Machine learning

Linear regression

Hamid Beigy

Sharif University of Technology

February 25, 2023

Table of contents

1. Introduction
2. Linear regression
3. Model selection
4. Sample size
5. Probabilistic Linear Regression
6. Overfitting
7. Regularization
8. Maximum a posteriori and regularization
9. Geometric Interpretation
10. Sequential learning
11. Multiple outputs regression
12. Bias-Variance trade-off

Introduction

Introduction

1. In regression, $c(x)$ is a continuous function. Hence the training set is in the form of

$$S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}, t_k \in \mathbb{R}.$$

2. When there is no noise, the task is interpolation and our goal is to find a function $f(x)$ that passes through these points. Hence

$$t_k = f(x_k) \qquad \forall k = 1, 2, \ldots, N$$

3. In polynomial interpolation, given N points, we find $(N-1)$st degree polynomial to predict the output for any x.
4. If x is outside of the range of the training set, the task is called extrapolation.
5. In regression, there is noise added to the output of the unknown function.

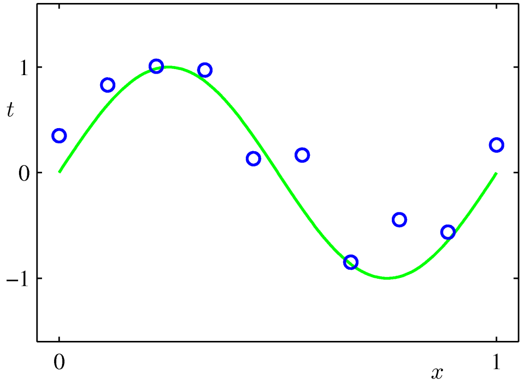
$$t_k = f(x_k) + \epsilon \qquad \forall k = 1, 2, \ldots, N$$

$f(x_k) \in \mathbb{R}$ is the unknown function and ϵ is the random noise.

Introduction(cont.)

1. In regression, there is noise added to the output of the unknown function.

$$t_k = f^*(x_k) + \epsilon \qquad \forall k = 1, 2, \dots, N$$



2. The explanation for the noise is that there are extra hidden variables, z_k, that we cannot observe.

$$t_k = f^*(x_k, z_k) + \epsilon \qquad \forall k = 1, 2, \dots, N$$

Linear regression

Linear regression

1. Our goal is to approximate the output by function $g(x)$.
2. The empirical error on the training set S is measured using loss/error/cost function.
   - Squared Error from target

$$\boxed{E_E(g(x_i)|S) = (t_i - g(x_i))^2 .}$$

   - Linear error from target

$$\boxed{E_E(g(x_i)|S) = |t_i - g(x_i)|.}$$

   - Mean square error from target

$$\boxed{E_E(g(x_i)|S) = \frac{1}{N}\sum_{i=1}^{N}(t_i - g(x_i))^2 .}$$

   - Sum of square error from target

$$\boxed{E_E(g(x_i)|S) = \frac{1}{2}\sum_{i=1}^{N}(t_i - g(x_i))^2 .}$$

3. The aim is to find $g(.)$ that minimizes the empirical error.
4. We assume that a hypothesis class for $g(.)$ with a small set of parameters such as linear function

$$\boxed{g(x) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_Dx_D}$$

Linear regression(cont.)

1. In the linear regression, when $D = 1$

$$\boxed{g(x) = w_0 + w_1 x}$$

2. Parameters w_0 and w_1 should minimize the empirical error

$$\boxed{E_E(w_0, w_1|S) = E_E(g(x)|S) = \frac{1}{2}\sum_{i=1}^{N}\left[t_k - (w_0 + w_1 x_k)\right]^2}$$

3. Taking derivative of error w.r.t w_0 and w_1 and setting equal to zero

$$\begin{aligned} w_0 &= \bar{t} - w_1\bar{x} \\ w_1 &= \frac{\sum_k t_k x_k - \bar{x}\bar{t}N}{\sum_k (x_k)^2 - N(\bar{x})^2} \\ \bar{t} &= \frac{\sum_{k=1}^{N} t_k}{N} \\ \bar{x} &= \frac{\sum_{k=1}^{N} x_k}{N} \end{aligned}$$

Linear Regression(cont.)

1. When the input variables form a $D-$dimensional vector, the linear regression model is

$$\boxed{g(x) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_Dx_D.}$$

2. Parameters $w_0, w_1, \ldots, w_D$ should minimize the empirical error

$$\boxed{E_E(g(x)|S) = \frac{1}{2}\sum_{i=1}^{N}(t_i - g(x_i))^2.}$$

3. Taking derivative of error w.r.t ws and setting equal to zero,

Linear Regression(cont.)

1. Taking derivative of error w.r.t *w*s and setting equal to zero,

$$\sum_{k=1}^{N} t_k = Nw_0 + w_1 \sum_{k=1}^{N} x_{k1} + w_2 \sum_{k=1}^{N} x_{k2} + \ldots + w_D \sum_{k=1}^{N} x_{kD}$$

$$\sum_{k=1}^{N} x_{k1} t_k = w_0 \sum_{k=1}^{N} x_{k1} + w_1 \sum_{k=1}^{N} (x_{k1})^2 + w_2 \sum_{k=1}^{N} x_{k1} x_{k2} + \ldots + w_D \sum_{k=1}^{N} x_{k1} x_{kD}$$

$$\sum_{k=1}^{N} x_{k2} t_k = w_0 \sum_{k=1}^{N} x_{k2} + w_1 \sum_{k=1}^{N} x_{k1} x_{k2} + w_2 \sum_{k=1}^{N} (x_{k2})^2 + \ldots + w_D \sum_{k=1}^{N} x_{k2} x_{kD}$$

$$\vdots$$

$$\sum_{k=1}^{N} x_{kD} t_k = w_0 \sum_{k=1}^{N} x_{kD} + w_1 \sum_{k=1}^{N} x_{k1} x_{kD} + w_2 \sum_{k=1}^{N} x_{k2} x_{kD} + \ldots + w_D \sum_{k=1}^{N} (x_{kD})^2$$

Linear Regression(cont.)

1. Define the following vectors and Matrix
   - Data matrix

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1D} \\ 1 & x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & & & \vdots & \\ 1 & x_{N1} & x_{N2} & \dots & x_{DD} \end{bmatrix}$$

   - The k^{th} input vector

$$\mathbf{x}_k = (1, x_{k1}, x_{k2}, \dots, x_{kD})^\top$$

   - The weight vector

$$\mathbf{w} = (w_0, w_1, w_2, \dots, w_D)^\top$$

   - The target vector

$$t = (t_1, t_2, t_3, \dots, t_N)^\top$$

Linear Regression(cont.)

1. The empirical error is equal to

$$\boxed{E_E(g(x)|S) = \frac{1}{2}\sum_{k=1}^{N}\left(t_k - \mathbf{w}^\top \mathbf{x}_k\right)^2.}$$

2. The gradient of $E_E(g(x)|S)$ is

$$\boxed{\begin{aligned}\nabla_{\mathbf{w}} E_E(g(x)|S) &= \sum_{k=1}^{N}\left(t_k - \mathbf{w}^\top \mathbf{x}_k\right)\mathbf{x}_k^\top \\ &= \sum_{k=1}^{N} t_k \mathbf{x}_k^\top - \mathbf{w}^\top \sum_{k=1}^{N} \mathbf{x}_k \mathbf{x}_k^\top = 0\end{aligned}}$$

3. Solving for W, we obtain

$$\boxed{\mathbf{w}^* = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{t}.}$$

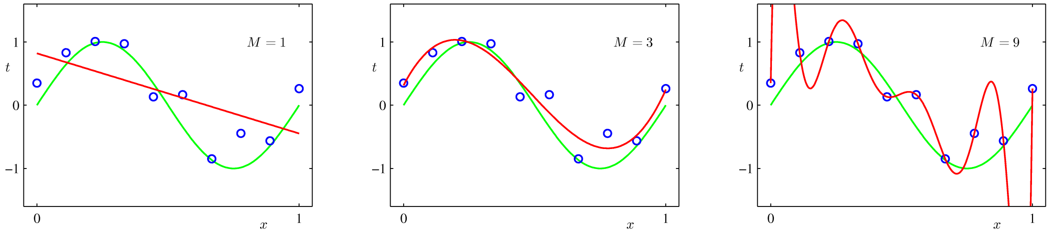
4. When $\mathbf{X}^\top\mathbf{X}$ is invertible, the problem has a unique solution.
5. When $\mathbf{X}^\top\mathbf{X}$ is not invertible, the pseudo inverse is used and the problem has several solution.

Regression(cont.)

1. If the linear model is too simple, the model can be a polynomial (a more complex hypothesis set)

$$g(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M.$$

2. M is the order of the polynomial.
3. Choosing the right value of M is called model selection.

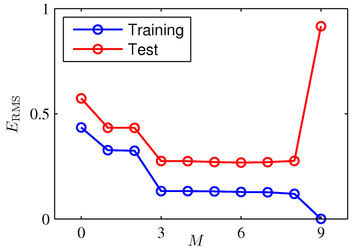


4. For $M = 1$, we have a too general model
5. For $M = 9$, we have a too specific model

Model selection

Regression(Model selection)

1. The goal of model selection is to achieve good generalization by making accurate predictions for new data.
2. The generalization ability of a model is measured by a separate test data generated using exactly the same process used for generating training data.
3. The model is chosen using a validation data set.



4. Two models sometimes are compared using root mean square (RMS) error.

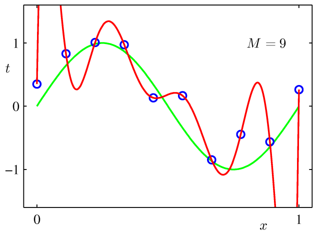
$$\boxed{E_{RMS} = \sqrt{2E_E(W^*|S)/N}}$$

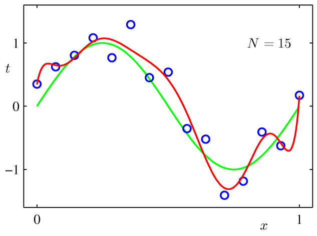
5. This allows comparison on different sizes of data sets.

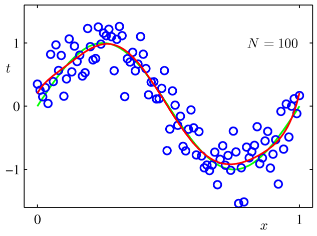
Sample size

Regression(Sample size)

1. For a given model complexity, the over-fitting problem become less severe as the size of the data set increases.







Linear Regression(cont.)

1. We can extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables, of the form

$$g(x) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(x)$$

   - $\phi_j(x)$ are known as basis functions.
   - M is total number of parameters.
   - w_0 is called bias parameter.

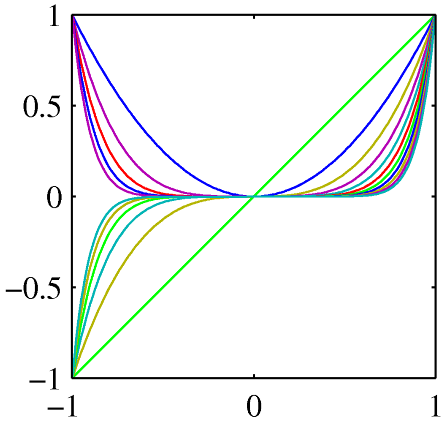
2. Usually a dummy basis function $\phi_0(x) = 1$ is used

$$g(x) = \sum_{j=0}^{M} w_j \phi_j(x) = W^\top \Phi(x)$$

3. $\mathbf{w} = (w_0, w_1, \dots, w_{M-1})^\top$ and $\mathbf{\Phi} = (\phi_0, \phi_1, \dots, \phi_{M-1})^\top$.

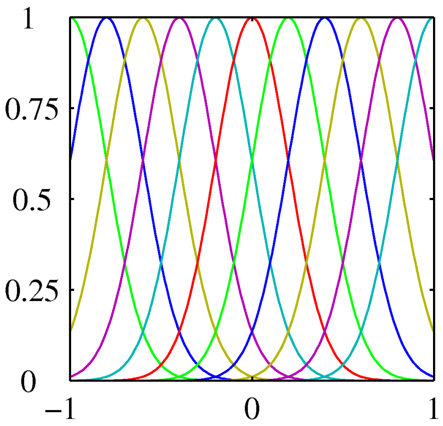
Linear Regression(cont.)

1. In pre-processing phase, the features can be expressed in terms of the basis functions $\{\phi_j(x)\}$.
2. Examples of basis functions
   - Polynomial basis function $\phi_j(x) = x^j$.



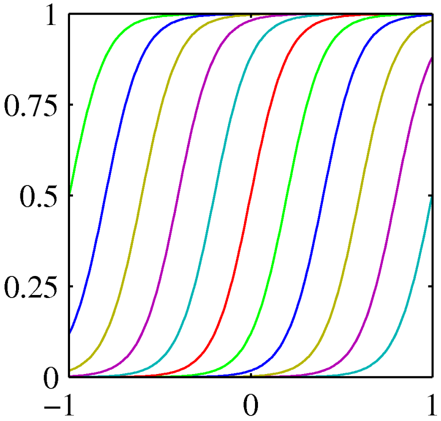
Linear Regression(cont.)

1. Examples of basis functions
   - Gaussian basis function $\phi_j(x) = exp\left\{\frac{(x-\mu_j)^2}{2s^2}\right\}$.
   - μ_j is location of the basis function.
   - s is the spatial scale of the basis function.



Linear Regression(cont.)

1. Examples of basis functions
   - Logistic basis function $\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right)$.
   - $\sigma(a) = \frac{1}{1+exp(-a)}$.



2. Fourier basis function
3. Wavelets basis function

Linear Regression(cont.)

1. The empirical error is equal to

$$E_E(g(x)|S) = \frac{1}{2}\sum_{k=1}^{N}\left(t_k - W^\top \Phi(X_k)\right)^2.$$

2. The gradient of $E_E(g(x)|S)$ is

$$\begin{aligned}\nabla_W E_E(g(x)|S) &= \sum_{k=1}^{N}\left(t_k - W^\top \Phi(X_k)\right)\Phi(X_k)^\top \\ &= \sum_{k=1}^{N} t_k \Phi(X_k)^\top - W^\top \sum_{k=1}^{N}\Phi(X_k)\Phi(X_k)^\top = 0\end{aligned}$$

3. Solving for W, we obtain $W^* = \left(\Phi^\top \Phi\right)^{-1}\Phi^\top t$.

$$\Phi = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_{M-1}(x_2) \\ \vdots & & & \vdots & \\ \phi_0(x_N) & \phi_1(x_N) & \phi_2(x_N) & \dots & \phi_{M-1}(x_N) \end{bmatrix}$$

Linear Regression(cont.)

1. The quantity $\Phi^{\dagger} = \left(\Phi^{\top}\Phi\right)^{-1}\Phi^{\top}$ is known as Moore-Penrose pseudo-inverse.
2. The bias value (w_0)

$$w_0 = \frac{1}{N}\sum_{k=1}^{N}\left\{t_k - \sum_{j=1}^{M-1} w_j\phi_j(X_k)\right\}.$$

   and equals to the difference between target values and the weighted sum of the basis function values.

Probabilistic Linear Regression

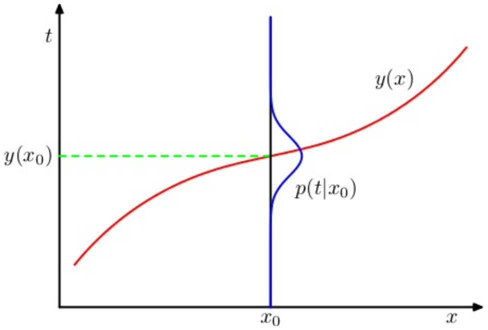
Probabilistic Linear Regression

1. Given N training examples $S = \{(x_1, t_1), \dots, (x_N, t_N)\}$, $x_k \in \mathbb{R}^D$, output $t_k \in \mathbb{R}$.
2. In probabilistic linear regression, outputs t_k's are generated from a probabilistic model.
3. We assume that the target variable t is given by a deterministic function $f(x)$ with additive Gaussian noise so that

$$t = f(x) + \epsilon$$

   where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is a zero mean Gaussian random variable with precision (inverse variance) β.
4. Thus, t is also a Gaussian random variable with mean t and precision β.

$$p(t|x) = \mathcal{N}(t|f(x), \beta^{-1}).$$



Probabilistic Linear Regression

1. Since training examples are drawn i.i.d from the same distribution, the likelihood equals

$$L(W) = \prod_{k=1}^{N} \mathcal{N}(t_k | g(x; w), \beta^{-1}) = \prod_{k=1}^{N} \frac{\sqrt{\beta}}{\sqrt{2\pi}} exp\left\{ -\frac{\beta\left(t_k - g(x_k)\right)^2}{2} \right\}.$$

2. The log-likelihood becomes

$$\ln L(W) = \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \frac{\beta}{2}\sum_{k=1}^{N}\left(t_k - g(x_k)\right)^2.$$

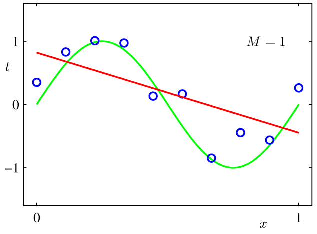
3. Clearly, maximizing $\ln L(W)$ equals to minimizing $-\ln L(W)$.
4. Removing constant terms yields

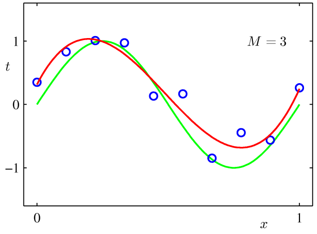
$$W_{ML} = \underset{W}{argmin} \quad \frac{1}{2}\sum_{k=1}^{N}\left(t_k - g(x_k)\right)^2.$$

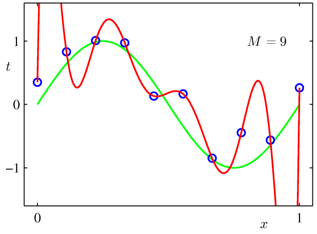
Overfitting

Over-fitting

1. Over-fitting occurs when a model begins to memorize training data rather than learning to generalize from a trend.
   - Too many parameters relative to the amount of training data
   - For example, an order-N polynomial can be exact fit to $N+1$ data points.

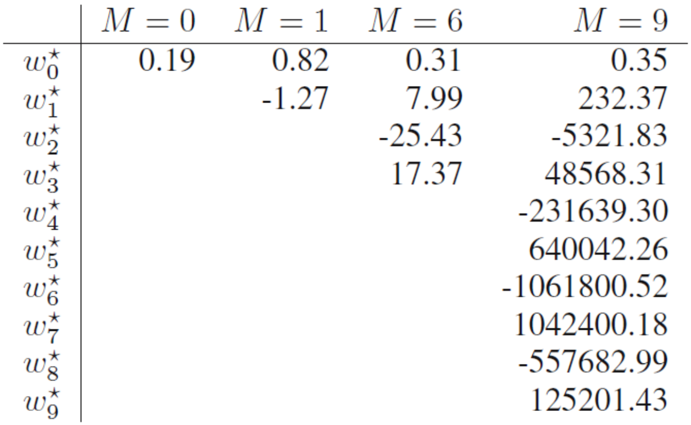






Avoiding over-fitting

1. Ways of detecting/avoiding over-fitting.
    - Use more training data
    - Evaluate on a parameter tuning set
    - Regularization
    - Take a Bayesian approach
2. In a Linear Regression model, overfitting is characterized by large parameters.
3. As M increase, the magnitude of the coefficients typically gets larger.



	$M = 0$	$M = 1$	$M = 6$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Regularization

Regularization

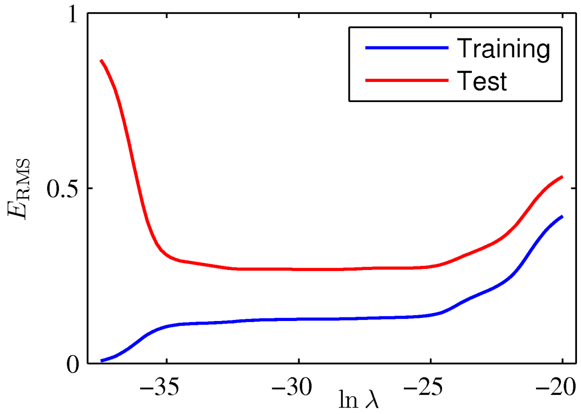
1. Introduce a penalty term for the size of the weights.
   - Unregularized Regression
$$E_E(g(x)|S) = \frac{1}{2}\sum_{i=1}^{N}(t_i - g(x_i))^2 .$$
   - Regularized Regression
$$E_E(g(x)|S) = \frac{1}{2}\sum_{i=1}^{N}(t_i - g(x_i))^2 + \lambda\Omega(W).$$
   - L_2-Regularization or Ridge Regularization
$$E_E(g(x)|S) = \frac{1}{2}\sum_{i=1}^{N}(t_i - g(x_i))^2 + \frac{\lambda}{2}||W||^2.$$
   - Large λ leads to higher complexity penalization

Regularization (cont.)

1. Least squares regression with L_2-regularization

$$\begin{aligned} \nabla_W E_E(g(x)|S) &= \nabla_W \left(\frac{1}{2} \sum_{i=1}^{N} (t_i - g(x_i))^2 + \frac{\lambda}{2} ||W||^2 \right) \\ (X^\top X + \lambda I) W &= X^\top t \\ W^* &= (X^\top X + \lambda I)^{-1} X^\top t \end{aligned}$$

2. Graph of the root-mean-square error (RMSE) versus $\ln \lambda$ for the $M = 9$ polynomial.



Regularization (cont.)

1. L_2-Regularization has Closed form solution and can be solved in polynomial time

$$E_E(g(x)|S) = \frac{1}{2}\sum_{i=1}^{N}(t_i - g(x_i))^2 + \frac{\lambda}{2}||W||^2.$$

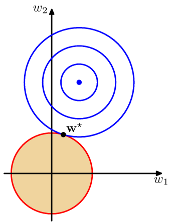
2. L_1-Regularization can be approximated in polynomial time

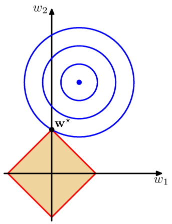
$$E_E(g(x)|S) = \frac{1}{2}\sum_{i=1}^{N}(t_i - g(x_i))^2 + \lambda||W||_1.$$

3. L_0-Regularization is NP-complete optimization problem

$$E_E(g(x)|S) = \frac{1}{2}\sum_{i=1}^{N}(t_i - g(x_i))^2 + \lambda\sum_{j=1}^{M-1}\delta(w_j \neq 0).$$

The L_0-norm represents the optimal subset of features needed by a Regression model.



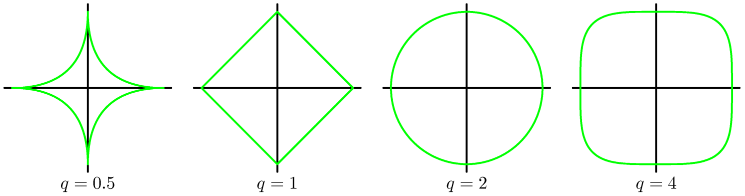


Regularization (cont.)

1. A more general regularizer is sometimes used, for which the regularized error takes the form

$$E_E(g(x)|S) = \frac{1}{2}\sum_{i=1}^{N}(t_i - g(x_i))^2 + \frac{\lambda}{2}\sum_{j=1}^{M-1}|w_j|^q.$$

2. When $q = 2$, it is called Ridge regularizer
3. When $q = l$, it is called Lasso regularizer



Maximum a posteriori and regularization

Maximum a posteriori estimation

1. Assume that W is
$$p(W) = \mathcal{N}(0, \sigma_0^2 I_D).$$
I_D denotes the $D \times D$ identity matrix.
2. This is equivalents to assume that the prior selects each component of W independently from a $\mathcal{N}(0, \sigma_0^2)$.
3. This prior density can be written as
$$p(W) = \frac{1}{(2\pi)^{D/2}\sigma_0^D} exp\left\{-\frac{1}{2\sigma_0^2}||W||_2^2\right\}.$$
4. Assume that noise precision is known.
5. The posterior density of W given set S takes the form
$$L(W) \propto exp\left(-\frac{1}{2\sigma_0^2}||W||_2^2\right) \times \prod_{k=1}^{N} exp\left(-\frac{\beta}{2}(t_k - g(x_k))^2\right).$$

Maximum a posteriori estimation

1. The log-likelihood becomes

$$\ln L(W) = -\frac{1}{2\sigma_0^2}||W||_2^2 - \frac{\beta}{2}\sum_{k=1}^{N}(t_k - g(x_k))^2 + const.$$

2. Clearly, maximizing $\ln L(W)$ equals to minimizing $-\ln L(W)$.
3. Removing constant terms yields

$$W_{MAP} = \underset{W}{argmin} \quad \frac{1}{2}\sum_{k=1}^{N}(t_k - g(x_k))^2 + \frac{1}{2\sigma_0^2\beta}||W||_2^2.$$

4. Setting $\lambda = \frac{1}{\sigma_0^2\beta}$, results in L_2- regularization.

Geometric Interpretation

Geometry of least squares

1. Consider an $N-$dimensional space whose axes are given by t_i, where $t = (t_1, t_2, \ldots, t_N)^\top$ is a vector in this space.
2. Each basis function $\phi_j(x_n)$ is evaluated at the N data points can also be represented as a vector in the same space. $\phi_j(x_n)$ corresponds to j^{th} column of Φ, i.e. $\phi_j(x_n) = (\phi_j(x_1), \phi_j(x_2), \ldots, \phi_j(x_N))^\top$.
3. Assume that $M < N$, then M vectors $\phi_j(x_n)$ will span a linear subspace of dimensionality M which embedded in N dimensions.
4. Let $y \in \mathbb{R}^N$ be a vector whose nth element is given by $g(x_n)$.
5. Hence, there exists some weight vector W such that

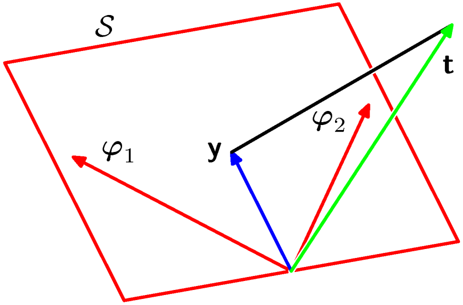
$$\hat{t} = \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \\ \vdots \\ \hat{t}_N \end{bmatrix} = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \phi_2(x_1) & \ldots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \phi_2(x_2) & \ldots & \phi_{M-1}(x_2) \\ \vdots & & & \vdots & \\ \phi_0(x_N) & \phi_1(x_N) & \phi_2(x_N) & \ldots & \phi_{M-1}(x_N) \end{bmatrix} \begin{bmatrix} W_0 \\ W_1 \\ \vdots \\ W_N \end{bmatrix} = \Phi W$$

6. We seek a $\hat{t} \in \mathbb{R}^N$ that lies in this linear subspace and is close as possible to t, i.e,

$$\hat{t} = \underset{y \in subspace(\Phi(X))}{argmin} ||t - y||_2$$

Geometry of least squares

1. The least square solution for W corresponds to that choice of y that lies in this subspace and is close to t.



2. In order to minimize the norm of the residual error $(t - y)$, we want the residual error vector be orthogonal to every column of $\Phi(X)$ so that $\phi_j(x_n)^\top(t - y) = 0$, for $j = 0, 1, 2, \ldots, M - 1$. (why?)

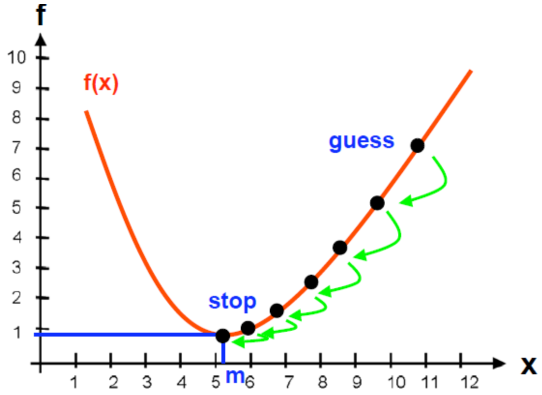
$$\begin{aligned} \phi_j(x_n)^\top(t - y) = 0 \quad &\Rightarrow \quad \Phi(X)^\top(t - \Phi(X)W) = 0 \\ &\Rightarrow \quad W = (\Phi(X)^\top\Phi(X))^{-1}\Phi(X)t. \end{aligned}$$

3. Hence, our projected value corresponds to an orthogonal projection of t onto column space of this subspace.

Sequential learning

Sequential learning

1. Batch techniques, such as the maximum likelihood solution, which involve processing the entire training set in one go, can be computationally costly for large data sets.
2. When data set is sufficiently large, it may be worthwhile to use sequential algorithms.
3. In sequential algorithms, the data points are considered one at a time, and the model parameters updated after each such presentation.
4. We want to choose W so as to minimize $E_E(W|S)$.
5. The algorithm starts with some initial guess for W and repeatedly updates W to make $E_E(W|S)$ smaller until hopefully converges to a value of W that minimizes $E_E(W|S)$.
6. This method is called gradient descent and is shown graphically for 1-dimensional problem.



Sequential learning (cont.)

1. Using the gradient descent rule, we have
$$W_j^{(k+1)} = W_j^{(k)} - \eta \nabla_W E_E(W|S).$$
2. η called learning rate and controls the step size of movement.
3. There are two methods for updating weights
   - Batch gradient descent method
   $$W_j^{(k+1)} = W_j^{(k)} - \eta \nabla_W E_E(W|S).$$
   - Stochastic gradient descent method
   $$\begin{aligned} W_j^{(k+1)} &= W_j^{(k)} - \eta \nabla_W E_E(W|(x_n, t_n)) \\ &= W_j^{(k)} + \eta\,(t_n - g(x_n))\,\Phi(x_n) \end{aligned}$$
   This rule is called least-mean-squares (LMS) algorithm and also known as the Widrow-Hoff learning rule.

Multiple outputs regression

Multiple outputs regression

1. We have considered the case of a single target variable t.
2. In some applications, we may wish to predict $K > 1$ target variables, which we denote collectively by the target vector $\mathbf{t}$.
3. This could be done by introducing a different set of basis functions for each component of $\mathbf{t}$, leading to multiple, independent regression problems.
4. A more interesting, and more common, approach is to use the same set of basis functions to model all of the components of the target vector so that

$$g(x) = W^\top \phi(x).$$

   - $g(x)$ is a $K-$dimensional column vector.
   - W is an $M \times K$ matrix of parameters.
   - $\Phi(x)$ is an $M-$dimensional column vector with elements $\phi_j(x)$, with $\phi_0(x) = 1$.

5. Suppose we take the conditional distribution of the target vector to be an isotropic Gaussian of the form

$$p(\mathbf{t}|\mathbf{x}, \beta) = \mathcal{N}(\mathbf{t}|\mathbf{W}^\top \phi(\mathbf{x}), \beta^{-1}\mathbf{I}).$$

6. If we have a set of observations $t_1, \ldots, t_N$, we can combine these into a matrix T of size $N \times K$ such that the n^{th} row is given by $t_n^\top$.
7. We can also combine the input vectors $x_1, \ldots, x_N$ into a matrix X.

Multiple outputs regression

1. Since training examples are drawn i.i.d from the same distribution, the likelihood equals

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

2. We can maximize this function with respect to W, giving

$$W_{ML} = (\Phi^\top \Phi)^{-1} \Phi^\top T.$$

3. If we examine this result for each target variable t_k, we have

$$W_k = (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{t_k}.$$

   $\mathbf{t_k}$ is an $N-$dimensional column vector with components t_{nk} for $n = 1, 2, \ldots, N$.

4. The solution to the regression problem decouples between the different target variables, and we need only compute a single pseudo-inverse matrix, which is shared by all of the vectors W_k.

Bias-Variance trade-off

Bias-Variance trade-off

1. In regression, random noise ϵ is added to the output of the unknown function $f(x)$.

$$t_k = f(x_k) + \epsilon \qquad \forall k = 1, 2, \ldots, N$$

2. Given a set of training examples,

$$S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}, t_k \in \mathbb{R}.$$

   we fit a hypothesis $g(x)$ to the data to minimize the following error function.

$$E_E(g(x)|S) = \frac{1}{2}\sum_{i=1}^{N}(t_i - g(x_i))^2 .$$

3. Now, given a new data point x, we would like to understand the expected prediction error

$$\mathbb{E}\left[(t - g(x))^2\right] .$$

4. Assume that x generated by the same process as the training set. We decompose $\mathbb{E}\left[(t - g(x))^2\right]$ into bias, variance, and noise.

Bias-Variance trade-off

1. From definition of variance we have

$$\mathbb{E}\left[(x-\mu_x)^2\right] = \mathbb{E}\left[x^2\right] - \mu_x^2$$

2. Hence we have

$$\mathbb{E}\left[x^2\right] = \mathbb{E}\left[(x-\mu_x)^2\right] + \mu_x^2$$

3. The error for new input x can be decomposed as

$$\begin{aligned}
\mathbb{E}\left[(t-g(x))^2\right] &= \mathbb{E}\left[(t-f(x)+f(x)-g(x))^2\right] \\
&= \mathbb{E}\left[(t-f(x))^2\right] + \mathbb{E}\left[(f(x)-g(x))^2\right] + 2\,\mathbb{E}\left[(t-f(x))(f(x)-g(x)\right] \\
&= \mathbb{E}\left[(t-f(x))^2\right] + \mathbb{E}\left[(f(x)-g(x))^2\right]
\end{aligned}$$

4. The last term is zero. (Why? Show it.)
5. Since $t = f(x) + \epsilon$, Hence $\epsilon = t - f(x)$, and we have

$$\begin{aligned}
\mathbb{E}\left[(t-f(x))^2\right] + \mathbb{E}\left[(f(x)-g(x))^2\right] &= \mathbb{E}\left[\epsilon^2\right] + \mathbb{E}\left[(f(x)-g(x))^2\right] \\
&= Var(\epsilon) + \mathbb{E}\left[(f(x)-g(x))^2\right]
\end{aligned}$$

Bias-Variance trade-off

1. Now consider the last term, we have

$$\begin{aligned}\mathbb{E}\left[(f(x)-g(x))^2\right] &= \mathbb{E}\left[(f(x)-\mathbb{E}[g(x)]+\mathbb{E}[g(x)]-g(x))^2\right] \\ &= \mathbb{E}\left[(f(x)-\mathbb{E}[g(x)])^2\right]+\mathbb{E}\left[(\mathbb{E}[g(x)]-g(x))^2\right] \\ &+ 2\,\mathbb{E}[(f(x)-\mathbb{E}[g(x)])(\mathbb{E}[g(x)]-g(x))]\end{aligned}$$

2. The first term is the squared bias.
3. The second term is the variance and equals to $(\mathbb{E}[g(x)]-g(x))^2 = \mathbb{E}\left[g(x)^2\right]-\mathbb{E}[g(x)]^2$.
4. The last term is zero. (Why? Show it.)
5. Hence, we have

$$\begin{aligned}\mathbb{E}\left[(t-g(x))^2\right] &= Bias^2(g(x))+Var(g(x))+Var(\epsilon) \\ &= Bias^2(g(x))+Var(g(x))+\beta^{-1}\end{aligned}$$

Bias-Variance trade-off

1. In summary, we have

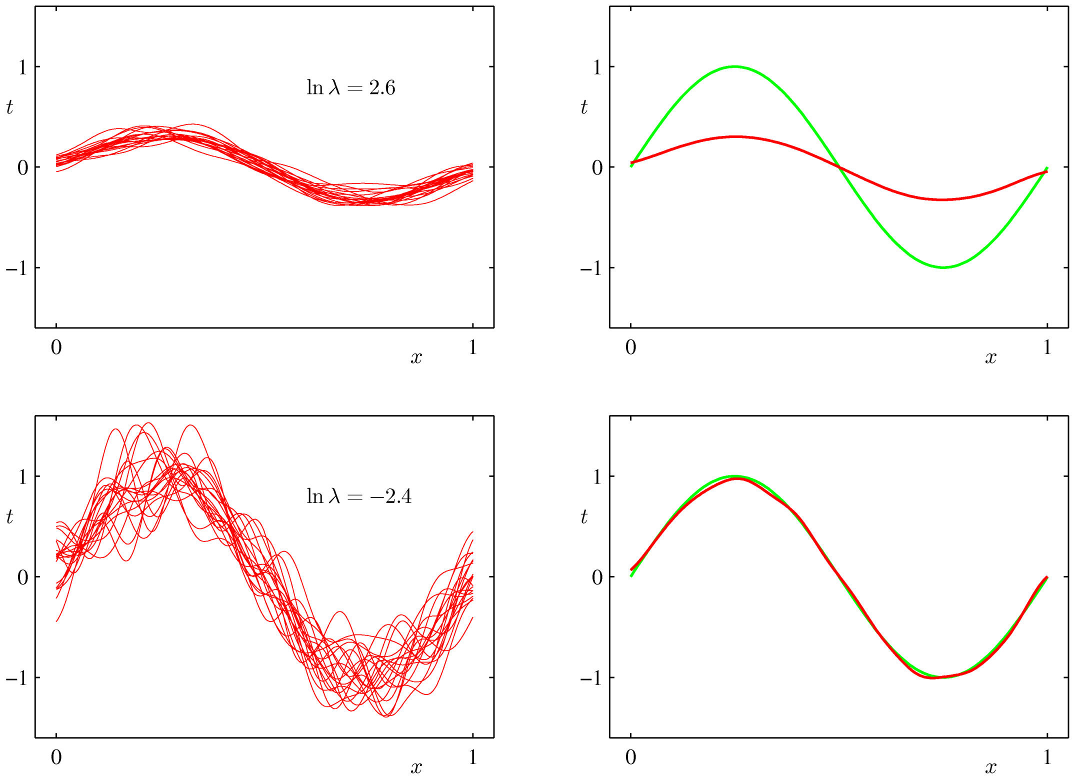
$$\mathbb{E}\left[(t - g(x))^2\right] = Bias^2(g(x)) + Var(g(x)) + \sigma^2$$

2. The first term describes the average error of $g(x)$.
3. The second term quantifies how much $g(x)$ deviates from one training set S to another one. This depends on both the estimator and the training set.This term is consequence of over-fitting.
4. The last term is the variance of the added noise. This error cannot be removed no matter what estimator we use. Note that the variance of the noise can not be minimized.

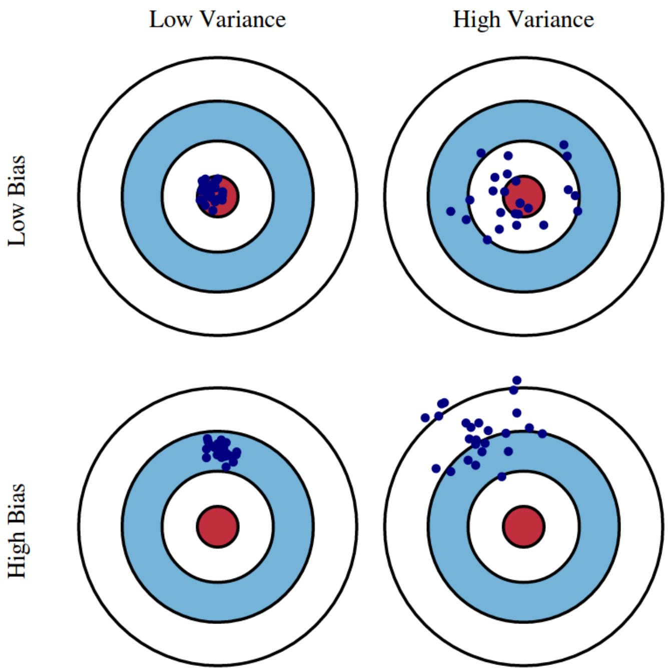
Bias-Variance trade-off

1. In context of polynomial regression, as M increases, a small changes in the data sets causes a greater change in the fitted function; thus variance increases.
2. Our goal is to minimize the expected loss. There is a trade-off between bias and variance.
   - Very flexible models have low bias and high variance.
   - Relative rigid models have high bias and low variance.
3. The model with optimal predictive capability is one that leads to the best balance between bias and variance.
4. If there is bias, there is under-fitting. why?
5. If there is variance, there is over-fitting. why?

Bias-Variance trade-off



Bias-Variance trade-off



Readings

1. Chapter 3 of Pattern Recognition and Machine Learning Book (Bishop 2006).
2. Chapter 7 of Machine Learning: A probabilistic perspective (Murphy 2012).
3. Chapter 11 of Probabilistic Machine Learning: An introduction (Murphy 2022).

References i

- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag.
- Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.
- — (2022). *Probabilistic Machine Learning: An introduction*. The MIT Press.

Questions?