INF367A: Probabilistic machine learning

Lecture 7: Linear models

Pekka Parviainen

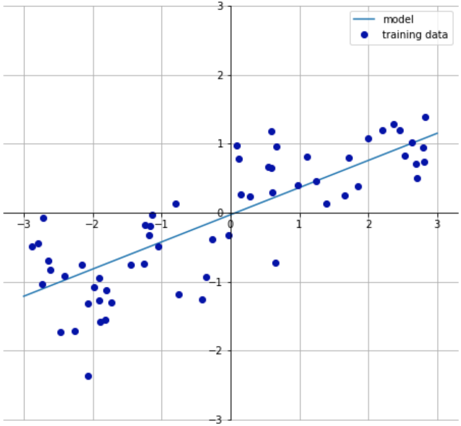
University of Bergen

18.2.2020

Outline

Linear regression (recap from INF264)

- Simple and widely studied models
- Often computationally convenient



Source: `https://medium.com/pharos-production/machine-learning-linear-models-part-1-312757aab7bc`

Regression

- Training data consist of n pairs of observations $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_n, y_n)$ where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector (predictors, input variables, independent variables, covariates) and $y_i \in \mathbb{R}$ is a response variable (label, output variable, dependent variable)
- Note that we will refer observations with a subscript, i.e., $\mathbf{x}_i$ and features with a superscript in parentheses. That is, $x^{(j)}$ is the jth element of vector $\mathbf{x}$
- Regression, we predict response values using a function f:

$$y_i = f(\mathbf{x}_i) + \epsilon_i,$$

where ϵ_i is the error (or residual) for ith observations

Multivariate linear regression

- Learn a linear function $f : \mathbb{R}^d \to \mathbb{R}$ for arbitrary d
- Instead of a line, we have a hyperplane
- Data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has n instances $\mathbf{x}_i^T$ as it rows and $\mathbf{y} \in \mathbb{R}^n$ has the corresponding labels y_i
- $\mathbf{X}$ is often called the design matrix
- Weights are stored in $\mathbf{w} \in \mathbb{R}^d$
- Useful trick: $\mathbf{x}$ can automatically include a constant term, $\mathbf{x} = (1, x^{(1)}, x^{(2)}, \ldots, x^{(d)})^T$, such that the intercept is automatically included:

$$\mathbf{w}^T\mathbf{x} = w^{(0)} + w^{(1)}x^{(1)} + \ldots + w^{(d)}x^{(d)}$$

Note that $x^{(j)}$ denotes the jth feature

Multivariate linear regression

- We have
$$y_i = \mathbf{w}^T \mathbf{x}_i + \epsilon_i,$$
- Goal: choose weights $\mathbf{w}$ to minimize the sum of squared errors
$$\begin{aligned} \sum_{i=1}^{n} \epsilon_i^2 &= \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \\ &= ||\epsilon||_2^2 \end{aligned}$$

Multivariate linear regression

- Setting gradient to zero and solving the equations, we get

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y},$$

  where A^{-1} denotes a matrix inverse of matrix A ($AA^{-1} = I$).
- If columns of $\mathbf{X}$ are linearly independent then the matrix $\mathbf{X}^T\mathbf{X}$ is of full rank and has an inverse

Multivariate linear regression with regularization

- To reduce overfitting, one may want to penalize complexity
- For example, the objective function for linear regression with L2-regularizer can be written as

  $$||(\mathbf{y} - \mathbf{w}^T\mathbf{X})||_2^2 + \lambda||\mathbf{w}||_2^2$$

  where hyperparameter $\lambda \geq 0$ specifies the strength of regularization
- Setting gradient to zero and solving the equations we get

  $$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

Likelihood

- Likelihood is the probability of data given parameters

$$\ell(\mathbf{w}) = P(\mathbf{X} \,|\, \mathbf{w}),$$

where $\mathbf{X}$ is the data

- Assuming that errors are independent and follow Gaussian distribution, that is, $\epsilon_i \sim N(0, 1/\beta)$, the likelihood of our linear model is

$$\begin{aligned} \ell(\mathbf{w}) &= \prod_{i=1}^{n} N(\mathbf{y}_i - \mathbf{w}^T \mathbf{x}_i \,|\, 0, 1/\beta) \\ &= \prod_{i=1}^{n} N(\mathbf{y}_i \,|\, \mathbf{w}^T \mathbf{x}_i, 1/\beta) \end{aligned}$$

Maximum likelihood estimation

- Maximizing likelihood, that is, finding $\mathbf{w}$ that maximize $\ell(\mathbf{w})$ is equivalent with maximizing log-likelihood

$$\begin{aligned} \ell\ell(\mathbf{w}) &= \log \ell(\mathbf{w}) \\ &= \sum_{i=1}^{n} \log N(\mathbf{y}_i - \mathbf{w}^T \mathbf{x}_i \,|\, 0, 1/\beta) \\ &= -\frac{1}{2}(\mathbf{y} - \mathbf{w}^T\mathbf{X})^T \Sigma^{-1} (\mathbf{y} - \mathbf{w}^T\mathbf{X}) + C, \end{aligned}$$

  where $\Sigma^{-1} = \beta \mathbf{I}$ and C is a constant

- Maximum likelihood estimate is

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ell\ell(\mathbf{w}) = \arg\max_{\mathbf{w}} -(\mathbf{y} - \mathbf{w}^T\mathbf{X})^T(\mathbf{y} - \mathbf{w}^T\mathbf{X})$$

ML estimation

- Solving the problem on the previous slide gives us

$$\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- The ML estimation with Gaussian likelihood is equivalent to minimizing the squared error

$$\mathbf{w}^* = \hat{\mathbf{w}}$$

Bayesian estimation

Prior distribution

- We can place a Gaussian prior distribution on **w**:

$$P(\mathbf{w}|\alpha) = N(\mathbf{w}|\nu, \alpha^{-1}\mathbf{I})$$

$$= \prod_{i=1}^{d} N(w_i|\nu_i, \alpha^{-1}) = \left(\frac{\alpha}{2\pi}\right)^{\frac{d}{2}} e^{-\frac{\alpha}{2}\sum_i (w_i - \nu_i)^2}$$

- Typically, $\nu = \mathbf{0}$. Let use denote the hyperparameters by $\Gamma = (\alpha, \beta, \nu)$
- Posterior

$$\log P(\mathbf{w}\,|\,\Gamma, \mathbf{X}) = -\frac{\beta}{2}\sum_{i=1}^{n}\left[y_i - \mathbf{w}^T\mathbf{x}_i\right]^2 - \frac{\alpha}{2}(\mathbf{w}-\nu)^T(\mathbf{w}-\nu) + \text{const}$$

Posterior distribution

- Posterior distribution is obtained by completing the square (left as an exercise):

$$P(\mathbf{w}\,|\,\Gamma, \mathbf{X}) = N(\mathbf{w}\,|\,\mathbf{m}, \mathbf{S})$$

where

$$\mathbf{S} = \left(\beta \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i{}^T + \alpha \mathbf{I}\right)^{-1}, \qquad \mathbf{m} = \mathbf{S}(\beta \sum_{i=1}^{n} y_i \mathbf{x}_i + \alpha \nu)$$

Predictive posterior distribution

- Typically, the goal is to predict y given $\mathbf{x}$
- Mean prediction:

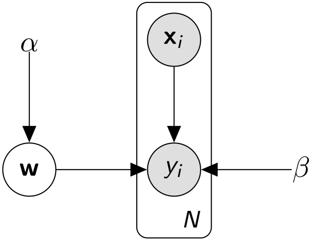
$$\widetilde{y} = \int \mathbf{w}^T\mathbf{x} \times P(\mathbf{w} | \Gamma, \mathbf{X})\mathrm{d}\mathbf{w} = \mathbf{m}^T\mathbf{x}$$

- Posterior predictive distribution:

$$\begin{aligned} P(y \mid \mathbf{x}, \Gamma, \mathbf{X}) &= \int N(y \mid \mathbf{w}^T\mathbf{x}, \beta^{-1}) P(\mathbf{w} \mid \Gamma, \mathbf{X})\mathrm{d}\mathbf{w} \\ &= \int N(y \mid \mathbf{w}^T\mathbf{x}, \beta^{-1}) N(\mathbf{w} \mid \mathbf{m}, \mathbf{S})\mathrm{d}\mathbf{w} \\ &= N(y \mid \mathbf{m}^T\mathbf{x}, \beta^{-1} + \mathbf{x}^T\mathbf{S}\mathbf{x}) \end{aligned}$$

Effect of hyperparamaters

Hyperparameters



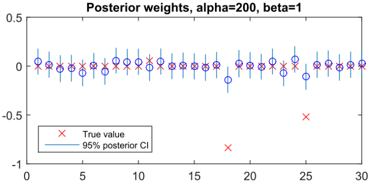
- α: *precision* of the *regression weights*
  - determines the amount of regularization
  - large precision $\rightarrow$ small variance $\rightarrow$ weights are close to zero
- β: *precision* of the noise

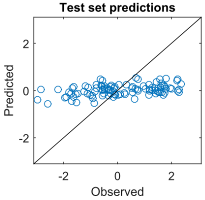
Example, impact of hyperparameters (1/3)

- Setup: simulate $y = \mathbf{w}_{true}^T \mathbf{x} + \epsilon$, where $\epsilon \sim N(0, \beta^{-1})$ and $\beta = 1$
- The goal is to investigate how hyperparameter α affects the posterior distribution of the parameters $\mathbf{w}$

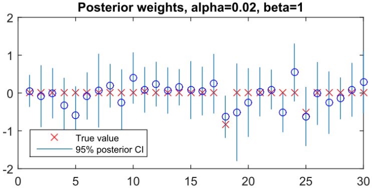
Example, impact of hyperparameters (2/3)

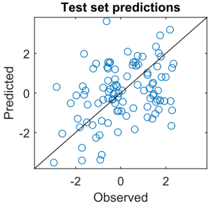
- Too large α, $Var(y - \tilde{y}) = 1.54$ (Original $Var(y) = 1.75$)





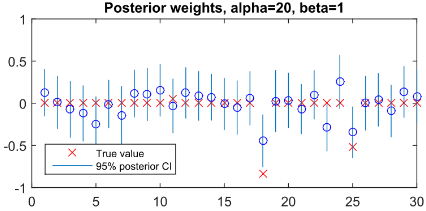
- Too small α, $Var(y - \tilde{y}) = 2.48$

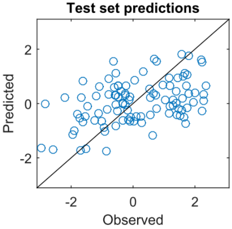




Example, impact of hyperparameters (3/3)

- About good α, $Var(y - \widetilde{y}) = 1.46$
- A compromise between bias and variance





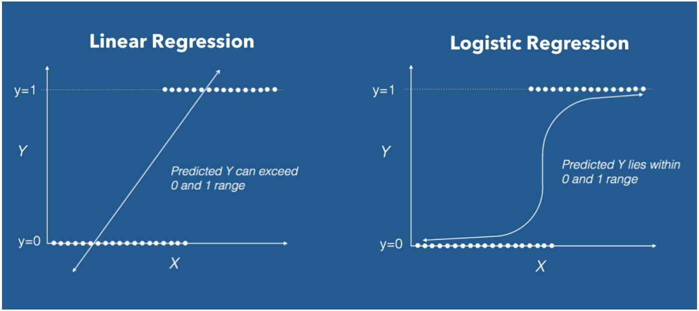
Determining hyperparameters

- Fully Bayesian approach: If you do not know some quantity, place a prior on it.
  - Specify $P(\alpha)$ and $P(\beta)$ and compute the posterior $P(\mathbf{w}, \alpha, \beta \,|\, \mathbf{X})$
  - No-closed form solution $\Rightarrow$ need to approximate
  - More later ...
- Or use model selection
  - More next week ...

Logistic regression

Linear models for classification (Recap from INF264)

- Using standard linear regression is problematic in predicting categorical responses
- We can use a variant called logistic regression instead



Source:

https://www.machinelearningplus.com/wp-content/uploads/2017/09/linear_vs_logistic_regression.jpg

Logistic regression

- Consider binary classification problem
  - $y_i \in \{0, 1\}$
- Let p denote our prediction of the probability that $P(y = 1|\mathbf{x})$
- Logistic linear regression

$$\log \frac{p}{1-p} = \mathbf{w}^T \mathbf{x}$$

- Or equivalently

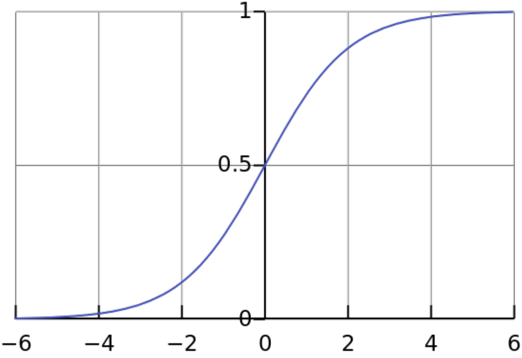
$$P(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}),$$

where $\sigma(\cdot)$ is the so-called logistic sigmoid

$$\sigma(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$

Sigmoid function

$$\sigma(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$$



Source: Wikipedia

Logistic regression for classification

- When used in classification, the decision boundary is defined by $P(y=1|\mathbf{x}) = P(y=0|\mathbf{x}) = 0.5$. This corresponds to a hyperplane
$$\mathbf{w}^T\mathbf{x} = 0$$
.
- Classification rule:
$$\begin{aligned} \mathbf{w}^T\mathbf{x} > 0 &\quad\rightarrow\quad y = 1 \\ \mathbf{w}^T\mathbf{x} < 0 &\quad\rightarrow\quad y = 0 \end{aligned}$$

Learning parameters

- Conditional likelihood of data $\mathbf{y}$ given $\mathbf{X}$ is

$$
\begin{aligned}
P(\mathbf{y}|\mathbf{w},\mathbf{X}) &= \prod_{i=1}^{n} P(y_i = 1|\mathbf{w},\mathbf{x}_i)^{y_i}(1 - P(y_i = 1|\mathbf{w},\mathbf{x}_i))^{1-y_i} \\
&= \prod_{i=1}^{n} \sigma(\mathbf{w}^T\mathbf{x}_i)^{y_i}(1 - \sigma(\mathbf{w}^T\mathbf{x}_i))^{1-y_i}
\end{aligned}
$$

- Maximizing the likelihood is equivalent to maximizing the log-likelihood

$$
\ell\ell(\mathbf{w}) = \sum_{i=1}^{n} \Big(y_i \log \sigma(\mathbf{w}^T\mathbf{x}_i) + (1 - y_i) \log(1 - \sigma(\mathbf{w}^T\mathbf{x}_i)) \Big)
$$

- Equivalent to minimizing logarithmic loss (log-loss)

$$
L(\mathbf{w}) = -\sum_{i=1}^{n} \Big(y_i \log \sigma(\mathbf{w}^T\mathbf{x}_i) + (1 - y_i) \log(1 - \sigma(\mathbf{w}^T\mathbf{x}_i)) \Big)
$$

Bayesian logistic regression

Prior and posterior

- Gaussian prior

$$P(\mathbf{w}\,|\,\alpha) = N_d(\mathbf{w}\,|\,\mathbf{0}, \alpha^{-1}\mathbf{I}) = \alpha^{\frac{d}{2}}(2\pi)^{-\frac{d}{2}} e^{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}}$$

  where α is the precision.

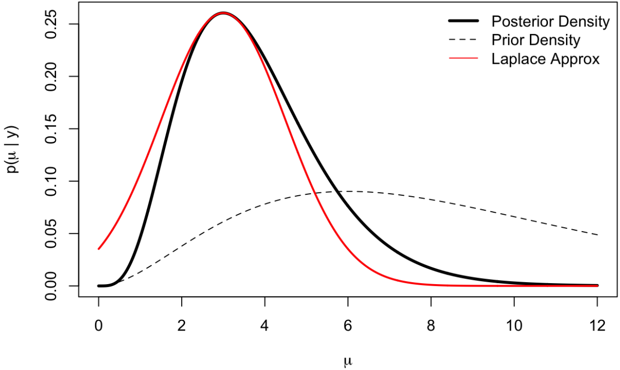
- Given $D = \{(\mathbf{x}_i, c_i), i = 1, \ldots, n\}$ the posterior equals

$$\begin{aligned} P(\mathbf{w}\,|\,\alpha, D) &= \frac{P(D\,|\,\mathbf{w}, \alpha)P(\mathbf{w}\,|\,\alpha)}{P(D\,|\,\alpha)} \\ &= \frac{1}{P(D|\alpha)} P(\mathbf{w}\,|\,\alpha) \prod_{i=1}^{n} P(c_i|\mathbf{x}_i, \mathbf{w}) \end{aligned}$$

  (not of standard form, Laplace approximation is feasible to compute).

Laplace approximation

- Use a Gaussian distribution to approximate the true posterior



Source: `https://bookdown.org/rdpeng/advstatcomp/laplace-approximation.html`

Taylor approximation

- A function $f(x)$ can be approximated in the neighborhood of point a using the following polynomial:

$$f(x) \approx f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2$$

Function of several variables

- Gradient: if $f \equiv f(x_1, \ldots, x_n)$

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

- Hessian matrix (matrix of second partial derivatives):

$$H_f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Laplace approximation to the posterior (1/2)

- Approximate the true posterior $P(\mathbf{w} \,|\, D)$ using a Gaussian distribution:
$$P(\mathbf{w} \,|\, D) \approx C \cdot e^{-\tilde{E}(\mathbf{w})}$$
where $\tilde{E}(\mathbf{w})$ is a quadratic polynomial in $\mathbf{w}$
- Suppose $E(\mathbf{w}) = -\log P(\mathbf{w} \,|\, D)$ (the negative log-posterior)
- Let $\bar{\mathbf{w}} = \arg\max_{\mathbf{w}} P(\mathbf{w} \,|\, D)$ be the mode of the posterior distribution
- Approximate $E(\mathbf{w})$ in the neighborhood of the mode using the Taylor approximation:

$$\tilde{E}(\mathbf{w}) = E(\bar{\mathbf{w}}) + \frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^T H_{\bar{\mathbf{w}}}(\mathbf{w} - \bar{\mathbf{w}})$$

Laplace approximation to the posterior (2/2)

- We can rewrite $\tilde{E}(\mathbf{w})$ as follows

$$\tilde{E}(\mathbf{w}) \quad = \quad \frac{1}{2}(\mathbf{w}-\mu)^T\Sigma^{-1}(\mathbf{w}-\mu)+C,$$

  where $\mu = \bar{\mathbf{w}}$ and $\Sigma = H_{\bar{\mathbf{w}}}^{-1}$
- We observe that we are dealing with a Gaussian distribution

Laplace approximation in practice

- In practice:
  1. Find the minimum of $E(\mathbf{w})$ (mode of the posterior) by numerical optimization, e.g., Newtons method:

$$\mathbf{w}^{new} = \mathbf{w} - H_w^{-1} \nabla E$$

  2. When converged, compute the Hessian $H_{\bar{\mathbf{w}}}$ of the $E(\mathbf{w})$ at $\bar{\mathbf{w}}$
  3. The posterior approximation is

$$q(\mathbf{w} \,|\, \alpha, D) = N(\mathbf{w} \,|\, \mathbf{m}, \mathbf{S}), \quad \mathbf{m} = \bar{\mathbf{w}}, \quad \mathbf{S} = H_{\bar{\mathbf{w}}}^{-1}.$$

Laplace approximation for logistic regression

- Negative log-posterior:

$$E(\mathbf{w}) = \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n} \log \sigma(\mathbf{w}^T\mathbf{h}_i),$$

  where $\mathbf{h}_i = (2y_i - 1)\mathbf{x}_i$

- Gradient:

$$\nabla E = \alpha\mathbf{w} - \sum_{i=1}^{n}(1 - \sigma_i)\mathbf{h}_i,$$

  where $\sigma_i = \sigma(\mathbf{w}^T\mathbf{h}_i)$

- Hessian:

$$\mathbf{H} = \alpha\mathbf{I} + \sum_{i=1}^{n}\sigma_i(1 - \sigma_i)\mathbf{x}_i\mathbf{x}_i^T$$

  Note that $\mathbf{x}_i\mathbf{x}_i^T$ is an outer product resulting in a $d \times d$ matrix

When not to use Laplace approximation?

- Laplace approximation assumes a Gaussian distribution
- Gives a good approximation when the posterior is "nearly" Gaussian
- Can be terrible when the posterior is "far" from being Gaussian
  - When the posterior is skewed
  - When the posterior is multimodal

Further readings

- Bishop: 3.1, 3.3, 4.5