ECE 469/ECE 568 Machine Learning

Textbook:
Machine Learning: a Probabilistic Perspective by Kevin Patrick Murphy

Southern Illinois University

September 18, 2024

A recap for the last lecture

We discuss cross-validation, algorithms and their implementation

- Hold-out cross-validation
- K-fold cross-validation
- Leave-one-out cross-validation (LOOCV)

Polynomial regression: Extending linear models with basis functions

- Linear models trained on non-linear functions of the input features can also be used in ML.
- This approach allow us to fit a much wider range of data while retaining the advantages of using a linear model.
- Thus, simple linear regression models can be extended by constructing polynomial features from the coefficients.
- Consider an example of using a linear model to fit two-dimensional data $\mathbf{x} = [x_1, x_2]$ as follows:

$$f(\mathbf{w}, \mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

- We may use a parabolic function of $\mathbf{x}$, yet make the model linear in $\mathbf{w}$ as follows:

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

Polynomial regression: Extending linear models with basis functions

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

- Notice that this model is still linear on the weights, and non-linear on the data.
- Hence, it can be solved via the same learning techniques that we discussed for linear models.
- This can also be seen as transforming the original 2D data-set to a new 5D data-set as

$$\mathbf{z} = [z_1 = x_1, z_2 = x_2, z_3 = x_1 x_2, z_4 = x_1^2, z_5 = x_2^2]$$

- Therefore, the underlying linear model becomes

$$f(\mathbf{w}, \mathbf{z}) = w_0 + w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5$$

- Notice that the resulting polynomial model is in the same class of linear models that we discussed earlier.

Polynomial regression: Extending linear models with basis functions

- Generally, we can exploit this technique to use linear learning models within a higher-dimensional space.
- The main benefit is that by transforming original data into a higher-dimensional space, we can make sure that the linear model has the flexibility to fit a much broader range of data.

Linear Basis Function Models for ML

- The simplest linear model for regression is a a linear combination of input variables.

$$g(\mathbf{x}, \mathbf{w}) = w_0 x_0 + w_1 x_1 + \cdots + w_M x_M \text{ with } x_0 = 1$$

- Key property: This is a linear function of the unknown parameters $\mathbf{w} = [w_1, \cdots, w_M]$
- This is also a linear function of input variables $\mathbf{x} = [x_0, \cdots, x_M]$ <– limitation in the model
- Remedy –> extend this by considering linear combinations of fixed nonlinear functions of the input variables.

$$g(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{M} w_i \phi_i(\mathbf{x})$$

- Here, $\phi_i(\mathbf{x})$ for $i = 1, \cdots, M$ are called a set of basis functions, which are typically non-linear in the input variables $\mathbf{x}$.

Linear Basis Function Models for ML

- By introducing a dummy basis function $\phi_0(\mathbf{x}) = 1$, we can rewrite the model as

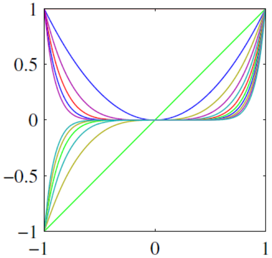
$$g(\mathbf{x}, \mathbf{w}) = \sum_{i=0}^{M} w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

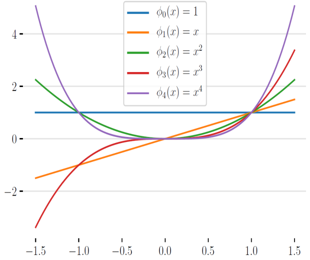
  where $\mathbf{w} = [w_0, \cdots, w_M]^T$ and $\boldsymbol{\phi}(\mathbf{x}) = [\phi_0(\mathbf{x}), \cdots, \phi_M(\mathbf{x})]^T$.

- Note that this model is called a class of linear models because this function is linear in $\mathbf{w}$.

Linear Basis Function Models for ML

- There are a number of functions that can be used as basis functions.
- If $\phi_i(x) = x^i$ for $i = 0, \cdots M$ would provide a polynomial of degree M.



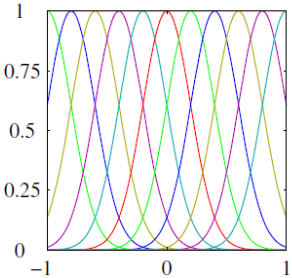


Since polynomial functions are global functions of the input variable $\mathbf{x}$, any changes in one region of input space affect all other regions. This is a limitation of the model.

Linear Basis Function Models for ML

- A class of Gaussian basis functions is defined as

$$\phi_i(x) = \exp\left(-\frac{(x-\mu_i)^2}{2s^2}\right)$$



In Gaussian basis functions, μ_j captures the location of the basis function in the input space and s accounts for the underlying spatial scale.

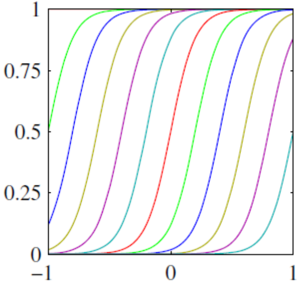
Linear Basis Function Models for ML

- The sigmoidal basis functions takes the form of

$$\phi_i(x) = \sigma\left(\frac{x-\mu_i}{s}\right)$$

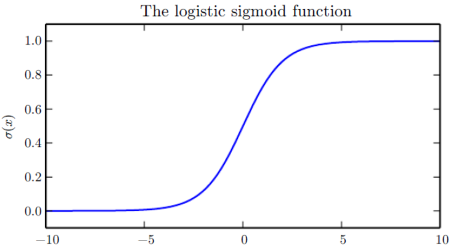
where $\sigma(\cdot)$ is the logistic sigmoid function given by

$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

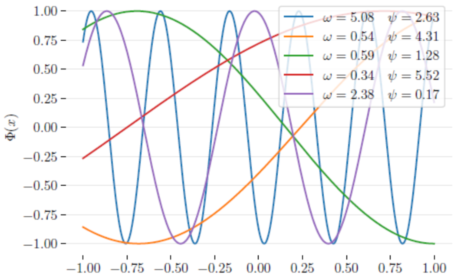


Linear Basis Function Models for ML

- Logistic sigmoid function saturates when its argument x is very positive or very negative.
- Thus, this function becomes very flat and insensitive to small changes in its input.
- Logistic sigmoid functions are used for classification algorithms in machine learning.



Linear Basis Function Models for ML



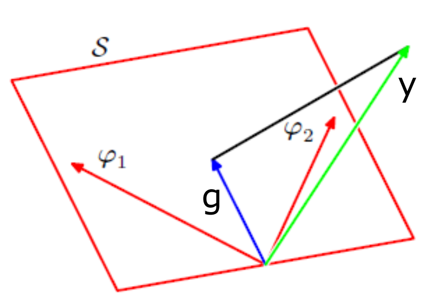
- Fourier basis functions provides an expansion in sinusoidal functions.

$$\phi_i(x) = \cos(\omega_j x + \psi_j)$$

- Typically, for regression on the interval [-1, 1], we can use a truncated Fourier series

$$\phi_i(x) = \begin{cases} 1, & i = 0 \\ \cos(\pi x i), & i \text{ even} \\ \sin(\pi x i), & i \text{ odd} \end{cases}$$

Linear Basis Function Models for ML



- Let us now consider the geometric interpretation of the least-squares solution.
- Let φ_j the jth column of $\mathbf{\Phi}$.
- Now, $\mathbf{y} = [y_1, y_2, \cdots, y_N]$ is a vector in the N-dimensional vector space.
- Note that there are $M+1$ basis functions $\phi_0, \cdots \phi_M$.
- If $M+1 < N$, then the M vectors $\phi_j(\mathbf{x}_n)$ spans a linear subspace $\mathcal{S}$ of dimensionality $M+1$.

Linear Basis Function Models for ML

- Recall that

$$g(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{M} w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

- Thus, $\mathbf{g}$ is an arbitrary linear combination of the vectors φ_j.
- The mean squared-error $\frac{1}{N}\sum_{i=1}^{N}(y_n - g(\mathbf{w}, \mathbf{x}_n))^2$ is proportional to the squared Euclidean distance between $\mathbf{g}$ and $\mathbf{y}$.
- The least-square solution tries to minimize this Euclidean distance between $\mathbf{g}$ and $\mathbf{y}$.
- Hence, the least-squares solution for $\mathbf{w}$ corresponds to that choice of $\mathbf{g}$ that lies in subspace $\mathcal{S}$ and that is closest to $\mathbf{y}$.
- This solution is indeed the orthogonal projection of $\mathbf{y}$ onto the subspace $\mathcal{S}$.

$$g^*(\mathbf{x}, \mathbf{w}) = \mathbf{w}_{\mathrm{opt}}^T \boldsymbol{\phi}(\mathbf{x}),$$

where $\mathbf{w}_{\mathrm{opt}} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \ \mathbf{y}$.

Linear Basis Function Models for ML

- The optimal weight vector is given by

$$\mathbf{w}_{\text{opt}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y}$$

where the (n, j)th element of is given by

$$[\mathbf{\Phi}]_{n,j} = \phi_j(\mathbf{x}_n)$$

- Hence, the matrix $\mathbf{\Phi}$ can be written as

$$\mathbf{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_M(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_M(\mathbf{x}_2) \\ \vdots & \vdots & \ldots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_M(\mathbf{x}_N) \end{bmatrix}$$

- $\mathbf{w}_{\text{opt}}$ can also be written as $\mathbf{w}_{\text{MLE}} = \mathbf{\Phi}^\dagger\mathbf{y}$, where $\mathbf{\Phi}^\dagger = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T$ is the Moore-Penrose pseudo-inverse of the matrix $\mathbf{\Phi}$.
- This is indeed the least-square solution and termed as normal equations for linear model with basis function in machine learning.

Regularization with linear basis function models

- Recall the idea of adding a regularization term to a cost function to control over-fitting:

$$J(\mathbf{w}) = E_{SE}(\mathbf{w}) + \lambda E_R(\mathbf{w}),$$

  where $E_{SE}(\mathbf{w})$ is the usual squared error, $E_R(\mathbf{w})$ is a regularization function, and λ is a positive regularization parameter.

- Let the regularization function be a sum-of-squares of the weight vector elements:

$$E_R(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{w}}{N} = \frac{||\mathbf{w}||^2}{N}$$

- This choice is useful as the modified error function still remains a quadratic function of $\mathbf{w}$, and hence, it has an exact minimizer.

- For the linear basis function model, the squared-error cost function becomes:

$$E_{SE}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n))^2$$

Regularization with linear basis function models

- The regularized cost/error function becomes:

$$J(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n))^2 + \lambda\frac{||\mathbf{w}||^2}{N}$$

- By computing the gradient and setting it to zero, we can obtain the optimal $\mathbf{w}^*$ that minimizes $J(\mathbf{w})$ as

$$\mathbf{w}^* = (\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{I})^{-1}\mathbf{\Phi}^T\mathbf{y}$$

- A more general regularizer will be in the form of

$$E_R(\mathbf{w}) = \sum_{j=1}^{M}|w_j|^p$$

- Then, $p = 2$ case gives us the quadratic regularizer.
- Moreover, $p = 1$ case gives us the “LASSO" regularization.

Regularization with linear basis function models

- When p is large, some of the coefficients in $\mathbf{w}$ will approach zero, and thus resulting a sparse model in which the corresponding basis functions do not play any role.
- Then we can minimize the unregularized error function $E_{SE}(\mathbf{w})$ with respect to $\mathbf{w}$ subject to a constraint $\sum_{j=1}^{M} |w_j|^p < \eta$.

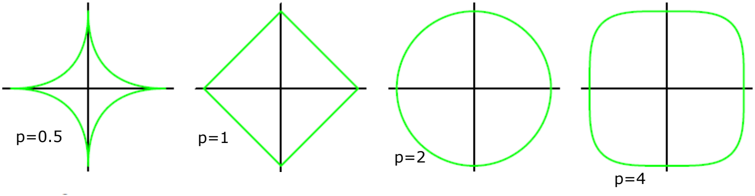
$$\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n))^2$$

$$\text{subject to } \sum_{j=1}^{M} |w_j|^p < \eta$$

- We can solve this minimizer by forming Lagrangian function and considering λ/N as the Lagrangian multiplier.

Regularization with linear basis function models

- With regularization, we can learn complex models on data sets of limited size without severe over-fitting. This is achieved by limiting the effective model complexity.
- Impact of p on the contours of the regularization term $E_R(\mathbf{w})$ is shown below.



Regularization with linear basis function models

- Impact of regularization with respect to the unconstrained error function can be depicted as follows: