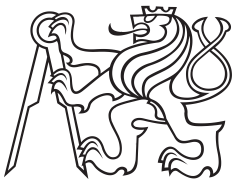
Master Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Methods for group anomaly detection

Bc. Štěpán Šubík

Supervisor: Doc. Ing. Václav Šmídl, Ph.D.
Field of study: Open Informatics
Subfield: Data Science
May 2022

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šubík** Jméno: **Štěpán** Osobní číslo: **459937**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Specializace: **Datové vědy**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Metody detekce anomálií v množinových datech

Název diplomové práce anglicky:

Methods for group anomaly detection

Pokyny pro vypracování:

Seznamte se s problémem detekce anomálií jako úlohou odhadu pravděpodobnostního rozložení dat považovaných za normální třídu. Představte základní metody pro úlohu detekce anomálií na vektorových datech a jejich vlastnosti. Představte teorii detekce anomálií pro množinová data, základní pozornost věnujte metodám s pravděpodobnostním modelem. Diskutujte použití existujících metod pro detekci anomálií vektorových dat jako stavebních prvků pro detekci množinových dat. Využijte knihovny generativních modelů pro vektorová data a rozšiřte je o pravděpodobnostní modely kardinality množiny. Použijte alespoň tři modely vektorových dat a tři modely kardinality. Ověřte chování všech kombinací těchto metod v Monte Carlo studii nad souborem standardních problémů v dané problematice. Vyhodnoťte výsledky metod a srovnejte je se standardními metodami jako je například support measure machine.

Seznam doporučené literatury:

1. Chandola, V., Banerjee, A. and Kumar, V., 2009. Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), pp.1-58.
2. Muandet, K., Fukumizu, K., Dinuzzo, F. and Schölkopf, B., 2012. Learning from distributions via support measure machines. In Advances in neural information processing systems (pp. 10-18).
3. Vo, B.N., Dam, N., Phung, D., Tran, Q.N. and Vo, B.T., 2018. Model-based learning for point pattern data. Pattern Recognition, 84, pp.136-151.
4. Chalapathy, R., Toth, E. and Chawla, S., 2018, September. Group anomaly detection using deep generative models. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 173-189). Springer, Cham.

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Václav Šmídl, Ph.D. centrum umělé inteligence FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **12.02.2021** Termín odevzdání diplomové práce: **20.05.2022**

Platnost zadání diplomové práce: **30.09.2022**

doc. Ing. Václav Šmídl, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

. _____ _____

Datum převzetí zadání Podpis studenta

Acknowledgements

First and foremost, I would like thank to my supervisor, Doc. Ing. Václav Šmídl, Ph.D., for all his patience and valuable advices. I would also like to thank to my parents and sisters and the rest of my family for their constant encouragement and support all through my studies. Lastly, my appreciation also goes to my girlfriend and friends, who made this period of life special.

Declaration

I declare that this work is all my own work and I have cited all sources I have used in the bibliography.

Prague, May 20, 2022

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 20. května 2022

Abstract

Anomaly detection in vector data is a common machine learning problem, which can be approached as normal class probability distribution estimation. It is also possible to use this attitude when detecting group data, which is the subject of this master thesis. In this thesis, we use a solution where we divide the problem into two parts, i. e., to learn the likelihood model of vectors and the cardinality model of the set. The learned probability distributions are then used to evaluate each newly observed data set. We experimentally verified the results of this approach.

Keywords: group data, anomaly detection, gaussian mixtures, variational autoencoder

Supervisor: Doc. Ing. Václav Šmídl, Ph.D.

Abstrakt

Detekce anomálií ve vektorových datech je běžný problém strojového učení, ke kterému lze přistupovat jako k úloze odhadu pravděpodobnostního rozložení dat považovaných za normální třídu. Tento přístup je možné použít i v případě detekce anomálií v množinových datech, jejíž studium je předmětem diplomové práce. V práci používáme řešení, kdy úlohu rozdělíme na dva dílčí části, tedy na naučení se věrohodnostního modelu vektorů a modelu kardinality množiny. Naučená pravděpodobnostní rozdělení potom slouží k ohodnocení každé nově pozorované množiny dat. Výsledky tohoto přístupu jsme experimentálně ověřili.

Klíčová slova: detekce anomálií, množinová data, Gaussovské směsi, variační autoenkodér

Překlad názvu: Metody detekce anomálií v množinových datech

Contents

Figures

Tables

Introduction

In this diploma thesis, we are studying anomaly detection as a problem of normal class probability density estimation. First, we introduce methods for anomaly detection in vector data. Particularly we focus on probability density estimation using a Gaussian mixture model (GMM) and variational autoencoder (VAE). With these estimators, we progress from vector data to group data, so that cardinality of instances is no longer uniform but may differ. To rank data instances unbiasedly towards cardinality, we need to extend earlier established probability density estimators that are used for feature density estimation with new cardinality distribution models. We focus on Poisson, log-normal and discrete uniform distributions. Feature and cardinality distributions are building blocks of the ranking function for group anomaly detection. Eventually, we evaluate our group anomaly detection models in Monte Carlo experiment and compare the results of our novel approach with the performance achieved by alternative anomaly detection method.

Chapter 1

Preliminary to problem formulation

Anomaly detection is a classic problem in machine learning (ML), in which we identify patterns in data that do not conform to behaviour that is regarded as normal or expected [13]. Most anomaly detection techniques were developed to solve a specific problem formulation for a particular application domain. For instance, anomaly detection techniques can spot an emerging defect in an aircraft engine [3, 15], detect a hacker sneaking into an enterprise network [17], identify a fraudulent transaction in a banking system [23], or even reveal a presence of a malignant tumour in an MRI scan [11].

In this chapter, we intend to familiarize the reader with the world of anomalies and develop an intuition about the characteristics of group anomaly detection. First, in Section 1.1, we describe the factors that differentiate anomaly detection problem formulations from each other. In Section 1.2, we define anomaly detection in vector data and follow up with anomaly detection in group data. Related works are summarized in Section 1.3, and eventually, we specify the objective of this thesis in Section 1.4.

1.1 Aspects of anomaly detection

In order to choose the most suitable ML algorithm capturing meaningful patterns for our anomaly detection problem, we should understand the nature of input data and the type of anomalies.

1.1.1 Nature of input data

Input data is generally a collection of data records. Further in the text, we also refer to data records as observations. Depending on the particular dataset, data records may be in the form of vectors or groups of constituent vectors. Input data are then called vector data or group data[1], respectively.

A vector data record is described by one or multiple attributes and is referred to as univariate or multivariate, respectively. An attribute is either qualitative or quantitative. The latter is further divided into discrete and

[1]Chapter 3 is based on point process theory, in which group data are commonly called point pattern data, and data records point patterns. We will use these terms interchangeably.

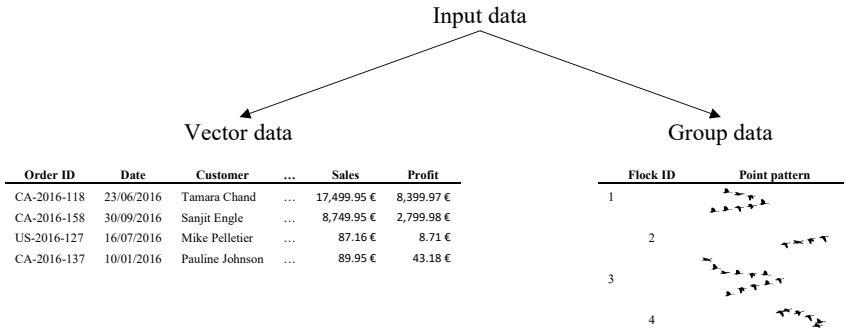


Figure 1.1: Sales records as an example of vector data (left), and migrating geese formations illustrating group data (right). Sales records are multivariate. "Order ID", "Date", and "Customer" are qualitative attributes, whereas "Sales" and "Profit" are continuous quantitative attributes. Geese formations are here well-defined by the size of the flock and the position of every goose.

continuous. Attributes of a multivariate record are either all of the same type or a mixture of different types.

Group data instances consist of several constituent vectors [2]. Although the cardinalities of these instances may differ, all constituent vectors have the exact attributes. Possible types of attributes in group data are similar to those listed when discussing vector data.

Examples of both vector and group data are shown in Fig. 1.1.

1.1.2 Types of anomalies

Anomalies can be broadly divided into three mutually exclusive categories: atomic univariate, atomic multivariate and aggregate anomalies [10] (atomic anomalies, in general, are in some works referred to as point anomalies [5]). Examples of all three categories are shown in Fig. 1.2, and we describe them in detail in the following paragraphs.

Atomic univariate anomalies are single data records with a deviant value for one or possibly multiple attributes. They are irrelevant of relationships between attributes or observations. These abnormal individual values of attributes are easy to detect even with a human eye. There may be several occurrences of atomic univariate anomalies in input data, yet every such observation must be an anomaly in its own right. Additionally, if a single data record holds multiple unusual attribute values, it is an anomaly with respect to each corresponding attribute (see Fig. 1.2).

Atomic multivariate anomalies are single data records whose values of attributes are not deviant individually. Still, they are anomalous either due to abnormal combinations of attribute values in particular observations (see Fig. 1.2) or differences in observations that are linked together, e.g. unusually low summer temperature in time series measurement that would have been

[2] In point process theory, constituent vectors are called features. We will use both terms interchangeably.

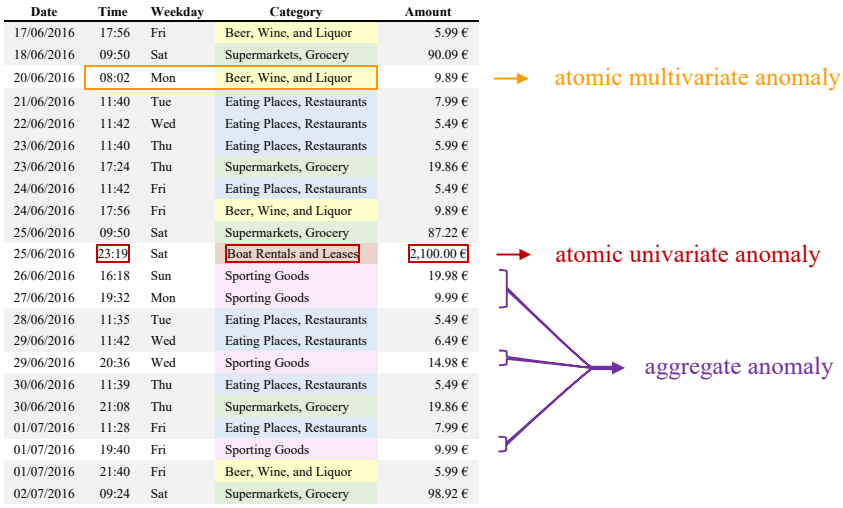


Figure 1.2: Records of debit card expenses before leaving for vacation. Grey records represent repetitive normal patterns that repeat throughout the year, whereas white ones are anomalies. The record highlighted in red is an atomic univariate anomaly because the attribute "Amount" value is unusually high. Also, this expense was paid at an unusual time, so this observation is an anomaly with respect to each attribute "Amount", "Time", and also "Category" individually. The record highlighted in red represents an atomic multivariate anomaly because records with the attribute value "Beer, Wine, and Liquor" usually occur on Friday afternoons and not on Monday mornings. Notice that atomic multivariate anomaly is more challenging to spot. Although individual expenses for sports equipment appear to be normal, it is an example of an aggregate anomaly when observed collectively.

normal in winter. These anomalies are more difficult to spot and detect as they are hidden in the multi-dimensionality of the dataset.

Fundamental characteristics of aggregate anomalies are relationships between observations and between attributes. Aggregate anomalies are collections of various cardinalities that deviate as a whole, while constituent vectors separately are usually not anomalous (see Fig. 1.2). Due to their complicated and complex nature, aggregate anomalies are the most difficult to discover.

The specific subtype of aggregate anomalies relevant to group anomaly detection is called distribution based aggregate anomalies [10]. These anomalies occur in datasets consisting of clusters, with the anomalous cluster exhibiting anomalous behaviour with respect to normal cluster distribution. For instance, in Fig. 1.2, we could divide expense records from the whole year into 52 clusters, each containing records from a single week (notice that clusters vary in cardinality since the number of expense records differs from week to week). Thus, in group anomaly detection, we solve a specific variant of distribution based aggregate anomalies, where we anticipate the input data to be divided into individual clusters, i.e. group data instances.

1.1.3 Semi-supervised and unsupervised techniques

Similarly to other ML problems, anomaly detection usually consists of a training phase, a validation phase for tuning hyperparameters and a testing phase. Based on the availability of labelled observations in training data, there are two ML approaches that anomaly detection techniques can take: semi-supervised and unsupervised [5].

Semi-supervised anomaly detection techniques require only normal data records to be labelled. Thence ML algorithm learns the model corresponding to normal behaviour and uses this model to identify anomalies in new observations. In this thesis, we consider solely semi-supervised techniques, so further on, when discussing ML algorithms or ML models, we always assume they were learned in a semi-supervised manner.

Unsupervised anomaly detection techniques do not require any labelled data records. These techniques usually make an implicit assumption that normal observations are far more frequent than anomalies in input data.

1.1.4 Outputs

According to the particular ML algorithm that is used, the output can be in the form of an anomaly score or probability.

ML algorithms that output an anomaly score assign every data record numerical value that describes to which extent it is considered an anomaly.

Probabilistic ML algorithms approximately learn the probability distribution of input data and consequently evaluate observations with respect to their probabilities. In this thesis, we study anomaly detection as a problem of normal class probability density estimation. Thus, outputs of our ML algorithms have the form of probabilities (or ranking function based on probabilities in the case of group anomaly detection).

1.2 Anomaly detection in vector data, transition to group data

The simplest setting for anomaly detection is the case with vector input data and atomic univariate anomalies. First, we train the probabilistic ML algorithm to estimate input data probability distribution so that it maximises the probability of training data. Output probabilities of previously unseen observations are then compared to a predefined threshold and labelled as normal or anomalous.

Things become tricky when detecting group anomalies in group data, which we refer to in this thesis as group anomaly detection. Remember that group data instances vary in their cardinality, so we should not just multiply the probabilities of individual vectors. Otherwise, output probability is biased towards cardinality because as the number of vectors in the group data instance grows, the corresponding joint probability value decreases rapidly to

zero. Authors of [22] propose a novel ranking function that considers both feature probability density and cardinality distribution.

1.3 Related works

In this section we provide an overview of approaches to group anomaly detection.

In [14] an anomaly detection technique based one-class support measure machines (OCSMMs) is proposed. OCSMMs generalize one-class support vector machines (OCSVMs) to a space of probability measures. Deep generative models such as adversarial autoencoder (AAE) and variational autoencoder (VAE) are used in [4] and performance of multi-level variation autoencoder (ML-VAE) is further evaluated in [2]. Authors of [8] introduce a technique relying on graph autoencoders.

1.4 Objective

The main objective of this thesis is to evaluate a novel ranking function for group anomaly detection introduced in [22]. We start with anomaly detection in vector data, getting familiar with probability density estimators such as Gaussian mixture model and variational autoencoder. Following up with group anomaly detection, we employ the ranking function which is build up from feature and cardinality distributions. We use probability density estimators introduced for vector data to learn the feature density. We choose to approximate cardinality using Poisson, log-normal and discrete uniform distributions. Eventually, in the experimental part, we will compare the results of our approach with a performance achieved by the standard anomaly detection method.

Chapter 2

Anomaly detection in vector data

In the previous chapter, we have informally described anomaly detection in vector data. Let us start this chapter with setting uniform notation in Section 2.1 and follow with proper problem formulation in Section 2.2. In Section 2.3 and Section 2.4, we present three approaches to learning vector data probability model.

2.1 Notation

In vector data, we consider vectors $\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$ from a D-dimensional vector space $\mathscr{X} \subset \mathbb{R}^D$, i. e. $\mathbf{x} \in \mathscr{X}$. We are further interested in putting more D-dimensional vectors $\mathbf{x}$ in a matrix, represented by $\mathbf{X}$. Dataset of N records consisting of vectors $\mathbf{x}_n$, $n = 1, \ldots, N$, is denoted as $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

The probability of the vector $\mathbf{x}$ falling into the interval $(\mathbf{x}, \mathbf{x} + \delta\mathbf{x})$ is given by $p(\mathbf{x})\,\delta\mathbf{x}$ for $\delta\mathbf{x} \to 0$, and $p(\mathbf{x})$ is the probability density over $\mathbf{x}$.

2.2 Problem formulation

Consider the same setting as in Section 1.2: detection of atomic univariate anomalies in the dataset $\mathcal{D}$. After the training and validation phases, the ML algorithm has learned a model corresponding to normal behaviour in the form of probability density function $p(\mathbf{x})$[1], so that every observation can be evaluated with its likelihood[2]. The final decision on whether an observation from the testing dataset is normal or anomalous reads as

$$\chi\left[p(\mathbf{x}) \geq \tau\right] = \begin{cases} \text{normal} & p(\mathbf{x}) \geq \tau \\ \text{anomaly} & \text{otherwise} \end{cases}, \tag{2.1}$$

where χ denotes an indicator function labelling observations as either normal or anomaly, $p(\mathbf{x})$ is the likelihood of an observation $\mathbf{x}$, and τ is a delimiting

[1] For compactness, the condition on the normal class is omitted, i.e. $p(\mathbf{x})$ is used instead of $p(\mathbf{x}|\theta)$, where θ are the parameters of the normal class probability density function.

[2] We will use the term likelihood not only for a realization of a likelihood function $\mathcal{L}(\theta|\mathbf{x})$, a function of θ given an observation $\mathbf{x}$, but also for $p(\mathbf{x}|\theta)$, a probability density of an observation $\mathbf{x}$ when the true values of the model parameters are θ.

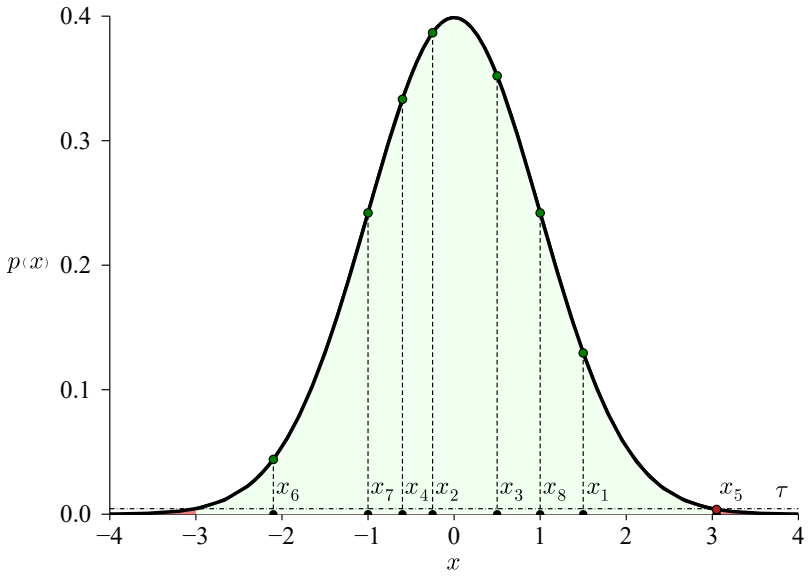


Figure 2.1: Plot of the probability density function $p(x)$ as a standard normal distribution $\mathcal{N}(0,1)$. The black semi-circles denote observed real numbers $x_1, \ldots, x_8$ with corresponding green/red points depicting greater/smaller likelihoods with respect to the threshold τ, which is represented by the dash-dotted line. According to (2.1), the observations with green/red likelihoods are labelled as normal/anomaly, respectively. In our dataset, there are seven normal observations, and the only observation x_5 is anomalous.

threshold, which is either predefined or learned during training. Note that in (2.1), observations are ranked according to their likelihoods and then compared to the threshold τ. We call functions with the ranking capability, e.g. the likelihood and log-likelihood functions and any monotonic transformation of the likelihood function ranking functions. These functions assign scores to observations that are only useful for their ranking. Provided some ranking function, we can detect anomalies even if the ML model does not output likelihoods but scores, as shown in Chapter 3.

Example. Anomaly detection in vector data

Suppose that we have observed eight independent and identically distributed (i.i.d) real numbers $x_1, \ldots, x_8$, i. e. $x \in \mathbb{R}$ drawn from the standard normal distribution $\mathcal{N}(0,1)$, and we want to detect anomalous observations. Having the prior knowledge about the distribution being normal, we choose to use the threshold τ given by the three-sigma rule [18] so that observations beyond the distance of three standard deviations from the mean are declared to be anomalous.

Since observations $x_1, \ldots, x_8$ are i.i.d., the likelihood of every observation is computed as $p(x) = \mathcal{N}(x|0,1)$. Then, we can compare the likelihoods of observations to the threshold τ and decide on each of them whether that observation is normal or anomalous, just like in (2.1).

The task and its solution are outlined in Fig. 2.1.

2.3 Gaussian mixture model

A Gaussian mixture model (GMM) is a model composed of several Gaussians. In this section, we derive GMM similarly as in [1]. We first refresh the properties of Gaussian distribution, then show how is the GMM built and eventually discuss learning the GMM model.

2.3.1 Gaussian distribution

The Gaussian distribution is a continuous probability distribution in one-dimensional space described by a univariate normal distribution

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{2.2}$$

with two parameters, the mean $\mu \in \mathbb{R}$ and the variance $\sigma^2 > 0$. x is one-dimensional real-valued variable. From (2.2), it can be seen that the requirements for a valid probability distribution hold, i. e. $\mathcal{N}\left(x|\mu,\sigma^2\right) > 0$ and that Gaussian is normalized $\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) \mathrm{d}x = 1$. The average value of random variable X under the Gaussian distribution is given by

$$\mathbb{E}\left[X\right] = \int_{-\infty}^{\infty} x\mathcal{N}\left(x|\mu,\sigma^2\right) \mathrm{d}x = \mu. \tag{2.3}$$

From the second order moment

$$\mathbb{E}\left[X^2\right] = \int_{-\infty}^{\infty} x^2\mathcal{N}\left(x|\mu,\sigma^2\right) \mathrm{d}x = \mu^2 + \sigma^2, \tag{2.4}$$

and (2.3) it follows that the variance is given by

$$\mathrm{var}\left[X\right] = \mathbb{E}\left[X^2\right] - \mathbb{E}\left[X\right]^2 = \sigma^2. \tag{2.5}$$

Generalization of the one-dimensional univariate normal distribuiton to higher-dimensional multivariate normal distribution reads as

$$\mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}\right) = \frac{1}{\sqrt{\left(2\pi\right)^D \left|\boldsymbol{\Sigma}\right|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}, \tag{2.6}$$

where $\boldsymbol{\mu} \in \mathbb{R}^D$ is the mean and $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$ is the covariance. $\mathbf{x}$ is a D-dimensional column vector and $|\boldsymbol{\Sigma}| \equiv \det \boldsymbol{\Sigma}$ is the determinant of $\boldsymbol{\Sigma}$.

2.3.2 Building up the Gaussian mixture model

Usually, we want to model real datasets whose records form not just sole but more dominant clumps. Single Gaussian distribution is unable to capture such a structure. We can make a linear superposition of more basic distributions to approximate such data more accurately. This kind of probabilistic model is known as mixture distribution. A Gaussian mixture can model nearly any

continuous probability density with a linear combination of a finite number of Gaussian distributions as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.7}$$

where each probability density $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, a so-called mixture component, has parameters mean $\boldsymbol{\mu}_k \in \mathbb{R}^D$ and covariance $\boldsymbol{\Sigma}_k \in \mathbb{R}^{D\times D}$. The parameter $\pi_k \in \mathbb{R}$ is called the mixing coefficient.

If we integrate both sides of (2.7) with respect to $\mathbf{x}$, note that $p(\mathbf{x})$ and all K Gaussian components are normalized, we obtain

$$\sum_{k=1}^{K} \pi_k = 1, \tag{2.8}$$

which together with the probability distribution requirements $p(\mathbf{x}) \geqslant 0$ and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \geqslant 0$ implies

$$0 \leqslant \pi_k \leqslant 1. \tag{2.9}$$

From (2.8) and (2.9), it follows that mixing coefficients sum to one and are greater or equal to zero, so they satisfy the requirements to be valid probabilities.

If we apply the sum and product rules, the marginal density of $\mathbf{x}$ is given by

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}, k) = \sum_{k=1}^{K} p(\mathbf{x}|k)\, p(k). \tag{2.10}$$

Comparing (2.7) with (2.10), we can see that $p(k) = \pi_k$ and $p(\mathbf{x}|k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $p(k)$ is the prior probability of picking k^{th} component and $p(\mathbf{x}|k)$ is the probability of observing $\mathbf{x}$ given the k^{th} component.

The posterior probability $p(k|\mathbf{x})$ is called responsibility, as it describes how much is the k^{th} component responsible for generating observation $\mathbf{x}$. We can express the responsibility using Bayes' theorem as

$$\begin{aligned} \gamma_k(\mathbf{x}) &\equiv p(k|\mathbf{x}) \\ &= \frac{p(\mathbf{x}|k)\, p(k)}{\sum_l p(\mathbf{x}|l)\, p(l)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_l \pi_l \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \end{aligned} \quad . \tag{2.11}$$

For the reason that will become apparent in Subsection 2.3.4, we build up a formulation of Gaussian mixture in terms of discrete latent variable $\mathbf{z}$. $\mathbf{z}$ is a K-dimensional binary random variable with a 1-of-K representation in which only a single element z_k is equal to 1, and all other elements are equal to 0, i. e. $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$. Therefore there are K possibilities (or positions) for that single nonzero element in $\mathbf{z} = (z_1, \ldots, z_K)$. Our goal is to redefine (2.11) so that responsibility is a function of the latent variable $\mathbf{z}$.

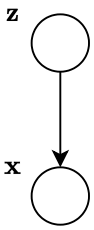


Figure 2.2: Graphical representation of a mixture model, according to which is the joint probability distribution expressed in the form $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})\, p(\mathbf{x}|\mathbf{z})$.

We start with expressing the joint probability distribution $p(\mathbf{x}, \mathbf{z})$ in terms of a marginal distribution $p(\mathbf{z})$ and a conditional distribution $p(\mathbf{x}|\mathbf{z})$, corresponding to the mixture model in Fig. 2.2. For this reason, we will derive their forms concerning the Gaussian mixture defined in (2.7).

We state that

$$p(z_k = 1) = \pi_k \tag{2.12}$$

so that the prior probability of the latent variable taking the value z_k is equivalent to the mixing coefficient of the k^{th} component. Because $\mathbf{z}$ uses a 1-of-K representation, the marginal distribution can be written as

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}. \tag{2.13}$$

Similarly, the probability of observing $\mathbf{x}$ given particular latent variable z_k is equivalent to the probability of observing $\mathbf{x}$ given the Gaussian distribution of the k^{th}component

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.14}$$

which can also be written in the form with latent variable $\mathbf{z}$

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}. \tag{2.15}$$

If we substitute corresponding probabilities in the joint probability distribution with (2.13) and (2.15) and marginalize it over $\mathbf{z}$, we can derive the marginal distribution of $\mathbf{x}$ as

$$\begin{aligned} p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) &= \sum_{\mathbf{z}} \left[\prod_{k=1}^{K} \pi_k^{z_k} \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k} \right] \\ &= \sum_{\mathbf{z}} \left[\prod_{k=1}^{K} \left(\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)^{z_k} \right] \\ &= \sum_{j=1}^{K} \left[\prod_{k=1}^{K} \left(\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)^{I_{kj}} \right] \\ &= \sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right) \end{aligned} \tag{2.16}$$

where $I_{kj} = 1$ if $k = j$ an 0 otherwise, exploiting the 1-of-K representation of $\mathbf{z}$. Notice that the marginal distribution of $\mathbf{x}$ in (2.16) is a Gaussian mixture of the form (2.7).

We can finally define the responsibility so that it depends on the latent variable $\mathbf{z}$

$$\begin{aligned}\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)\, p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)\, \mathcal{N}(\mathbf{x}|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}\end{aligned}. \tag{2.17}$$

Having multiple records in our dataset $\mathcal{D}$, we can compute component k's responsibility for explaining the observation $\mathbf{x}_n$, $n = 1, \ldots, N$, individually for every observation, denoting it as $\gamma(z_{nk})$. For this purpose, we represent the records from $\mathcal{D}$ by the matrix $\mathbf{X} \equiv \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

2.3.3 Setting model parameters

Usually, the model parameters $\boldsymbol{\mu} \equiv \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$, $\boldsymbol{\Sigma} \equiv \{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K\}$ and $\boldsymbol{\pi} \equiv \{\pi_1, \ldots, \pi_K\}$, as well as the number of components K are unknown and our goal is to determine their optimal values given data $\mathbf{X}$. Assuming that observations $\mathbf{X}$ are drawn independently from the distribution and that the optimal number of components is K, the formulation

$$p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \left[\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right] \tag{2.18}$$

is the Gaussian mixture likelihood function. By applying a logarithm to each side of the equation, we get the log-likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left[\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]. \tag{2.19}$$

We want to set the values of the parameters so that it maximizes (2.19). If we were dealing with just a single Gaussian, we would differentiate its log-likelihood with respect to the parameters, equate it to zero, and find the optimal values for these parameters. In the Gaussian mixture model, due to the summation over k inside the logarithm in (2.19), the maximum likelihood for the parameters no longer has a closed-form analytical solution. However, we can maximize the log-likelihood using iterative numerical optimization techniques [9] or alternatively applying the expectation-maximization (EM) algorithm [1], which is summarized in Subsection 2.3.4.

2.3.4 Expectation-maximization algorithm for Gaussian mixture

Given a Gaussian mixture model of K components, the goal of the EM algorithm is to obtain maximum likelihood estimates of model parameters

$\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\pi}$ by means of maximizing the log-likelihood function (2.19). The algorithm is comprised of four steps: initialization, expectation or E step, maximization or M step and evaluation.

1. Initialize the Gaussian mixture parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\pi}$ and evaluate the initial value of the log-likelihood .

2. **E step.** Evaluate the responsibilities using the current parameter values

$$\gamma\left(z_{nk}\right)=\frac{\pi_k \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K}\pi_j \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\right)}. \tag{2.20}$$

3. **M step.** Recompute the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}}=\frac{1}{N_k}\sum_{n=1}^{N}\gamma\left(z_{nk}\right)\mathbf{x}_n, \tag{2.21}$$

$$\boldsymbol{\Sigma}_k^{\text{new}}=\frac{1}{N_k}\sum_{n=1}^{N}\gamma\left(z_{nk}\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}}, \tag{2.22}$$

$$\boldsymbol{\pi}_k^{\text{new}}=\frac{N_k}{N}, \tag{2.23}$$

   where

$$N_k=\sum_{n=1}^{N}\gamma\left(z_{nk}\right). \tag{2.24}$$

4. Evaluate the Gaussian mixture log-likelihood

$$\ln p\left(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=\sum_{n=1}^{N}\ln\left[\sum_{k=1}^{K}\pi_k \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)\right] \tag{2.25}$$

   and check that the log-likelihood or parameters converge, i. e., that the change in log-likelihood or the parameters falls below some threshold. If the convergence criterion is not satisfied, return to step 2.

2.4 Variational autoencoder

A variational autoencoder (VAE) is a neural network architecture that was introduced in [12]. This section starts with a classical autoencoder description and follows with a variational autoencoder. Next, we derive some mathematical fundamentals used in VAE construction, and eventually, we build the VAE that we will use for learning the probability model.

2.4.1 Motivation: from autoencoders to variational autoencoders

Why do we start the discussion about variational encoders by talking about classical autoencoders? Simply because they consist of the same blocks, and

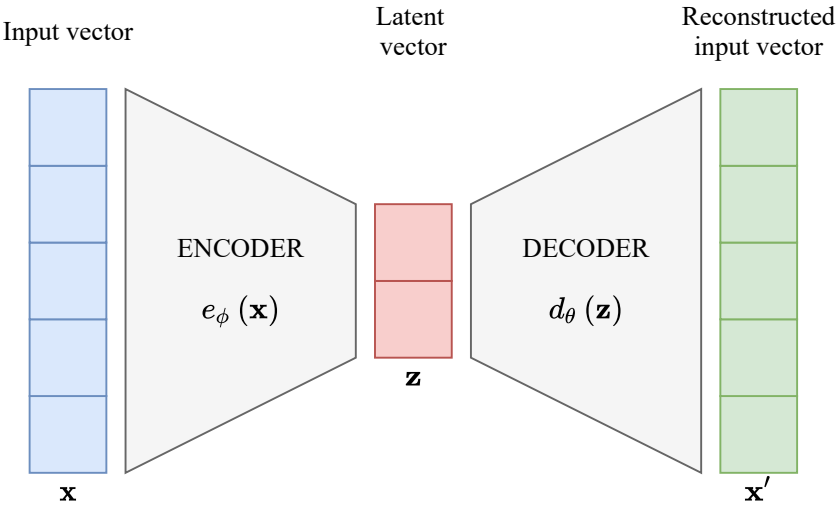


Figure 2.3: An autoencoder consists of an encoder $e_\theta(\mathbf{x})$ which maps an input vector $\mathbf{x}$ into a lower-dimensional latent vector $\mathbf{z} = e_\theta(\mathbf{x})$ and a decoder $d_\phi(\mathbf{z})$ which reconstructs the input vector $\mathbf{x}' = d_\phi(\mathbf{z})$.

both are related to data compression. By comparing these two architectures, the reader should better understand how a variational autoencoder works.

An autoencoder, as well as a variational encoder, are built from two neural networks, representing an encoder and a decoder. The encoder $e_\theta(\mathbf{x})$ takes an input vector $\mathbf{x}$ and encodes it producing a lower-dimensional representation of $\mathbf{x}$ in encoded space, also called latent space. The decoder $d_\phi(\mathbf{z})$ then reconstructs the input vector from its latent representation so that $\mathbf{x}$ and $\mathbf{x}'$ are as similar as possible, i. e. the reconstruction loss is minimized. Typically, the encoder and decoder networks are trained jointly via backpropagation. The difference between autoencoder and variational autoencoder lies in the latent space representation.

Standard autoencoder learns to encode an input vector $\mathbf{x}$ into a compact latent vector representation $\mathbf{z}$ from which is the input vector well reconstructed, as shown in Fig. 2.3. The loss function is only determined by the reconstruction loss. Such architecture is useful for dimensionality reduction so that the latent vector representation retains the meaningful attributes of the input vector. Notice that if we trained an autoencoder on, for instance, the MNIST dataset, the encoding for each image type would form distinct clusters because distinct encodings make it easier for the decoder to reconstruct corresponding input [19].

By contrast, as shown in Fig. 2.4, variational encoders encode an input vector $\mathbf{x}$ to a lower-dimensional latent distribution given by mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$. We will see the properties which control the latent distribution shape in Subsection 2.4.4. Since variational autoencoder intrinsically works with the distribution over the latent variable $p(\mathbf{z})$, we can say, that a variational encoder aims to maximize the probability of each $\mathbf{x}$ in the training

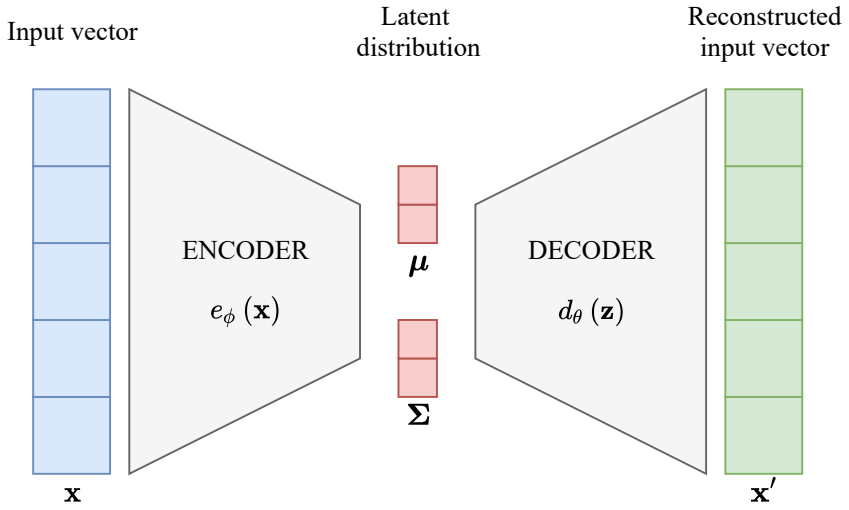


Figure 2.4: A variational autoencoder's encoder maps an input vector $\mathbf{x}$ into a lower-dimensional latent distribution described by its parameters $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\} = e_\theta(\mathbf{x})$. Corresponding latent variable $\mathbf{z}$ is then fed into a decoder to reconstruct the input vector $\mathbf{x}' = d_\phi(\mathbf{z})$.

data according to

$$p(\mathbf{x}) = \int_{\mathcal{Z}} p(\mathbf{x}, \mathbf{z})\, \mathrm{d}\mathbf{z} = \int_{\mathcal{Z}} p(\mathbf{x}|\mathbf{z})\, p(\mathbf{z})\, \mathrm{d}\mathbf{z}. \tag{2.26}$$

In other words, we are looking for such latent variable $\mathbf{z}$ that it explains the observation $\mathbf{x}$. Only after finding such $\mathbf{z}$ the probability $p(\mathbf{x})$ will be maximized. Unfortunately, when the dimension of $\mathbf{z}$ is high, the computation of $p(\mathbf{x})$ becomes intractable.

2.4.2 Evidence lower bound

Let us consider a dataset of N vectors $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. $\boldsymbol{X}$ denotes a random variable, and each $\mathbf{x}_n$, $n = 1, \ldots, N$, is its multi-dimensional vector realization. We further denote latent random variable $\boldsymbol{Z}$ and its multidimensional vector realization as $\mathbf{z}$. We are looking for the latent variable $\mathbf{z}$, which explains observation $\mathbf{x}$, i. e. we are interested in the posterior distribution

$$p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}|\mathbf{z})\, p(\mathbf{z})}{p_\theta(\mathbf{x})} = \frac{p_\theta(\mathbf{x}|\mathbf{z})\, p_\theta(\mathbf{z})}{\int_{\mathcal{Z}} p_\theta(\mathbf{x}, \mathbf{z})\, \mathrm{d}\mathbf{z}}, \tag{2.27}$$

where θ are parameters of the distribution. As in Subsection 2.4.1, the marginal probability $p_\theta(\mathbf{x})$ might be intractable. However, we can use variational inference, a paradigm for estimating a posterior distribution when computing it explicitly is intractable. Instead of computing $p_\theta(\mathbf{z}|\mathbf{x})$ using the Bayes theorem, variational inference attempts to approximate the posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$ with another distribution $q_\phi(\mathbf{z})$, characterized by its own parameters ϕ. $q_\phi(\mathbf{z})$ should be easier to evaluate than the posterior

distribution, and by optimization of ϕ, we look for $q(\mathbf{z}|\phi)$, which is similar to $p_\theta(\mathbf{z}|\mathbf{x})$.

We need to refresh one last thing before diving into evidence lower bound (ELBO) derivation, the Jensen's inequality. It states that for any concave function f we have

$$f(\mathbb{E}[\boldsymbol{X}]) \geq \mathbb{E}[f(\boldsymbol{X})]. \tag{2.28}$$

Since we have already introduced q_ϕ, and we are familiar with Jensen's inequality, we derive ELBO as follows

$$\begin{aligned} \ln p_\theta(\mathbf{x}) &= \ln\left(\int_{\mathcal{Z}} p_\theta(\mathbf{x},\mathbf{z})\,\mathrm{d}\mathbf{z}\right) \\ &= \ln\left(\int_{\mathcal{Z}} p_\theta(\mathbf{x},\mathbf{z})\,\frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})}\mathrm{d}\mathbf{z}\right) \\ &= \ln\left(\mathbb{E}_q\left[\frac{p_\theta(\mathbf{x},\boldsymbol{Z})}{q_\phi(\boldsymbol{Z})}\right]\right) \\ &\geq \mathbb{E}_{q_\phi}\left[\ln\frac{p_\theta(\mathbf{x},\boldsymbol{Z})}{q_\phi(\boldsymbol{Z})}\right] \end{aligned}. \tag{2.29}$$

The inequality in (2.29) stems from Jensen's inequality, which we applied on logarithm, a concave function. In the thesis, we will refer to ELBO as $\mathcal{L}$, so that

$$\mathcal{L} = \mathbb{E}_{q_\phi}\left[\ln\frac{p_\theta(\mathbf{x},\boldsymbol{Z})}{q_\phi(\boldsymbol{Z})}\right]. \tag{2.30}$$

The final inequality in (2.29) implicates, that instead of maximizing $p_\theta(\mathbf{x})$, we can maximize $\mathcal{L}$.

2.4.3 Kullback–Leibler divergence

The Kullback–Leibler (KL) divergence has its origins in information theory, whose primary goal is to quantify how much information is in data according to

$$\mathcal{I}(x) = -\ln p(x), \tag{2.31}$$

where x is a realization of a random variable X. The intuition behind this concept is that if the probability of a certain event is high, then its information $\mathcal{I}$ is low. For instance, suppose that x is given by the sentence "When it rains, paths are wet". Corresponding probability $p(x) \approx 1$ means that such event is almost certain, and therefore its information is almost zero, $\mathcal{I} \approx 0$.

The most important metric in information theory is called entropy. For a discrete random variable X, it is given by

$$\mathcal{H}(x) = -\sum p(x)\ln p(x). \tag{2.32}$$

Then the KL divergence $\mathcal{D}_{KL}\left[p \parallel q\right]$ is a measure of dissimilarity between two probability distribution p and q with respect to the distribution p. Therefore, it is given by

$$\mathcal{D}_{KL}\left[p\left(x\right) \parallel q\left(x\right)\right] = -\sum p\left(x\right) \ln q\left(x\right) + \sum p\left(x\right) \ln p\left(x\right) \tag{2.33}$$

$$= \sum p\left(x\right) \ln \frac{p\left(x\right)}{q\left(x\right)} \tag{2.34}$$

for a discrete random variable realization x. Similarly, we can define KL divergence for a continuous random variable realization x as

$$\mathcal{D}_{KL}\left[p\left(x\right) \parallel q\left(x\right)\right] = \int_{-\infty}^{\infty} p\left(x\right) \ln \frac{p\left(x\right)}{q\left(x\right)} \mathrm{d}x. \tag{2.35}$$

In this thesis, we will use a general notation of KL divergence with an expected value

$$\mathcal{D}_{KL}\left[p\left(x\right) \parallel q\left(x\right)\right] = \mathbb{E}_q\left[\ln \frac{p\left(X\right)}{q\left(X\right)}\right], \tag{2.36}$$

where the q subscript denotes that the expected value is computed with respect to the probability distribution q.

The KL divergence has two important properties:

1. $\mathcal{D}_{KL}$ is not a symmetrical quantity, i. e. $\mathcal{D}_{KL}\left[p \parallel q\right] \not\equiv \mathcal{D}_{KL}\left[q \parallel p\right]$,
2. $\mathcal{D}_{KL}$ is non-negative, i. e. $\mathcal{D}_{KL}\left[p \parallel q\right] \geq 0$ and $\mathcal{D}_{KL}\left[p \parallel q\right] = 0$ if, and only if, $p\left(x\right) = q\left(x\right)$.

2.4.4 Building up the variational encoder

In Subsection 2.4.2, we asserted that we would use the distribution $q_\phi\left(\mathbf{z}\right)$ to estimate intractable posterior distribution $p_\theta\left(\mathbf{z}|\mathbf{x}\right)$. So, let us examine the KL divergence between these two distributions

$$\begin{aligned}
\mathcal{D}_{KL}\left[q_\phi\left(\mathbf{z}\right) \parallel p_\theta\left(\mathbf{z}|\mathbf{x}\right)\right] &= \mathbb{E}_{q_\phi}\left[\ln \frac{q_\phi\left(\boldsymbol{Z}\right)}{p_\theta\left(\boldsymbol{Z}|\mathbf{x}\right)}\right] \\
&= \mathbb{E}_{q_\phi}\left[\ln q_\phi\left(\boldsymbol{Z}\right)\right] - \mathbb{E}_{q_\phi}\left[\ln p_\theta\left(\boldsymbol{Z}|\mathbf{x}\right)\right] \\
&= \mathbb{E}_{q_\phi}\left[\ln q_\phi\left(\boldsymbol{Z}\right)\right] - \mathbb{E}_{q_\phi}\left[\ln \frac{p_\theta\left(\mathbf{x}, \boldsymbol{Z}\right)}{p_\theta\left(\mathbf{x}\right)}\right] \\
&= \mathbb{E}_{q_\phi}\left[\ln q_\phi\left(\boldsymbol{Z}\right)\right] - \mathbb{E}_{q_\phi}\left[\ln p_\theta\left(\mathbf{x}, \boldsymbol{Z}\right)\right] + \mathbb{E}_{q_\phi}\left[\ln p_\theta\left(\mathbf{x}\right)\right] \\
&= \ln p_\theta\left(\mathbf{x}\right) - \mathbb{E}_{q_\phi}\left[\ln \frac{p_\theta\left(\mathbf{x}, \boldsymbol{Z}\right)}{q_\phi\left(\boldsymbol{Z}\right)}\right] = \ln p_\theta\left(\mathbf{x}\right) - \mathcal{L}
\end{aligned} \tag{2.37}$$

From (2.37), we observe that the difference between $\ln p_\theta\left(\mathbf{x}\right)$ and $\mathcal{L}$ is precisely the KL divergence between $q_\phi\left(\mathbf{z}\right)$ and $p_\theta\left(\mathbf{z}|\mathbf{x}\right)$. Therefore, the equality $\ln p_\theta\left(\mathbf{x}\right) = \mathcal{L}$ holds if, and only if the KL divergence between $q_\phi\left(\mathbf{z}\right)$ and $p_\theta\left(\mathbf{z}|\mathbf{x}\right)$ equals zero. We can rearrange the result of (2.37) so that

$$\ln p_\theta\left(\mathbf{x}\right) = \mathcal{D}_{KL}\left[q_\phi\left(\mathbf{z}\right) \parallel p_\theta\left(\mathbf{z}|\mathbf{x}\right)\right] + \mathcal{L}. \tag{2.38}$$

Now, we revisit (2.30) and rearrange its formulation to contain the KL divergence between $q_\phi(\mathbf{z})$ and $p_\theta(\mathbf{z})$

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{q_\phi}\left[\ln \frac{p_\theta(\mathbf{x}, \boldsymbol{Z})}{q_\phi(\boldsymbol{Z})}\right] \\ &= \mathbb{E}_{q_\phi}\left[\ln p_\theta(\mathbf{x}|\boldsymbol{Z})\right] + \mathbb{E}_q\left[\ln \frac{p_\theta(\boldsymbol{Z})}{q_\phi(\boldsymbol{Z})}\right] \\ &= \mathbb{E}_{q_\phi}\left[\ln p_\theta(\mathbf{x}|\boldsymbol{Z})\right] - \mathcal{D}_{KL}\left[q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z})\right]\end{aligned} . \tag{2.39}$$

Substituting the result of (2.39) into (2.38) and keeping the reformulated ELBO on the right-hand side, we obtain

$$\ln p_\theta(\mathbf{x}) - \mathcal{D}_{KL}\left[q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z}|\mathbf{x})\right] = \mathbb{E}_{q_\phi}\left[\ln p_\theta(\mathbf{x}|\boldsymbol{Z})\right] - \mathcal{D}_{KL}\left[q_\phi(\mathbf{z}) \parallel p_\theta(\mathbf{z})\right], \tag{2.40}$$

where $q_\phi(\mathbf{z})$ can be any distribution. But since we are interested in inferring $p_\theta(\mathbf{x})$, a distribution over $\mathbf{x}$, it is reasonable to construct the distribution q_ϕ which does depend on $\mathbf{x}$, i. e. $q_\phi(\mathbf{z}|\mathbf{x})$,

$$\ln p_\theta(\mathbf{x}) - \mathcal{D}_{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x})\right] = \mathbb{E}_{q_\phi}\left[\ln p_\theta(\mathbf{x}|\boldsymbol{Z})\right] - \mathcal{D}_{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})\right]. \tag{2.41}$$

The equation (2.41) lays the foundation for variational autoencoders and we will spend some time with it. Remember, that we derived the equation (2.41) from ELBO, whose formulation (2.39) is on the righ-hand side. Therefore, since we know that VAE aims to maximize $p_\theta(\mathbf{x})$, which can be achieved by maximizing ELBO, the ultimate goal it to maximize both sides of (2.41). When $\mathbf{x}$ is fixed, the left-hand side is maximized by minimizing the KL divergence. And since $p_\theta(\mathbf{z}|\mathbf{x})$ is intractable, this term will become small if q_ϕ is high-capacity. Assuming that we use arbitrarily high-capacity model, the probability distributions $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z}|\mathbf{x})$ will match and the KL divergence will become zero. The right-hand side of (2.41) can be maximized by stochastic gradient descent. Notice, that by maximization of the right-hand side, in the first term, we maximize $\ln p_\theta(\mathbf{x}|\mathbf{z})$, where $\mathbf{z}$ is the expected value with respect to the distribution $q_\phi(\mathbf{z}|\mathbf{x})$. In other words, $q_\phi(\mathbf{z}|\mathbf{x})$ is encoding $\mathbf{x}$ into $\mathbf{z}$ and $p_\theta(\mathbf{x}|\mathbf{z})$ is decoding it to reconstruct the input $\mathbf{x}$. The second term is a regularization term, which pushes the encoding distribution $q_\phi(\mathbf{z}|\mathbf{x})$ to be similar to the distribution $p_\theta(\mathbf{z})$.

We can make the optimization of the right hand side easier if we make the right decisions regarding forms of distributions $p_\theta(\mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x})$. The usual choice is to say that

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\Sigma}_\phi(\mathbf{x})\right), \tag{2.42}$$

$$p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}), \tag{2.43}$$

where $\boldsymbol{\mu}_\phi$ and $\boldsymbol{\Sigma}_\phi$ are arbitrary deterministic functions with parameters ϕ that can be learned from training data. In practice, $\boldsymbol{\mu}_\phi(\mathbf{x})$ and $\boldsymbol{\Sigma}_\phi(\mathbf{x})$

are implemented as neural networks with $\mathbf{\Sigma}_\phi$ constrained to be a diagonal matrix. The decoding distribution $p_\theta(\mathbf{x}|\mathbf{z})$ has arbitrary form and it is also implemented via neural network.

The last term on the right hands side of (2.41) is due to our choice of distribution a KL-divergence between two multivariate Gaussians, which can be solved in a closed form as

$$\begin{aligned}&\mathcal{D}_{KL}\left[\mathcal{N}\left(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0\right) \| \mathcal{N}\left(\boldsymbol{\mu}_1, \mathbf{\Sigma}_1\right)\right] \\ &= \frac{1}{2}\left(\operatorname{tr}\left(\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_0\right) + \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\right)^{\mathrm{T}}\mathbf{\Sigma}_1^{-1}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\right) - D + \ln\frac{\det\mathbf{\Sigma}_1}{\det\mathbf{\Sigma}_0}\right), \quad (2.44)\end{aligned}$$

where D is the dimension of Gaussians [7]. In our case, the equation boils down to

$$\begin{aligned}&\mathcal{D}_{KL}\left[\mathcal{N}\left(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \mathbf{\Sigma}_\phi(\mathbf{x})\right) \| \mathcal{N}\left(\mathbf{z}|\mathbf{0}, \mathbf{I}\right)\right] \\ &\qquad = \frac{1}{2}\left(\operatorname{tr}\left(\mathbf{\Sigma}_\phi(\mathbf{x})\right) + \boldsymbol{\mu}_\phi(\mathbf{x})^{\mathrm{T}}\boldsymbol{\mu}_\phi(\mathbf{x}) - D - \ln\det\mathbf{\Sigma}_\phi(\mathbf{x})\right). \quad (2.45)\end{aligned}$$

Constraining $\mathbf{\Sigma}_\phi(\mathbf{x})$ to be a diagonal matrix, we can simplify the notation so that $\left(\mathbf{\Sigma}_\phi\right)_{ii} = \sigma_i^2$ and then

$$\begin{aligned}&\mathcal{D}_{KL}\left[\mathcal{N}\left(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \mathbf{\Sigma}_\phi(\mathbf{x})\right) \| \mathcal{N}\left(\mathbf{z}|\mathbf{0}, \mathbf{I}\right)\right] = \\ &\qquad = \frac{1}{2}\left(\sum_{i=1}^{D}\sigma_i^2 + \sum_{i=1}^{D}\mu_i^2 - D - \ln\prod_{i=1}^{D}\sigma_i^2\right) \\ &\qquad = \frac{1}{2}\left(\sum_{i=1}^{D}\sigma_i^2 + \sum_{i=1}^{D}\mu_i^2 - D - \sum_{i=1}^{D}2\ln\sigma_i\right). \\ &\qquad = \frac{1}{2}\sum_{i=1}^{D}\left[\sigma_i^2 + \mu_i^2 - 1 - 2\ln\sigma_i\right]\end{aligned} \quad (2.46)$$

To compute the first term on the right hand side of (2.41), we could use sampling to estimate the expected value $\mathbb{E}_{q_\phi}$. However, to train the model using back-propagation, we need to be able to differentiate the final output with respect to each parameter in the network. This cannot be done for random sampling. We can overcome it with the reparameterization trick [12], which moves the random sampling to the input layer. Instead of sampling from $\mathcal{N}\left(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \mathbf{\Sigma}_\phi(\mathbf{x})\right)$ directly, we first sample from $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and then compute $\mathbf{z} = \boldsymbol{\mu}_\phi(\boldsymbol{X}) + \epsilon \cdot \mathbf{\Sigma}_\phi^{1/2}(\boldsymbol{X})$, so that we eventually take a gradient of

$$\mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\ln p_\theta\left(\mathbf{x}|\mathbf{z} = \boldsymbol{\mu}_\phi(\boldsymbol{X}) + \epsilon \cdot \mathbf{\Sigma}_\phi^{1/2}(\boldsymbol{X})\right) - \mathcal{D}_{KL}\left[q_\phi(\mathbf{z}|\boldsymbol{X}) \| p_\theta(\mathbf{z})\right]\right]\right] \quad (2.47)$$

The VAE that we have just build is depicted in Fig. 2.5.

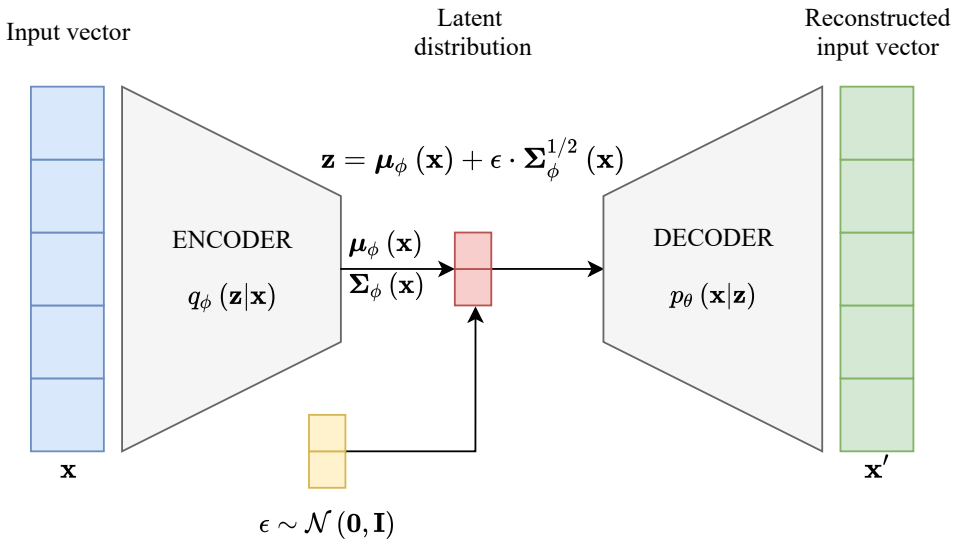


Figure 2.5: Architecture of variational encoder.

Chapter 3

Anomaly detection in group data

This section picks up the threads of group anomaly detection that we have presented in Chapter 1. In Section 3.1, we extend the notation introduced for vector data to refer to group data appropriately. The need for a different approach to treating group data is outlined in Section 3.2. The proposed ranking function for group anomaly detection is discussed in Section 3.3 and followed by the description of building blocks of the ranking function in Section 3.4 and Section 3.5.

3.1 Notation

Definition of group data follows the notation from [21]. We consider point patterns $\mathcal{X}$ of features $\mathbf{x} = (x_1, \ldots, x_D)^T$ from a feature space $\mathscr{X} \subset \mathbb{R}^D$, i. e. $\mathbf{x} \in \mathcal{X}$ and $\mathcal{X} \subset \mathscr{X} \subset \mathbb{R}^D$. We are further interested in matrix representation of point patterns $\mathbf{X} \in \mathbb{R}^{C \times D}$, $C = |\mathcal{X}|$, where C denotes the cardinality achieved by counting features of $\mathcal{X}$. A permutation of the feature axis with a permutation π is denoted by $\mathbf{X}_\pi$, i. e. $\mathbf{X} \neq \mathbf{X}_\pi$ but $\mathbf{X} \equiv \mathcal{X} \equiv \mathbf{X}_\pi$. Dataset consisting of G finite point patterns $\mathbf{X}_n$, $n = 1, \ldots, N$, of potentially varying cardinalities is denoted as $\mathcal{D} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$.

3.2 Motivation

First of all, consider what if the cardinality of all group data was identical. Then the likelihood of the point pattern $\mathbf{X} \in \mathbb{R}^{C \times D}$ with the cardinality C would be given by the joint probability density of constituent vectors

$$p(\mathbf{X}) = p(\mathbf{x}_1, \ldots, \mathbf{x}_C) = \prod_{c=1}^{C} p(\mathbf{x}_c), \tag{3.1}$$

where the joint probability density acts like a valid ranking function. Then we could simply decide for every observation according to (2.1).

However, all point patterns in the dataset are not necessarily of the same cardinality, which makes the usage of joint probability density as a ranking

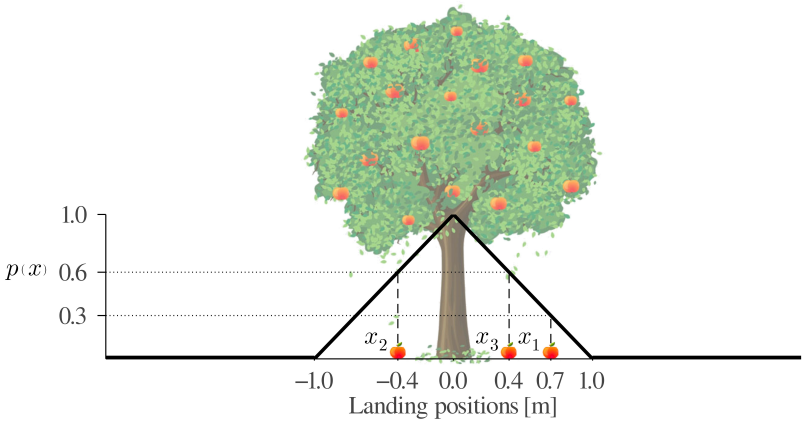


Figure 3.1: Probability density function of landing positions $p(x)$. Position $x_1 = 0.7$ m is two times less likely than equally likely positions $x_2 = -0.4$ and $x_3 = 0.4$.

function in group data misleading. With an example inspired by [22], we illustrate that group data should not be ranked according to the joint probability density of its constituent vectors.

Example. Group anomaly detection with unsuitable ranking function

Suppose, that we are given two point patterns of fallen apples from different days $\mathcal{X}_1 = \{x_1\}$, $\mathcal{X}_2 = \{x_2, x_3\}$ and a probability density function $p(x)$ of variable $x \in \mathbb{R}$ learned from normal training data instances as shown in Fig. 3.1. Assuming that apples fall independently from each other, and that daily observations are independent from day to day, we want to decide, which point pattern is more likely to be anomalous.

If we ranked observations $\mathcal{X}_1$ and $\mathcal{X}_2$ according to the joint probability density, then we would obtain the following probabilities of point patterns

$$p(x_1) = 0.3,$$
$$p(x_2, x_3) = p(x_2)\,p(x_3) = 0.36,$$

and since $p(x_1) < p(x_2, x_3)$, we would conclude that the point pattern $\mathcal{X}_1$ is more likely to be anomalous. However, if the observations had been measured in centimeter instead of meters, we would have come to contradictory result since $p(x_1) = 0.003 > 0.000036 = p(x_2, x_3)$. This inconsistency arises from comparing two unequally sized point patterns using an improper measure of goodness of fit, i.e. joint probability density as the likelihood function.

3.3 Novel ranking function

To rank group data properly, we will use a novel ranking function that considers the feature density and the cardinality distribution of the point pattern, as suggested in [22]. Moreover, according to Proposition 1 in [22],

which we quote below, we can determine the feature probability density from all features cellectively irrespective of the point pattern to which a particular feature belongs. We can also estimate the parameters of assumed cardinality distribution given all point pattern cardinalities.

Proposition. *Let $\mathcal{X}_1, \ldots, \mathcal{X}_N$ be N i. i. d. realizations of an IID-cluster with parameterized cardinality distribution $p_{c\xi}$ and feature density p_φ. Then the maximum likelihood estimate of ξ and φ is given by*

$$\hat{\xi} = \mathrm{MLE}\left(p_{c\xi}; |\mathcal{X}_1|, \ldots, |\mathcal{X}_N|\right), \tag{3.2}$$

$$\hat{\varphi} = \mathrm{MLE}\left(p_\varphi; \uplus_{n=1}^{N} \mathcal{X}_n\right), \tag{3.3}$$

where $\uplus$ denotes disjoint union.

The ranking function then reads as

$$r\left(\mathbf{X}\right) \propto p_c\left(|\mathcal{X}|\right) \frac{\prod_{\mathbf{x}\in\mathcal{X}} p\left(\mathbf{x}\right)}{\left(\|p\|_2^2\right)^{|\mathcal{X}|}}, \tag{3.4}$$

where $\prod_{\mathbf{x}\in\mathcal{X}} p\left(\mathbf{x}\right)$ is the likelihood of point pattern features, $p_c\left(|\mathcal{X}|\right)$ denotes the likelihood of a point pattern whose cardinality is $|\mathcal{X}|$, and $\|p\|_2$ is the feature probability density L2-norm. Note that we omit the parameters ξ and φ in our distribution notation.

Now, similarly to (2.1) for vector data, we can label group data according to

$$\chi\left[r\left(\mathbf{X}\right) \geq \tau\right] = \begin{cases} \text{normal} & r\left(\mathbf{X}\right) \geq \tau \\ \text{anomaly} & \text{otherwise} \end{cases}. \tag{3.5}$$

The finding that we can treat the features independently of point patterns is crucial because it allows us to manipulate features exactly as we did with vector data. Therefore, we can use the methods discussed in Chapter 2 when learning the feature probability density. However, first, we look into models of cardinality distribution, which we will use in our ranking function.

3.4 Models of cardinality distribution

At this point, the reader should be aware of the ranking function $r\left(\mathbf{X}\right)$, but yet, we did not discuss its building blocks. This section introduces one of the ranking function components, the cardinality distributions p_c.

3.4.1 Poisson distribution

The Poisson distribution is a discrete probability distribution that expresses the probability of a given number of event occurrences within an interval. It is described by the probability mass function

$$\mathrm{Pois}\left(x|\lambda\right) = \frac{e^{-\lambda}\lambda^x}{x!}, \tag{3.6}$$

with a single parameter $\lambda > 0$, which is responsible for the shape of the distribution, and one-dimensional variable x indicating the number of event occurrences.

The parameter λ signifies the average number of event occurrences and is given as

$$\mathbb{E}[X] = \sum_{x=0}^{\infty} x\mathrm{Pois}(x|\lambda) = \lambda. \tag{3.7}$$

One drawback of the Poisson distribution is that it makes strong assumptions regarding the distribution of the underlying data. In particular, that the mean equals the variance

$$\mathrm{var}[X] = \lambda. \tag{3.8}$$

While these assumptions are tenable in some settings, they are less appropriate for modelling point pattern cardinalities, which we will experimentally evaluate.

In our case, we do not know the parameter λ, but we want to estimate it from the group data $\mathcal{D}$ of N point patterns, which can be simply done using maximum likelihood estimation

$$\hat{\lambda} = \frac{1}{N}\sum_{n=1}^{N} |\mathcal{X}_n| . \tag{3.9}$$

3.4.2 Log-normal distribution

The log-normal distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed. Its probability density function is given by

$$\mathrm{lognormal}\left(x|\mu,\sigma^2\right) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln(x)-\mu}{\sigma}\right)^2}, \tag{3.10}$$

where, similarly to the Gaussian distribution, the parameters $\mu \in \mathbb{R}$ and $\sigma^2 > 0$ are the mean and variance, respectively. x is one-dimensional real-valued variable.

Although there have been some attempts to discretize the log-normal distribution [20], we decided to use the original log-normal distribution and evaluate its behaviour experimentally.

To estimate the values of the parameters μ and σ^2 from our group data $\mathcal{D}$ of N point patterns, we use the maximum likelihood estimation

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} |\mathcal{X}_n| , \tag{3.11}$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N} \left(\ln\left(|\mathcal{X}_n|\right) - \hat{\mu}\right) . \tag{3.12}$$

3.4.3 Discrete uniform distribution

The discrete probability distribution is a symmetric probability distribution wherein a finite number of values are equally likely to be observed. Its probability mass function reads as

$$\mathcal{U}(x|a,b) = \frac{1}{n}, \tag{3.13}$$

where the parameters a and b,$a \leq b$, are the limit points of the discrete interval; and $n = b - a + 1$ is the number of integer values within the limit points. Every discrete variable $x = a, \ldots, b$ has equal probability $\frac{1}{n}$.

3.5 Feature density models

In this section, we discuss the other building block of the ranking function $r(\mathbf{X})$, the feature density models and their normalization via L2-norm.

3.5.1 VAE and GMM

Assuming, that the point pattern data are i. i. d., we can treat all point pattern features independently from the point patterns that they belong to.

Hence, we can estimate the probability distribution of the point pattern features exactly the same way as with vector data.

3.5.2 L2-norm

In the denominator of (3.4), we need to compute a square of the feature probability density L2-norm, which reads as

$$\|p\|_2^2 = \int p(\mathbf{x})\, p(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{3.14}$$

Assuming that $\mathbf{x} \in \mathbb{R}^D$, where D is high, the integral over $\mathbf{x}$ becomes intractable, and we need to turn to a numerical integration method to estimate the integral. With Monte Carlo estimation, we can write

$$p(\mathbf{x}) \approx \frac{1}{N} \sum_{n=1}^{N} \delta(\mathbf{x}_n - \mathbf{x}). \tag{3.15}$$

Suppose we now substitute (3.15) for the feature probability density in (3.14). In that case, we get a tractable form of the L2-norm squared

$$\begin{aligned} \|p\|_2^2 &= \int p(\mathbf{x}) \frac{1}{N} \sum_{n=1}^{N} \delta(\mathbf{x}_n - \mathbf{x})\, \mathrm{d}\mathbf{x} \\ &= \frac{1}{N} \sum_{n=1}^{N} \int p(\mathbf{x})\, \delta(\mathbf{x}_n - \mathbf{x})\, \mathrm{d}\mathbf{x}, \\ &= \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{x}_n) \end{aligned} \tag{3.16}$$

where $\mathbf{x}_n$ can be either sample from $p(\mathbf{x})$ or, in the case of i. i. d. data, the original data vector.

Chapter 4

Experiments

The models discussed in this thesis can not only be used for anomaly detection in group data, but also in various scenarios when point patterns vary in their cardinalities.

In this chapter, we first describe the dataset that will be used for testing our models in Section 4.1, then we discuss the metric for models evaluation. In Section 4.3, the setting of the models is discussed and eventually we evaluate the results in Section 4.4

4.1 MIL dataset

MIL dataset is gathered dataset combining twenty smaller datasets for tasks varying form text categorization to image classification. Individual datasets vary in size, have different number of bags and features, etc., as can be seen in Table 4.1. We will evaluate our models using the MIL dataset [6].

4.2 Metrics used for model quality assessment

In order to comparatively assess quality of models, we need to use appropriate metric. The receiver operating characteristic (ROC) [16] curve is such a performance measurement. It is used for binary classification problems at various threshold settings. ROC curve is a probability curve that plots the true positive rate (TPR) against false positive rate (FPR).

TPN value tells us what is the proportion of positive observations that were correctly classified.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$

FPR value is on the other hand the proportion of negative observations that were incorrectly classified

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}.$$

The area under the curve (AUC) [16] is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the

dataset	p. p.	attr.	normal	anom.	medi.	mean
BrownCreeper	546.0	38.0	350.0	196.0	17.0	18.7
CorelAfrican	2000.0	9.0	1900.0	100.0	3.0	4.0
CorelBeach	2000.0	9.0	1900.0	100.0	3.0	4.0
Elephant	200.0	230.0	100.0	100.0	7.0	7.0
Fox	200.0	230.0	100.0	100.0	6.0	6.6
Musk1	91.0	166.0	45.0	46.0	4.0	5.1
Musk2	99.0	166.0	61.0	38.0	10.0	57.0
Mut1	185.0	7.0	61.0	124.0	56.0	55.8
Mut2	39.0	7.0	27.0	12.0	44.0	50.9
News1	100.0	200.0	50.0	50.0	58.0	54.4
News2	100.0	200.0	50.0	50.0	30.0	30.9
News3	100.0	200.0	50.0	50.0	52.0	51.8
Protein	190.0	9.0	166.0	24.0	145.5	138.4
Tiger	200.0	230.0	100.0	100.0	6.0	6.1
UCSB-BC	57.0	708.0	31.0	26.0	36.0	34.8
Web1	72.0	5863.0	52.0	20.0	24.0	29.8
Web2	74.0	6519.0	56.0	18.0	23.0	29.9
Web3	74.0	6306.0	60.0	14.0	24.0	33.0
Web4	74.0	6059.0	20.0	54.0	23.5	30.5
WinterWren	545.0	38.0	437.0	108.0	17.0	18.7

Table 4.1: MIL datasets with corresponding number of point patterns, number of attributes per vector, number of normal and anomalous point patterns, median and mean of point pattern cardinalities.

ROC curve. Let us consider three particular values of AUC. When AUC $= 1$, then the model is perfectly able to distinguish between positive and negative observation. Contrary, AUC $= 0$ implicates the model which classifies positive observations as negative and negative observations as positive. The model whose AUC $= 0.5$ achieves similar performance asif the classes were guessed randomly.

We will use AUC to effectively evaluate model performance.

4.3 Experiments setting

In Chapter 3, we indicated the scenario under which we will undertake the experiments. We have two models of feature probability density: VAE and GMM; and three models of cardinality distribution: Poisson, lognormal and discrete uniform. We will evaluate the performance for each combination of feature probability density and cardinality distribution, which gives us six distinct models in total. These models will be ranked according to the logarithmic form of (3.4) which reads as

$$r'(\mathbf{X}) \propto \ln p_c(|\mathcal{X}|) + \sum_{\mathbf{x}\in\mathcal{X}} \ln p(\mathbf{x}) - |\mathcal{X}| \ln\left(\|p\|_2^2\right). \tag{4.1}$$

Additionally, we also evaluate models which take into account only feature probability distribution without its normalization. Again, we have two models of probability density and three models of cardinality, which gives us six distinct models in total. Then the ranking function reads as

$$r''(\mathbf{X}) \propto \ln p_c(|\mathcal{X}|) + \sum_{\mathbf{x}\in\mathcal{X}} \ln p(\mathbf{x}). \tag{4.2}$$

Alternatively, we check the performace of models which use a some function on our point pattern data, particularly fun $= \{\text{mean}, \max\}$

$$r'''(\mathbf{X}) \propto \text{fun}\left[\ln p(\mathbf{x})\right]_{\mathbf{x}\in\mathcal{X}}. \tag{4.3}$$

Since we have two feature probability density models, we obtain four distinct models.

All together, we evaluate 16 models, each on 20 datasets. Every model is trained and evaluated 10 times per dataset and the results per dataset are averaged.

The problem is implemented in Julia programming language and we use GroupAD.jl package.

4.4 Results

First, let us look at the best performing models per the feature probability estimator. Such comparison for every dataset is shown in Fig. 4.2. An evident conclusion is that models based on VAE perform in general better than the ones using GMM. The gap between model performance is the biggest when tested on the datasets Web1, Web2, Web3 and Web4. These datasets are characteristic by more than ten times higher number of attributes than other models. Therefore, it is apparent that VAEs do a better job when estimating high-dimensional probability distributions. Also, both models did a poor job on Fox, Musk1 and Musk2 datasets. Fox and Musk1 are distinguishing due to identical number of normal an anomalous observations in the dataset, whereas Musk2 consists of relatively small amount of data with point patterns having various cardinalities. On the other hand, VAE and GMM models perform well on datasets BrownCreeper, CorelBeach, News1, News2, News3 and WinterWren. Corelbeach can be trainer finely since the model consist of multiplicatively more normal point patterns than other models. We leave the rest of mentioned good performing models to be explored by the reader.

Performance of the VAE based models are shown in Fig. 4.2. In general, all the evaluated models have similar results. However, the models that are evaluated according to $r'(\mathbf{X})$ from (4.1) have better results on datasets Mut1 and Mut2 and slightly better performance when evaluated on CoreAfrican, CoreBeach, and Elephant. The dataset Mut1 is significant by high number of anomaly point patterns and Mut2 stands out because it varies in the number of cardinalities. We turn back to the dataset Mut1 once again, note, that alternative scoring functions $r'''(\mathbf{X})$ from (4.3) that take into account mean or

even maximum of likelihoods performs well on this dataset. The Winterwren dataset is the most promiment example where ranking according to (4.1) gives significantly worse results than all other scoring functions. This dataset is characteristic only by the fact that the number of normal point patterns is four times higher than number of anomalies.

Last remark regarding VAE based models. In Section 3.4, we pointed out, that the Poisson distribution might not be suitable cardinality distribution function since its mean equals the variance. Note, that according to our experiments, all the cardinality distributions perform similarly.

Similarly, the best test AUC curves for GMM based models per dataset are shown in Fig. 4.3. Again, the models which evaluates according to $r'(\mathbf{X})$ from (4.1) achieves significantly better result on Elephant dataset. On the other hand, in GMM based models it is more frequent, that normalization pulls the score down and better performance is achieved by models with feature density and cardinality distribution that are not normalized. Such models follow the scoring $r''(\mathbf{X})$ from (4.2).

Note, that for Winterwren dataset, GMM based models can achieved similarly good results as VAE based models only when using alternative scoring according to $r'''(\mathbf{X})$ from (4.3).

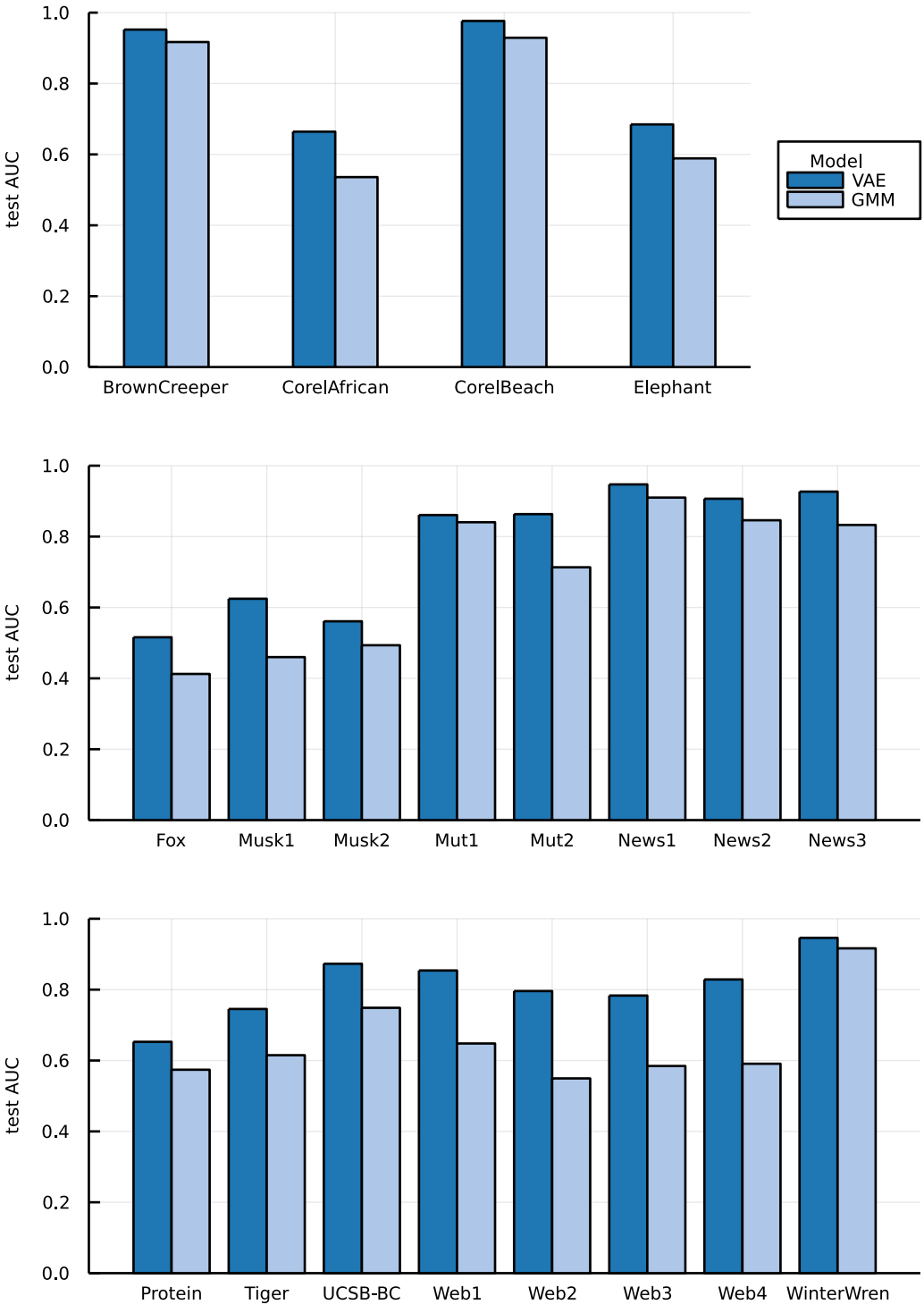


Figure 4.1: The best test AUC curve per dataset w. r. t. to the feature density estimator.

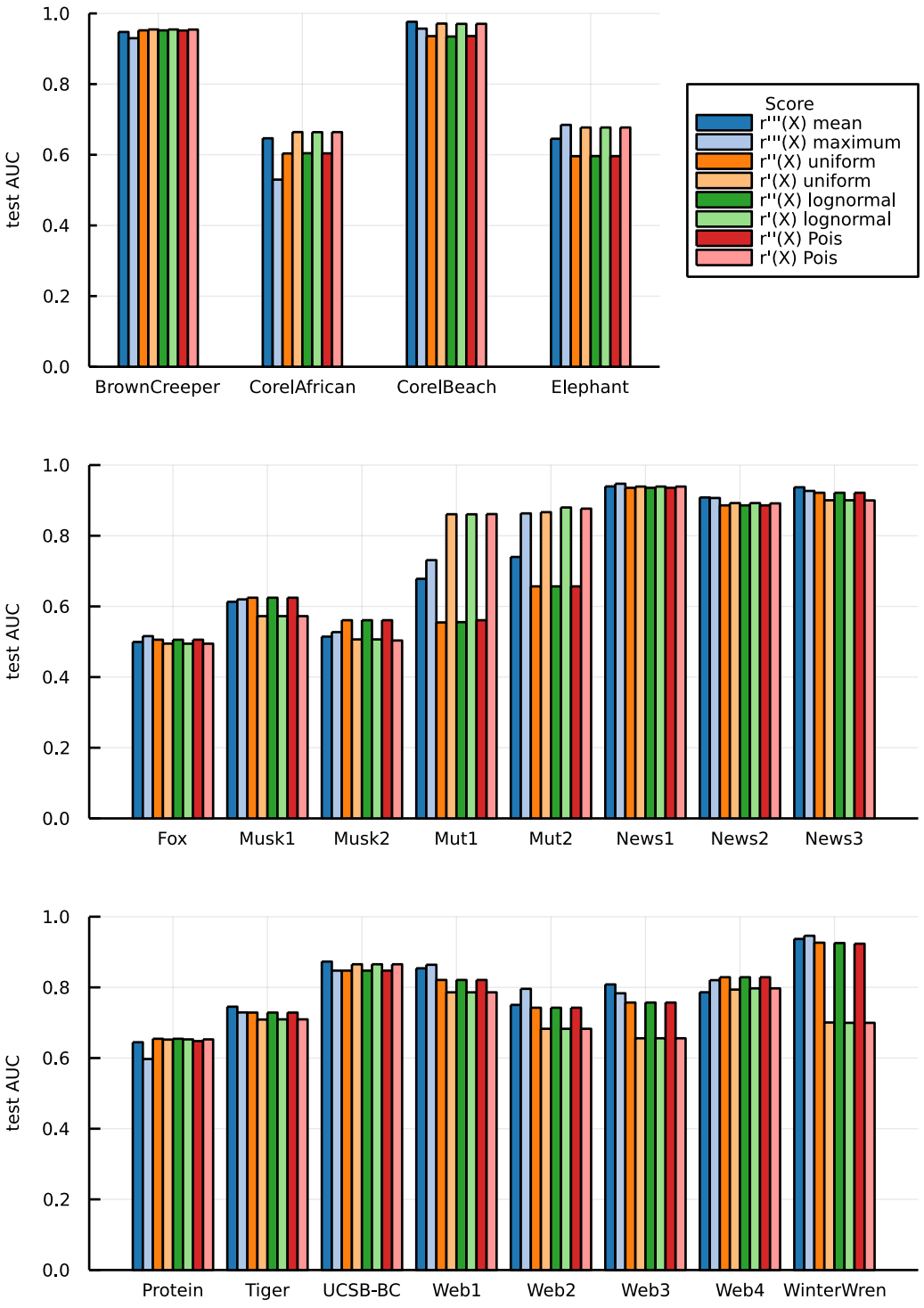


Figure 4.2: Test AUC curve for VAE using various scoring functions, per dataset.

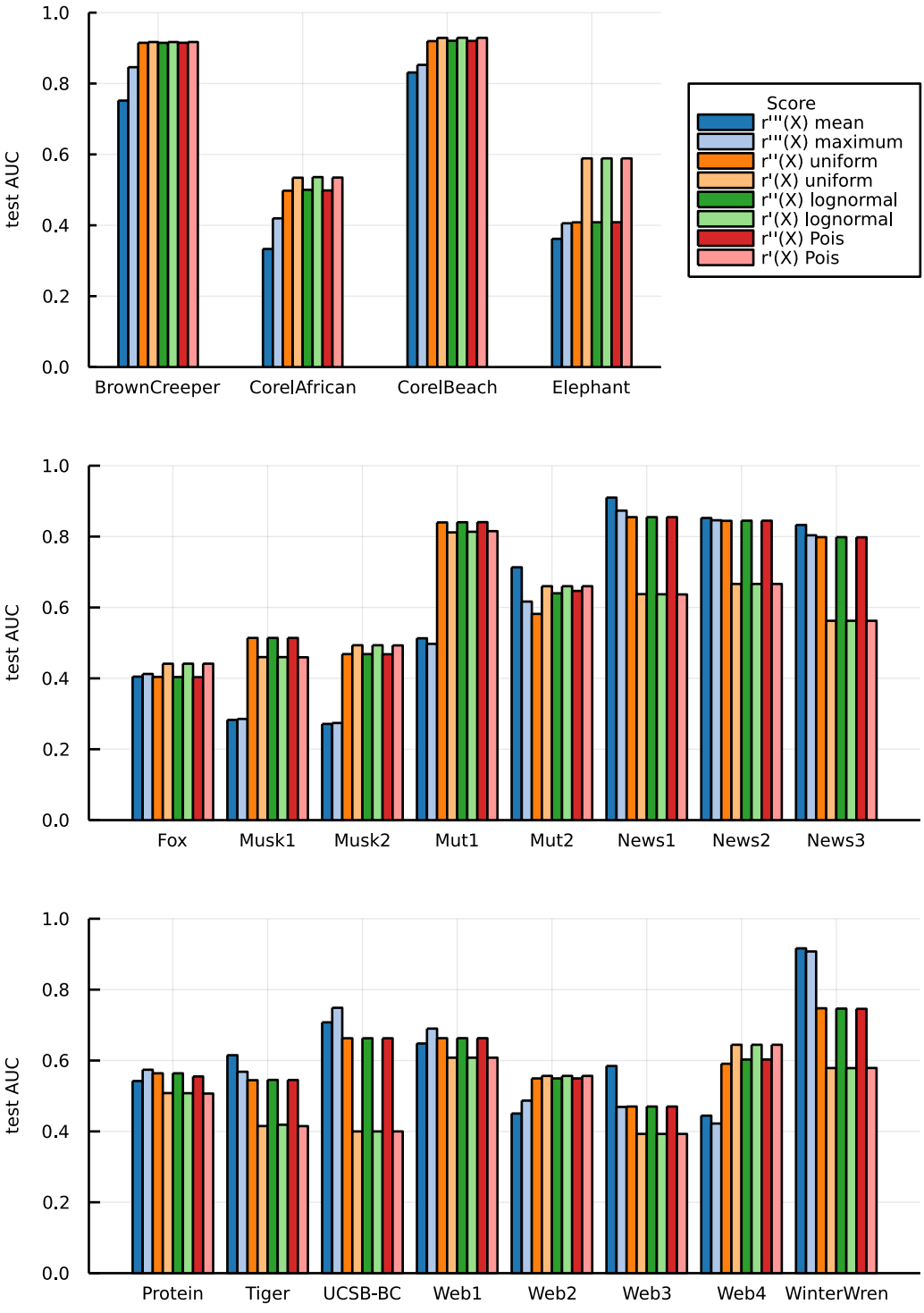


Figure 4.3: Test AUC curve for GMM using various scoring functions, per dataset.

Chapter 5

Conclusion

In this thesis, we studied anomaly detection as a problem of normal class probability density estimation. We introduced the framework which is used for anomaly detection on vector data and extended it so, that it could be used also for anomaly detection on group data.

Eventually, we experimentally evaluated the ranking function for group anomaly detection that we introduced in the theoretical part and compared its performance with alternative scoring functions. The ranking function achieves promising results which in some scenarios outperform alternative approaches as shown in experiment. There are also scenarios when it is better to use alternative scoring functions, however, this is beyond the scope of this thesis and we keep it for the future work.

Bibliography

[1] Christopher M. Bishop. Pattern Recognition and Machine Learning. page 738, jan 2006.

[2] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 2095–2102, 2018.

[3] Tom Brotherton and Tom Johnson. Anomaly detection for advanced military aircraft using neural networks. *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*, 6:63113–63123, 2001.

[4] Raghavendra Chalapathy, Edward Toth, and Sanjay Chawla. Group Anomaly Detection using Deep Generative Models. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11051 LNAI:173–189, apr 2018.

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, jul 2009.

[6] Veronika Cheplygina, David M.J. Tax, and Marco Loog. Multiple instance learning with bag dissimilarities. *Pattern Recognition*, 48(1):264–275, jan 2015.

[7] Carl Doersch. Tutorial on Variational Autoencoders. jan 2021.

[8] Pierluca D'oro, Ennio Nasca, Jonathan Masci, and Matteo Matteucci. Group Anomaly Detection via Graph Autoencoders. *NIPS Workshop*, pages 1–8, 2019.

[9] Roger Fletcher. *Practical methods of optimization*. Wiley, 2 edition, may 2000.

[10] Ralph Foorthuis. On the nature and types of anomalies: a review of deviations in data. *International Journal of Data Science and Analytics 2021*, pages 1–35, aug 2021.

[11] Changhee Han, Leonardo Rundo, Kohei Murao, Tomoyuki Noguchi, Yuki Shimahara, Zoltán Ádám Milacski, Saori Koshino, Evis Sala, Hideki Nakayama, and Shin'ichi Satoh. MADGAN: unsupervised medical anomaly detection GAN using multiple adjacent brain MRI slice reconstruction. *BMC Bioinformatics*, 22(2):1–20, apr 2021.

[12] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, Banff, AB, Canada, apr 2014. International Conference on Learning Representations, ICLR.

[13] Kishan G. Mehrotra, Chilukuri K. Mohan, and HuaMing Huang. *Anomaly Detection Principles and Algorithms*. Springer, 2017.

[14] Krikamol Muandet and Bernhard Schölkopf. One-Class Support Measure Machines for Group Anomaly Detection. *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013*, pages 449–458, mar 2013.

[15] Anvardh Nanduri and Lance Sherry. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). *2016 Integrated Communications Navigation and Surveillance (ICNS)*, pages 5C2–1– 5C2–8, may 2016.

[16] Sarang Narkhede. Understanding auc-roc curve. *Towards Data Science*, 26(1):220–227, 2018.

[17] Animesh Patcha and Jung Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470, aug 2007.

[18] Friedrich Pukelsheim. The three sigma rule. *American Statistician*, 48(2):88–91, 1994.

[19] Irhum Shafkat. Intuitively understanding variational autoencoders, 2018.

[20] Paul Sheridan and Taku Onodera. A Preferential Attachment Paradox: How Preferential Attachment Combines with Growth to Produce Networks with Log-normal In-degree Distributions. *Scientific Reports*, 8(1):1–11, feb 2018.

[21] Maximilian Soelch, Adnan Akhundov, Patrick van der Smagt, and Justin Bayer. On Deep Set Learning and the Choice of Aggregations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11727 LNCS:444–457, mar 2019.

[22] Ba-Ngu Vo, Nhan Dam, Dinh Phung, Quang N. Tran, and Ba-Tuong Vo. Model-based learning for point pattern data. *Pattern Recognition*, 84:136–151, dec 2018.

[23] Wei Wei, Jinjiu Li, Longbing Cao, Yuming Ou, and Jiahang Chen. Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, 16(4):449–475, jul 2013.