Statistical Machine Learning

Linus Falk

November 18, 2022

0 Contents

Lecture 2: Linear regreesion

wednesday 02 nov 10:15

1 Linear regression discussion points

Different reasons we look at the square error, will be discussed.

Cost landscape in parameter space.

minimizes can be hard to find

- In what type of problems is the squared prediction error an unsuitable performance measure:

It symmetric, we maybe don't want to under overestimate the outcome/prediction. Think of medication or battery. Another way yo do it pinball loss. two alternatives

$$L(x,y,\theta) = \left[Pinballloss\right](y - f(x,\theta))\alpha, y \geq f(x,\theta) or \\ (y - f(x,\theta))(1-\alpha), y < f(x,\theta)) \tag{1}$$

- How would you visualize a regression model with two inputs.

A plane for a linear regression model. A surface lives in a subspace of the d parameters space.

- **When is there a unique linear regression model that minimizes the average squared-error loss**

IF we get to little data we cant generate a unique model for an example. Think of just having one data point and fitting a line to it. All interpolate the training data

Necessary for uniqueness that $n \geq (d+1)$, (not sufficient)

Linear regression model

$$\overline{X} = \begin{bmatrix} -x_1^T - \\ \vdots \\ -x_n^T - \end{bmatrix} \tag{2}$$

$X^T X$ is invertible. (d+1) when they are linearly independent we can find a unique solution. (rand($\overline{X}$) = d+1)

- 4

Parameter space Numerical search e.g. gridding. Evaluate J at every gridpoint. Might not find the actual minimum. Might work when d i small ≤ 3. Another way is gradient search.

- Consider alternative ways to regularize the least-square method.

$J(\theta) = 1/nnorm(\overline{y} - \overline{X\theta})$

Positive quadratic function in θ (convex) local minimum $\rightarrow$ global minimum of J(θ)

Local min (*) $\nabla_0 J(\theta) = 0 <-> \overline{X^T}\overline{X} = X^T\overline{y}$

**Notes if small amount of data, choose fewer d, but which one to choose?

Different norms for example norm 2 for ridge regression affects the cost. Lasso is another example using norm 1.

Regularization reduces the sensitivity.

- How does one interpret the 'best' regression function?

Finding the balancing point of the distribution for each point.

- In what types of problem can it still produce poor performance

Balancing point between two "hills" give a prediction between them. Data will never come there. Multi modal distribution. The population splits into two parts. In the blood pressure it could be difference between male and female there could therefore be necessary add a new feature.

Family of Gaussian distributions

Why not model the best regression function

Lecture 3: Classification

tuesday 08 nov 10:15

2 Classification

We return to the example with blood pressure and cholesterol. We got the patient data

- x_1 change in blood pressure during exercise
- x_2 total cholesterol
- y stroke within 5 years {-1, +1}

Given this data x_*, we want to predict strokes within five years y_* using the training data set.

The expected new error of a model

At an unknown future point, the miss classification error of the model is

$$\mathbf{1}\{y_* \neq f(x_*; \theta)\} \tag{3}$$

This is easy to interpret and easy to analyze. We now want to find θ to minimize our expected error:

$$\mathbf{E}\left[\mathbf{1}\{y_* \neq f(x_*; \theta\}\right] \tag{4}$$

but as in previous case p(x,y) is Unknown! We return to:

Learning a linear classifier

We can span a linear half plane by

$$\{x \quad x^T\theta \quad \geq 0\} \tag{5}$$

for a given vector θ

Our linear classifier:

$$f(x;\theta) = \text{sign}(x^T\theta) = \begin{cases} +1, & x^T\theta \geq 0 \\ -1, & x^T\theta < 0 \end{cases} \tag{6}$$

Learn a model, the θ parameters by minimizing the cos function (average loss)

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(x_i, y_i; \theta) \tag{7}$$

with the aim to reduce the expected error from a new observation (new error)

$$\mathbf{E}_{new\theta}) = \mathbf{E}_* \left[E(f(x_*;\theta), y_*)\right] \tag{8}$$

Computational challenge

This is a challenge to compute and it doesn't take the consideration of how close this classifier lines is put to observations. Average loss:

$$J(\theta) = \frac{1}{n}\sum_{i=0}^{n} L(x_i, y_i; \theta), \text{where} L(x_i, y_i; \theta) = \mathbf{1}\{y \neq \text{sign}(x^T\theta)\} \tag{9}$$

Loss function and classifier margin

The margin of the (linear) classifier is defined as:

$$y \times x^T\theta \tag{10}$$

Comparing the miss classification loss functions. The second, logistic loss:

$$\begin{gathered} L(x, y; \theta) = \begin{cases} 0, & yx^T \geq 0 \\ 1, & yx^T\theta < 0 \end{cases} \\ L(x, y; \theta) = \ln\left[1 + e^{-yx^T\theta}\right] \end{gathered} \tag{11}$$

The linear classifier f(x; θ) = sign($x^t\theta$) learned by minimizing

$$J(\theta)\frac{1}{n}\sum_{i=0}^{n} L(x_i, y_i; \theta) \tag{12}$$

where the logistic loss is a convex function of θ(can be minimized). Like in previous regression problems we can use regularization to reduce the sensitivity of the learned model $\hat{\theta}$. We may regularize the cost by:

$$J(\theta) + \lambda||\theta||_2^2 \tag{13}$$

Discussion points* Classification

- **n what type of problems is the missclassification error an unsuitable performance measure?**
  Bad error method when it's not symmetric, could be better to classify risk of stroke instead of not. In medicine there is often a big asymmetry. Two types of error.

- **How does the missclassification loss of a linear classifier change if you rescale the parameters θ ?**
  It doesn't change, linear.

- **How does the logistic loss increase with the (negative) margin of missclassified points.**

$$L(x, y; \theta) = \ln \left[1 + e^{-yx^T\theta}\right] \approx -y \times x^T\theta \tag{14}$$

  Linearly for very "negative" numbers. LOOK at this again

- **The logistic loss is convex in θ. What does this imply in parameter space?**
  Two theories, two universes, convex and not convex in optimization.

$L(x, y; \theta)$ convex function $\rightarrow J(\theta)\frac{1}{n}\sum_i^n L(x, y; \theta)$
also convex func in $\theta J(w\theta + (1-w)\theta') \leq wJ(\theta) + (1-w)J(\theta')$
$\rightarrow$ all minima of J, arg min J. Form a convex set

The best classifier

Search through all functions f(x) that can minimize the expected value of an new observation:

$$E_{new} = \mathbf{E}_* \left[\mathbf{1}\{y_* \neq f(x_*)\}\right] \tag{15}$$

The conditional distribution for y=+1: p(y=1|x)
The model that minimizes $E_{new} = \mathbf{E}_* \left[\mathbf{1}\{y_* \neq f(x_*)\}\right]$ is given by the conditional distribution:

$$\mathrm{f}_0(x) = \mathrm{argmax} p(y|x) \tag{16}$$

Discussion point* The best classifier

- **The plot illustrates p(y = 1|x), how does p(y = -1|x) look? How does p(x) look?**

$$p(y = -1|x) = \left[\text{complementary event}\right] = 1 - p(y = 1|x) \tag{17}$$

  p(x) probability were there might be data points.

- **How does one interpret the 'best' classifier?**
  what does this arg max p(y|x) mean. y $\in \{-1, 1\}$

$$\begin{cases} +1, & p(1|x) > p(-1|x) \\ -1, & p(1|x) \leq p(-1|x) \end{cases} \tag{18}$$

  Alternative 1:

$$p(1|x) > 1 - p(1|x) \leftrightarrow p(1|x) > \frac{1}{2} \tag{19}$$

  Alternative 2, threshold:

$$\frac{p(1|x)}{(0|x)} > 1 \rightarrow \text{threshold } \frac{p(1|x)}{(0|x)} > \text{T} \tag{20}$$

- **In what types of problems can it still produce poor performance?**
  Same answer as before, medical example. Missing out stroke patients is serious.

Alternative loss: the likelihood perspective

The best classifier $f_0(x)$ depends on: p(y|x). Let us now model that directly. We have the family of distribution models:

$$p(y|x;\theta) = \frac{e^{yx^T\theta}}{1 + e^{yx^T\theta}} \tag{21}$$

Which gives a model for $f_0(x)$ also known as logistic regression

How surprising is the training data?

For the model θ, the surprise of the training data point $(x_i, y_i$ is given by:

$$L(x_i, y_i; \theta) = -\ln p(y_i|x_i;\theta) \tag{22}$$

which is also known as the negative log-likelihood loss

$$\hat{\theta} = \arg\min \frac{1}{n} \sum_{i=1}^{n} L(x_i, y_i; \theta) \tag{23}$$

that is the maximum likelihood model of conditional distribution

$$p(y|x;\hat{\theta}) = \left[1 + e^{-yx^T\hat{\theta}}\right]^{-1} \tag{24}$$

For logistic distribution model, $\hat{\theta}$(training data)) matches the logistic loss minimizer.

Discussion points* Alternative loss

- **How does one interpret the parameters of a linear model?**
  $f(x;\theta) = x^T\theta \rightarrow \theta_0 + \theta_1 x_1 \ldots$
  what mean θ_1 models association with between x_1 and y when all other x_i are fixed. Logistic distrubution:

$$ln(\frac{p(y=1|x)}{p(y=-1|x)}) = x^T\theta \tag{25}$$

  θ affects the low odds for 'stroke'.

  Special case, classes are linearly separable.

$$L = \ln\left[1 + e^{-yx^T\theta}\right] \tag{26}$$

  Can make the loss smaller and smaller by blowing up the parameters. Making it sharper and sharper. J(θ) has no minimal point.

- **Why is the surprisal equivalent to the logistic loss?**

$$p(y|x) = \frac{e^{yx^T\theta}}{1 + e^{yx^T\theta}} = \frac{1}{e^{-yx^T\theta} + 1} = \left[1 + e^{-yx^T\theta}\right]^{-1} \tag{27}$$

$$L = -\ln p(x, y; \theta) = -\ln\left[1 + e^{-yx^T\theta}\right]^{-1} = \ln\left[\ldots\right] \tag{28}$$

  negative log likelihood

- **What are some advantages/disadvantages of using parametric distributional modelling?**
  Instead of hard predictions we can get a probability. Some times called soft classification. The cons are : its an uncalibrated model. Gap in predictions.

- **How does one extend the model to handle M > 2 classes?**
  The idea how to do it:

$$\begin{aligned} y &\in 1, 2, \ldots, M \\ *\ln\frac{p(y=1|x)}{p(y=M|x)} &= x^T\theta_1 \\ \ln\frac{p(y=2|x)}{p(y=M|x)} &= x^T\theta_2 \\ &\vdots \\ \ln\frac{p(y=M-1|x)}{p(y=M|x)} &= x^T\theta_{M-1} \end{aligned} \tag{29}$$

$$**1\sum_{y=1}^{M} p(y|x)) = 1 \tag{30}$$

$$(*) \rightarrow p(y=m|x) = p(y=M|x)e^{x^T\theta_m} = \frac{e^{x^T\theta_m}}{1 + \sum_{k=1}^{M-1} e^{x^T\theta_k}} \tag{31}$$

Lecture 4: Classification

wednesday 09 nov 10:15

Material from 2020 course in SML at Uppsala university

3 Multivariate Gaussian density

The p-dimensional Gaussian probability density function with mean vector μ and covariance matrix Σ is,

$$N(x|\mu, \Sigma = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{32}$$

where $\mu : p \times 1$ vector and $\Sigma : p \times p$ positive definite matrix.
If we let $\mathbf{x} = (x_1, \ldots, x_p)^T tilde N(\mu, \Sigma)$

- μ_j is the mean of x_j
- Σ_{jj} is the variance of x_j
- $\Sigma_{ij}(i \neq j)$ is the covariance between x_i and x_j

4 Linear discriminant analysis, LDA

Here we need,

- The prior class probabilities π_m, p(y=m), m $\in \{1, ..., m\}$
- The conditional probability densities of the input x, $f_m(x)$, p(x |y=m) for each class m.

This will give us the model:

$$g_m(x) = \frac{\pi_m f_m(x)}{\sum_{m=1}^{M} \pi_m f_m(x)} \tag{33}$$

For **first task** a natural estimator is the proportion of training samples in the m th class.:

$$\hat{pi}_m = \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}\{y_m = m\} = \frac{n_m}{n} \tag{34}$$

where n is the size of the training set and n_m the number of training samples of class m.

for the **second task** a simple model is to assume that $f_m(x)$ is a multivariate normal density with mean vector μ_m and covariance matrix Σ_m

$$f_m(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_m)^T \sum_m^{-1}(x-\mu_m)} \tag{35}$$

if we further assume that all classes share the same covariance matrix,

$$\Sigma = \text{def}\Sigma_1 = \ldots = \Sigma_M \tag{36}$$

the remaining parameters of the model are: $\mu_1, \mu_2, \ldots, \mu_M, \Sigma$

These parameters are naturally estimated as the sample means and sample covariance, respectively:

$$\begin{aligned} \hat{\mu}_m &= \frac{1}{n_m} \sum_{i:y_i=m} x_i \quad m = 1, \ldots, M \\ \hat{\Sigma} &= \frac{1}{n-M} \sum_{m=1}^{M} \sum_{i:y_i=m} (x_i - \hat{\mu}_m)(x_i - \hat{\mu}_m)^T \end{aligned} \tag{37}$$

Modeling the class probabilities using the normal assumptions and these parameter estimates is referred to as **Linear Discriminant Analysis (LDA)**

The LDA classifier assigns a test input x to class m for which

$$\hat{\delta}_m = x^T \hat{\Sigma}^{-1} \hat{\mu}_m - \frac{1}{2} \hat{mu}_m + \log \hat{\pi}_m \tag{38}$$

is largest, where

$$\begin{aligned} \hat{\pi}_m &= \frac{n_m}{n}, \quad m = 1, \ldots, M \\ \hat{\mu}_m &= \frac{1}{n_m} \sum_{i:y_i=m} (x_i - \hat{\mu}_m)(x_i - \hat{mu}_m)^T \\ \hat{\Sigma} &= \frac{1}{n-M} \sum_{m=1}^{M} \sum_{i:y_i=m} (x_i - \hat{\mu}_m)(x_i - \hat{\mu}_m)^T \end{aligned} \tag{39}$$

4.1 Quadratic discrimination analysis

Question: do we have to assume a common covariance matrix? **No**, estimating a separeate covariance matrix for each class leads to the method: Quadratic discrimination analysis or QDA for short. Which one to chose has to do with the bias-variance trade of or in other words the risk of over or under-fitting. Comparing LDA to QDA:

- has more parameters
- is more flexible
- has higher risk of overfitting (large variance)

Example: Difference between LDA and QDA

- **If the optimal boundary is linear, do we expect LDA or QDA to perform better on the training set? What do we expect on the test set?**
  We can always assume that QDA performs better to the test set since it is more flexible. If the optimal decision boundary is linear LDA will perform better on test data since it would not overfit.
- **If the optimal decision boundary is nonlinear, do we expect LDA or QDA to perform better on the training set? What do we expect on the test set?**
  If the optimal decision boundary is non linear We expect QDA to perform better on the test set.
- **In general, as the sample size n increases, do we expect the test error rate of QDA relative to LDA to increase, decrease or be unchanged? Why?**
  We can assume that the QDA error rate will reduce when n increases, with more n it will come close and closer to the optimal decision boundary while when n is small the risk of overfitting is increasing.
- **True or false: Even if the optimal decision boundary for a given problem is linear, we will probably achieve a smaller test error rate using QDA rather than LDA because QDA is flexible enough to model a linear decision boundary. Justify your answer.**
  False if n is small we have a risk of overfitting

  Questions copied from UU sml course material

5 Parametric and non parametric models

The models we have discussed earlier have all been of the type parametric models: linear regression, logistic regression, LDA and QDA. These models are all parametrized with a "fixed-dimensional parameter". We are now taking a look at Non-parametric models that allow the model to grow with the amount of data that are available.

5.1 k-NN

k-nearest neighbor is an example of non-parametric model. It classifies a test input x to a class according the k training samples that are nearest to our input x.

Lecture 5: Bias-variance trade-off and cross validation

monday 14 nov 10:15

6 Evaluating a supervised machine learning method

A machine learning model is not suppose to give only good result on the training data set, it should also give good result on unseen data out in the "wild"/in production. How we ensure this or try to ensure this befor putting it into production is described in this lecture.

The confusion matrix revisited

You always have predicted condition and true condition. 4 possible values but many different measurements. True positive, false negative, false true and so on. False positive rate, true positive rate are those in ROC plot.

$$\begin{aligned} \text{True positive rate: } TPR &= \frac{TP}{P} = \frac{TP}{FN+TP} \in [0, \quad 1] \\ \text{False positive rate: } FPR &= \frac{FP}{N} = \frac{FP}{FP+TN} \in [0, \quad 1] \\ \text{Precision: Prec} &= \frac{TP}{P^*} = \frac{TP}{FP+TP} \in [0, \quad 1] \end{aligned} \tag{40}$$

AUC

Area under curve or AUC, a performance measure for classifier, here taking all possible thresholds into account (can apply to ROC and Precision-recall curve). The area under the curve gives an idea how accurate the model is.

Another important curve/plot to look at to determine model performance is the precision recall curve.

Loss functions are used in learning

Loss is the difference between model prediction and model outcome. In parameterize methods we have loss function. During training we minimize the average loss on our training data. Some examples of loss functions:

$$\begin{aligned} \text{Linear regression: } & L(\hat{y}(x;\theta),y))(y_i - \hat{y}(x;\theta)))^2 \\ \text{Logistic regression: } & L(\hat{y}(x,\theta),y) = \log(1+e^{-y\times\hat{y}(x;\theta)}) \end{aligned} \tag{41}$$

Evaluating our model then we are interested in the error function. Examples of error functions;

$$\begin{aligned} \text{Classification: } I\{\hat{y}(x)=y\}) & \begin{cases} 1, & \hat{y}=y \\ 0, & \text{otherwise} \end{cases} \\ \text{Regression: } & = (\hat{y}(x)-y)^2 \end{aligned} \tag{42}$$

The error function doesn't need to be the same as the loss function as in the linear regression case. The loss function is very method specific so we cant compare it between different method. We need this in the project to compare the result between them. That's why we need this error function.

$$E(\hat{y}(x), y) = \begin{cases} \mathbf{I}\{\hat{y} = y\} \\ \text{False Positive rate} \\ \text{Area under the curve (AUC)} \\ (\hat{y}(x) - y)^2 \end{cases} \tag{43}$$

Evaluate performance on new data. What is gonna happen when we show the model new data : E_{New}

$$E_new = \mathbf{E}_* \left[E(\hat{y}(x_*), y_*) \right] = \int E(\hat{y}(x_*), y_*) p(x_*, y_*) \, dx_* dy_* \tag{44}$$

But we dont know what $p(x_*, y_*)$ is. Maybe we can learn that from the data we got.

We can approximate integrals using enough samples (law of large numbers etc). Remember that its important to use data that is samples comes from real world distribution or production. Because it is this distribution we want to approximate. But can we approximate the E_{New} with our training data... NO! because then we use the same distribution twice. Take the polynomial fitting as an example.

Holding out

We start with splitting our data into two groups: training data T and validation data. Then we have our new E_{New} :

$$E_{New} \approx E_{hold-out} = \frac{1}{n_p} \sum_{i=1}^{n_v} E(\hat{y}(x_i; T), y_i) \tag{45}$$

This is a good method because its easy to implement but it requires our validation set to be big to get a good approximation of our E_{New} but at the same time we need a large set to train on to get good predictions. Another down side is that we don't use all our data for training.

Important!

Split randomly between training and validation data

k-fold cross validation

Keep testing over and over again. We split the data into k batches and hold out batch L when estimating the model. Then use L to estimate E_{New} and average over all k estimates. A 90 10 split is a good start:

$$E_{New} \approx E_{k-fold} \frac{1}{k} \sum_{\epsilon=1}^{k} E_{hold-out}^{(\epsilon)} \tag{46}$$

This gives a better approximation of E_{new} but is computational heavy. This method is for one method at a time, not for comparing. This method is good to compare how well our hyper parameters are chosen. Like k in nearest neighbors or λin regularization. But !

We cant use E_{k-fold} to estimate E_{New}

We set aside a test set and use this **ONLY** to estimate E_{New}

We cant calculate E_{New} exactly, only approximate it.

$$\begin{aligned} \overline{E}_{train} &= \mathbf{E}_T\left[E_{train}\right] \\ \overline{E}_{New} &= \mathbf{E}_T\left[E_{New}\right] \end{aligned} \tag{47}$$

Where $\overline{\mathbf{E}}_T\left[\cdot\right]$ is the average over **Training data** T. So note here! k-fold cross-validation approximates $\overline{E}_{New}$ rather than E_{New} and it is usually so that:

$$\overline{E}_{train} < \overline{E}_{New} \tag{48}$$

We can from this define a gap between these: the generalization gap

$$\overline{E}_{New} = \overline{E}_{train} + \text{generalization gap} \tag{49}$$

Model complexity is the models ability to adapt to pattern in the data.

Higher complexity giver a lower training error but increases the generalization gap. Lower complexity gives lower generalization gap but increases training error. Think of the polynomial example again.

7 Bias and variance

Bias is the failure to capture the reality with our model. The variance is due to inherent randomness in the world. If z_0 is the real position and $\overline{z}$ is our measured average then $\overline{z} - z_0$ is our bias. The variance is because of noise in the measurements, so the variance is the expected error: $\mathbf{E}\left[(z - \overline{z}^2)\right]$.

All of our measurements errors can be broken down to two errors:

$$\begin{aligned} \mathbf{E}\left[(z - z_0)\right]^2 = \mathbf{E}\left[((z - \overline{z}) + (\overline{z} - z_0))^2\right] = \\ \mathbf{E}\left[(z - z_0)^2\right] + 2(\mathbf{E}\left[z\right] - \overline{z})(\overline{z} - z_0) + (\overline{z} - z_0)^2 \end{aligned} \tag{50}$$

Since the middle term is zero we are left with the variance and the bias.

Bias-variance decomposition

We can use the decomposition of the error to our prediction, $\overline{E}_{New}$

$$\overline{E}_{New} = \mathbf{E}_*\left[\mathbf{E}_T\left[\hat{y}(x_*;T) - \overline{f}(x_*))^2\right]\right] + \mathbf{E}_*\left[(\overline{f}(x_*) - f_0(x_*))^2\right] + \sigma^2 \tag{51}$$

- **Bias** due to that the model cannot represent the true f_0 $\mathbf{E}_* \left[(\overline{f}(x_*) - f_0(x_*))^2 \right] + \sigma^2$
- **Variance** due to variability in the training data $\mathbf{E}_* \left[\mathbf{E}_T \left[\hat{y}(x_*; T) - \overline{f}(x_*) \right)^2 \right] \right]$
- σ Irreducible error, always some noise in the background

In other words: the bias is the inability of a method to describe the complicated patterns we want to describe. Low model complexity. Variance is how sensitive a method is to the training data.