Polynomial Regression

Polynomial regression is a regular linear regression with a change of basis.

Example.
From linear model to a quadratic model:

\[
\begin{bmatrix}
1 \\
x_1 \\
x_2 \\
\vdots \\
x_p \\
\end{bmatrix}_{(p+1)\times 1} \quad \xrightarrow{\text{change of basis}} \quad \begin{bmatrix}
1 \\
x_1^2 \\
x_1x_2 \\
\vdots \\
x_1x_p \\
x_2^2 \\
x_2x_3 \\
\vdots \\
x_2x_p \\
x_p^2 \\
\end{bmatrix}_{(2p+1)\times 1}
\]

(1)

In general, if the degree is d and the number of features is p, then there will be pd+1 coefficients in a dth degree polynomial model.

Generalized Linear Model

Recall that in linear model,

\[
h_{\vec{b}}(\vec{x}) = \vec{b}^T \cdot \vec{x}.
\]

In generalized linear model,

\[
h_{\vec{b}}(\vec{x}) = \vec{b}^T \cdot \phi(\vec{x}),
\]

where \( \phi(\vec{x}) \) is the transformation of \( \vec{x} \).

In matrix form,

\[
\Phi = \begin{bmatrix}
\phi(\vec{x}_1) \\
\phi(\vec{x}_2) \\
\vdots \\
\phi(\vec{x}_n)
\end{bmatrix}.
\]

(2)

The analytic solution is

\[
\vec{b} = (\Phi^T \Phi)^{-1} (\Phi^T \vec{y}).
\]
Example.

If \( p = 1 \), \( h_\theta(x) = b_0 + b_1 x + b_2 x^2 + ... + b_d x^d = \hat{\theta}^T \cdot \phi(x) \), where \( \phi(x) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \).

**Generalization Error**

If there are \( n \) data points, they can be fit perfectly in a model with degree \( d = n - 1 \). This could lead to overfitting.

![Figure 1: Three models of different complexity on a price vs. size dataset. From CS66 lec08 slides page 11.](image)

There are trade-offs between two types of errors.

**Structure Error**

Hypothesis space is too restricted to model the true relationship.

- Having more data does not help
- Need more flexible model

**Estimation (Approximation) Error**

Hypothesis space is able to model the true relationship, but the correct model is hard to identify due to large hypothesis space, small \( n \), or noise

- Add more data
- Need to reduce hypothesis space
In practice, we favor building a complex model first and then reducing the hypothesis space to introduce more flexibility.

**Regularization**

Regularization is a technique for controlling model complexity and preventing overfitting by adding a penalty to the cost function. This effectively penalizes large coefficient terms. We prefer smaller weights for $b_j$ because doing so prevents any single feature from dominating the prediction, and helps to identify useless features that have close-to-zero weights after regularization. This is important for feature selection.

**Three Common Regularizers**

$L_0$, $L_1$, and $L_2$ are all commonly used regularizers for constraining a model. $L_2$ has several advantages over other regularizers. First, it is differentiable and thus can be more easily added to a gradient descent algorithm. Second, because its form is a sum of squared weights, and squaring values < 1 returns smaller values, small weights will be kept within the vicinity of 0 while weights > 1 will spread out, which motivates the selection of features.

**Cost Function with Regularization**

Using $L_2$ norm,

$$J(\vec{b}) = \frac{1}{2} \sum_{i=1}^{n} (\vec{b}^T \vec{x}_i - y_i)^2 + \frac{\lambda}{2} \sum_{j=1}^{p} b_j^2,$$

where $\lambda \geq 0$ is a regularization parameter. Small $\lambda$ results in more fit to the training data and consequently more error to the test data, while large $\lambda$ keeps weights small and leads to more generalization.
Stochastic Gradient Descent with Regularization

The coefficients update as follows:
\[ b_0 \leftarrow b_0 - \alpha (\vec{b}^T \vec{x}_i - y_i) \]
\[ b_j \leftarrow b_j - \alpha ((\vec{b}^T \vec{x}_i - y_i)x_{ij} + \lambda b_j) \]
\[ = (1 - \alpha \lambda)b_j - \alpha ((\vec{b}^T \vec{x}_i - y_i)x_{ij}), \]
where \( \alpha \) and \( \lambda \) are hyperparameters.

The cost function is
\[ J(\vec{b}) = (\vec{X} \vec{b} - \vec{y})^T (\vec{X} \vec{b} - \vec{y}) + \lambda \vec{b}^T \vec{b}. \]

And the analytic solution is
\[
\hat{\vec{b}} = (\vec{X}^T \vec{X} + \lambda \vec{I})^{-1} \vec{X}^T \vec{y}, \quad \text{where } \vec{I} = \begin{bmatrix}
0 & 0 & 0 & \ldots & 0 \\
0 & 1 & 0 & \ldots & 0 \\
0 & 0 & 1 & \ldots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & 1
\end{bmatrix}.
\]

Logistic Regression

In logistic regression, output is restricted to \([0, 1]\). Compared to linear regression, a logistic model is better able to capture the uncertainty and probabilistic nature of a dataset with binary outcomes. Let 1 encode Y and 0 encode N. We predict
\[
\hat{y} = \begin{cases} 
1 & \text{if } \vec{b}^T \vec{x} \geq 0.5 \\
0 & \text{otherwise}
\end{cases}
\]