Ludwig-Maximilians-Universität München
Lehrstuhl für Datenbanksysteme und Data Mining
Prof. Dr. Thomas Seidl

Knowledge Discovery and Data Mining I

Winter Semester 2018/19

Agenda

1. Introduction

2. Basics

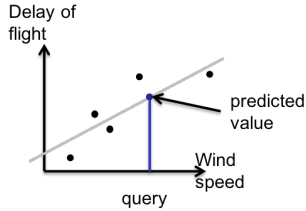
3. Unsupervised Methods

4. Supervised Methods
   - 4.1 Classification
   - 4.2 Regression
     - 4.2.1 Piece-Wise Linear Regression

5. Advanced Topics

Numerical Prediction

- Related problem to classification: numerical prediction
  - Determine the numerical value of an object
  - Method: e.g., regression analysis
  - Example: Prediction of flight delays



- Numerical prediction is *different* from classification
  - Classification refers to predict categorical class label
  - Numerical prediction models continuous-valued functions
- Numerical prediction is *similar* to classification
  - First, construct a model
  - Second, use model to predict unknown value
  - Major method for numerical prediction is regression:
    - Linear and multiple regression
    - Non-linear regression

Examples

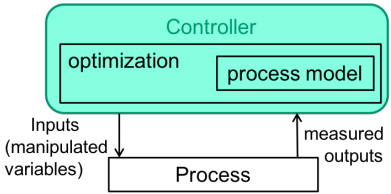
Example: Housing values in suburbs of Boston

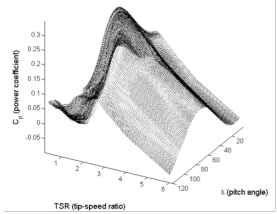
- Inputs:
  - Number of rooms
  - Median value of houses in the
  - Neighborhood
  - Weighted distance to five Boston employment centers
  - Nitric oxides concentration
  - Crime rate per capita
  - ...
- Goal: Compute a model of the housing values, which can be used to predict the price for a house in that area.

Examples

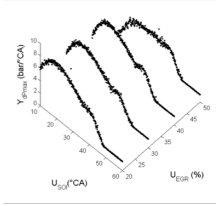
Control engineering

- Control the inputs of a system in order to lead the outputs to a given reference value
- Required: A model of the process





Wind turbine

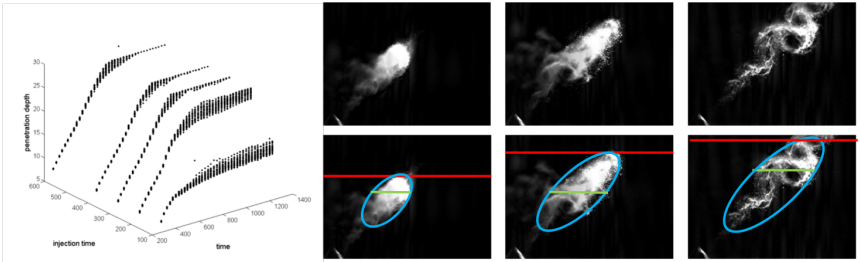


Diesel engine

Examples

Fuel injection process

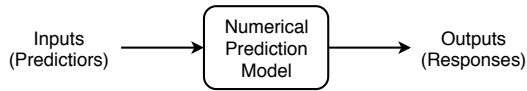
- Database of spray images
- Inputs: Settings in the pressure chamber
- Outputs: Spray features, e.g., penetration, depth, spray, width, spray area
- Goal: Compute a model which predicts the spray features, for input settings which have not been measured.



Numerical Prediction

Numerical Prediction

- Given: A set of observations
- Compute: A generalized model of the data which enables the prediction of the output as a continuous value



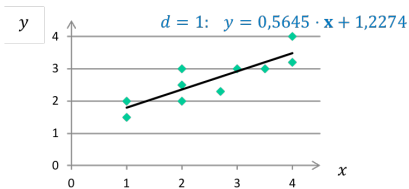
Quality Measures

- Accuracy of the model
- Compactness of the model
- Interpretability of the model
- Runtime efficiency (training, prediction)

Linear Regression

- Given a set of N observations with inputs of the form $x = (x_1, \ldots, x_d)^T \in \mathbb{R}^d$ and outputs $y \in \mathbb{R}$
- Approach: Minimize the *Sum of Squared Errors (SSE)*
- Numerical Prediction: Describe the outputs y as a linear equation of the inputs

$$\hat{y} = f(x) = w_0 + \sum_{i=1}^{d} w_i x_i = (1, x_1, \ldots, x_d)^T w$$



- Train the parameters $w = (w_0, w_1, \ldots, w_d)^T$ according to

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

Linear Regression

- Matrix notation: Let $X \in \mathbb{R}^{N \times (d+1)}$ be the matrix containing the inputs, $Y \in \mathbb{R}^N$ the outputs, and w the resulting coefficients:

$$X = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & \dots & x_{N,d} \end{pmatrix}, \qquad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}, \qquad w = \begin{pmatrix} w_0 \\ \vdots \\ w_d \end{pmatrix}$$

- Goal: Find the coefficients w, which minimize the SSE

$$\begin{aligned} \min_w \frac{1}{2} \sum_{i=1}^{N} (y_i - f(x_i))^2 &= \min_w \frac{1}{2} ||Xw - Y||_2^2 \\ &= \min_w \frac{1}{2} (Xw - Y)^T (Xw - Y) \end{aligned}$$

Linear Regression

- The optimal coefficients can be derived by setting the first derivative to zero (cf. classification with linear discriminant functions)

$$w = (X^T X)^{-1} X^T Y$$

- For $d = 1$, the regression coefficients w_0 and w_1 can be computed as

$$w_1 = \frac{Cov(x, y)}{Var(x)} = \frac{\tilde{x}^T \tilde{y}}{\tilde{x}^T \tilde{x}}, \qquad w_0 = \bar{y} - w_1 \bar{x}$$

  where $\tilde{x} = x - \bar{x}$ and $\tilde{y} = y - \bar{y}$

- Note: If $\bar{x} = \bar{y} = 0$ (i.e., the data is centered), then $w_1 = \frac{x^T y}{x^T x}$ and $w_0 = 0$

Polynomial Regression

- Second order polynomial for $d = 1$:

$$\hat{y} = f(x) = w_0 + w_1 x_1 + w_2 x_2^2 = (1, x_1, x_2^2)^T w$$

with

$$X = \begin{pmatrix} 1 & x_{1,1} & x_{1,1}^2 \\ \vdots & \vdots & \vdots \\ 1 & x_{N,1} & x_{N,1}^2 \end{pmatrix} \quad \text{and} \quad w = (X^T X)^{-1} X^T Y$$

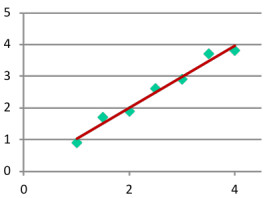
- Second order polynomial for $d = 2$:

$$\hat{y} = f(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2$$

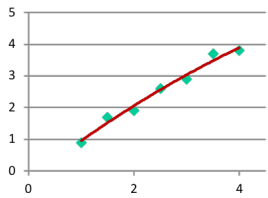
Polynomial Regression

- The number of coefficients increases exponentially with k and d
- Model building strategies: Forward selection, backward elimination
- The order of the polynomial should be as low as possible, high order polynomials tend to overfit the data

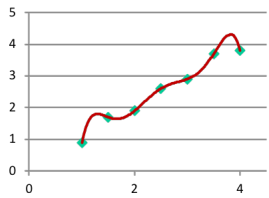
Linear model



Polynomial model 2^{nd} order



Polynomial model 6^{th} order



Nonlinear Regression

- Different nonlinear functions can be approximated
- Transform the data to a linear domain

$$\hat{y} = \alpha e^{\gamma x} \implies \ln \hat{y} = \ln \alpha + \gamma x$$
$$\implies \hat{y}' = w_0 + w_1 x$$

(for $\hat{y}' = \ln \hat{y}$, $w_0 = \ln \alpha$ and $w_1 = \gamma$)

- The parameters w_0 and w_1 are estimated with SSE
- The parameters α and γ are obtained, describing an exponential curve which passes through the original observations
- Problem: SSE determines normally distributed errors in the transformed space $\rightsquigarrow$ skewed error distribution in the original space

Nonlinear Regression

- Different nonlinear functions can be approximated
- Outputs are estimated by a function with nonlinear parameters, e.g., exponential, trigonometric
- Example type of function:

$$\hat{y} = w_0 + w_1 e^{w_2 x} + \sin(w_3 x)$$

- Approach: The type of nonlinear function is chosen and the corresponding parameters are computed
- No closed form solution exists $\rightsquigarrow$ numerical approximation:
  - Gauss Newton, Gradient descent, Levenberg-Marquardt

Linear and Nonlinear Regression

Problems

- Linear regression: Most of the real world data has a nonlinear behavior
- Polynomial regression: Limited, cannot describe arbitrary nonlinear behavior
- General nonlinear regression: The type of nonlinear function must be specified in advance

Agenda

1. Introduction

2. Basics

3. Unsupervised Methods

4. Supervised Methods
   - 4.1 Classification
   - 4.2 Regression
     - 4.2.1 Piece-Wise Linear Regression

5. Advanced Topics

Piecewise Linear Functions

Piecewise Linear Functions

For a partitioning of the input space $\mathcal{C} = \{C_1, \ldots, C_k\}$, a piecewise linear function is defined by coefficient vectors $\mathbf{w_i}$, and offset β_i for $i = 1, \ldots, k$.

$$f(\mathbf{x}) = \begin{cases} \mathbf{w}_1^T \mathbf{x} + \beta_1, & \mathbf{x} \in C_1 \\ \vdots & \\ \mathbf{w}_k^T \mathbf{x} + \beta_k, & \mathbf{x} \in C_k \end{cases}$$

Piece Linear Functions

Properties

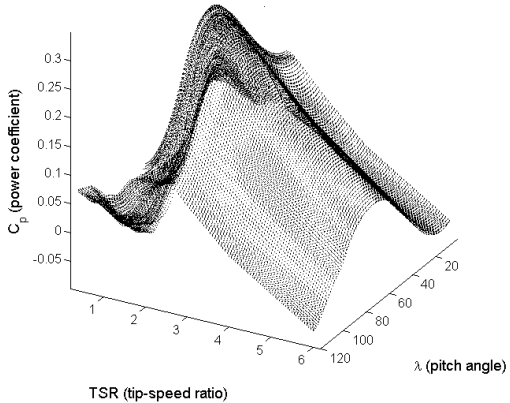
- Simple approach
- Can approximate any function
- *Accuracy increases* with increasing number of partitions
- *Compactness & Interpretability decreases* with increasing number of partitions

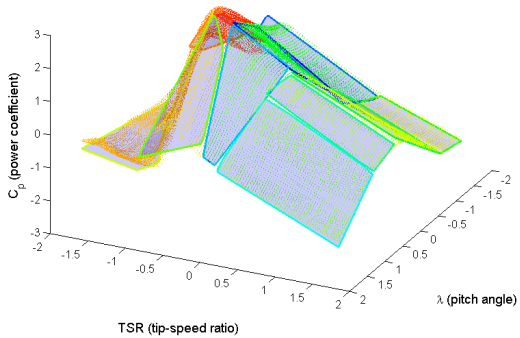
Challenge

Find an appropriate partitioning of the input space

Regression Tree

Greedy divide and conquer: recursive partioning of the input space.



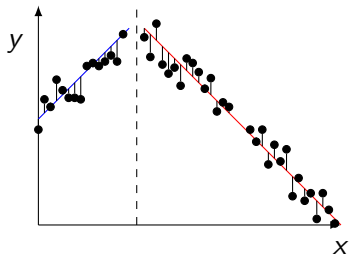


Regression Tree

General Approach

- Given: Set of observations $T = (X, Y)$ with $X = \{x_1, \ldots, x_n\}$, and $Y = \{y_1, \ldots, y_n\}$
- Find a split of T into T_1, T_2 with minimal summed impurity $imp(T_1) + imp(T_2)$.
- If the stopping criterion is not reached: repeat for T_1 and T_2
- If the stopping criterion is reached: undo the split

Impurity Measure



Variance of the Residuals

$$imp(T) = \frac{1}{|T|} \sum_{(\mathbf{x},y)\in T} (y - f(\mathbf{x}))^2$$

where f is the approximator function, i.e. here a linear function. For constant f, this measure coincides with the variance of the output.

Stopping Criterion: Impurity Ratio

Impurity Ratio Stopping Criterion

The recursive splitting is stopped if one of the following holds

- The sample size of a node is below some specified threshold
- The split is *not significant* for threshold τ_0, when

$$\tau = \frac{imp(T_1) + imp(T_2)}{imp(T)} \geq \tau_0$$

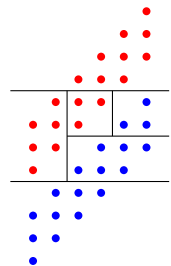
Choosing τ_0

τ_0 controls the overfitting/underfitting trade-off:

- τ_0 too large $\implies$ stopping too early $\implies$ model not accurate enough
- τ_0 too small $\implies$ stopping too early $\implies$ model overfits to observations

Split Strategy

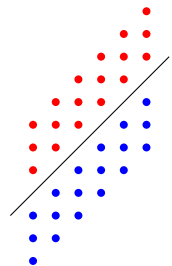
- The split strategy determines how the training samples are partitioned, whether the split is actually performed is decided by the stopping criterion.
- The most common splits are *axis parallel*:
  - Split = a value in one input dimension
  - Compute the impurity of all possible splits in all input dimensions and choose at the end the split with the lowest impurity
  - For each possible split compute the two corresponding models and their impurity $\rightsquigarrow$ expensive to compute



Five axis-parallel splits to separate red from blue samples.

Strategy for Oblique Splits

- More intuitive to use oblique splits
- An oblique split is a linear separator in the input space instead of a split value in an input dimension
- The optimal split (with minimal impurity measure) cannot be efficiently computed $\rightsquigarrow$ Heuristic approach required

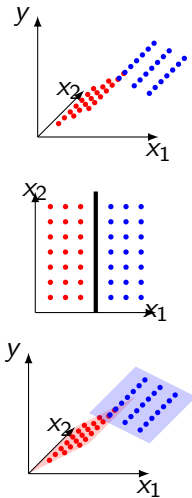


A single oblique split can separate red from blue samples.

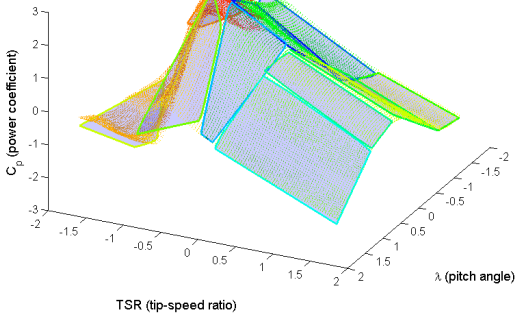
Strategy for Oblique Splits

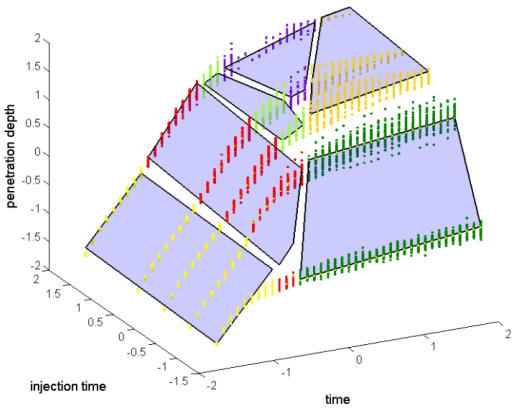
Heuristic Approach

1. Compute a clustering in the full (input + output) space, such that the samples are as well as possible described by linear equations
2. Project the clusters onto the input space
3. Use the clusters to train a linear classifier in the input space. Split = separating hyperplane in input space
4. Compute linear models for the two linearly separated clusters



Example: Oblique Splits





Continuous Splits

Motivation

- At the boundaries of the partitions the prediction may be discontinuous.
- In some applications this might by undesirable, e.g. when using the prediction to control engine speed, a rapid jump may cause damage to the engine

Solution

Hinging Hyperplane Models (not detailed in this lecture).

$$f(\mathbf{x}) = \sum_{i=1}^{k} h_i(\mathbf{x}) \qquad h_i(\mathbf{x}) = \begin{cases} \left\langle \mathbf{w}^{(i,+)}, \tilde{\mathbf{x}} \right\rangle, & \left\langle \mathbf{w}^{(i)}, \tilde{\mathbf{x}} \right\rangle > 0 \\ \left\langle \mathbf{w}^{(i,-)}, \tilde{\mathbf{x}} \right\rangle, & \left\langle \mathbf{w}^{(i)}, \tilde{\mathbf{x}} \right\rangle \leq 0 \end{cases} \qquad \begin{array}{l} \tilde{\mathbf{x}} = (1, x_1, \ldots, x_d) \\ \mathbf{w}^{(i)} = \mathbf{w}^{(i,+)} - \mathbf{w}^{(i,-)} \end{array}$$