Machine Learning

Kiar Fatah

February 29, 2020

Contents

1 Lecture 2 **7**

1.1 Decision Tree 7

1.2 How to build a Tree 7

1.3 How to ask the best questions? 7

1.3.1 Gini impurity 7

1.3.2 Entropy 7

1.3.3 Information gain 8

1.3.4 Best question 8

1.4 Overfitting 8

1.4.1 Pruning 8

1.4.2 Validation set 8

2 Lecture 3 **8**

2.1 Introduction 8

2.2 Curse of Dimensionality 9

2.3 The Bias-Variance Trade-off 9

3 Lecture 4 **10**

3.1 Linear Regression, A Parametric Method 10

3.2 RANdom SAmpling Consensus 10

3.3 Disadvantages with RANSAC 11

3.4 k-NN Regression, A Non-parametric 11

3.5 Parametric or Non-parametric Methods? 12

3.6 Shrinkage Methods 12

3.7 Ridge Regression 12
3.8 The Lasso 13

4 Lecture 5 **13**

4.1 Axiomatic definition of probabilities 13
4.2 Random (Stochastic) Variables 13
4.3 Types of Random Variables 14
4.4 Joint Probabilities 14
4.5 Marginalization 14
4.6 Conditional Probabilities 14
4.7 Common Distributions 15
4.8 Central Limit Theorem 15
4.9 Expectation 15
4.10 General Machine Learning Problem 16
4.11 Bayes's Rule 16
4.12 Probabilistic Regression 16
4.13 Selecting the most probable hypothesis 17

5 Lecture 6 **17**

5.1 Introduction 17
5.2 Discriminative vs Generative Models 17
5.3 Parametric vs Non-parametric Inference 17
5.4 Maximum Likelihood (ML) Estimate 18
5.5 Curse of Dimensionality 18
5.6 Naive Bayes Classifier 18

6 Lecture 7 **19**

6.1 Maximum a Posteriori Estimation 19

6.2 Limitation of Linear Regression 19

6.3 Bayesian estimation 20

6.4 Bayesian Linear Regression 20

6.5 Occam's Razor 20

6.6 Limitations of Bayesian Non-parametric Methods 21

6.7 Expectation Maximization 21

6.8 EM properties 21

6.9 Lecture 8 21

7 Training a Linear Separator **21**

7.1 Percepton Learning 21

7.2 Delta rule 22

7.3 Linear Separation 22

7.4 Structural Risk Minimization 22

7.5 Support Vector Machine 23

7.6 Kernels 24

7.7 Slack 24

7.8 Lecture 10 25

7.9 Introduction 25

7.10 The Wisdom of Crowds 25

7.11 Combining Classifiers 25

7.12 Ensemble Method: Baggning 26

7.13 Ensemble Method: Forest 26

7.14 Ensemble Method: Boosting 26

7.15 Adaboost Algorithm 27

If you can't explain something in simple terms, you don't understand it.

Richard Feynman

1 Lecture 2

1.1 Decision Tree

A decision tree is a map of the possible outcomes of a series of related choices.

1.2 How to build a Tree

1. Choose the best question (according to the information gain), and split the input data into subsets.
2. Terminate: call branches with a unique class labels leaves (no need for further questions).
3. Grow: recursively extend other branches (with subsets bearing mixtures of labels).

1.3 How to ask the best questions?

The machine will automate the questions for the decision tree, however how does it obtain the best questions to ask? It does so by with the help of Gini impurity and information gain.

1.3.1 Gini impurity

Gini impurity is a measurement of uncertainty in a node. It is calculated by how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.

$$1 - \sum_i p_i^2. \tag{1.1}$$

1.3.2 Entropy

Entropy is also a measurement of uncertainty in a node and can be applied instead of Gini impurity. It is calculated as

$$\sum_i -p_i log_2 p_i. \tag{1.2}$$

The information entropy is measured in bits, corresponding to the logarithmic base 2. Tossing a coin results in the entropy 1, hence it has 1 bit of information. For a real dice and a fake dice the value is 2.58 and 2.16 bits of information respectively.

1.3.3 Information gain

Information gain is how much the question reduces the uncertainty. The information is given by the first set of values for the internal node subtracted by the average value of the impurity of the split set.

1.3.4 Best question

The question with the highest value on the information gain will be the one to ask.

1.4 Overfitting

Overfitting is when the learned models are overly specialised for the training samples. This occurs when the data is too noisy, not representative and when the model is too complex. This can be tackled by choosing a simpler model.

1.4.1 Pruning

The idea of reduced error pruning is to consider each node in the tree as a candidate for removal. A node is removed if the resulting pruned tree performs at least as well as the original tree over a separate validation dataset. The pruning is done on validation set, until it is harmful for the model.

1.4.2 Validation set

Due to pruning there is a need for validation dataset that is not training nor test data set. Therefore if there is access to a rich data set, it is separated to three categories. In essence the validation set is used to estimate prediction error for model selection (i.e. to determine hyperparameters).

2 Lecture 3

2.1 Introduction

How does one determine the right model, f, from the data? The simple concept for classification is by calculating the misclassification rate in consideration to the data. Assume the

following data is given $\mathcal{D} = (\boldsymbol{x_1}, y), ...$ is given. The misclassification rate is given by

$$err(f, \mathcal{D}) = \frac{1}{N} \sum Ind(f(\boldsymbol{x}) \neq y) \tag{2.1}$$

2.2 Curse of Dimensionality

1. Easy problems in low-dimensions are harder in high-dimensions, e.g training more complex models with limited sample data.
2. In high-dimensions everything is far from everything else, this is an issue in the nearest neighbour model, this causes issues with Nearest Neighbours.
3. Any method that attempts to produce locally varying functions in small isotropic neighbourhoods will run into problems in high dimensions.

2.3 The Bias-Variance Trade-off

Let us imagine we could repeat the modelling for many times – each time by gathering new set of training samples, D. The resulting models will have a range of predictions due to randomness in the underlying data set.

1. Variance: Refers to the amount by which $\hat{f}$, the prediction function, would change if we estimated it using a different training data set.
2. Bias: Refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model.

This can be showcased in figure 1 In essence the relationship between complexity, bias and variance can be summarised as

1. Low model complexity implies high bias, low variance.
2. High model complexity implies low bias, high variance.

Note that bias and variance are equally important as we are always dealing with a single realisation of the data set.

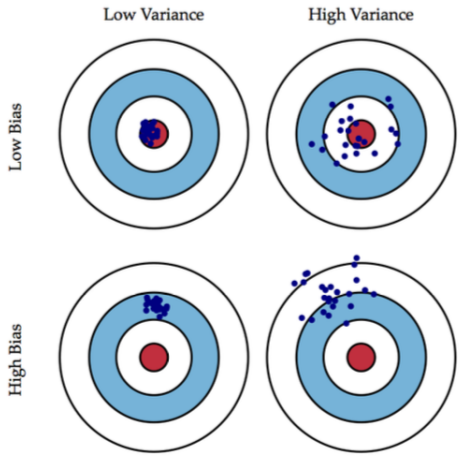

Figure 1: Variance versus Bias

3 Lecture 4

3.1 Linear Regression, A Parametric Method

Simple linear regression assumes theres a linear relationship between the output and input. Therefore a linear equation is assumed, parametric method. However, the coefficients are unknown and thus needs to be approximated.

The measurement of error is mean square. It is a convex function. Therefore the coefficients that minimises the sum of the squared error is given by the vector that sets the gradient of the sum of squared error to zero.

$$E_{in}(w) = ||Xw - Y||^2 \implies \frac{\partial E_{in}}{\partial w} = 2X^T(Xw - Y) = 0 \implies w = (X^TX)^{-1}X^TY \quad (3.1)$$

whereas w is the coefficients, X the data, Y the label and E_{in} the sum of squared errors.

3.2 RANdom SAmpling Consensus

Random sampling consensus or RANSAC is applied when the data set S contains outliers

1. Randomly select a (minimum number of) sample of s data points from S and instantiate the model from this subset.
2. Determine the set of data points S_i which are within a distance threshold t of the model. The set S_i is the consensus set of samples and defines the inliers of S.

3. If the subset of S_i is greater than some threshold T, re-estimate the model using all the points in Si and terminate
4. If the size of S_i is less than T, select a new subset and repeat the above.
5. After N trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i

In essence RANSAC takes two random data points and fits a line. This is repeated until there is a good result of data points within the margin of the fitted line. RANSAC is applied when the dataset contains outliners.

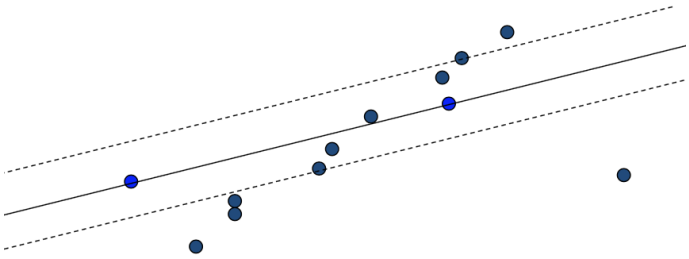

Figure 2: Line fitting between two data points.

3.3 Disadvantages with RANSAC

The threshold of the margin in the line is vulnerable.

1. If it is to high every fit is ranked equally.
2. If it is to low it leads to an unstable fit.

However, this issues are tackled within MLESAC that evaluates the quality of the consensus dataset by calculating its likelihood for better predictions.

3.4 k-NN Regression, A Non-parametric

The k-NN regression method is closely related to the k-NN classifier. Given a value for K and a prediction point x_0, k-NN regression first identiffies the K training observations that

are closest to x_0 represented by N_0. It then estimates $f(x_0)$ using the average of all the training responses in N_0. In other words

$$f(\hat{x_0}) = \frac{1}{K} \sum_{x_i \in N_0} y_i. \tag{3.2}$$

Note larger values of k provide a smoother and less variable fit (lower variance!)

3.5 Parametric or Non-parametric Methods?

1. If the parametric form is close to the true form of f, the parametric approach will outperform the non-parametric
2. As a general rule, parametric methods will tend to outperform non-parametric when there is a small number of observations per predictor (i.e. in a high dimension).
3. Interpretability stand point: Linear regression preferred to KNN if the test MSEs are similar or slightly lower.

3.6 Shrinkage Methods

A common way to fit models is by applying least square, however an alternative is it is possible to fit a model containing all p predictors using a technique that constrains or regularizes the coefficient estimates, or equivalently that shrinks the coefficient estimates towards zero.

1. Among a large number of variables X in the model there are generally many that have little (or no) effect on Y
2. Leaving these variables in the model makes it harder to see the big picture, i.e. the effect of the "important variables"
3. Would be easier to interpret the model by removing unimportant variables (setting the coefficients to zero)

3.7 Ridge Regression

Similar to least squares but minimizes different quantity

$$RSS + \lambda \sum_i w_i^2 \tag{3.3}$$

Note that the second term is called shrinkage penalty.

1. Shrinkage penalty is a small value when the coefficients w_i are close to zero.
2. The parameter λ controls the impact of the two terms. Hence the selection is critical.
3. For a higher value on λ the variance decreases and bias increases.
4. Ridge regression is harder to interpret.

3.8 The Lasso

Similar to least squares but minimizes different quantity

$$RSS + \lambda \sum_i |w_i| \tag{3.4}$$

4 Lecture 5

4.1 Axiomatic definition of probabilities

Given an sample space ω of all possible outcomes, and an event E, or set of outcomes from ω then

1. $P(E) \geq 0$ for all $E \subseteq \omega$.
2. If $E = \omega$ then the probability of ω is one. In other words the probability that one event will occur is equal to one.
3. if the events are a ocuntable sequence of pairwise disjoint events then the probability of the union of each event is equal to the probability of the summation of every event.

4.2 Random (Stochastic) Variables

A random variable is neither random nor a variable, it is a function that does exactly what we need, $X : \Omega \to \mathbb{R}$. The probability distribution function, so called pdf, of a random variable maps the range to positive numbers, $Pr(x) : X \to \mathbb{R}_{>0}$.

1. The probability of a random variables describes how the probability density is distributed over the range of X,

$$P[0 \leq X \leq 1] = \int_{x=0}^{1} Pr(x)dx. \tag{4.1}$$

2. X is distributed Pr(x) is written ore compactly as

$$X \sim Pr(x). \tag{4.2}$$

An example is

1. Ω - All possible graded exams of all students in class.
2. $X : \Omega \rightarrow \mathbb{R}$ - Random variable mapping exam outcome to score.
3. $X \sim Pr(x)$ - Gaussian distribution.

4.3 Types of Random Variables

1. Discrete random variable: Countable set
2. Continuous random variable: Uncountable set

4.4 Joint Probabilities

Consider two random variables X and Y. Observe multiple paired instances, some paired outcomes will occur frequently. This information is encoded in the joint probability density function $Pr(x, y)$.

4.5 Marginalization

The probability distribution function, PDF, of any single variable can be recovered from a joint distribution by summing for the discrete case and integrating for the continuous case.

$$Pr(x) = \sum_{y} Pr(x, y), (Discrete) \tag{4.3}$$

4.6 Conditional Probabilities

The conditional probability of X given that Y takes a value y gives

$$P(A|B) = \frac{P(A, B)}{P(B)}. \tag{4.4}$$

If it turns out the events are independent then

$$P(A|B) = \frac{P(A)P(B)}{P(B)}. \tag{4.5}$$

Bayes's theorem gives us

$$P(A|B) = P(A|B)P(B) = P(B|A)P(A) \implies P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{4.6}$$

4.7 Common Distributions

1. Bernoulli - is a discrete distribution that models binary trials.
2. Categorical - is a discrete distribution that determines the probability of observing one of k possible outcomes. Bernoulli distribution is a special case of categorical distribution.
3. Gaussian or Univariate normal distribution - is a continuous distribution.

4.8 Central Limit Theorem

The distribution of a linear combination of a large number of independent, identically distributed (iid) variables will tend to normal, regardless of the underlying distribution.

4.9 Expectation

Given a function, $f[*]$, that returns a value for each possible value x^* of the variable x and a probability $Pr(x = x^*)$ that each value of x occurs. The expected output of that fuction is calculated according to

$$\mathbb{E}[X] = \mu_X = \int xPr(x)dx. \tag{4.7}$$

it is equal to the center of gravity of a distribution. However, for different choices of the function, $f[*]$, the interpretation is different. Hence the variance can be calculated as

$$Var[X] = \sigma_X^2 = \mathbb{E}[X - \mathbb{E}[X])^2], \tag{4.8}$$

and is interpreted as the spread of a distribution. The covariance is given by

$$\sigma_{X,Y} = \mathbb{E}[X - \mathbb{E}[X]](Y - \mathbb{E}[Y])], \tag{4.9}$$

it showcases how two variables vary together.

4.10 General Machine Learning Problem

if Y is discrete then it is classification, if it is continuous it is regression.

1. Learning: We estimate Pr($\boldsymbol{x}$, y) from data.
2. Inference: We estimate $Pr(y|\boldsymbol{X} = \boldsymbol{x})$ from data.

4.11 Bayes's Rule

1. $Pr(\boldsymbol{x}|Y = y)$: Likelihood represents the probability density of observing data $\boldsymbol{x}$ given the hypothesis Y = y
2. $Pr(Y = y)$: Prior represents the knowledge about Y before any observation.
3. $Pr(y|\boldsymbol{X} = \boldsymbol{x})$: Posterior represents the probability density of hypothesis y given observation $\boldsymbol{X} = \boldsymbol{x}$.
4. $Pr(\boldsymbol{X} = \boldsymbol{x})$: Evidence describes how well the model fits the evidence.

4.12 Probabilistic Regression

Regression via conditional probability:

1. Find the joint distribution of $\boldsymbol{X}$ and Y: Pr($\boldsymbol{X}$,y)
2. Compute the posterior of Y: $Pr(y|\boldsymbol{X} = x) = \frac{Pr(\boldsymbol{x},y)}{Pr(\boldsymbol{X}=x)}$
3. Compute conditional expectation: $\mathbb{E}[Y|X = x]$.

Explicit regression model:

1. Define a deterministic model $y = f(\boldsymbol{x}) + \epsilon$.
2. Describe probability distribution of the error $\epsilon = y - f(\boldsymbol{x})$.
3. Estimate the parameters in $f(\boldsymbol{x})$.

4.13 Selecting the most probable hypothesis

Maximum A Posteriori (MAP) is about choosing hypothesis from Y with highest probability given observed data x:

$$y_{MAP}(\boldsymbol{x}) = argmax_{y \in \mathcal{Y}} Pr(\boldsymbol{X} = \boldsymbol{x}|Y = y)Pr(Y = y) \tag{4.10}$$

Maximum Likelihood (ML) is If we do not know prior distribution, then choose hypothesis with highest likelihood of generating the observed data:

$$y_{MLE} = argmax_{y \in \mathcal{Y}} Pr(\boldsymbol{X} = \boldsymbol{x}|Y = y) \tag{4.11}$$

5 Lecture 6

5.1 Introduction

Previous lecture Bayes' theorem was mentioned. However how does one go by to obtain the prior, likelihood and evidence from data?

5.2 Discriminative vs Generative Models

Discriminative modeling	**Generative modeling**
This models $Pr(y\|\vec{x}, D)$ directly	This models $Pr(\vec{x}\|y, D)$
Example: Logistic Regression	Naive Bayes

5.3 Parametric vs Non-parametric Inference

The posterior, $Pr(y|\vec{x}) = Pr(y|\vec{x}, \theta)$, distribution in consideration data is characterized by parameters θ.

Parametric Inference	**Non-Parametric Inference**
Estimate θ using data, D	Estimate $Pr(\theta\|D)$
Computes $Pr(y\|\vec{x}, \theta)$ to make inference	Compute $Pr(y\|\vec{x}, D)$ from $Pr(y\|\vec{x}, \theta, D)Pr(\theta, D)$
Learning corresponds to estimating θ	The number of parameters can grow with data
MAP & ML estimation	Bayesian methods

5.4 Maximum Likelihood (ML) Estimate

The ML estimation allows for approximation of posterior, $Pr(y|\vec{x},\theta)$, and likelihood, $Pr(\vec{x}|y,\theta)$, by finding the parameter values that make the data most likely. The $\theta_M L$ is obtained through ML optimally which is defined as maximizing the likelihood of the data, D

$$\theta_{ML} = argmax_\theta P(D|\theta) = argmax_\theta \log P(D|\theta). \tag{5.1}$$

We can then approximate distributions given the data: $Pr(y|\vec{x},\theta) \approx Pr(y|\vec{x},\theta_{ML})$ and $Pr(\vec{x}|y,\theta) \approx Pr(\vec{x}|y,\theta_{ML})$. However note that a first source of confusion is ML parameter estimation is not ML regression/classification. The second one is Maximum a Posteriori (MAP) and Maximum Likelihood (ML) classification are different

$$y_{MAP} = argmax_y Pr(y|\vec{x},\theta_{ML}), \tag{5.2}$$

$$y_{ML} = argmax_y Pr(\vec{x}|y,\theta_{ML}), \tag{5.3}$$

even with parameters θ estimated with the ML optimality criterion

$$y_{ML} = argmax_\theta Pr(D|y,\theta) = argmax_\theta \prod_n P(x_n|y_n).s \tag{5.4}$$

5.5 Curse of Dimensionality

1. Volume of feature space exponential in number of features
2. More features implies potential for better description of the objects but need more and more data to model Pr(x,y) well

However, for Naive Bayes classifier

1. All features (dimensions) regarded as conditionally independent.
2. Instead of modelling one D-dimensional distribution, model D one-dimensional distributions.

5.6 Naive Bayes Classifier

$\boldsymbol{x}$ is a vector $(x_1,...,x_D)$ of attribute or feature values and Let $\mathcal{Y}$ = 1,2,...,K be the set of possible classes. The MAP classification becomes

$$y_{MAP} = argmax_{y\in\mathcal{Y}} Pr(y|(x_1,...,x_D)) = argmax_{y\in\mathcal{Y}} \frac{Pr((x_1,...,x_D)|y)Pr(y)}{Pr((x_1,...,x_D))} \tag{5.5}$$

,

$$argmax_{y\in\mathcal{Y}}\frac{Pr((x_1,...,x_D)|y)Pr(y)}{Pr((x_1,...,x_D))} = argmax_{y\in\mathcal{Y}}Pr((x_1,...,x_D)|y)Pr(y). \tag{5.6}$$

The Naive Bayes assumption is

$$Pr((x_1,...,x_D)|y) = \prod_d Pr(x_d|y). \tag{5.7}$$

The MAP classification with Naive Bayes becomes

$$y_{MAP} = argmax_{y\in\mathcal{Y}}Pr(y)\prod_d Pr(x_d|y). \tag{5.8}$$

When to use Naive Bayes

1. Moderate or large training set available
2. Feature dimensions are conditionally independent given class (or at least reasonably independent, still works with a little dependence)

One issue with Naive Bayes is what if none of the training instances with target value y have attribute x_i? Then $Pr(x_i|y) = 0 \implies Pr(y)\prod_d Pr(x_d|y) = 0$. A simple solution to this is to add pseudocounts to all counts so that no count is zero. This is a form of regularization or smoothing.

6 Lecture 7

6.1 Maximum a Posteriori Estimation

In the previous lecture it was mentioned that ML estimation chooses θ to maximize probability of the data, D. However, MAP estimation chooses the most likely θ given the data, D.

$$argmax_\theta(logPr(\theta) + \sum logPr(x_n|\theta). \tag{6.1}$$

whereas the term $logPr(\theta)$ works as a regularizer. See the example on the presentation for more in depth application of MAP!

6.2 Limitation of Linear Regression

1. With MAP estimation, the problem shifts to defining the parameters of the prior $Pr(\vec{w})$
2. We still have uncertainty in the posterior $Pr(y|\vec{x},\vec{w}^*)$ but this is not explicit.
3. We don't want $Pr(y|\vec{x},\vec{w})$, but instead $Pr(y|\vec{x},D)$

6.3 Bayesian estimation

1. ML: $D \rightarrow \theta_{ML} \rightarrow Pr(y|\vec{x}, \theta_{ML})$
2. MAP: $D, Pr(\theta) \rightarrow \theta_{MAP} \rightarrow Pr(y|\vec{x}, \theta_{MAP})$
3. Bayes: $D, Pr(\theta) \rightarrow Pr(\theta|D) \rightarrow Pr(y|\vec{x}, D)$

Consider θ as a random variable (same as MAP). Now characterize θ with the posterior distribution $Pr(\theta|D)$ given the data and compute the new predicting posterior $Pr(y|\vec{x}, D)$ marginalizing θ (predictive posterior)

$$Pr(y|\vec{x}, D) = \int_{\theta \in \Theta} Pr(y|\vec{x}, \theta) Pr(\theta|D) d\theta \tag{6.2}$$

6.4 Bayesian Linear Regression

If the data, D, is given the model is the same as MAP. Estimating $Pr(y|\vec{x}, D)$ is through

$$Pr(y|\vec{x}, D) = \int_{\mathcal{R}^D} Pr(y|\vec{x}, D, \vec{w}) Pr(\vec{w}|\vec{x}, D) d\vec{w} = \int_{\mathcal{R}^D} Pr(y|\vec{x}, \vec{w}) Pr(\vec{w}|D) d\vec{w} \tag{6.3}$$

Note that if $Pr(\vec{w})$ is Gaussian then $Pr(w|\vec{D})$ is Gaussian. Since $Pr(y|\vec{x}, D)$ is also Gaussian. The result is in closed-form. Read more about closed form solutions in the presentation.

6.5 Occam's Razor

Choose the simplest explanation for the observed data. Importent factors are

1. number of model parameters
2. number of data points
3. model fit to the data

More complex models fit the data very well (large $Pr(D|\theta)$ and $Pr(\theta|D)$) but only for small regions of the parameter space Θ.

6.6 Limitations of Bayesian Non-parametric Methods

1. closed form solution for $Pr(\vec{w}|D)$ is not always possible (conjugate priors)
2. can use approximations with high computational cost (sampling methods) or complex solutions (variational methods)
3. sometimes we will have a non-informative prior of $\vec{W}$, but Bayesian methods carry uncertainty estimates

6.7 Expectation Maximization

Fitting model parameters with missing (latent) variables

$$Pr(\vec{x}|\theta) = \sum_k \pi_k Pr() \tag{6.4}$$

1. Very general idea (applies to many different probabilistic models)
2. Augment data with latent variables: hi ∈ (1, . . . , K) is the assignment of data point xi to a component of the mixture
3. Optimize likelihood of the complete data over N data points

6.8 EM properties

6.9 Lecture 8

7 Training a Linear Separator

What does learning mean here? It means finding the best weights w_i. For the application of this it exists two optimal algorithms which are perceptron learning and delta rule.

7.1 Percepton Learning

Perceptron learning is an algorithm for spervised learning of binary classifiers. Thus it is only a function which can decide whether or not an input belongs to some specific class. Percepton learning involves the following

1. Incremental learning.
2. Weights only change when the output is wrong.
3. Update rule $w_i + \eta(t - o)x_i \rightarrow w_i$.
4. Always converges if the problem is solvable.

7.2 Delta rule

Delta rule is a gradient descent learning rule for updating the weights of the input, the input is continuous. The delta rule invovles

1. Incremental learning.
2. Weights always change.
3. $w_i + \eta(t - \boldsymbol{w}^T \boldsymbol{X})x_i$.
4. Converges only in the mean.
5. The algorithm will find an optimal solution even if the problem can not be fully solved.

7.3 Linear Separation

Linear separation has many acceptable solutions, however it generalizes poorly. Thus it works well for all training data but creates structural risk. Hence the future data samples might get mis-classified.

7.4 Structural Risk Minimization

To minimize structural risk hyperplanes with margins are introduced. The training data points will at least be a distance d from the plane. This results in less arbitrariness and therefore better generalization.

1. Wide margins restrict the possible hyperplanes to choose from.
2. Less risk to choose a bad hyperplane by accident.
3. Reduced risk for bad generalization.

Minimization of the structural risk ≡ maximation of the margin. Hence out of all hyperplanes which solve the problem, the one with the widest margin will probably generalize best. The mathematical formulation begins with introducing the hyperplane

$$\boldsymbol{w}^T\boldsymbol{x} = 0 \tag{7.1}$$

The hyperplane with a margin is represented as when it is a positive target ($t = 1$)

$$\boldsymbol{w}^T\boldsymbol{x} \geq 1 \tag{7.2}$$

when the target is negative ($t = -1$)

$$\boldsymbol{w}^T\boldsymbol{x} \leq -1 \tag{7.3}$$

It would be convenient if these two equation of the positive and negative target could be written as one equation. Which is possible

$$t\boldsymbol{w}^T\boldsymbol{x} \geq 1. \tag{7.4}$$

The qusetion arises, how wide is the margin? By selecting two points $\boldsymbol{p}$ and $\boldsymbol{q}$ on the two margins, the length between them can be expressed as

$$2d = \frac{\boldsymbol{w}^T}{|}|\boldsymbol{w}||(\boldsymbol{p} - \boldsymbol{q}). \tag{7.5}$$

This equation can be simplified

$$\frac{\boldsymbol{w}^T(\boldsymbol{p} - \boldsymbol{q})}{||\boldsymbol{w}||} = \frac{1}{||\boldsymbol{w}||} = \frac{2}{||\boldsymbol{w}||}. \tag{7.6}$$

The maximal margin corresponds to minimal length of the weight vector. In essence the support vector machine tries to do is structural risk minimization in reference too $\boldsymbol{w}^T\boldsymbol{w}$ with the constraints $t_i\boldsymbol{w}^T\boldsymbol{x}_i \geq 1, \forall 1$.

7.5 Support Vector Machine

Note almost everything becomes linearly separable when represented in high-dimensional spaces and ordinary low-dimensional data can be scattered into a high-dimensional space. However, two problems emerge. If there are too many parameters the result will generalize poorly and the computation is extensive. What support vector machine tries to do is transform the input to a suitable high-dimensional space and then choose the unique separating hyperplane that has maximal margins. The advantages of support vector machine are

1. The generalization of support vector machine is optimal.
2. Performs well with few training samples.

3. Support vector machine classifies quick.

The disadvantages of support vector machine are

1. Non-local weight calculation.
2. Hard to implement efficiently

7.6 Kernels

Kernels allows for utilizing the advantages of a high-dimensional space without actually representing anything high-dimensional. The condition is the only operation done in the high-dimensional space is to compute scalar products between pairs of items. The trick is the high-dimensional scalar product is computed using the original (low-dimensional) representation. Now the minimization becomes $\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$ with the constraints $t_i\boldsymbol{w}^T\phi(\boldsymbol{x_i}) \geq 1, \forall i$. This minimization can be done by Lagrange Multiplier Method

$$L = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_i \alpha_i(t_i\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - 1) \tag{7.7}$$

The task is now to minimize $\boldsymbol{w}$ and maximize $\alpha_i \geq 0$.

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} - \sum_i \alpha_i t_i \phi(\boldsymbol{x}_i) = 0 \implies \boldsymbol{w} = \sum_i \alpha_i t_i \phi(\boldsymbol{x}_i). \tag{7.8}$$

Now eliminate $\boldsymbol{w}$ in equation 7.7 and the result is the dual formulation problem

$$L = \sum_i \alpha_i - \frac{1}{2}\sum_i \alpha_i\alpha_j t_i t_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) \tag{7.9}$$

Now the actual task is to maximize the equation under the constraint $\alpha_i \geq 0, \forall i$

7.7 Slack

There will be data that are none-Separable. In many cases, especially when the data contain some sort of noise, it is desirable to allow a few data points to be miss-classified if it results in a substantially wider margin. Thus slack is introduce, the reformulation becomes to minimize $\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + c\sum_i \epsilon_i$ under the constraint $t_i\boldsymbol{w}^T\phi(\boldsymbol{x}_i \geq 1 - \epsilon_i)$. The dual formulation with slack is to maximize

$$L = \sum_i \alpha_i - \frac{1}{2}\sum_i \alpha_i\alpha_j t_i t_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) \tag{7.10}$$

with the constraint $0 \leq \alpha_i \leq C, \forall i$.

7.8 Lecture 10

7.9 Introduction

Ensemble learning is when you train weak classifiers or regressors and how to combine them to make them more powerful.

7.10 The Wisdom of Crowds

A crowd is wiser than any indiviual. The collective knowledge of a diverse and independent body of of people typically exceeds the knowledge of any single indiviual and can be harnessed by voting. There are four elements that are required to make a crowd wise

1. Diversity of opinion: People in a crowd should have a range of experience, education and opinions.
2. Independence: prediction by a persion in a crowd is not influence by other people in a crowd.
3. Decentralization: people have specalizationand local knowledge.
4. Aggregation: There is a mechanism for aggregating all predictions into one single prediction.

7.11 Combining Classifiers

The wisdom of crowd ideas will be exploited for specific tasks by

1. Combining classifiers predictions.
2. Aim to combine independent and diverse classifiers

However, labelled training data will be used to

1. Identify the expert classifiers in the pool.
2. Identify complementary classifiers.
3. Indicate how to best combine them.

7.12 Ensemble Method: Baggning

Baggning stands for Bootstrap Aggregating and is applied by using bootstrap replicates of training set by sampling with replacement. One each replicate learn one model - combined altogether. E.g decision tree

1. High variance classifiers produce differing decision boundaries which are highly dependent on training data.
2. Low bias classifiers produce decision boundaries which on average are good approximations to the true decision boundary.
3. Ensemble predictions using diverse high-variance, low-bias classifiers reduce the variance of the ensemble classifier.

The baggning method is in essence: given training data as input it is iterated to create a new set of training data S_b and used to estimate the regression or classification function f_b. The output for classification is then decided by

$$f_{bag}(\boldsymbol{b})argmax_{1\leq k\leq K}\sum_{b=1}^{B} ind(f_b(\boldsymbol{x})=k), \tag{7.11}$$

and regression by

$$f_{bag}(\boldsymbol{x})=\frac{1}{b}\sum_{b=1}^{B} f_b(\boldsymbol{x}). \tag{7.12}$$

Conclusion is bagging is a procedure to reduce the variance of our classifier (when labelled training data is limited). Thus implies it is a powerful algorithm for controlling overfitting. Note it only produces good results for high variance, low bias classifiers.

7.13 Ensemble Method: Forest

Equal to bagging but in addition has a random feature selection at each node. Trees are less correlated, i.e. even higher variance between weak learners. It is suited for multi-class problems.

7.14 Ensemble Method: Boosting

A diverse and complementary set of high-bias classifiers, with performance better than change, combined by voting

$$f_v(\boldsymbol{x})=sign(\sum_{t=i}^{T} h_t(\boldsymbol{x})), \tag{7.13}$$

can produce a classifier with a low-bias. Here $h_t \in \mathcal{H}$ where $\mathcal{H}$ is family of weak classifiers function. The input is normal data, however the output is

$$f_T(\boldsymbol{x}) = sign(\sum_{t=i}^{T} \alpha_t h_t(\boldsymbol{x})), or, \sum_{t=i}^{T} \alpha_t h_t(\boldsymbol{x})) \tag{7.14}$$

Where α_t is confidence/reliability. The core ideas are

1. Performance of classifiers $h_1, ..., h_t$ helps define h_{t+1}. Remember: Each $h_t \in H$.
2. Maintain weights for each training example.
3. Large $\boldsymbol{w}_i^t$ implies that $\vec{x}_i$ has stronger influence on h_t.
4. For each iteration the $\boldsymbol{w}_i^t$ increases or decreases if $\vec{x}_i$ is wrongly or correctly classified by h_t.

7.15 Adaboost Algorithm

Given labelled training data and weak classifiers. The adaboost algorithm works by

1. set the weight vector to $1/m$ for each j.
2. Compute the reliability coefficient

$$\alpha_t = \log_e(\frac{1-\epsilon_t}{\epsilon_t}). \tag{7.15}$$

   Note that $epsilon_t$ must be less than 0.5 and break out of the loop if it is approximately 0.5.
3. Update the weights using $w_j^{t+1} = w_j^t \exp(-\alpha_t y_j h_t(\boldsymbol{x}_j))$
4. Normalize the weights so that they sum to 1.