

Introduction to Machine Learning

Bayesian Regression

Varun Chandola

Computer Science & Engineering
State University of New York at Buffalo
Buffalo, NY, USA
chandola@buffalo.edu



University at Buffalo
Department of Computer Science
and Engineering
School of Engineering and Applied Sciences



Linear Regression

- Problem Formulation

- Learning Parameters

Bayesian Linear Regression

Bayesian Regression

- Estimating Bayesian Regression Parameters

- Prediction with Bayesian Regression

Handling Outliers in Regression

Generative vs. Discriminative Classifiers

Probabilistic Interpretation of Logistic Regression

Logistic Regression - Training

- Using Gradient Descent for Learning Weights

- Using Newton's Method

- Regularization with Logistic Regression

- Handling Multiple Classes

- Bayesian Logistic Regression

Linear Regression

- ▶ There is one scalar **target** variable y (instead of hidden)
- ▶ There is one vector **input** variable x
- ▶ Inductive bias:

$$y = \mathbf{w}^\top \mathbf{x}$$

Linear Regression Learning Task

Learn \mathbf{w} given training examples, $\langle \mathbf{X}, \mathbf{y} \rangle$.

Probabilistic Interpretation

- ▶ y is assumed to be normally distributed

$$y \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \sigma^2)$$

- ▶ or, equivalently:

$$y = \mathbf{w}^\top \mathbf{x} + \epsilon$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$

- ▶ y is a *linear combination* of the input variables
- ▶ Given \mathbf{w} and σ^2 , one can find the probability distribution of y for a given \mathbf{x}

Learning Parameters - MLE Approach

- Find \mathbf{w} and σ^2 that maximize the likelihood of training data

$$\begin{aligned}\hat{\mathbf{w}}_{MLE} &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \\ \hat{\sigma}_{MLE}^2 &= \frac{1}{N} (\mathbf{y} - \mathbf{X} \hat{\mathbf{w}})^\top (\mathbf{y} - \mathbf{X} \hat{\mathbf{w}})\end{aligned}$$

Putting a Prior on \mathbf{w}

- ▶ “Penalize” large values of \mathbf{w}
- ▶ A zero-mean Gaussian prior

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} | 0, \tau^2 I)$$

- ▶ What is posterior of \mathbf{w}

$$p(\mathbf{w} | \mathcal{D}) \propto \prod_i \mathcal{N}(y_i | \mathbf{w}^\top \mathbf{x}_i, \sigma^2) p(\mathbf{w})$$

- ▶ Posterior is also Gaussian

Parameter Estimation for Bayesian Regression

- Prior for \mathbf{w}

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \tau^2 \mathbf{I}_D)$$

- Posterior for \mathbf{w}

$$\begin{aligned} p(\mathbf{w}|\mathbf{y}, \mathbf{X}) &= \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \\ &= \mathcal{N}(\bar{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X} + \frac{\sigma^2}{\tau^2} \mathbf{I}_D)^{-1} \mathbf{X}^\top \mathbf{y}, \sigma^2 (\mathbf{X}^\top \mathbf{X} + \frac{\sigma^2}{\tau^2} \mathbf{I}_D)^{-1}) \end{aligned}$$

- Posterior distribution for \mathbf{w} is also Gaussian
- What will be MAP estimate for \mathbf{w} ?

Prediction with Bayesian Regression

- ▶ For a new \mathbf{x}^* , predict y^*
- ▶ Point estimate of y^*

$$y^* = \hat{\mathbf{w}}_{MLE}^\top \mathbf{x}^*$$

- ▶ Treating y as a Gaussian random variable

$$p(y^*|\mathbf{x}^*) = \mathcal{N}(\hat{\mathbf{w}}_{MLE}^\top \mathbf{x}^*, \hat{\sigma}_{MLE}^2)$$

$$p(y^*|\mathbf{x}^*) = \mathcal{N}(\hat{\mathbf{w}}_{MAP}^\top \mathbf{x}^*, \hat{\sigma}_{MAP}^2)$$

Full Bayesian Treatment

- ▶ Treating y and \mathbf{w} as random variables

$$p(y^*|\mathbf{x}^*) = \int p(y^*|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathbf{X}, \mathbf{y})d\mathbf{w}$$

- ▶ This is also *Gaussian*!

Impact of outliers on regression

- ▶ Linear regression training gets impacted by the presence of outliers
- ▶ The square term in the exponent of the Gaussian pdf is the culprit
 - ▶ Equivalent to the square term in the loss
- ▶ How to handle this (*Robust Regression*)?
- ▶ Probabilistic:
 - ▶ Use a different distribution instead of Gaussian for $p(y|\mathbf{x})$
 - ▶ Robust regression uses Laplace distribution

$$p(y|\mathbf{x}) \sim \text{Laplace}(\mathbf{w}^\top \mathbf{x}, b)$$

- ▶ Geometric:
 - ▶ *Least absolute deviations* instead of least squares

$$J(\mathbf{w}) = \sum_{i=1}^N |y_i - \mathbf{w}^\top \mathbf{x}|$$

Generative vs. Discriminative Classifiers

- ▶ Probabilistic classification task:

$$p(Y = \textit{benign} | \mathbf{X} = \mathbf{x}), p(Y = \textit{malicious} | \mathbf{X} = \mathbf{x})$$

- ▶ How do you estimate $p(y|\mathbf{x})$?

$$p(y|\mathbf{x}) = \frac{p(y, \mathbf{x})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$

- ▶ Two step approach - Estimate generative model and then posterior for y (Naïve Bayes)
- ▶ Solving a more general problem [2, 1]
- ▶ Why not directly model $p(y|\mathbf{x})$? - **Discriminative approach**

Which is Better?

- ▶ Number of training examples needed to learn a PAC-learnable classifier \propto *VC-dimension of the hypothesis space*
- ▶ VC-dimension of a probabilistic classifier \propto Number of parameters [2] (or a small polynomial in the number of parameters)
- ▶ Number of parameters for $p(y, \mathbf{x}) >$ Number of parameters for $p(y|\mathbf{x})$

Discriminative classifiers need lesser training examples to for PAC learning than generative classifiers

Logistic Regression

- ▶ $y|\mathbf{x}$ is a *Bernoulli* distribution with parameter $\theta = \text{sigmoid}(\mathbf{w}^\top \mathbf{x})$
- ▶ When a new input \mathbf{x}^* arrives, we toss a coin which has $\text{sigmoid}(\mathbf{w}^\top \mathbf{x}^*)$ as the probability of heads
- ▶ If outcome is heads, the predicted class is 1 else 0
- ▶ Learns a linear boundary

Learning Task for Logistic Regression

Given training examples $\langle \mathbf{x}_i, y_i \rangle_{i=1}^D$, learn \mathbf{w}

- ▶ MLE Approach
- ▶ Assume that $y \in \{0, 1\}$
- ▶ What is the likelihood for a bernoulli sample?
 - ▶ If $y_i = 1$, $p(y_i) = \theta_i = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x}_i)}$
 - ▶ If $y_i = 0$, $p(y_i) = 1 - \theta_i = \frac{1}{1 + \exp(\mathbf{w}^\top \mathbf{x}_i)}$
 - ▶ In general, $p(y_i) = \theta_i^{y_i} (1 - \theta_i)^{1 - y_i}$

Log-likelihood

$$LL(\mathbf{w}) = \sum_{i=1}^N y_i \log \theta_i + (1 - y_i) \log (1 - \theta_i)$$

- ▶ No closed form solution for maximizing log-likelihood

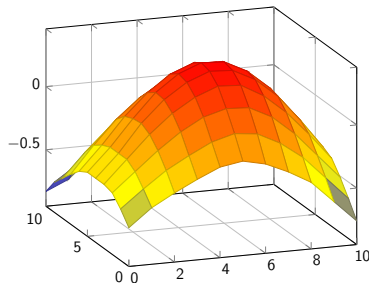
Using Gradient Descent for Learning Weights

- ▶ Compute gradient of LL with respect to \mathbf{w}
- ▶ A convex function of \mathbf{w} with a unique global maximum

$$\frac{d}{d\mathbf{w}} LL(\mathbf{w}) = \sum_{i=1}^N (y_i - \theta_i) \mathbf{x}_i$$

- ▶ Update rule:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \eta \frac{d}{d\mathbf{w}_k} LL(\mathbf{w}_k)$$



Using Newton's Method

- ▶ Setting η is sometimes *tricky*
- ▶ Too large – incorrect results
- ▶ Too small – slow convergence
- ▶ Another way to speed up convergence:

Newton's Method

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \eta \mathbf{H}_k^{-1} \frac{d}{d\mathbf{w}_k} LL(\mathbf{w}_k)$$

What is the Hessian?

- ▶ Hessian or **H** is the second order derivative of the objective function
- ▶ Newton's method belong to the family of **second order optimization algorithms**
- ▶ For logistic regression, the Hessian is:

$$H = - \sum_i \theta_i (1 - \theta_i) \mathbf{x}_i \mathbf{x}_i^T$$

Regularization with Logistic Regression

- ▶ **Overfitting** is an issue, especially with large number of features
- ▶ Add a *Gaussian prior* $\sim \mathcal{N}(\mathbf{0}, \tau^2)$
- ▶ Easy to incorporate in the gradient descent based approach

$$LL'(\mathbf{w}) = LL(\mathbf{w}) - \frac{1}{2}\lambda \mathbf{w}^\top \mathbf{w}$$

$$\frac{d}{d\mathbf{w}} LL'(\mathbf{w}) = \frac{d}{d\mathbf{w}} LL(\mathbf{w}) - \lambda \mathbf{w}$$

$$H' = H - \lambda I$$

where I is the identity matrix.

Handling Multiple Classes

- ▶ $p(y|\mathbf{x}) \sim \text{Multinoulli}(\boldsymbol{\theta})$
- ▶ Multinoulli parameter vector $\boldsymbol{\theta}$ is defined as:

$$\theta_j = \frac{\exp(\mathbf{w}_j^\top \mathbf{x})}{\sum_{k=1}^C \exp(\mathbf{w}_k^\top \mathbf{x})}$$

- ▶ Multiclass logistic regression has C weight vectors to learn

Bayesian Logistic Regression

- ▶ How to get the posterior for \mathbf{w} ?
- ▶ Not easy - *Why?*

Laplace Approximation

- ▶ We do not know what the true posterior distribution for \mathbf{w} is.
- ▶ Is there a close-enough (approximate) Gaussian distribution?



A. Y. Ng and M. I. Jordan.

On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes.

In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *NIPS*, pages 841–848. MIT Press, 2001.



V. Vapnik.

Statistical learning theory.

Wiley, 1998.