CS-567 Machine Learning

Nazar Khan

PUCIT

Lecture 06
Bayesian Curve Fitting

Polynomial Curve Fitting
Bayesian Perspective

- ML estimation of $\mathbf{w}$ maximises the likelihood function $p(\mathbf{t}|\mathbf{x}, \mathbf{w})$ to find the w for which the observed data is most likely.
- By using a prior $p(\mathbf{w})$, we can employ Bayes' theorem

$$\underbrace{p(\mathbf{w}|\mathbf{x}, \mathbf{t})}_{\text{posterior}} \propto \underbrace{p(\mathbf{t}|\mathbf{x}, \mathbf{w})}_{\text{likelihood}} \underbrace{p(\mathbf{w})}_{\text{prior}}$$

- Now maximise the posterior probability $p(\mathbf{w}|\mathbf{x}, \mathbf{t})$ to find the most probable $\mathbf{w}$ given the data $(\mathbf{x}, \mathbf{t})$.
- This technique is called **maximum posterior** or **MAP**.

Polynomial Curve Fitting
Bayesian Perspective

- Let the prior on parameters $\mathbf{w}$ be a zero-mean Gaussian

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2}\right)^{(M+1)/2} \exp\{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\}$$

- Negative logarithm of posterior becomes

$$-\ln p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}$$

which is the same as the *regularized sum-of-squares error* function with $\lambda = \alpha/\beta$.

Polynomial Curve Fitting
Bayesian Perspective

- **So**, assuming $t \sim \mathcal{N}$ **and** $\mathbf{w} \sim \mathcal{N}$, MAP estimation leads to regularized sum-of-squared errors minimisation.
- **Equivalently**, minimising regularized sum-of-squared errors implies $t \sim \mathcal{N}$ **and** $\mathbf{w} \sim \mathcal{N}$ (*i.e.*, noise and the parameters were normally distributed).
- If precision on noise and parameters were α and β respectively, then regularizer $\lambda = \alpha/\beta$.
- MAP estimation allows us to determine optimal α and β whereas regularised-SSE minimisation depends on a user-given λ.

Programming Assignment 1

Curve Fitting

- In this assignment, we will learn how to fit a polynomial to data points $\{x, t\}_1^N$ using
  1. Maximum Likelihood (ML) estimation – find $\mathbf{w}$ that maximises $p(T|X, \mathbf{w})$.
  2. Maximum Posterior (MAP) estimation – find $\mathbf{w}$ that maximises $p(\mathbf{w}|X, T)$
- The goal is to reproduce Figures 1.4 till 1.7 from Chapter 1 of Bishop's book.
- The main Matlab files in this assignment are
  - `generate_data.m` generates data from the $\sin(2\pi)$ function and adds random noise.
  - `evaluate_polynomial.m` evaluates polynomial $\mathbf{w}$ at points in vector $\mathbf{x}$.
  - `fit_polynomial_ML.m` fits a polynomial to data X, T via Maximum Likelihood (ML) estimation.

Programming Assignment 1

Curve Fitting

  - `fit_polynomial_MAP.m` fits a polynomial to data X, T via Maximum Posterior (MAP) estimation.
- You have to fill in the missing pieces of code in
  - `evaluate_polynomial.m`
  - `fit_polynomial_ML.m`
  - `fit_polynomial_MAP.m`
- To generate all results required for this assignment, run the provided script `get_all_results.m`.
- **Submission**: Paste `your_roll_number_curve_fitting.zip` to `\\printsrv\Teacher Data\Dr.Nazar Khan\Teaching\Fall2016\CS 567 Machine Learning\Submissions\PA1`
- **Deadline**: Wednesday, December 07, 2016 before 5:30 pm.
- The .zip file should ONLY contain

Programming Assignment 1
Curve Fitting

- `evaluate_polynomial.m`
- `fit_polynomial_ML.m`
- `fit_polynomial_MAP.m`

and

- `Figure_1_4.png`
- `Figure_1_5.png`
- `Figure_1_6.png`
- `Figure_1_7.png`
- `polynomial_fitting_ML_VS_MAP.png`