Pattern Recognition and Machine Learning Chapter 3 Summary

CSCI-GA 3033 Bayesian Machine Learning
New York University
Yilun Kuang

Sep 20, 2022

3.1 Linear Basis Function Models

3.1.1 - 3.1.3 MLE, Least Squares, Geometry, Sequential Learning

There is a class of linear basis function $\mathcal{F} := \{y(\mathbf{x}, \mathbf{w}) | y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \phi(\mathbf{x})\}$ for regression, where $\mathbf{w}$ comes from a finite-dimensional space and $\phi(\cdot)$ could be identity, "Gaussian", sigmoid, Fourier, wavelet basis etc. For the target generating process $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$, where $y(\mathbf{x}, \mathbf{w}) \in \mathcal{F}$ and ϵ is an additive Gaussian noise, the MLE is equivalent to the squared loss minimization. The $\mathbf{w}_{\text{ML}}$ solutions follows from the standard least square normal equation.

The prediction $\mathbf{y}$ is the projection of $\mathbf{t}$ onto the subspace spanned by the basis function $\{\phi_j\}_{j \in \mathcal{J}}$, and learning can be done sequentially using SGD methods.

3.1.4 - 3.1.5 Regularized Least Squares and Multiple Outputs

The regularized least square solution also follows from the normal equation, where L1 / Lasso induces sparsity and L2 / Ridge penalize weights norm uniformly. The additive regularization penalty can be casted into the primal-dual formulation of a standard constrained optimization problem.

3.2 Bias-Variance Decomposition

From a classical frequentist perspective, the risk functional $R(f)$ can be decomposed into contributions from the bias, variance, and irreducible error term. For Bayesian, overfitting and model complexity is related to averaging with respect to the posterior measure of the weights. It would be interesting to see how double descent can be brought into the discussion.

3.3 Bayesian Linear Regression

With specified prior and likelihood function, maximizing the posterior with respect to the weights $\mathbf{w}$ is equivalent to MLE w/ regularization (via the prior distribution). In the infinite data regime, the posterior measure will converge to the delta function. For localized basis like Gaussians, the model will extrapolate outside the domain of the basis function. Gaussian Process (GP) can be used to address this issue. GP also has connections to (NNGP) kernels, a theoretical proxy for lots of mathematical theories of deep learning.

3.4 Bayesian Model Comparison

Instead of performing cross-validation over a set of models $\{\mathcal{M}_i\}$, the Bayesian approach compute the marginal likelihood / model evidence $p(\mathcal{D}|\mathcal{M}_i)$ (integrating over all parameters), where $p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_i)$. The ratio between two marginal likelihoods $\frac{p(\mathcal{D}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_j)}$ is called the Bayes factor.

The model evidence is formally given by

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}, \mathcal{M}_i) p(\mathbf{w}|\mathcal{M}_i) d\mathbf{w} = \mathbb{E}_{\mathbf{w}\sim P}[p(\mathcal{D}|\mathbf{w}, \mathcal{M}_i)]$$

So the model evidence is the average of the probability of generating data $\mathcal{D}$ from a model $\mathcal{M}_i$ with respect to the prior distribution P over weights $\mathbf{w}$. It can be shown that marginal likelihood will favor models with intermediate complexity by numerically approximating the above integral.

3.5 The Evidence Approximation

For a complete Bayesian treatment, we also need to integrate over hyper-priors α and β in addition to integrating over $\mathbf{w}$. To avoid the analytically intractable integration, we can perform *evidence approximation*, i.e. maximizing the marginal likelihood by first integrating over the parameters $\mathbf{w}$ to get $\hat{\alpha}$ and $\hat{\beta}$.

3.6 Limitations of Fixed Basis Functions

Fixed basis functions can suffer from large linear approximation errors and the curse of the dimensionality. In a way, fixed basis functions have the following linear approximation error

$$\inf_{U;\ \dim(\mathrm{U})=\mathrm{N}} \sup_{f\in\mathcal{F}} \left\| f - \mathrm{Proj}_{U_N}(f) \right\|^2$$

while neural networks have the following non-linear approximation error

$$\sup_{f\in\mathcal{F}} \inf_{U;\ \dim(\mathrm{U})=\mathrm{N}} \left\| f - \mathrm{Proj}_{U_N}(f) \right\|^2$$