DEPAUL UNIVERSITY

DSC478: Programming Machine Learning Applications

Roselyne Tchoua

rtchoua@depaul.edu

School of Computing, CDM, DePaul University

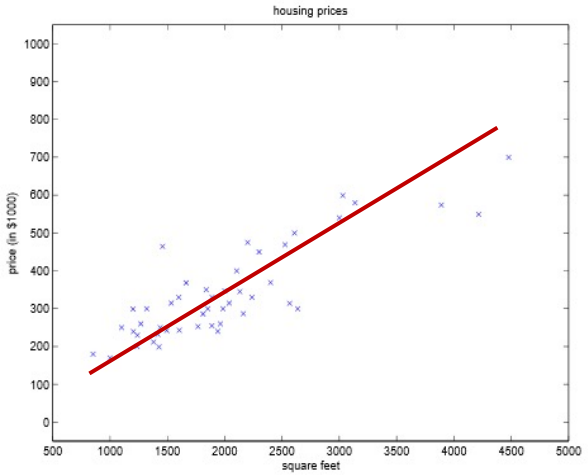
Numerical Prediction

- (Numerical) prediction/estimation/forecasting is similar to classification
  - construct a model
  - use model to predict continuous or ordered value for a given input
- Prediction is different from classification
  - Classification refers to predicting categorical class label
  - Prediction models **continuous-valued** functions
- Major method for prediction: regression
  - model the relationship between one or more *independent* or **predictor** variables and a ***dependent*** or **response** variable
- Regression analysis
  - Linear and multiple regression
  - Non-linear regression
  - Other regression methods: generalized linear model, Poisson regression, log-linear models, regression trees

Linear Regression

- Linear regression: involves a response variable y and a single predictor variable x → $y = w_0 + w_1 x$

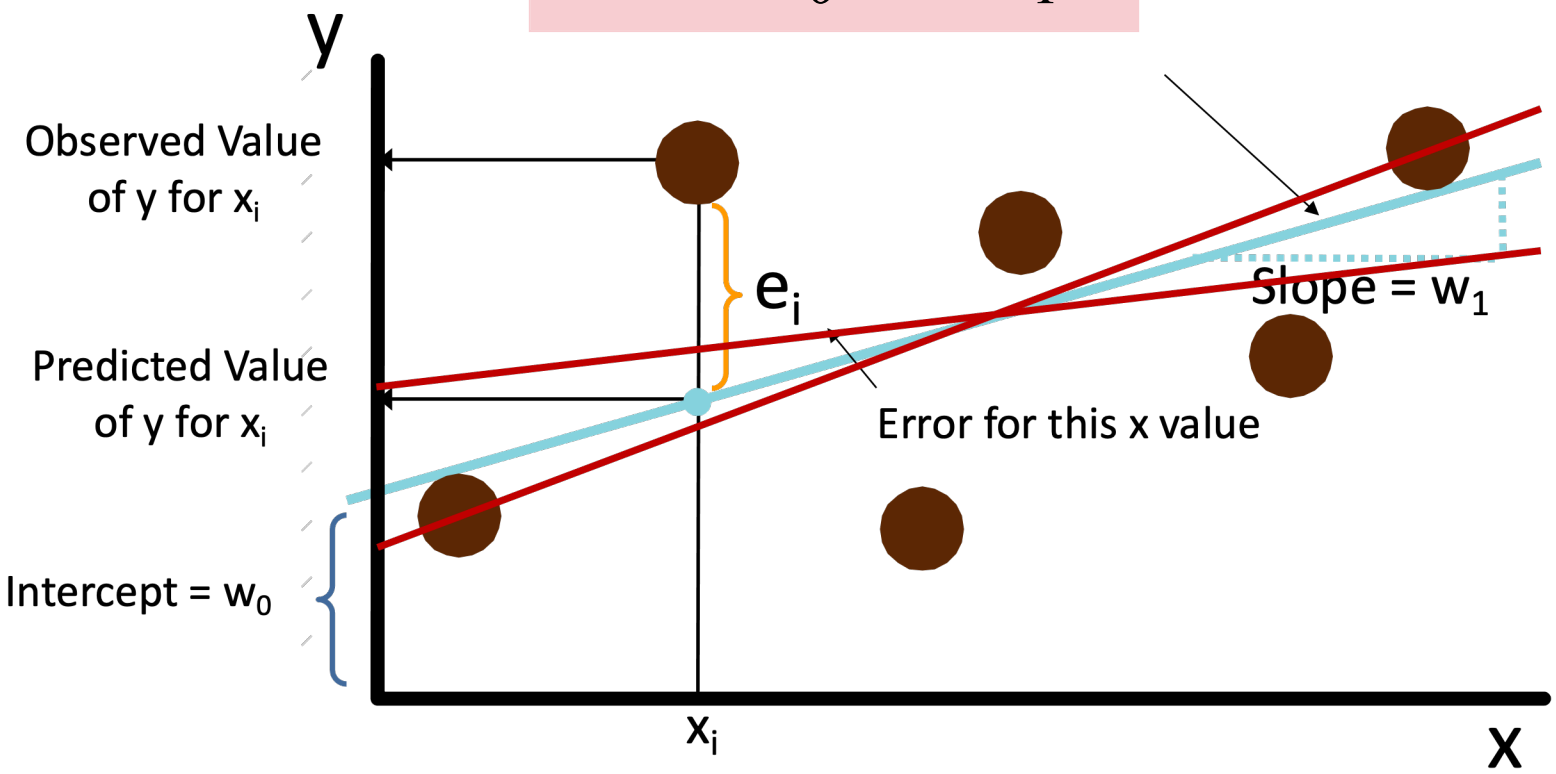
x	y
Living area ($feet^2$)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮



- Goal: Using the data estimate weights (parameters) w_0 and w_1 for the line such that the prediction error is minimized
  - The weights w_0 (y-intercept) and w_1 (slope) are regression coefficients

Linear Regression

$$y = w_0 + w_1 x$$



Linear Regression

Method of least squares:

- Estimates the best-fitting straight line
- w_0 and w_1 are obtained by minimizing the sum of the squared errors (a.k.a. residuals)

$$SSE = \sum_i e_i^2 = \sum_i (y_i - \hat{y}_i)^2$$
$$= \sum (y_i - (w_0 + w_1 x_i))^2$$

w_1 can be obtained by setting the partial derivative of the SSE to 0 and solving for w_1, ultimately resulting in:

$$w_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

Multiple Linear Regression

Multiple linear regression involves more than one predictor variable

- Features represented as $x_1, x_2, \ldots, x_d$
- Training data is of the form $(\mathbf{X}^1, y^1), (\mathbf{X}^2, y^2), \ldots, (\mathbf{X}^n, y^n)$
  (each $\mathbf{x}^j$ is a row vector in matrix **X**, i.e., a row in the data)
- For a specific value of a feature x_i in data item $\mathbf{X}^j$ we use: x_i^j
- Ex. For 2-D data, the regression function is: $\hat{y} = w_0 + w_1 x_1 + w_2 x_2$

x_1	x_2	y
Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

Least Squares Generalization

Multiple dimensions: to simplify add a new feature x_0 = **1** to feature vector **x**:

x_0	x_1	x_2	y
	Living area (feet^2)	#bedrooms	Price (1000$s)
1	2104	3	400
1	1600	3	330
1	2400	3	369
1	1416	2	232
1	3000	4	540
⋮	⋮	⋮	⋮

$$\hat{y} = f(x_0, x_1, \ldots, x_d) = w_0 x_0 + \sum_{i=1}^{d} w_i x_i = \sum_{i=0}^{d} w_i x_i = \mathbf{w}^T.\mathbf{x}$$

Least Squares Generalization

$$\hat{y} = f(x_0, x_1, \ldots, x_d) = f(\mathbf{x}) = w_0 x_0 + \sum_{i=1}^{d} w_i x_i = \sum_{i=0}^{d} w_i x_i = \mathbf{w}^{\mathrm{T}} . \mathbf{x}$$

Calculate the error function (SSE) and determine **w**:

$$E(\mathbf{w}) = \left(\mathbf{y} - f(\mathbf{x})\right)^2 = \left(\mathbf{y} - \sum_{i=0}^{d} w_i \cdot x_i\right)^2 = \sum_{j=1}^{n} (y^i - \sum_{i=0}^{d} w_i \cdot x_i^j)^2$$

$$= (\mathbf{y} - \mathbf{X}\mathbf{w})^{\mathrm{T}} \bullet (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$\mathbf{y}$ = vector of all training responses y^j

$\mathbf{X}$ = matrix of all training samples $\mathbf{x}^j$

$$\mathbf{w} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

Closed form solution to

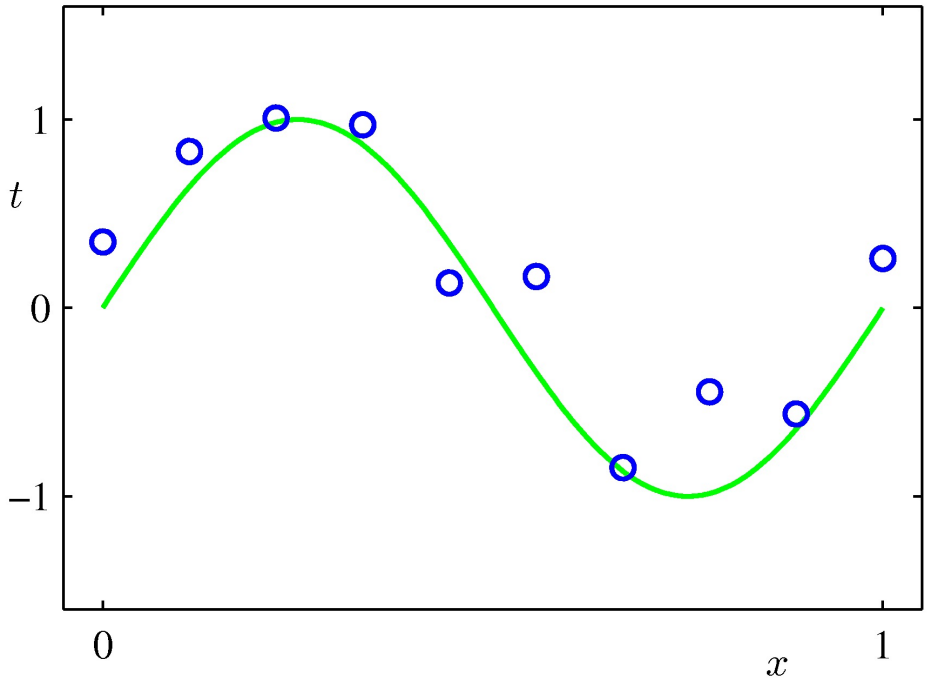
$$\frac{\partial}{\partial \mathbf{w}} E(\mathbf{w}) = 0$$

$\hat{y}^{test} = \mathbf{w} \cdot \mathbf{x}^{test}$ for test sample $\mathbf{x}^{test}$

Extending Application of Linear Regression

- The inputs **X** for linear regression can be:
  - Original quantitative inputs
  - Transformation of quantitative inputs, e.g., log, exp, square root, square, etc.
  - Polynomial transformation
    - example: $y = w_0 + w_1 \cdot x + w_2 \cdot x^2 + w_3 \cdot x^3$
  - Dummy coding of categorical inputs
  - Interactions between variables
    - example: $x_3 = x_1 \cdot x_2$
- This allows use of linear regression techniques to fit much more complicated non-linear datasets.

Example of Polynomial Regression



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Regularization

- Complex models (lots of parameters) are often prone to overfitting
- Overfitting can be reduced by imposing a constraint on the overall magnitude of the parameters (i.e., by including coefficients as part of the optimization process)
- Two common types of regularization in linear regression:
  - L_2 regularization (a.k.a. ridge regression). Find **w** which minimizes:

$$\sum_{j=1}^{N}(y_j - \sum_{i=0}^{d} w_i \cdot x_i)^2 + \lambda \sum_{i=1}^{d} w_i^2$$

    - λ is the regularization parameter: bigger λ imposes more constraint

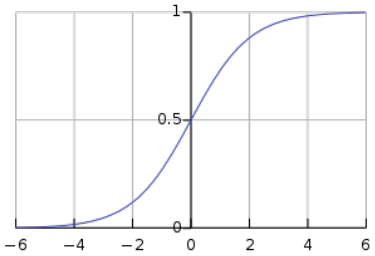
  - L_1 regularization (a.k.a. lasso). Find **w** which minimizes:

$$\sum_{j=1}^{N}(y_j - \sum_{i=0}^{d} w_i \cdot x_i)^2 + \lambda \sum_{i=1}^{d} |w_i|$$

N = samples
D = features

Other Regression Models

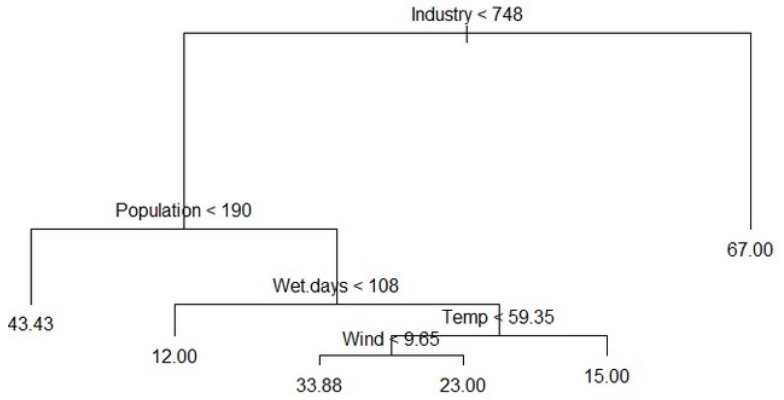
- Generalized linear models (error distribution models other than a normal distribution)
  - Foundation on which linear regression can be applied to modeling categorical response variables
  - Variance of y is a function of the mean value of y, not a constant
  - Logistic regression models the probability of some event occurring as a linear function of a set of predictor variables
  - Poisson regression models the data that exhibit a Poisson distribution
- Log-linear models (for categorical data: the form of a function whose logarithm equals a linear combination of the parameters of the model)
  - Approximate discrete multidimensional prob. distributions
  - Also useful for data compression and smoothing
- Regression trees and model trees
  - Trees to predict continuous values rather than class labels



Logistic regression

Other Regression Models: Trees

- Regression tree: proposed in CART system (Breiman et al. 1984)
  - CART: Classification And Regression Trees
  - Each leaf stores a continuous-valued prediction
  - It is the average value of the predicted attribute for the training instances that reach the leaf
- Model tree: proposed by Quinlan (1992)
  - Each leaf holds a regression model—a multivariate linear equation for the predicted attribute
  - A more general case than regression tree



Decision Tree of Pollution Dataset

- Regression and model trees tend to be more accurate than linear regression when instances are not represented well by simple linear models

Evaluating Numeric Prediction

- Prediction Accuracy
  - Difference between predicted scores and the actual results (from evaluation set)
  - Typically, the accuracy of the model is measured in terms of variance (i.e., average of the squared differences)

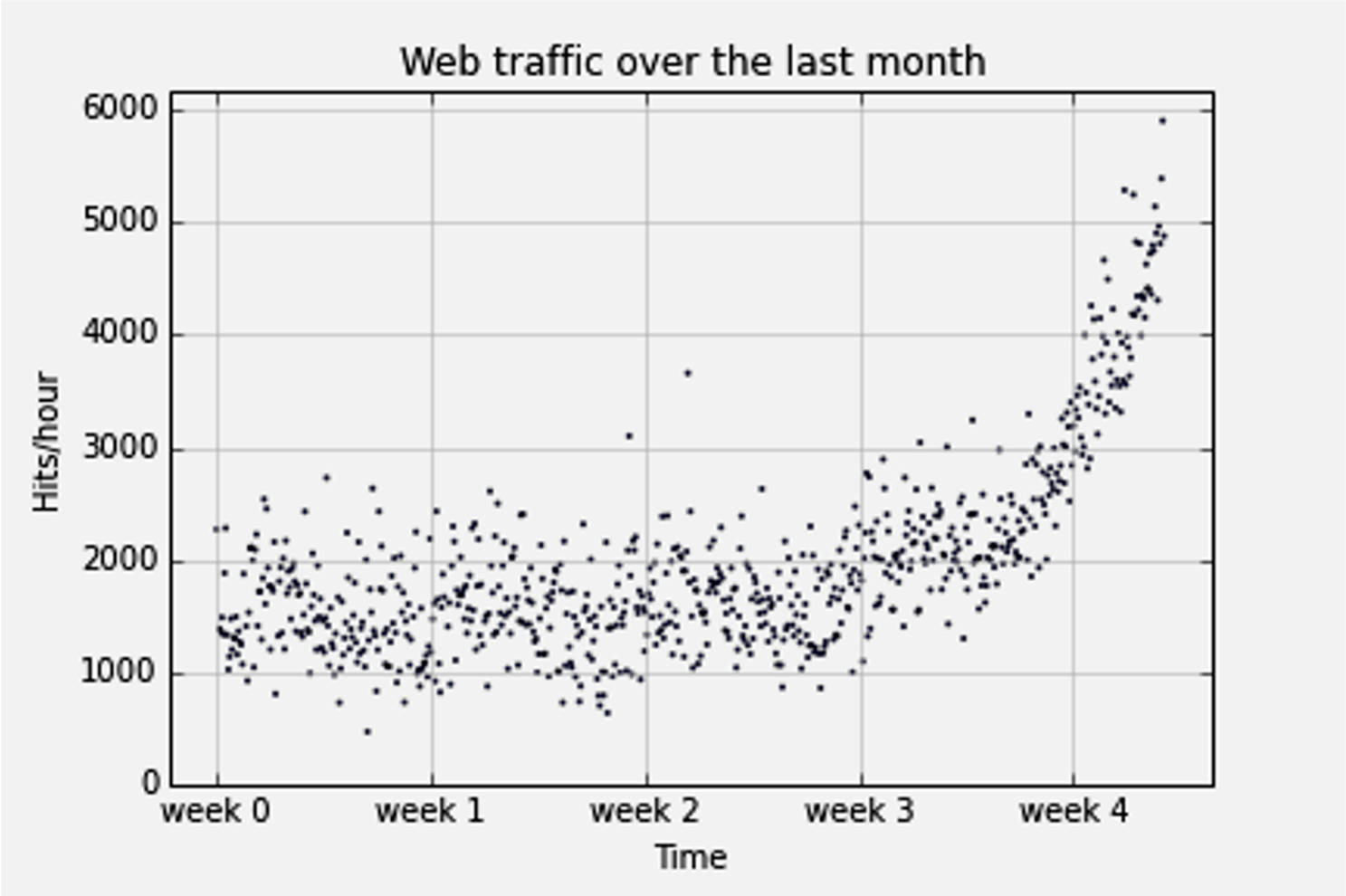
- Common Metrics (p_i = predicted target value for test instance i, a_i = actual target value for instance i)
  - **Mean Absolute Error:** Average loss over the test set

$$MAE = \frac{(p_1 - a_1) + \ldots + (p_n - a_n)}{n}$$

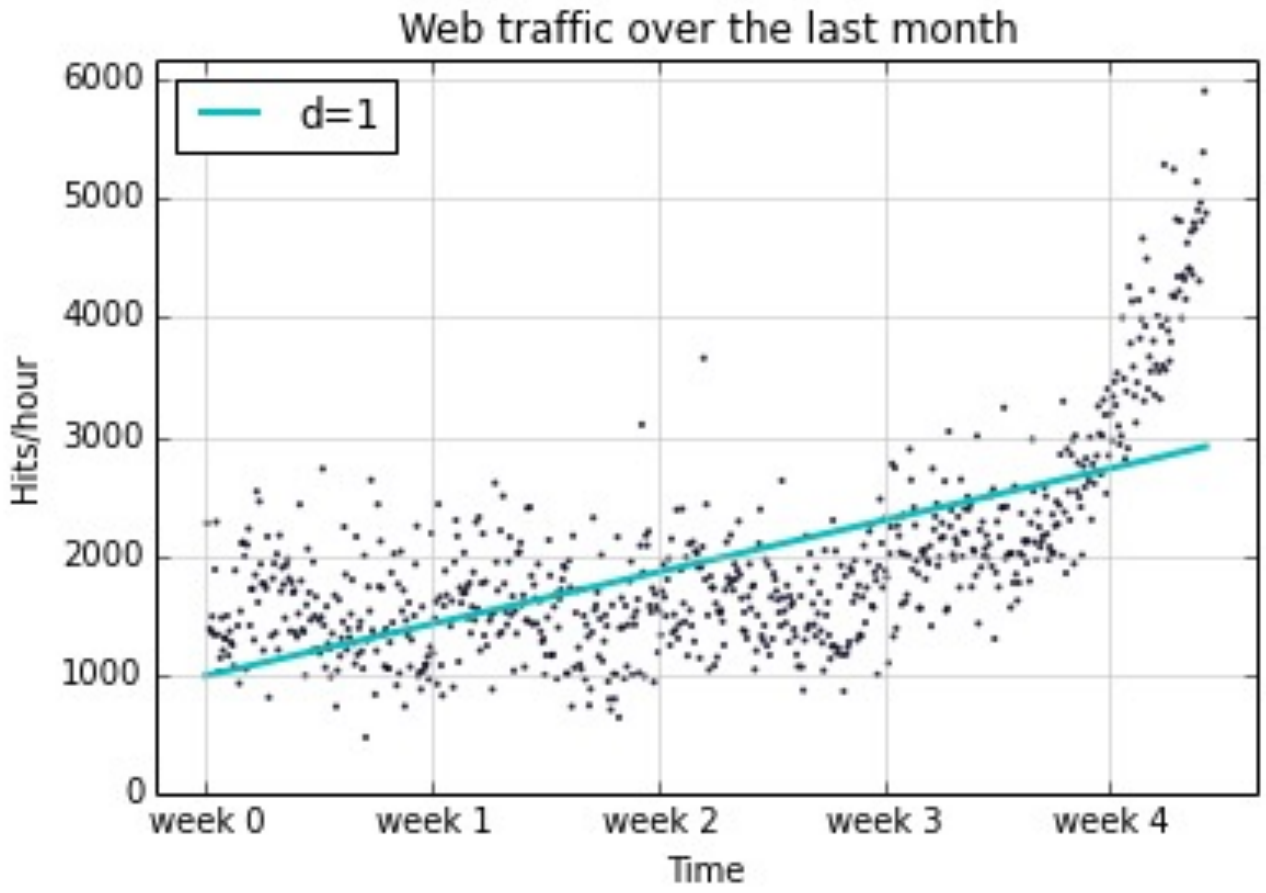
  - **Root Mean Squared Error**: compute the standard deviation (i.e., square root of the co-variance between predicted and actual ratings)

$$RMSE = \sqrt{\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}}$$

Example: Web Traffic Data

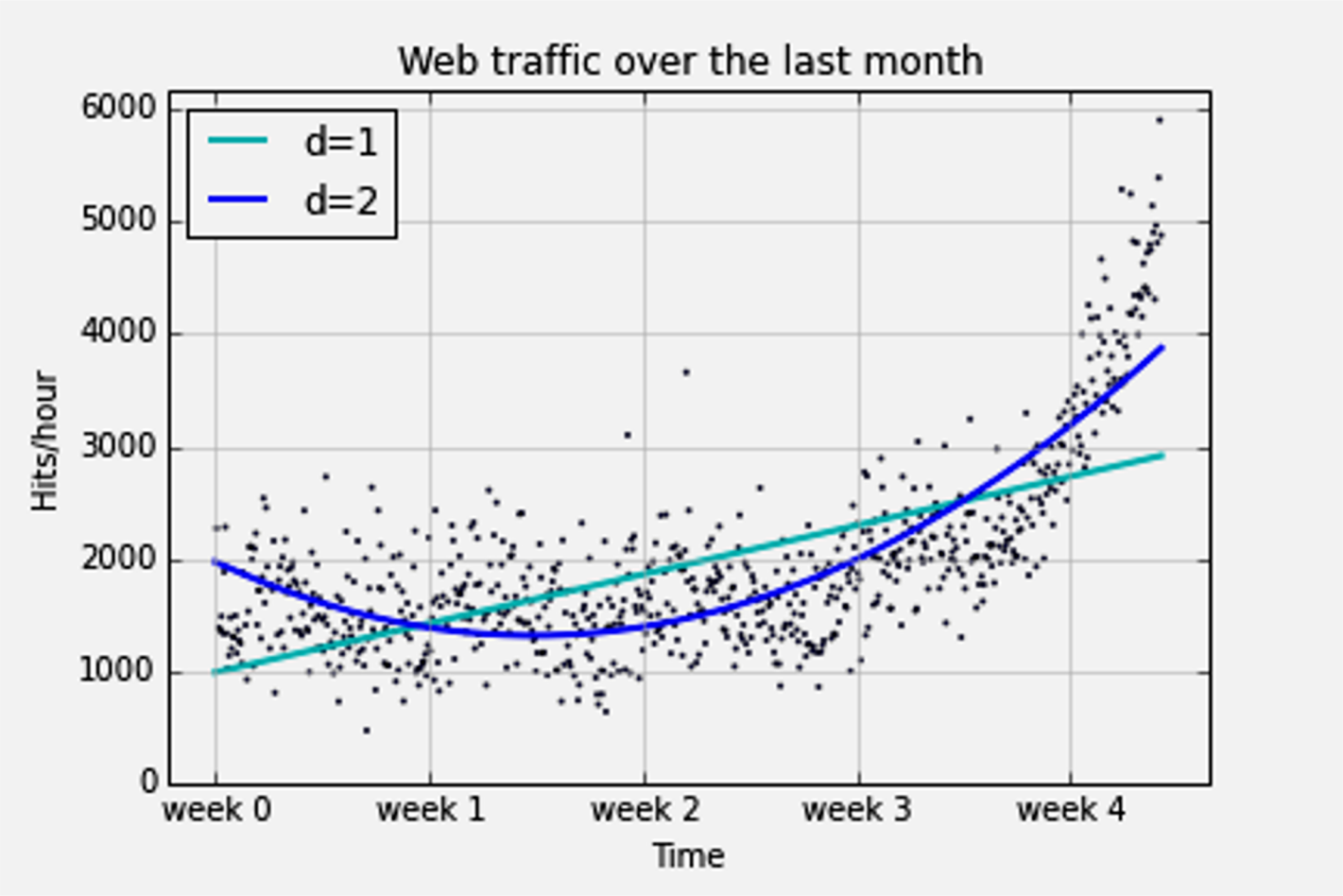


1D Poly Fit

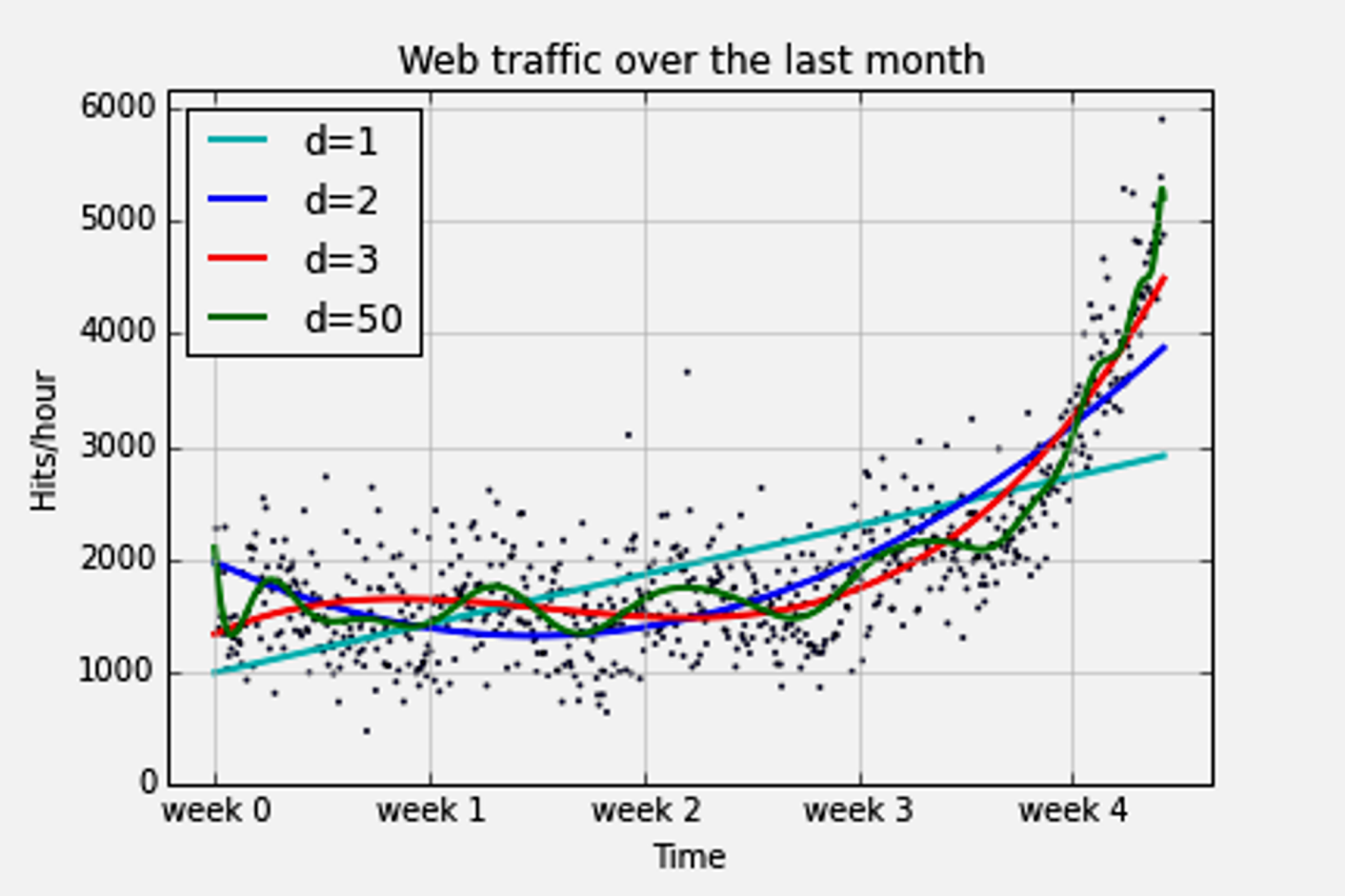


Example of too much "bias" → underfitting

1D and 2D Poly Fit



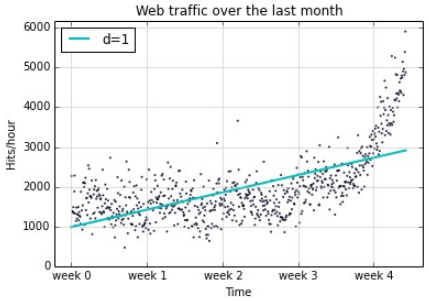
Example: 1D, 2D, 3D, 50D Poly Fit



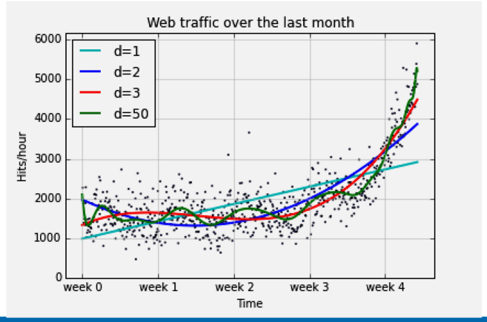
Example of too much "variance" → overfitting

Bias-Variance Tradeoff

- Bias: Any assumption made about your data
  - High-bias model: pays no or little attention to data

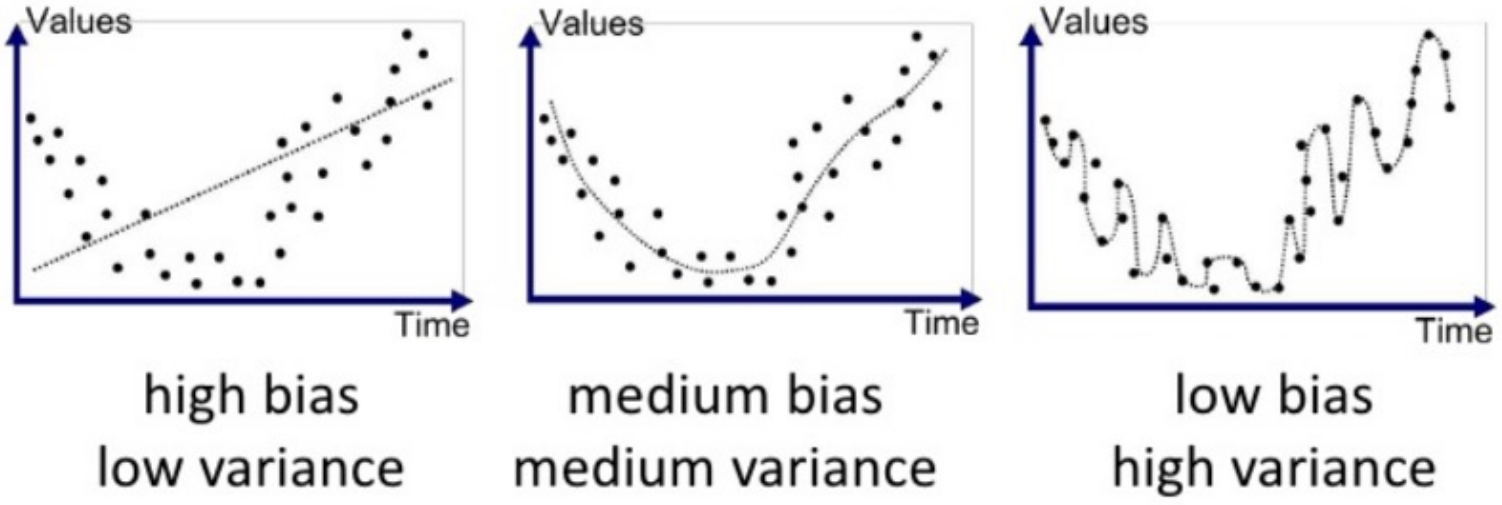


- Variance: Changes in predictions with different training sets (very different predictions for similar test points)
  - High variance model: pays too much attention to the data – overfits!



Bias-Variance Tradeoff

Tension between the two: often decreasing bias (increasing model complexity) will increase variance, decreasing variance will increase bias



Low← Model Complexity → High

Source: https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

Addressing Bias-Variance Tradeoff

- Possible ways of dealing with high bias
  - Get additional features
  - More complex model **(e.g., adding polynomial terms such as x_1^2, x_2^2, $x_1.x_2$, etc.)**
  - **Use smaller regularization coefficient λ.**
  - **Note:** getting more training data won't necessarily help in this case

- Possible ways dealing with high variance
  - Use more training instances
  - Reduce the number of features
  - Use simpler models
  - **Use a larger regularization coefficient λ.**