CSE 847 (Spring 2016): Machine Learning— Regression
Instructor: Jiayu Zhou

1 Revisit the Polynomial Curve Fitting problem

2 Linear Basis Function

- Linear model $y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_D x_D$
- There is an obvious limitation of the linear model (what is it?), we can overcome the limitation by introducing fixed non-linear transformations, using basis functions $\phi_j(\mathbf{x})$.

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

  - Add additional dummy basis $\phi_0(\mathbf{x}) = 1$ to match bias x_0:

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\Phi}(\mathbf{x})$$

  - In the curve fitting problem we used $\phi_j(x) = x^j$, polynomial basis functions.
  - Applying basis functions can be considered as the feature engineering process, from which we obtain new (and hopefully better) features from original feature space.

- Other examples of basis functions
  - Gaussian basis functions (spline functions) $\phi_j = \exp\left\{-\frac{x-\mu_j}{2s^2}\right\}$
  - Sigmoidal basis function $\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right)$, where $\sigma(a) = \frac{1}{1+\exp(-a)}$
  - Fourier basis (Fourier transformation)

3 From Gaussian noise to Least Squares

- Assumption: observation from a deterministic function with added Gaussian noise:

$$t = y(\mathbf{x}; \mathbf{w}) + \epsilon \quad \text{where} \quad p(\epsilon|\beta) = \mathcal{N}(\epsilon|0, \beta^{-1})$$

- Equivalently: $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t - y(\mathbf{x}; \mathbf{w})|0, \beta^{-1}) = \mathcal{N}(t|y(\mathbf{x}; \mathbf{w}), \beta^{-1})$
- Given a dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{t}\}$, where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are features, and $\mathbf{t} = \{t_1, \ldots, t_N\}$ are targets.
- When samples are drawn from *i.i.d*, the probability of data is given by:

$$p(\mathcal{D}; \mathbf{w}, \beta) = p(\mathbf{X}, \mathbf{t}; \mathbf{w}, \beta) = p(\mathbf{t}|\mathbf{X}; \mathbf{w}, \beta)p(\mathbf{X})$$

- With the linear prediction function $y(\mathbf{x}_n; \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)$, the likelihood function is given by:

$$p(\mathbf{t}|\mathbf{X}; \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n | y(\mathbf{x}_n; \mathbf{w}), \beta^{-1}) = \prod_{n=1}^{N} \mathcal{N}(t_n | \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}).$$

- And the log-likelihood is:

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{X}; \mathbf{w}, \beta) &= \ln \prod_{n=1}^{N} \mathcal{N}(t_n | \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) = \sum_{n=1}^{N} \ln \mathcal{N}(t_n | \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \\ &= -\frac{N}{2} \ln(2\pi) + \frac{N}{2} \ln \beta - \beta \frac{1}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2\end{aligned}$$

- Our goal is to maximize the log-likelihood. Removing constants, we get the following:

$$\max_{\mathbf{w}, \beta} \ln p(\mathbf{t}|\mathbf{X}; \mathbf{w}, \beta) = \min_{\mathbf{w}, \beta} \frac{\beta}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2 - \frac{N}{2} \ln \beta$$

  We note the problem of $\mathbf{w}$ is independent from β:

$$\mathbf{w}_{\mathrm{ML}} = \mathrm{argmin}_{\mathbf{w}} \frac{1}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

  Plugin in $\mathbf{w}_{\mathrm{ML}}$, the optimal solution of the inverse of noise precision β is given by:

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N} \sum_{n=1}^{N} \{t_n - \mathbf{w}_{\mathrm{ML}}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2,$$

  which is the residual variance of the target values around the regression function.

- Introducing design matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N \times M}$, N samples and M features (after feature engineering):

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \dots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \dots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

  The least squares can then be represented as:

$$\min_{\mathbf{w}} \frac{1}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2 = \min_{\mathbf{w}} \left\{ \frac{1}{2} \|\boldsymbol{\Phi}\mathbf{w} - \mathbf{t}\|_2^2 = \nabla E_D \right\}.$$

4 Solving Least Squares

4.1 Unregularized Version

- Solve least squares problem using SVD: $\min_{\mathbf{w} \in \mathbb{R}^n} ||\boldsymbol{\Phi}\mathbf{w} - \mathbf{t}||_2^2$.

– Assume the SVD of $\mathbf{\Phi}$ is given by

$$\mathbf{\Phi} = (U_1 \;\; U_2) \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T,$$

where $\Sigma \in \mathbb{R}^{N \times M}$ is diagonal, $U_1 \in \mathbb{R}^{N \times M}$ and $U_2 \in \mathbb{R}^{N \times (M-N)}$ have orthonormal columns and $V \in \mathbb{R}^{M \times M}$ is orthogonal.

$$\begin{aligned} ||\mathbf{\Phi w} - \mathbf{t}||_2^2 &= \left\| (U_1 \;\; U_2) \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T \mathbf{w} - \mathbf{t} \right\|_2^2 = \left\| \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T \mathbf{w} - (U_1 \;\; U_2)^T \mathbf{t} \right\|_2^2 \\ &= \left\| \begin{pmatrix} \Sigma y \\ 0 \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right\|_2^2 = \|\Sigma y - b_1\|_2^2 + \|b_2\|_2^2, \end{aligned}$$

where $y = V^T \mathbf{w}$, $b_1 = U_1^T \mathbf{t}$, and $b_2 = U_2^T \mathbf{t}$. The optimal y is given by $y = \Sigma^{-1} b_1$. Thus, the least squares solution is given by

$$\mathbf{w} = Vy = V\Sigma^{-1} b_1 = V\Sigma^{-1} U_1^T \mathbf{t} = \sum_{i=1}^{M} \frac{u_i^T \mathbf{t}}{\sigma_i} v_i.$$

which is a weighted average of right singular vectors.

– What if $\sigma_i \to 0$?

- Alternatively solution: set the gradient of least squares to 0

$$\nabla_{\mathbf{w}} \frac{1}{2} \|\mathbf{\Phi w} - \mathbf{t}\|_2^2 = \mathbf{\Phi}^T(\mathbf{\Phi w} - \mathbf{t}) = 0 \Rightarrow \mathbf{w}^* = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{t}$$

  – Therefore $\mathbf{\Phi w}^* = \mathbf{\Phi}(\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{t}$.
  – Geometry: This leads to a projection to the space spanned by the columns of $\mathbf{\Phi}$ (why?)
  – When $\mathbf{\Phi}^T \mathbf{\Phi}$ is close to singular, we would have numerical problems.

- We call $\mathbf{\Phi}^\dagger \equiv (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T$ pseudo-inverse (generalization of inverse to non-square matrix). See for a square invertable matrix $\mathbf{\Phi}$, we have $\mathbf{\Phi}^\dagger = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T = \mathbf{\Phi}^{-1} (\mathbf{\Phi}^T)^{-1} \mathbf{\Phi}^T = \mathbf{\Phi}^{-1}$

  – When rank$(\mathbf{\Phi}) = r \leq \min(M, N)$, let $\mathbf{\Phi} = U_r \Sigma_r V_r^T$ be the economical SVD

$$\mathbf{w} = (V_r \Sigma_r U_r^T U_r \Sigma_r V_r^T)^{-1} V_r S_r U_r^T \mathbf{t} = \sum_{i=1}^{r} \frac{u_i^T \mathbf{t}}{\sigma_i} v_i$$

  – How about when $\sigma_i \to 0$?

- Since SVD is too expensive, typically we use gradient descent. Conceptually for each iteration:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \nabla E_D = \mathbf{w}^t - \eta(\mathbf{\Phi}^T \mathbf{\Phi w} - \mathbf{\Phi}^T \mathbf{t})$$

  How to efficient compute the gradient $\mathbf{\Phi}^T \mathbf{\Phi w} - \mathbf{\Phi}^T \mathbf{t}$?

  – Recall one classifical problem in dynamic programming – Matrix Chain Multiplication.
    * $A \in \mathbb{R}^{10 \times 30}, B \in \mathbb{R}^{30 \times 5}, C \in \mathbb{R}^{5 \times 60}$

- What if data is very large, and compute the gradient is nearly impossible? We can use stochastic gradient. Each time we see one data point n, we have loss function $E_n = (t_n - \mathbf{w}^{(t)T}\phi_n)^2$, then we perform gradient descent.

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \nabla E_n = \mathbf{w}^t - \eta \boldsymbol{\phi}_n(\mathbf{w}^{(t)T}\boldsymbol{\phi}_n - t_n)$$

  where $\phi_n = \boldsymbol{\phi}(\mathbf{x}_n)$

- How about data is located in different servers ($\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_s\}$)?

$$\nabla E_D = \mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{w} - \mathbf{t}) = \sum_{i=1}^{N} \boldsymbol{\phi}_i(\boldsymbol{\phi}_i^T\mathbf{w} - t_i) = \sum_{i\in\mathcal{S}_1} \boldsymbol{\phi}_i(\boldsymbol{\phi}_i^T\mathbf{w} - t_i) + \cdots + \sum_{i\in\mathcal{S}_s} \boldsymbol{\phi}_i(\boldsymbol{\phi}_i^T\mathbf{w} - t_i)$$

4.2 Multiple Outputs

- Predict K targets simultaneously $\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{W}^T\boldsymbol{\phi}(\mathbf{x})$
- Assume the observation is from a multi-response deterministic function with added Isotropic Gaussian distribution $\mathbf{t} = \mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}) + \boldsymbol{\epsilon}$, then $p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{t}|\mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}), \beta^{-1}\mathbf{I})$.
- Given a set of observations $t_1, \ldots, t_N$, we can combine these into a matrix $\mathbf{T} \in \mathbb{R}^{N\times K}$, and the likelihood function is given by

$$\begin{aligned}\ln p(\mathbf{T}|\mathbf{X}, \mathbf{W}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(\mathbf{t}_n|\mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\mathbf{I}) \\ &= \frac{NK}{2}\ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2}\sum_{n=1}^{N}\|\mathbf{t}_n - \mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}_n)\|^2\end{aligned}$$

- Maximize the log-likelihood gives $\mathbf{W}_{\mathrm{ML}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{T}$.
- Compare the solution from solving a multi-response least sqaures, and the solution from solving a set of least squares independently. Are they the same?

4.3 Regularization

- Adding regularization to control overfitting

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{\Phi}\mathbf{w} - \mathbf{t}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 \tag{1}$$

- Setting gradient to zero, and $\mathbf{\Phi} = USV^T = U\begin{bmatrix}\Sigma \\ 0\end{bmatrix}V$

$$\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{w} - \mathbf{\Phi}^T\mathbf{t} + \lambda\mathbf{w} = 0 \Rightarrow \mathbf{w} = V(\Sigma^2 + \lambda I)^{-1}SU^T\mathbf{t} \Rightarrow \mathbf{w} = \sum_{i=1}^{M}\frac{\sigma_i u_i^T\mathbf{t}}{\sigma_i^2 + \lambda}v_i$$

  Note: V has to be square matrix ($\mathbb{R}^{M\times M}$) in order to be an orthogonal matrix (and thus $I = VV^T$).

- Probabilistic interpretation $p(\mathbf{w}|\mathbf{t},\mathbf{X},\beta,\lambda) \propto p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta)p(\mathbf{w}|\lambda)$

$$p(\mathbf{w},\mathbf{t}|\mathbf{X},\beta,\lambda) = p(\mathbf{w}|\mathbf{t},\mathbf{X},\beta,\lambda)p(\mathbf{t}) = p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta)p(\mathbf{w}|\lambda)$$

- A general regularization framework

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{\Phi}\mathbf{w}-\mathbf{t}\|_2^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|\mathbf{w}_j|^q = \min_{\mathbf{w}} \frac{1}{2}\|\mathbf{\Phi}\mathbf{w}-\mathbf{t}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_q^q$$

For ridge regression $q = 2$ and Lasso regression $q = 1$. When $q < 1$ the problem is non-convex (more sparsity)

- Interpretation of regularizations via constrained optimization problem
  - From regularized problems to constrained problems. For convex problems, a regularized problem has an equivalent constrained problem.
  - Sparsity from ℓ_p-norm constrained problems: when $p \leq 0$.

5 Decision Theory for Regression

- Choosing a specific estimate $y(\mathbf{x}) = y(\mathbf{x};\mathbf{w})$ of the value of t for each input $\mathbf{x}$, given the squared loss $L(t, y(\mathbf{x})) = \{t - y(\mathbf{x})\}^2$

$$\mathbb{E}[L] = \iint L(t, y(\mathbf{x}))p(\mathbf{x},t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t = \iint \{t - y(\mathbf{x})\}^2 p(\mathbf{x},t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t$$

We will need to find the optimal $y(x)$ using the loss function

$$\min_y \mathbb{E}[L] = \min_y \iint \{t - y(\mathbf{x})\}^2 p(\mathbf{x},t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t$$

To obtain the optimal prediction function $y(x)$, we set

$$\begin{aligned}
&\frac{\delta\mathbb{E}[L]}{\delta y} = 2\int \{y(\mathbf{x}) - t\}p(\mathbf{x},t)\ \mathrm{d}t = 0 \\
\Rightarrow &\int y(\mathbf{x})p(\mathbf{x},t)\ \mathrm{d}t - \int tp(\mathbf{x},t)\ \mathrm{d}t = 0 \\
\Rightarrow &y(\mathbf{x})p(\mathbf{x}) - \int tp(\mathbf{x},t)\ \mathrm{d}t = 0 \\
\Rightarrow &y(\mathbf{x}) = \frac{\int tp(\mathbf{x},t)\ \mathrm{d}t}{p(\mathbf{x})} = \mathbb{E}_t[t|\mathbf{x}]
\end{aligned}$$

The optimal prediction function given by the sqaured loss is the conditional expectation of t.