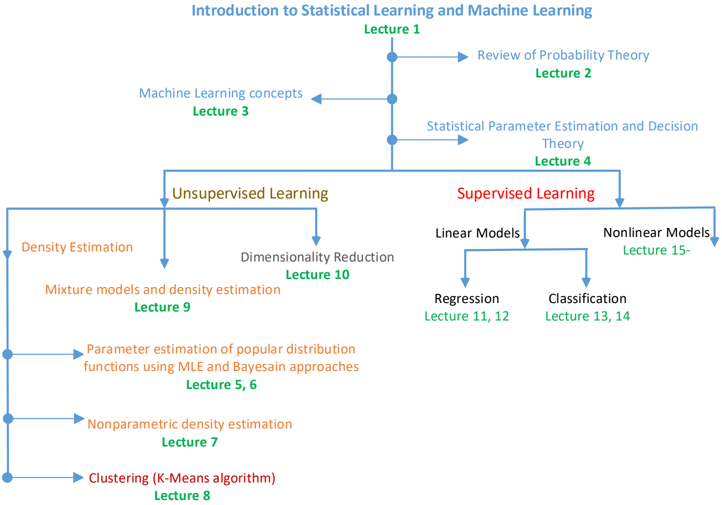
Statistical Learning and Machine Learning

Lecture 12 - Linear Models for Regression 2

October 6, 2021

Course overview and where do we stand

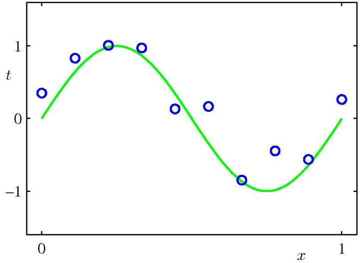


Objectives of the lecture

- Linear models for regression
    - Review of least squares approach
    - Sequential updates of weight vector in least squares
    - Multiple inputs
    - Bayesian linear regression

Goal of Regression

- The goal of regression is to predict the value of one or more continuous *target* variables t given the value of a D-dimensional vector $\boldsymbol{x}$ of *input* variables.
- Supervised learning: Training data consisting of N observations $\{\boldsymbol{x}_n\}$ for $n = 1, \ldots, N$ along with target values $\{t_n\}$ are available.
- Output: A function $y(\boldsymbol{x})$ whose values for new inputs $\boldsymbol{x}$ constitute the predictions t.



Linear Basis Function Models

Two ways to define linear models:

- Linear w.r.t. to both input $\boldsymbol{x}$ and parameters $\boldsymbol{w}$

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x_1 + \ldots, + w_D x_D = w_0 + \sum_{j=1}^{D} w_j x_j = \boldsymbol{w}^T \boldsymbol{x}$$

  where $\boldsymbol{x} = (x_0, x_1, \ldots, x_D)^T$ and $x_0 = 1$ in the final form.

- Non-linear functions $\phi_j(\cdot), j = 1, \ldots, M-1$ (basis functions) w.r.t. to the input, with $\phi_0(\boldsymbol{x}) = 1$

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

  where $\boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^T$ and $\boldsymbol{w} = (w_0, \ldots, w_{M-1})^T$.

Examples

Linear basis function in both $\boldsymbol{w}$ and $\boldsymbol{x}$

$$\phi_j(\boldsymbol{x}) = x_j, \text{ for } j = 1, \ldots, D$$

with $\phi_0(\boldsymbol{x}) = 1$.
In vector form, we get

$$\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_0, \ldots, \phi_{M-1})^T = \boldsymbol{x}$$

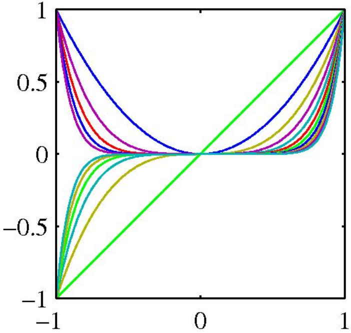
where $M - 1 = D$. In this case, the output function becomes

$$y(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

Example basis functions

Polynomial basis function:

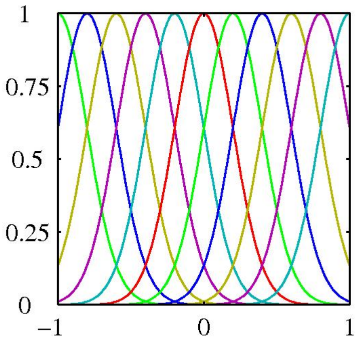
$$\phi_j(x) = x^j$$



Example basis functions

Radial basis function:

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

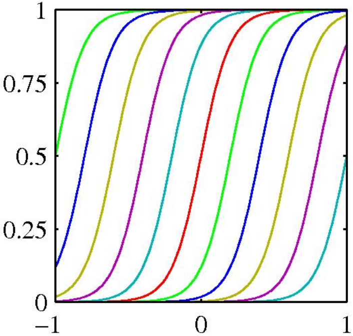


Example basis functions

Sigmoidal basis function:

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$

where $\sigma(a) = 1/(1 + exp(-a))$.



Least Squares

- Goal: Given a set of i.i.d. data points $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ and the corresponding $t_n,\ n = 1, \ldots, N$, we want to estimate the parameters $\boldsymbol{w}$ of the regression model.
- Which model?

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

- Cost function to be minimized: Sum of the squares of the individual errors:

$$E_D(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y(\boldsymbol{x}_n, \boldsymbol{w}) - t_n\right)^2 = \frac{1}{2}\sum_{n=1}^{N}\left(\boldsymbol{w}^T\phi(\boldsymbol{x}_n) - t_n\right)^2$$

- Minimization: By setting $\frac{\partial E_D(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0$:

$$\boldsymbol{w} = \left(\Phi^T\Phi\right)^{-1}\Phi^T\boldsymbol{t}$$

where $\Phi \in \mathbb{R}^{N\times M}$ is formed by using the vectors $\phi(\boldsymbol{x}_n)$ as rows and $\boldsymbol{t} = [t_1, \ldots, t_N]^T$.

Linear Regression: Sequential updates

We obtain a sequential (or *on-line*) learning algorithm for updating $\boldsymbol{w}$ by applying **stochastic gradient descent (SGD)**:

- If the error function has the form $E(\boldsymbol{w}) = \sum_n E_n(\boldsymbol{w})$ then:

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} - \eta \nabla E_n$$

  where τ denotes the iteration number, η is a learning rate parameter and ∇ is the gradient operator.

- For the least-squares error case:

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} + \eta \left(t_n - \boldsymbol{w}^{(\tau)T} \phi(\boldsymbol{x}_n) \right) \phi(\boldsymbol{x}_n).$$

  The value of η needs to be chosen appropriately to ensure convergence of the algorithm.

Linear Regression: Sequential updates

A more effective sequential learning algorithm for updating $\boldsymbol{w}$ is based on the Newton-Raphson iterative optimization scheme, which uses a local quadratic approximation of the log-likelihood function:

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} - \boldsymbol{H}^{-1} \nabla E_n(\boldsymbol{w})$$

where $\boldsymbol{H}$ is the Hessian matrix having elements $H_{ij} = \frac{\partial^2 E_n(\boldsymbol{w})}{\partial w_i \partial w_j}$.

$$\begin{aligned}
\nabla E_n(\boldsymbol{w}) &= \sum_{n=1}^{N} (\boldsymbol{w}^T \phi(\boldsymbol{x}_n) - t_n) \phi(\boldsymbol{x}_n) = \Phi^T \Phi \boldsymbol{w} - \Phi^T \boldsymbol{t} \\
\boldsymbol{H} = \nabla \nabla E_n(\boldsymbol{w}) &= \sum_{n=1}^{N} \phi_n \phi_n^T = \Phi^T \Phi
\end{aligned}$$

Linear Regression: Sequential updates

Because $E_n(\boldsymbol{w})$ is a quadratic function, the Newton-Raphson method gives the exact solution in one step if applied on the entire data set.

The update takes the form:

$$\begin{aligned}\boldsymbol{w}^{(\tau+1)} &= \boldsymbol{w}^{(\tau)} - (\Phi^T\Phi)^{-1}\left(\Phi^T\Phi\boldsymbol{w}^{(\tau)} - \Phi^T\boldsymbol{t}\right) \\ &= (\Phi^T\Phi)^{-1}\Phi^T\boldsymbol{t}\end{aligned}$$

Linear Regression: Multiple outputs

When we want to regress to multiple target values $\boldsymbol{t}_n \in \mathbb{R}^K$:

$$\boldsymbol{y}(\boldsymbol{x}_n, \boldsymbol{W}) = \boldsymbol{W}^T \phi(\boldsymbol{x}_n)$$

where $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K] \in \mathbb{R}^{M \times K}$.

When the targets are multi-dimensional random variables $\boldsymbol{t}$, we need to estimate $p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{W}, \beta)$. We will consider the case:

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

Linear Regression: Multiple outputs

Given a set of i.i.d. data points $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ and the corresponding target vectors $\boldsymbol{T} \in \mathbb{R}^{N \times K}$, having as its n-th row the target vector $\boldsymbol{t}_n$.

The log-likelihood function is given by:

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

Setting $\frac{\partial \ln p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{W}, \beta)}{\partial \boldsymbol{W}} = 0$:

$$\boldsymbol{W}_{ML} = \Phi^{\dagger} \boldsymbol{T}.$$

Bayesian Linear Regression

We consider the model parameters $\boldsymbol{w}$ as parameters drawn by a distribution $p(\boldsymbol{w})$ (we assume β is known).

The likelihood function $p(\boldsymbol{t}|\boldsymbol{w})$ is an exponential of quadratic function of $\boldsymbol{w}$:

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

The corresponding conjugate prior has the form:

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_0, \boldsymbol{S}_0).$$

Bayesian Linear Regression

Reminder: $$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

The posterior distribution is:

$$p(\boldsymbol{w}|\boldsymbol{t}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_N, \boldsymbol{S}_N)$$

where

$$\begin{aligned} \boldsymbol{m}_N &= \boldsymbol{S}_N(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \beta\Phi^T\boldsymbol{t}) \\ \boldsymbol{S}_N^{-1} &= \boldsymbol{S}_0^{-1} + \beta\Phi^T\Phi. \end{aligned}$$

Cases:

- If $\boldsymbol{S}_0 = \alpha^{-1}\boldsymbol{I}$ with $\alpha \to 0$, then $\boldsymbol{m}_N \to \boldsymbol{w}_{ML}$
- If $N = 0$ then $p(\boldsymbol{w}|\boldsymbol{t})$ becomes $p(\boldsymbol{w})$

When $p(\boldsymbol{w}|\alpha) = \mathcal{N}(\boldsymbol{w}|0, \alpha^{-1}\boldsymbol{I})$:

$$
\begin{aligned}
\boldsymbol{m}_N &= \beta \boldsymbol{S}_N \Phi^T \boldsymbol{t} \\
\boldsymbol{S}_N^{-1} &= \alpha \boldsymbol{I} + \beta \Phi^T \Phi
\end{aligned}
$$

and

$$
\ln p(\boldsymbol{w}|\boldsymbol{t}) = -\frac{\beta}{2} \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}^T \phi(\boldsymbol{x}_n) \right)^2 - \frac{\alpha}{2} \boldsymbol{w}^T \boldsymbol{w} + \text{const.}
$$

Maximization of the posterior distribution w.r.t. $\boldsymbol{w}$ is equivalent to the minimization of the sum-of-squares error with an additional quadratic regularization term with regularization term $\lambda = \alpha/\beta$.

Bayesian Linear Regression: Predictive distribution

In practice, we want to make predictions of t for new values of $\boldsymbol{x}$:

$$p(t|\boldsymbol{t}, \alpha, \beta) = \int p(t|\boldsymbol{w}, \beta) p(\boldsymbol{w}|\boldsymbol{t}, \alpha, \beta) d\boldsymbol{w}$$

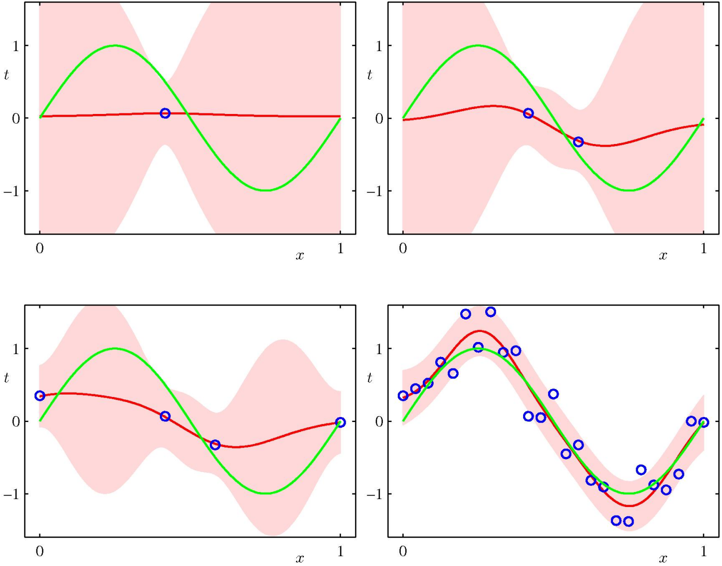
- Conditional distribution: $p(t|\boldsymbol{w}, \beta) = \mathcal{N}(t|y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1})$
- Posterior: $p(\boldsymbol{w}|\boldsymbol{t}, \alpha, \beta) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_N, \boldsymbol{S}_N)$

The result is a Gaussian:

$$p(t|\boldsymbol{x}, \boldsymbol{t}, \alpha, \beta) = \mathcal{N}\left(t|\boldsymbol{m}_N^T \phi(\boldsymbol{x}), \sigma_N^2(\boldsymbol{x})\right)$$

where: $\sigma_N^2(\boldsymbol{x}) = \beta^{-1} + \phi(\boldsymbol{x})^T \boldsymbol{S}_N \phi(\boldsymbol{x})$.

Bayesian Linear Regression: Predictive distribution



Bayesian Linear Regression: Predictive distribution