30. What is the curse of dimensionality? Why it can be problem?

We often deal with the problems in high-dimensional vector spaces using the intuition from our 3-dimensional space and it often works well in the pure vector space. However, not all intuitions developed in spaces of low dimensionality can generalize to spaces of many dimensions. In some cases, our intuitions formed in a space of three dimensions, can fail badly in the situation of spaces of higher dimensionality, (especially when more structure like inner product or metric are added to the vector space). The problem of the fraction of the volume of the sphere in a D-dimensional space lies between radius r = 1 and r = 1- is an example.



(lec record week4a around 8:15)

It will be a big problem for the Linear Basis Function Model. For this model, in one dimension case, we choose 10 basis functions to model the data, which is not complicated. However, in the D dimension case, we have 10 basis functions in each direction, then we need number of basis functions to cover the whole space. That is the number of basis functions grows rapidly, often exponentially, with the dimensionality D, even with small number of basis functions in low-dimension case.



31. What are the three models for decision problems? How they are different?

(a) Discriminant function

Find a function f(x) which maps each input x directly onto a class label. In this case, probabilities play no role.

(b) Discriminative models

Firstly solve the **inference**(?) problem of determining the posterior class probabilities , and then use decision theory to assign each new x to one of the classes.



In this approach, we directly model the posterior probabilities

(c) Generative models

We should derive the posterior class probabilities instead of directly modeling them, and then still use decision theory to assign class label to each new x.

There are two approaches:

(1) Directly model the joint distribution and then normalize to acquire the posterior probabilities.



(2) Firstly solve the inference problem of determining the class-conditional densities for each class. Meanwhile, solve the inference problem of determining the prior class probabilities . Then use Bayes’ rule to the derive the posterior class probabilities .



The main difference among these 3 models is on the use of probabilities.

In discriminant function approach, no probabilities are used. The posterior class probabilities are used to classify new data in both generative model and discriminative model, but there are two main differences between them:

1). The way they get the posterior class probabilities.

In the discriminative model, we directly model the posterior class probabilities .



In the generative model, we derive the posterior class probabilities from modeling class-conditional densities and class prior or joint distribution first.



2). In the generative model, we also model the distribution of input data, while the discriminative model only models the posterior class probabilities without model the distribution of input data.

32. What are the deficiencies of the least squares approach in linear classification?

Textbook P185-186

1). Least-squares solutions lack robustness to outliers, i.e. is highly sensitive to outliers, and this applies equally to the classification application. Because the sum-of-squares error function also penalizes a lot on predictions (that are ‘too correct’ in) that (they) lie a long way from (on) the correct side of the decision boundary.

2). In some multi-class classification problems that linear decision boundaries can give excellent separation between the classes, the least squares approach will misclassify most of the points from some classes. The failure of least squares is because that it corresponds to maximum likelihood under the assumption of a Gaussian conditional distribution, whereas the target vectors in 1-of-Kbinary coding scheme clearly have a distribution that is far from Gaussian.

33. What’s the idea of Fisher’s linear discriminant? Derive the fisher’s linear discriminant that projects x to where .

(Textbook P186-188)

View linear classification as dimensionality reduction, i.e. projection data in D-dimensional space onto lower-dimensional space.

For classification, we want to find a projection which maximally preserves the class separation.

Then, the idea of Fisher’s linear discriminant is to maximize a function that will give a large separation between the projected class means while also giving a small variance within each class, thereby minimizing the class overlap.

(Strictly it is not a discriminant but rather a specific choice of direction for projection of the data down to lower dimension. However, the projected data can subsequently be used to construct a real discriminant by choosing threshold in the projected space.)

Derive: (Textbook P191)

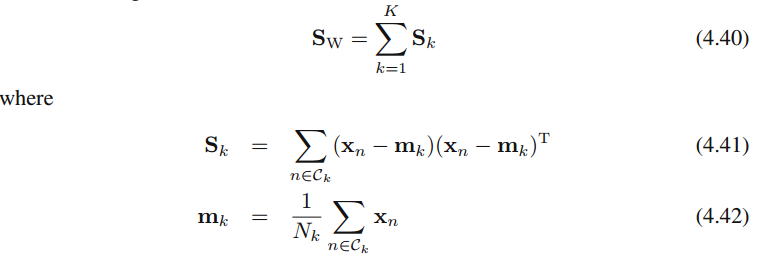
We shall assume that the dimensionality D of the input space is greater than the number K of classes. Then,



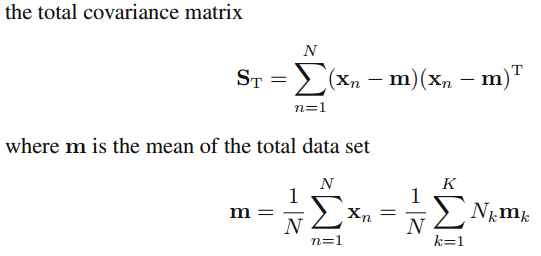
So that, we can have the projection from x to :



The within-class covariance matrix in the case of K classes:



Andis the number of data points in class

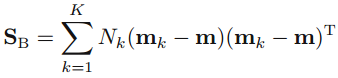


where N is the total number of data points.

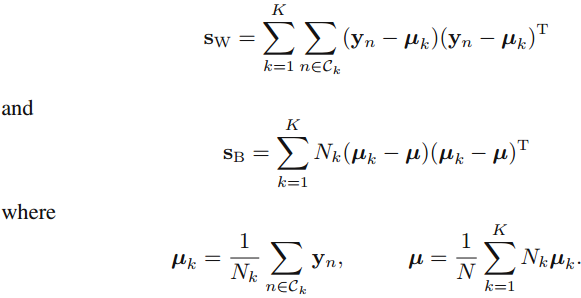
The total covariance matrix can be decomposed into the sum of the within-class covariance matrix plus an additional matrix, which we identify as a measure of the between-class covariance



Then, we can get



We can also define similar matrices in the projected D’-dimensional y-space:



Again we wish to construct a scalar criterion that is large when the between-class covariance is large and when the within-class covariance is small. We can the following as the Fisher criterion:



The weight matrix W is determined by thoseeigenvectors ofthat correspond to the D’largest eigenvalues.



The fisher’s linear discriminant is derived by the process listed above.

Outline of the derivation part

1. The projection formula

2. within-class covariance matrix for input data

3. derive between-class covariance for input data

4. define similar matrices in the projected D’-dimensional space

5. The Fisher criterion and how to get weight matrix W

34. What is the perceptron algorithm? Describe it in detail.

The perceptron corresponds to a two-class model in which feature vector φ(x), transformed from the input vector x using a fixed nonlinear transformation, (also including the bias ) and the generalized linear model of the form

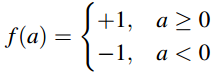


,

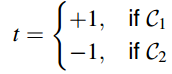


where the nonlinear activation function is

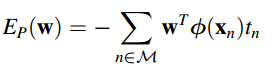
,



are used. Then, for perceptron, target coding scheme is



Using this target coding scheme, we will have for patterns correctly classified but this quantity <0 for misclassified patterns. Therefore, we sum up the quantity of all misclassified data points to define the error function, perceptron criterion, as



wheredenotes the set of all misclassified patterns. (It only sums up over misclassified data points)



After that, we try to minimize this error function, to get the value of w. We apply the stochastic gradient descent algorithm to this error function. The change in the weight vector w in one step for a misclassified pattern is then given by



Because the perceptron function y(x,w) is unchanged if we multiply w by a constant, we can set the learning rate parameter η equal to 1 and get



Then, the perceptron learning algorithm can have a simple interpretation. After we initialize the weight vector w with some value, we cycle through the training patterns in turn, and for each patternwe evaluate the function we defined above. If the pattern is correctly classified, then the weight vector remains unchanged, whereas if it is incorrectly classified, then for classwe add the vector φ()onto the current estimate of weight vector w while for class we subtract the vector φ() from w. We repeat these steps until the weight vector converges.



From the statement of perceptron convergence theorem, if the training data set is linearly separable, the perceptron learning algorithm is guaranteed to find an exact solution in a finite number of steps. However, for data sets that are not linearly separable, the perceptron learning algorithm will never converge.

Outline

1. The two-class model of perceptron algorithm

2. The error function

3. Applying stochastic gradient descent, the change of w in each step

4. A simple interpretation

5. Perceptron convergence theorem

35. What is the probabilistic generative model? Describe it in detail.

I think this is the problem for describing the probabilistic generative model for linear classification.

Textbook P43

Generative models:

We should derive the posterior class probabilities instead of directly modeling them, and then use decision theory to assign class label to each new x based on the posterior class probabilities.

There are two approaches to deriving the posterior class probabilities:

1) Directly model the joint distribution and then normalize to acquire the posterior probabilities.



2) Firstly solve the inference problem of determining the class-conditional densities for each class. Meanwhile, solve the inference problem of determining the prior class probabilities . Then use Bayes’ rule to the derive the posterior class probabilities .

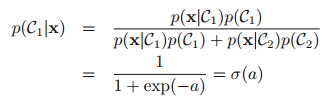


Textbook P197-198

Probabilistic Generative model for linear classification:

Here we shall adopt the generative approach in which we model the class-conditional densities p(x|Ck), as well as the class priors p(Ck), and then use these to compute posterior probabilities p(Ck|x)through Bayes’ theorem.

Consider first of all the case of two classes. The posterior probability for classC1can be written as



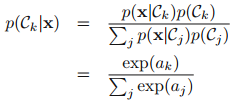
where



and is the logistic sigmoid function



For the case of multi-class, we have

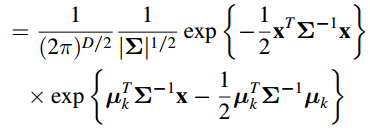


where



This is known as the normalized exponential, (also known as softmax function) and can be regarded as a multiclass generalization of the logistic sigmoid.

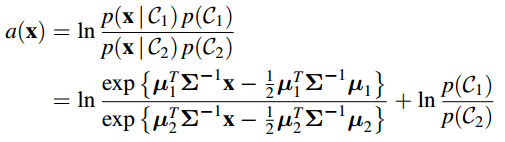
Then, we can assume that the class-conditional densities are Gaussian and all classes share the same covariance matrix. The density for class Ck is then



Then, for two classes,



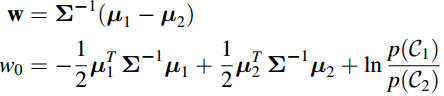
where



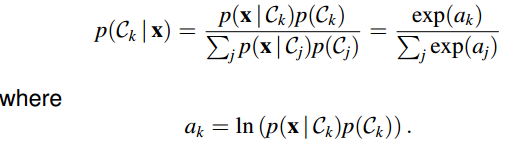
Then, we can eventually have



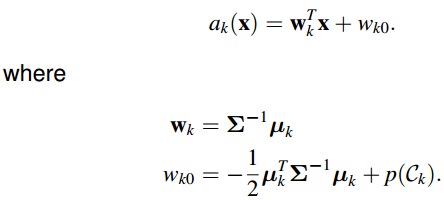
where



For the general cases of K classes using the normalized exponential

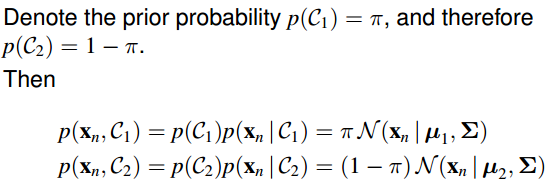


We can have



Then, we use maximum likelihood to determine the values of parameters of the class-conditional densities and prior class probabilities.

For the two-class case, given a data set {xn, tn}, n=1,2,…,N where tn=1 for class C1 and tn=0 for class C2.



Thus, we can have the likelihood function in the form



Consider first the maximization with respect toπ. The terms in the log likelihood function that depend onπare



Setting the derivative with respect toπequal to zero, we can have



whereN1denotes the total number of data points in classC1, andN2denotes the total number of data points in classC2.This result is easily generalized to the multiclass case where again the maximum likelihood estimate of the prior probability associated with class Ck is given by the fraction of the training set points assigned to that class.

Then, do maximization with respect to . We can pick out of the log likelihood function those terms that depend on and have



Setting the derivative with respect toµ1to zero, we obtain

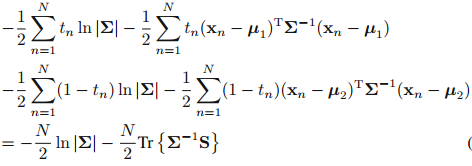


which is simply the mean of all the input vectors xn assigned to classC1. Similarly, we can also have

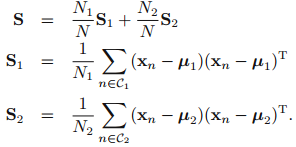


In the multi-class cases, we can also have the maximum likelihood solution for the mean of the Gaussian class-condition densities for class Ck is the mean of all the input vectors xn assigned to class Ck.

Finally, consider the maximum likelihood solution for the shared covariance matrix Σ. Picking out the terms in the log likelihood function that depend on Σ, we can have



where



Using the standard result for the maximum likelihood solution for a Gaussian distribution, we can have



which represents a weighted average of the covariances of the data associated with each class, in which the weighting coefficients are given by the prior probabilities of the classes.

In the multi-class case, we can have similar results of the maximum likelihood solution of the shared covariance matrix of the Gaussian class-condition densities.

Outline for the generative model for linear classification:

1. The basic model

2. Fit Gaussian to class-conditional densities

3. The maximum likelihood solution for Gaussian class-conditional densities

36. What is logistic regression? Describe it in detail?

(In my opinion, logistic regression model is a kind of discriminative model, i.e. directly using the functional form of the posterior class probabilities , and then determining its parameters directly by maximum likelihood. That is different from the method of generative modeling that firstly determining the parameters of class-conditional densities and class priors separately and then applying Bayes’ rule to get the posterior class probabilities .)



(To be deleted)

For the two-class classification problem of classifying data into class or class , logistic regression is that writing the posterior probability of class as this functional form , a logistic sigmoid function activating on a linear function of feature vector , and directly determining the parameters w of this model instead of the generative modeling that firstly determining the parameters of class-conditional densities and class prior, then using the Bayes’ rule to get the posterior class probability.



(For the two-class classification problem, the posterior class probability of can be written as



a logistic sigmoid function acting on a linear function of the feature vector , where is a vector of basis functions of x and w is the parameters of the model with . This model is called as logistic regression model. Actually, this is a model for classification instead of regression.



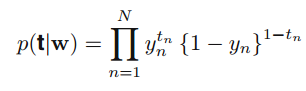
Then, we directly determine the parameters of this model instead of using the generative approach.) (To be deleted)

**(Also can probably include multi-class cases in the future)**

We can use maximum likelihood to get the parameters of the model.



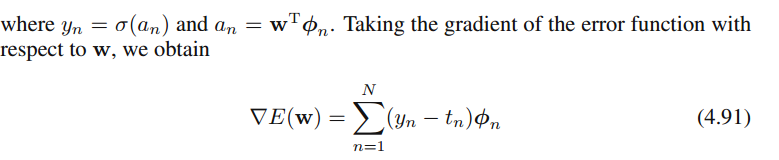
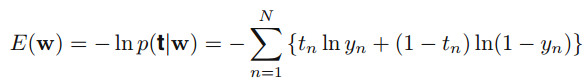
The likelihood function can be written as



Where and .



We can define the error function by taking the negative logarithm of the likelihood and have



Finally, we can use the sequential learning algorithm to get the optimal value of w. That is,

(1)



(2)



Where is the learning rate that must be chosen carefully to make the parameters vector w converge at last.



The advantage of this method is that in the 2-class classification case, if is in M dimensional space, then this model has only M parameters to determine, much less than that of the generative modeling with fitting Gaussian class conditional densities using maximum likelihood that requires the estimate of 2M parameters for the means + M(M+1)/2 parameters for the shared covariance matrix+1 parameter for the class prior.



37. What is the feature mapping in classification problems? Why we need this feature mapping sometimes?

(P204 4.3.1 para1)

Feature mapping is that, given an original input vector x, we make a fixed nonlinear transformation of the input using a vector of basis functions φ(x). Then, in classification problems, those classification models that work directly with the original input vector x can be also applicable with the feature vector φ(x). For example, as to the commonly-used generalized linear model on classification problems, for original input vector x, it is in the form,



where f() is the activation function. This model can be also applied for feature vector φ(x), with the form,



For the classification problem, sometimes, we will have nonlinear decision boundaries in the original x input space. But, by feature mapping, we may have linear decision boundaries in the projected feature space φ. Sometimes, classes that are not linearly separable in the original observation space x can be linearly separable in the feature space φ(x).

38. What is Laplace approximation? Why we need to do Laplace approximation sometimes? Describe how we do Laplace Approximation in detail.

P214

Laplace approximation is to find a Gaussian approximation to a probability density defined over a set of continuous variables.

Sometimes, especially in the case of Bayesian inference, the posterior distribution over parameters is no longer Gaussian. Then, in such circumstances, the exact Bayesian inference will be very complicated especially compared to the inference in Gaussian and sometimes even intractable. It is therefore necessary to introduce some form of approximation. Then, by Laplace approximation, especially to the posterior distribution over parameters, we can get a much easier and tractable Bayesian inference.

For a probability distribution , where Z is the normalization coefficient.



In the Laplace method the goal is to find a Gaussian approximation centered on a mode of the distribution p(z), i.e. a local maximum. Then, we can check the stationary point. At a stationary pointz0the gradient ∇f(z) will vanish.



On the other hand, a Gaussian distribution has the property that its logarithm is a quadratic function of the variables. We therefore consider a Taylor expansion of lnf(z) centred on the modez0so that we have



where Hessian matrix A is defined by



Then, we can have



The Gaussian approximation distribution q(z)is proportional to this approximation of f(z). Then, using the standard result for a normalized multivariate Gaussian, we can eventually have



This Gaussian distribution will be well defined provided A is positive definite.

To sum up, in order to apply the Laplace approximation we first need to find the mode of the original posterior distribution and then evaluate the Hessian matrix at that mode.

39. Describe Bayesian Logistic Regression in detail.

Exact Bayesian inference for logistic regression is intractable. Because evaluation of the posterior distribution would require normalization of the product of a prior distribution and a likelihood function that itself comprises a product of logistic sigmoid functions, one for every data point. Therefore, we consider the application of the Laplace approximation to the problem of Bayesian logistic regression.

Now I will describe how to get the Laplace approximation of the posterior distribution over parameters.

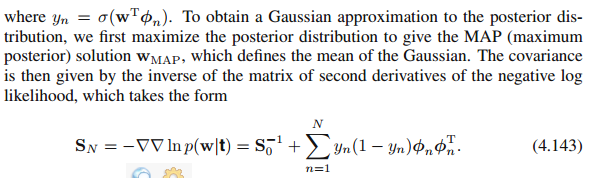
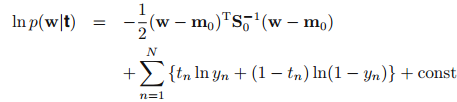
Because we seek a Gaussian approximation representation for the posterior distribution, it is natural to begin with a Gaussian prior in the form:



The posterior distribution over w is given by



Taking logarithm on both sides, together with the formula of log likelihood function of logistic regression, we can have



Then, we can get the Gaussian approximation of the posterior distribution p(w|t) in the form



Outline

1. The motivation of applying Laplace approximation to Bayesian Logistic regression

2. Derive the Laplace approximation of the posterior distribution