \Pr_\theta(y_i).$$

Structured Regression Models

Consider the RSS:

$$RSS(f) = \sum_{i=1}^N (y_i - f(x_i))^2$$

Difficulty of the problem:

- There are infinitely many solutions.
- May be a poor predictor of non-test data.
- There may be multiple observation pairs per x_i .

We need to place constraints on the solution space!

Structured Regression Models

Solution Constraints - How can we do it?

A couple of ways.....

- Encoded via parametric representation of the function we are trying to approximate.
 - Build into the learning method, implicitly or explicitly. Often called a “complexity restriction”.
-
- Constraints are often imposed in local neighborhoods.
 - Strong constraints – large neighborhood size. The more sensitive the solution is to the adopted constraint.

Structured Regression Models

Solution Constraints - How can we do it?

A couple of ways.....

- Encoded via parametric representation of the function we are trying to approximate.
 - Build into the learning method, implicitly or explicitly. Often called a “complexity restriction”.
-
- Constraints are often imposed in local neighborhoods.
 - Strong constraints – large neighborhood size. The more sensitive the solution is to the adopted constraint.


IMPORTANT NOTE: Before we have ambiguity arising from infinitely many solutions. Constraint restrict the class of feasible functions. However, the ambiguity remains.

Why? Because there are infinitely many constraints.

Roughness Penalty and Bayesian Methods

Penalized RSS:


$$PRSS(f; \lambda) = RSS(f) + \lambda J(f).$$



User-selected functional
Lots of possibilities.

Example - cubic smoothing spline

$$PRSS(f; \lambda) = \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda \int (f''(x))^2 dx.$$



Penalizes large second
derivatives.

Prior – we expect a smooth
solution. Tuning parameter
lambda conveys our belief
in the smoothness.

Kernel Methods and Local Regression

Specifying the local neighborhood by a **Kernel function**:

$K_\lambda(x_0, x)$ which assigns weights to points x in a region around x_0 .

Example – Gaussian Kernel

Weights die exponentially with their squared Euclidean Distance from the center point.

$$K_\lambda(x_0, x) = \frac{1}{\lambda} \exp \left[-\frac{\|x - x_0\|^2}{2\lambda} \right]$$

Incorporates variance of the Gaussian density and controls neighborhood size.

Residual Sum of Squares:

$$RSS(f_\theta, x_0) = \sum_{i=1}^N K_\lambda(x_0, x_i) (y_i - f_\theta(x_i))^2$$

Example – Nearest Neighbor (data-centric)

Basis Functions and Dictionary Methods

The model f is a linear expansion of **Basis functions**:

$$f_{\theta}(x) = \sum_{m=1}^M \theta_m h_m(x).$$

Example – Neural Networks:

$$f_{\theta}(x) = \sum_{m=1}^M \beta_m \frac{1}{1 + \exp(-x)} (\alpha_m^T x + b_m).$$

Considered an “adaptive basis function method”, in that the models are built up through a search mechanism.

AKA as dictionary methods – choosing from a dictionary of possibilities.

Model Selection and Bias-Variance tradeoff

These methods all employ a **smoothing** or **complexity parameter**:

- The multiplier on the penalty term.
- The width of the kernel.
- The number of basis functions.

How to choose complexity parameter?

We can't use RSS, because we would always choose the parameters that gives zeros residual. Would be a bad predictor for future data.

Model Selection and Bias-Variance tradeoff

Consider a given estimate \hat{f} and a set of predictors X , which yields the prediction $\hat{Y} = \hat{f}(X)$.

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

Model Selection and Bias-Variance tradeoff

Back to nearest neighbors: The expected prediction error (EPE) at x_0 :

$$\begin{aligned} EPE_k(x_0) &= E\left[(Y - \hat{f}_k(x_0))^2 \mid X = x_0\right] \\ &= \sigma^2 + \left[\text{Bias}^2(\hat{f}_k(x_0)) + \text{Var}_T(\hat{f}_k(x_0)) \right] \\ &= \sigma^2 + \left[f(x_0) - \frac{1}{k} \sum_{l=1}^k f(x_l) \right]^2 + \frac{\sigma^2}{k}. \end{aligned}$$

MSE

Irreducible error –
beyond out control.

Squared difference between the true
mean and the estimated. Likely to
increase with k .

The variance of an
average. As k
increases this
decreases.

Model Selection and Bias-Variance tradeoff

