# Machine Learning and Data Mining I: Lecture 3
# Polynomial Regression and Regularization

Morgan McCarty

06 July 2023

# 1 Polynomial Regression

## 1.1 Linear Regression Review

- Estimate the parameters of a regression function:
- $\hat{y}_i = w_0 + \Sigma_{j=1}^{N} w_j x_{ij}$ where:
  - $w = (X^T X)^{-1} X^T y$ where:
    - $w$ is the vector of weights
    - $X$ is the matrix of features
    - $y$ is the vector of observations
  - $y_i$ is the $i$th observation (of the vector of observations)
  - $x_{ij}$ is the $j$th feature of the $i$th observation
  - $N$ is the number of features
- We find the solution by minimizing the squared error on the training data.
- We want the model to generalize to new data (i.e. have low error on unseen [test] data).
- The parametric form of the regression function:
  - Linear in parameters
  - Can transform input features
  - Do we learn a straight line? Polynomial?

## 1.2 Polynomial Regression

- Transform the input features to higher order polynomials.
- i.e. $f_w(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \cdots + w_m x^m$
- Here $m$ is the degree of the polynomial (the highest power of $x$).
- We can choose $m$, it is a hyperparameter.
- Does our optimization problem change? No, we can still use the same linear regression model, but we have to transform the input features.

- Our previous design matrix $X$ was:

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Np} \end{bmatrix} \tag{1}$$

- Where $N$ is the number of observations and $p$ is the number of features.
- All terms are linear in the parameters (i.e. $w_0, w_1, \ldots, w_N$ i.e. of order 1).
- We can transform the input features to higher order polynomials:

$$X = \begin{bmatrix} 1 & x_{11} & x_{11}^2 & \cdots & x_{11}^m \\ 1 & x_{21} & x_{21}^2 & \cdots & x_{21}^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N1}^2 & \cdots & x_{N1}^m \end{bmatrix} \tag{2}$$

- Where $m$ is the degree of the polynomial and $N$ is the number of observations.
- The weight vector $w$ was previously:

$$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} \tag{3}$$

- Now it is:

$$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix} \tag{4}$$

- The difference is that we have more features, but they are all linear in the parameters. That is that the parameters are still of order 1 so we can still use linear regression.
- The observation vector $y$ remains the same:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{5}$$

- We can still use the same linear regression model:
- We will minimize the mean squared error:

$$\underset{w}{\operatorname{argmin}} \frac{1}{N} \Sigma_{i=1}^{N} (y_i - f_w(x_i))^2 \tag{6}$$

### 1.2.1 An Example

We have a training dataset sampled from the unit interval $[0, 1]$ with a target function $f(x) = 7.5\sin(2.5\pi x)$ and gaussian noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

$$y_i = f(x_i) + \epsilon_i \tag{7}$$

The training dataset has $N$ observations. $(x_i, y_i)$ for $i = 1, \dots, N$.

1. If we use Polynomial Regression with $m = 1$ (i.e. a straight line) the model fails to capture the underlying function.
   It has a massive training error and underfits the data. Meaning that the model is too simple to capture the underlying function.

2. If we use Polynomial Regression with $m = 6$ (i.e. a polynomial of degree 6) the model captures underlying function and the noise in the data.
   It has a negligible training error and overfits the data. Meaning that the model is too complex and fits to the noise, not just the function.

3. If we use Polynomial Regression with $m = 3$ (i.e. a polynomial of degree 3) the model captures underlying function without fitting to the noise in the data.
   It has a small training error and generalizes well to unseen data.

## 1.3 Overfitting

- The model learns idiosyncrasies (noise) in the training data resulting in poor generalization.
- When does this happen?
  - There are more model parameters than training data points.
  - Model is too complex and fits to the noise in the training data.
  - E.g. a degree $N$ polynomial exactly fits $N + 1$ data points.

## 1.4 Avoiding Overfitting

- More training data: always works, but is not always possible in practice.
- Cross Validation: split the training data into a training set and a test set. Train the model on the training set and evaluate it on the test set.
- Regularization: Mathematical framework for controlling the complexity of the model.

# 2 Regularization

## 2.1 Ex. Polynomial Regression

More complex models (i.e. higher order or with larger $m$) lead to larger magnitude of model parameters. Slight perturbations in input features lead to large changes in the output. We can avoid overfitting by discouraging large model parameter values.

## 2.2 Occam's Razor

"Among competing hypotheses, the one with the fewest assumptions should be selected."
In order to avoid overfitting, we should choose the simplest model that fits the data.

## 2.3 Regularized Linear Regression

- Our original loss function:

$$E(w) = \frac{1}{2}\Sigma_{i=1}^{N}(y_i - w^T x_i)^2 \tag{8}$$

- Where $x_i$ is the $i$th observation and $y_i$ is the $i$th target value.
- We can penalize the weights to augment the error function (i.e. add a penalty/regularization term):

$$E^{ridge}(w) = \frac{1}{2}\Sigma_{i=1}^{N}(y_i - w^T x_i)^2 + \frac{\lambda}{2}||w||^2, \lambda \geq 0 \tag{9}$$

- $||w||^2$ is the squared Euclidean norm (L2 norm) of the weight vector.
- E.g. for weight vector $w$:

$$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix} \tag{10}$$

$$||w||^2 = w_0^2 + w_1^2 + \cdots + w_m^2 \tag{11}$$

- $\lambda$ is the regularization parameter. It controls the strength of the regularization.
- $\lambda = 0$ is equivalent to the original loss function.
- $\lambda \to \infty$ is equivalent to setting all weights to 0.
- We can also use the L1 norm:

$$||w||_1 = |w_0| + |w_1| + \cdots + |w_m| \tag{12}$$

- This is called Lasso Regression. Other norms can be used as well.
- This would require swapping the L2 norm for the L1 norm in the loss function.