ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230260

Long-Tail Theory Under Gaussian Mixtures

Arman Bolatov[a], Maxat Tezekbayev[b], Igor Melnykov[c], Artur Pak[b], Vassilina Nikoulina[d] and Zhenisbek Assylbekov[e;*]

[a]Department of Computer Science, School of Engineering and Digital Sciences, Nazarbayev University, Kabanbay Batyr 53, Astana, Kazakhstan, 010000
[b]Department of Mathematics, School of Sciences and Humanities, Nazarbayev University, Kabanbay Batyr 53, Astana, Kazakhstan, 010000
[c]Department of Mathematics and Statistics, University of Minnesota Duluth, Duluth, MN, USA
[d]NAVER LABS Europe, 6-8 chemin de Maupertuis, 38240 Meylan, France
[e]Department of Mathematical Sciences, Purdue University Fort Wayne, Fort Wayne, IN, USA
ORCiD ID: Zhenisbek Assylbekov https://orcid.org/0000-0003-0095-9409

Abstract. We suggest a simple Gaussian mixture model for data generation that complies with Feldman's long tail theory (2020). We demonstrate that a linear classifier cannot decrease the generalization error below a certain level in the proposed model, whereas a nonlinear classifier with a memorization capacity can. This confirms that for long-tailed distributions, rare training examples must be considered for optimal generalization to new data. Finally, we show that the performance gap between linear and nonlinear models can be lessened as the tail becomes shorter in the subpopulation frequency distribution, as confirmed by experiments on synthetic and real data.

1 Introduction

In classical learning theory [17, 18], generalizing ability and model complexity are usually opposed to each other: the more complex the model,[1] the worse its generalizing ability on new data. This is well illustrated by typical curves of test and training errors as functions of the complexity of the model being trained. The training error tends to decrease whenever we increase the model complexity, that is, when we try harder to fit the data. With too much fitting, the model adapts itself too closely to the training data, and will not generalize well (i.e., have large test error).

However, modern machine learning models such as deep neural networks (DNNs) break this principle: they are usually complex enough to be able to memorize the entire training set, and nevertheless show excellent generalization ability. This phenomenon, called **benign overfitting**, was discovered empirically by Zhang et al. [20] and has since attracted the attention of many minds in the field of machine learning, both experimentalists and theorists. We refer the reader to the survey of Bartlett et al. [1] and Belkin [2] for a more comprehensive overview of benign overfitting.

In our opinion, the most adequate explanation for the *necessity* of overfitting is the **long tail theory** of Feldman [8], which considers learning from natural data (such as texts or images). The fact is that the distribution of such data usually consists of subpopulations, and the frequencies of subpopulations have a so-called long tail, i.e. examples from rare/atypical subpopulations will regularly occur in both training and test samples.

As an example, consider a typical dataset consisting of movie reviews, such as SST-2 [16]. In this dataset, each review (more precisely, each sentence) is labeled as positive or negative. If we look at a typical positive sentence,

The large-format film is well suited to capture these musicians in full regalia and the incredible IMAX sound system lets you feel the beat down to your toes.

we will notice that it contains positive phrases (underlined). At the same time, a typical negative sentence, for example

The images lack contrast, are murky, and are frequently too dark to be decipherable.

includes mostly negative phrases. However, the richness of human language allows one to write negative review sentences, which nevertheless abound in positive phrases:

Starts out with tremendous promise, introducing an intriguing and alluring premise, only to become a monumental achievement in practically every facet of inept filmmaking.

These kinds of negative reviews are not typical, and according to Feldman's long-tail theory, they constitute a separate subpopulation (or several subpopulations) in the class of negative reviews.

A similar situation is observed in the image domain. Consider the popular MNIST dataset [7], which consists of handwritten digits from 0 to 9. Typically, this dataset is used for 10-class classification, where each digit is a class. If we take one of the classes, say 3, then most of the examples in this dataset look like in Figure 1. However, there are rare and atypical examples of writing digit 3, such as in Figure 2, which are easily confused with other digits (for example, 7). Again, according to the long-tail theory, such rare examples should be allocated to a separate subpopulation (or several subopulations) within the class 3.

Feldman [8] showed that if the distribution over subpopulations has a long-tail (as in the examples above), then to achieve optimal

* Corresponding Author. Email: zhenisbek.assylbekov@gmail.com. Work done while at Nazarbayev University.
[1] By *complexity* of a model we mean its ability to fit an arbitrary dataset.

Please check ArXiv or contact the authors for any appendices or supplementary material mentioned in the paper.

Figure 1. Typical examples of writing 3 in MNIST dataset. Source: [9].

Figure 2. Atypical examples of writing 3 in MNIST dataset. Source: [9].

performance, the learning algorithm *needs* to memorize rare/atypical examples from the training set. This is formalized via a lower bound on the generalization error, which is proportional to the number of mislabeled training examples (and the proportionality coefficient depends on the long-tailed distribution over subpopulations). Thus, in order for the learning algorithm to be able to reduce the generalization error to a minimum, it *needs* to fit *all* examples from the training set, including rare/atypical ones. And this, in turn, entails the need to use more complex models (with a larger number of parameters), since simple and underparameterized models are not able to memorize such atypical cases. However, Feldman's work does not provide conditions that guarantee successful learning from natural data, i.e. there are no upper bounds on the generalization error.

At the same time, it should be noted that recently we have seen an increase in the number of works in which guarantees of successful learning for **interpolating methods** are mathematically proved. For example, Chatterji and Long [5] showed that an overparameterized max-margin linear classifier trained on a linearly separable-with-noise data can perfectly fit the training sample (interpolate), yet generalize to new data nearly optimally. A similar result was shown by Shamir [15], and extensions to neural networks with one hidden dense layer and one hidden convolutional layer were recently given by Frei et al. [11] and Cao et al. [4] respectively. We are mainly concerned with the assumptions on data generation made in these works: the setup of a *single*, albeit noisy, subpopulation within each class is completely different from what Feldman [8] suggested in his long-tail theory. Moreover, in such a setup, there are non-interpolating algorithms with the same (or better) generalization guarantees. Accordingly, memorizing rare noisy examples is *not* necessary to achieve optimal generalization error.

In this paper, we propose a simple Gaussian mixture model for data generation that is consistent with Feldman's long-tail theory. Further, we show that, within the framework of the proposed model, a linear classifier cannot reduce the generalization error below a certain limit, regardless of the number of parameters used. At the same time, there is a nonlinear model with a larger number of parameters, which can reduce the generalization error below this limit. Thus we show that fitting rare/atypical training examples is *necessary* for optimal generalization to new data. Finally, we prove that the performance gap between linear and non-linear models can be decreased as the tail shortens in the subpopulation frequency distribution. This result is confirmed by experiments on both synthetic and real data.

2 Data Generating Model

Motivating Example. To motivate our choice of data-generating model, let us go back to the movie review examples. For simplicity, let us imagine that we can identify positive and negative phrases in the reviews. Next, let us represent each review sentence as a single number

$$x = (\#\text{positive phrases}) - (\#\text{negative phrases})$$

It is intuitively clear that for most positive sentences, $x > 0$, and for most negative sentences, $x < 0$. However, as we mentioned in the Introduction, there are rare examples of negative review sentences that abound in positive phrases, i.e. for which $x > 0$.[2] This observation leads us to the following data distribution model: for all positive reviews, x is concentrated at the point $\mu_+ > 0$; for most negative reviews, x is concentrated at the point $\mu_-^{\text{maj}} < 0$; while there is a minority of negative reviews for which x is concentrated at the point $\mu_-^{\text{min}} > 0$ (Figure 3). In what follows, we formalize this model.

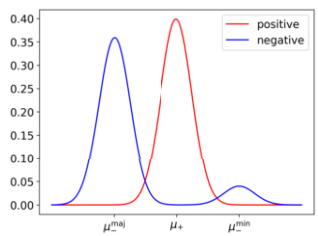

Figure 3. Simplified data distribution model.

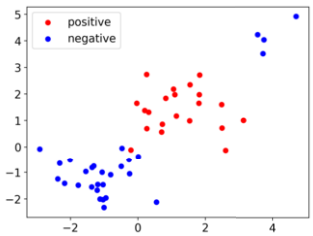

Figure 4. A sample from our data generating model.

Notation. We let $\mathbb{R}$ denote the real numbers. Bold-faced lowercase letters ($\mathbf{x}$) denote vectors in $\mathbb{R}^d$, bold-faced uppercase letters ($\mathbf{A}$, $\mathbf{X}$) denote matrices and random vectors, regular lowercase letters (x) denote scalars, regular uppercase letters (X) denote random variables. $\|\cdot\|$ denotes the Euclidean norm: $\|\mathbf{x}\| := \sqrt{\mathbf{x}^\top \mathbf{x}}$. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes multivariate Gaussian with mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$. 'p.d.f.' stands for 'probability density function', and 'c.d.f.' stands for 'cumulative distribution function'. The p.d.f. of $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is denoted by $f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. The p.d.f. and c.d.f. of $Z \sim \mathcal{N}(0, 1)$ are denoted by $\phi(z)$ and $\Phi(z)$ respectively. We also use the standard big O notation, such as $O(\cdot)$, $\widetilde{O}(\cdot)$, $\Omega(\cdot)$, $\Theta(\cdot)$, [6, Chapter 3].

The Model. Let $\mathbf{X} \in \mathbb{R}^d$ be the feature vector, and $Y \in \{-1, +1\}$ its class label. We assume that Y is a Rademacher random variable, i.e.

$$\Pr[Y = -1] = \Pr[Y = +1] = \frac{1}{2}. \tag{1}$$

For the positive class, we assume that the class-conditional distribution of $\mathbf{X}$ is a spherical (a.k.a. isotropic) Gaussian centered at $\boldsymbol{\mu} \in \mathbb{R}^d$, i.e.

$$(\mathbf{X} \mid Y = +1) \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I}). \tag{2}$$

Whereas for the negative class, the class-conditional distribution of $\mathbf{X}$ is a *mixture* of two spherical Gaussians centered at $-\boldsymbol{\mu}$ and $3\boldsymbol{\mu}$:

$$(\mathbf{X} \mid Y = -1, K = 1) \sim \mathcal{N}(-\boldsymbol{\mu}, \sigma^2\mathbf{I}), \tag{3}$$

$$(\mathbf{X} \mid Y = -1, K = 2) \sim \mathcal{N}(3\boldsymbol{\mu}, \sigma^2\mathbf{I}). \tag{4}$$

Here, the latent random variable K represents the mixture component. With $K = 1$, features are generated from the distribution of typical negative examples, whose proportion is $p > 1/2$ of all negative examples (i.e., this is the cluster of a majority of negative examples). With $K = 2$, features are generated from the distribution of

[2] And vice versa: there are rare examples of positive review sentences that abound in negative phrases. However, for ease of analysis, we will omit this case.

atypical/rare negative examples, whose proportion is $(1-p) < 1/2$ of all negative examples:

$$\Pr[K = 1 \mid Y = -1] = p, \qquad p > \frac{1}{2} \tag{5}$$

$$\Pr[K = 2 \mid Y = -1] = 1 - p. \tag{6}$$

We center the atypical examples at $3\boldsymbol{\mu}$ so that the distance between the means of neighboring Gaussians is $2\|\boldsymbol{\mu}\|$, and this simplifies the analysis. The assumption that the Gaussians are isotropic with equal covariances is also made to simplify the analysis. The centers of the Gaussians are located on the same straight line to prevent the linear separability of finite samples generated from our model. We emphasize that our goal is to build a simple data generating model that is consistent with Feldman's long-tail theory. Building a model that better agrees with real data is beyond the scope of this work and is a reasonable direction for further research. However, we believe that our model captures important features of the distribution of real data, such as the presence of rare subpopulations. We also emphasize that our model makes sense when $p > 1/2$, but not too close to 1 for rare examples to occur in a finite sample.

The distribution over $\mathbb{R}^d \times \{-1, +1\}$ given by (1)–(6) will be denoted by $\mathcal{D}$. Figure 4 shows a sample of size 50 from our data model with $d = 2$ and $p = 0.9$.

3 Classifiers

In this section, we consider two classifiers—Linear discriminant analysis and Mixture discriminant analysis—and examine their performance on data generated from our model. Let $\mathcal{P}$ be a distribution over $\mathbb{R}^d \times \{-1, +1\}$. For a classifier $h : \mathbb{R}^d \to \{-1, +1\}$, we define its generalization error (or misclassification error rate, or simply error) with respect to $\mathcal{P}$ as

$$\underset{\mathcal{P}}{\operatorname{err}}[h] := \Pr_{\mathbf{X}, Y \sim \mathcal{P}}[h(\mathbf{X}) \neq Y]. \tag{7}$$

When h is parameterized (as is the case for the classifiers that we consider), and the parameters are estimated based on a sample $S := \{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ of i.i.d. observations from $\mathcal{P}$, we denote the resulting classifier as h_S and consider its expected error $\mathbb{E}_{S \sim \mathcal{P}^n}[\operatorname{err}_{\mathcal{P}}[h_S]]$, where expectation is over samples S of size n from $\mathcal{P}$.

3.1 Linear Discriminant Analysis (LDA)

LDA [10] is a generative classifier whose simplest version makes almost the same assumptions about data distribution as our data generating model $\mathcal{D}$. The only difference is that for the negative class, not a mixture, but one Gaussian is used, i.e. instead of the assumptions (3)–(6), one has

$$(\mathbf{X} \mid Y = -1) \sim \mathcal{N}(\boldsymbol{\mu}_-, \sigma^2 \mathbf{I}). \tag{8}$$

The LDA classifier that has access to the true $\boldsymbol{\mu}$, σ, and p can be written as

$$h^{\text{LDA}}(\mathbf{x}) = \begin{cases} +1 & \text{if } f(\mathbf{x}; \boldsymbol{\mu}, \sigma^2_{\text{LDA}}\mathbf{I}) \geq f(\mathbf{x}; \boldsymbol{\mu}_-, \sigma^2_{\text{LDA}}\mathbf{I}) \\ -1 & \text{otherwise} \end{cases}, \tag{9}$$

where $\boldsymbol{\mu}_-$ and σ^2_{LDA} are functions of $\boldsymbol{\mu}$, σ^2, and p, that can be derived under the assumptions of the data generating model $\mathcal{D}$ (see Appendix A.2 [3]). It is well known [13, Section 4.3] that in this case the decision boundary of the LDA consists of a set of points equidistant from $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_-$, i.e. it is a hyperplane, which is the perpendicular bisector of the line segment connecting $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_-$. It is easy to see that the data distribution used in LDA is a special case of our model when $p = 1$, i.e. when the proportion of atypical examples $(1-p)$ is zero and the classes are linearly separable.[3] At the same time, in our data model, for $1 - p = \Omega(1/n)$, classes cannot be linearly separated, which means that LDA will fundamentally lack the ability to fit examples from the minority subpopulation of the negative class. This is formalized in the following lemma.

Lemma 1. *Let $S \sim \mathcal{D}^n$ be a random sample from our data generating model $\mathcal{D}$ with unknown $\boldsymbol{\mu}$ and known σ^2 and p. Let h_S^{LDA} be the LDA classifier trained on S with the method of moments under the assumptions* (1)*,* (2)*, and* (8)*. Then*

$$\begin{aligned} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}\left[\underset{\mathcal{D}}{\operatorname{err}}[h_S^{LDA}]\right] = \frac{1}{2}\Bigg[\Phi\left(-(2p-1)\frac{\|\boldsymbol{\mu}\|}{\sigma}\right) \\ + p\Phi\left(-(3-2p)\frac{\|\boldsymbol{\mu}\|}{\sigma}\right) + (1-p)\Phi\left((2p+1)\frac{\|\boldsymbol{\mu}\|}{\sigma}\right)\Bigg] \\ + \widetilde{O}\left(\sqrt{\frac{d}{n}}\right). \end{aligned} \tag{10}$$

Proof. See the full version [3]. □

The right-hand side (RHS) of (10) without the last term $\widetilde{O}\left(\sqrt{d/n}\right)$ is the misclassification error of the LDA classifier given by (9), i.e. when the true parameter $\boldsymbol{\mu}$ is known to the classifier. In practice, it is estimated from S adding a so-called estimation error in the RHS of (10).

For $p \in (1/2, 1)$, we have $(2p-1) \in (0, 1)$ and $3 - 2p \in (1, 2)$. Using the Chernoff bound $\Phi(-x) \leq \exp(-x^2/2)$ for $x > 0$, and $\Phi(x) = 1 - \Phi(-x)$, we can qualitatively assess the bound (10) as

$$\begin{aligned} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}\left[\underset{\mathcal{D}}{\operatorname{err}}[h_S^{\text{LDA}}]\right] \\ = \frac{1-p}{2} + \exp\left(-\Omega\left(\frac{\|\boldsymbol{\mu}\|^2}{2\sigma^2}\right)\right) + \widetilde{O}\left(\sqrt{\frac{d}{n}}\right). \end{aligned} \tag{11}$$

This confirms the impossibility of the LDA classifier to reduce the generalization error arbitrarily close to zero, no matter how far we place the Gaussians from each other. Moreover, we note that the first term in (11) does not depend on d, the dimensionality of the sample space. Therefore, regardless of the dimensionality, the LDA classifier will *not* be able to interpolate the training sample when $p < 1$, i.e. when there is a minority subpopulation in the negative class. This is in stark contrast with the previous studies on interpolating linear methods [5, 15].

3.2 Mixture Discriminant Analysis (MDA)

MDA [12] is a generative classifier that in general assumes that the data in each class is generated from a mixture of Gaussians. In our case, we can consider a version of MDA that makes precisely the assumptions (1)–(6) that our data generating model $\mathcal{D}$ makes. Hence, the MDA classifier (that knows the true values of the parameters) can

[3] with high probability over a choice of a finite sample given sufficiently large $\|\boldsymbol{\mu}\|$

be written as

$$h^{\text{MDA}}(\mathbf{x}) = \begin{cases} +1 & \text{if } \ \frac{1}{2} f(\mathbf{x}; \boldsymbol{\mu}, \sigma^2 \mathbf{I}) \geq \frac{p}{2} f(\mathbf{x}; -\boldsymbol{\mu}, \sigma^2 \mathbf{I}) \\ & \text{and } \ \frac{1}{2} f(\mathbf{x}; \boldsymbol{\mu}, \sigma^2 \mathbf{I}) \geq \frac{1-p}{2} f(\mathbf{x}; 3\boldsymbol{\mu}, \sigma^2 \mathbf{I}) \\ -1 & \text{otherwise} \end{cases} \tag{12}$$

Obviously, such an MDA classifier can take into account the presence of a minority subpopulation in the negative class, since it has the ability to fit a separate third Gaussian to this subpopulation. Not surprisingly, the MDA classifier (12) has a near-to-optimal generalizing ability, as presented in the following lemma.

Lemma 2. *Let $S \sim \mathcal{D}^n$ be a random sample from our data generating model $\mathcal{D}$ with unknown $\boldsymbol{\mu}$ and known σ^2 and p. Let h_S^{MDA} be the MDA classifier trained on S with the method of moments under the assumptions* (1)–(6)*. Then*

$$\begin{aligned} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n} \left[\mathop{\text{err}}_{\mathcal{D}} [h_S^{MDA}] \right] \leq \frac{1}{2} \Bigg[& \Phi \left(-\frac{\|\boldsymbol{\mu}\|}{\sigma} + \frac{\sigma \ln p}{2\|\boldsymbol{\mu}\|} \right) \\ & + \Phi \left(-\frac{\|\boldsymbol{\mu}\|}{\sigma} + \frac{\sigma \ln (1-p)}{2\|\boldsymbol{\mu}\|} \right) + p \cdot \Phi \left(-\frac{\|\boldsymbol{\mu}\|}{\sigma} - \frac{\sigma \ln p}{2\|\boldsymbol{\mu}\|} \right) \\ & + (1-p) \cdot \Phi \left(-\frac{\|\boldsymbol{\mu}\|}{\sigma} - \frac{\sigma \ln (1-p)}{\|\boldsymbol{\mu}\|} \right) \Bigg] + \widetilde{O} \left(\sqrt{\frac{d}{n}} \right). \end{aligned} \tag{13}$$

Proof. See the full version [3]. □

Arguing as in the case of LDA, we can estimate the order of the bound (13) as

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^n} \left[\mathop{\text{err}}_{\mathcal{D}} [h_S^{\text{MDA}}] \right] \leq \exp \left(-\Omega \left(\frac{\|\boldsymbol{\mu}\|^2}{2\sigma^2} \right) \right) + \widetilde{O} \left(\sqrt{\frac{d}{n}} \right). \tag{14}$$

Thus, by placing the Gaussians far enough apart, the optimal error of the MDA classifier can be made arbitrarily close to zero. We emphasize that this is only due to the ability of the MDA classifier to fit (memorize) examples from the minority subpopulation $\mathcal{N}(3\boldsymbol{\mu}, \sigma^2 \mathbf{I})$ of the negative class. Roughly speaking, MDA has the ability to allocate some of its parameters for fitting atypical examples, while LDA simply does not have such an opportunity.

Finally, we remark that the term $\widetilde{O}\left(\sqrt{d/n}\right)$ in the RHS of (13) is due to the error in estimating the model parameter $\boldsymbol{\mu}$ from the training sample S.

4 Performance Gap between LDA and MDA

Using the bounds (11) and (14) we can already estimate the expected difference between the LDA and MDA errors as

$$\begin{aligned} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n} \left[\mathop{\text{err}}_{\mathcal{D}} [h_S^{\text{LDA}}] - \mathop{\text{err}}_{\mathcal{D}} [h_S^{\text{MDA}}] \right] \\ \geq \frac{1-p}{2} - \exp \left(-\Omega \left(\frac{\|\boldsymbol{\mu}\|^2}{2\sigma^2} \right) \right) + \widetilde{O} \left(\sqrt{\frac{d}{n}} \right). \end{aligned} \tag{15}$$

However, a closer analysis gives us the following

Theorem 1. *Let $S \sim \mathcal{D}^n$ be a random sample from our data generating model $\mathcal{D}$, and let h_S^{LDA} be the LDA classifier trained on S under the assumptions* (1)*,* (2)*,* (8)*, and let h_S^{MDA} be the MDA classifier trained on S under the assumptions* (1)–(6)*. Then*

$$\begin{aligned} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n} \left[\mathop{\text{err}}_{\mathcal{D}} [h_S^{LDA}] - \mathop{\text{err}}_{\mathcal{D}} [h_S^{MDA}] \right] \\ \geq \frac{1-p}{2} - \exp \left(-\frac{\|\boldsymbol{\mu}\|^2}{2\sigma^2} \right) + \widetilde{O} \left(\sqrt{\frac{d}{n}} \right). \end{aligned} \tag{16}$$

Proof. See the full version [3]. □

The advantage of the bound (16) is that, in comparison with (15), here the second term in the RHS is written explicitly, i.e., without using the big O notation. This was done through careful analysis of the original bounds from Lemmas 1 and 2.

Theorem 1 implies the main conclusion of our work: *there is a performance gap between a simple model that is unable to memorize rare examples from the tail of the distribution, and a complex model that is able to fit such examples. Moreover, the gap can be made smaller when the proportion of atypical examples is smaller*. From (10) and (13), it is easy to see that for the "ideal" LDA and MDA (that have access to the true $\boldsymbol{\mu}$), we have

$$\mathop{\text{err}}_{\mathcal{D}} \left[h^{\text{LDA}} \right] \xrightarrow{p \to 1} \Phi \left(-\frac{\|\boldsymbol{\mu}\|}{\sigma} \right), \quad \mathop{\text{err}}_{\mathcal{D}} \left[h^{\text{MDA}} \right] \xrightarrow{p \to 1} \Phi \left(-\frac{\|\boldsymbol{\mu}\|}{\sigma} \right).$$

I.e., the gap between LDA and MDA is minimal at $p = 1$ (when there is no minority subpopulation in the negative class), which is expected.

Implications for Real Data. Unfortunately, our conclusion is practically impossible to test directly on real data, since we cannot be sure that its distribution resembles our model $\mathcal{D}$, and can be more complex. Moreover, our conclusion is drawn within the framework of generative classification models LDA and MDA, while in practice discriminative models such as logistic regression and multilayer neural networks are usually used, which make much fewer assumptions about the distribution of data. However, we will be able to test our conclusion *indirectly* in realistic settings if there is a way to identify training examples from rare subpopulations. Fortunately, this is exactly what the memorization score introduced by Feldman and Zhang does [8, 9].

For a learning algorithm A operating on a dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the amount of label memorization by A on example $(\mathbf{x}_i, y_i) \in S$ is defined as

$$\begin{aligned} \text{mem}[A, S, i] \\ := \Pr_{h \leftarrow A(S)} [h(\mathbf{x}_i) = y_i] - \Pr_{h \leftarrow A(S^{\setminus i})} [h(\mathbf{x}_i) = y_i], \end{aligned} \tag{17}$$

where $S^{\setminus i}$ denotes the dataset S with $(\mathbf{x}_i, y_i)$ removed and probability is taken over the randomness of the algorithm A (such as random initialization). One thing to keep in mind is that the memorization score itself must be calculated through a learner that *can* memorize, for example, an MDA with enough components, a neural network, or a nearest neighbor classifier.

As we can see, the memorization score will be high for examples that are difficult (or even impossible) to correctly classify using other examples in S, provided that the learning algorithm is flexible enough to (almost) completely fit the training set. For example, under our data generating model, for p close enough to 1 (so that $1 - p = \Theta(1/n)$), these are precisely the points generated by the minority subpopulation of the negative class. Accordingly, the shortening of the tail, i.e. making p closer to 1 can be simulated by discarding examples from the training set for which the memorization score is high. But the distribution of the test sample will not change, since we are not discarding top memorized examples from it. This is because the memorization score can only be calculated on the training sample and, accordingly, the most memorized examples can only be discarded from the long tail of the training sample.

For clarity, let us re-denote the distribution given by formulas (1)–(6) as $\mathcal{D}_p$. Then we are interested in the expected error of the classifier, which was trained on a sample from $\mathcal{D}_q$, but is tested on a sample from D_p, i.e. $\mathbb{E}_{S\sim\mathcal{D}_q}\left[\mathrm{err}_{\mathcal{D}_p}[h_S]\right]$. Fortunately, the analysis of this case resembles the analysis of the case when $q = p$. Denoting $q := 1 - \frac{1}{t}$, $t > 2$, we can prove the following (asymptotic in t) results for the LDA and MDA classifiers.

Theorem 2. *Let $S \sim \mathcal{D}^n_{1-1/t}$. Then for the LDA and MDA classifiers trained on S but evaluated on examples from $\mathcal{D}_p$ we have*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}_{1-1/t}}\left[\mathop{\mathrm{err}}_{\mathcal{D}_p}[h_S^{LDA}]\right] = \frac{1+p}{2}\Phi\left(-\frac{\|\boldsymbol{\mu}\|}{\sigma}\right) + \frac{1-p}{2}\underbrace{\Phi\left(\frac{3\|\boldsymbol{\mu}\|}{\sigma}\right)}_{\text{(A)}} + \Theta\left(\frac{1}{t}\right) + \widetilde{O}\left(\sqrt{\frac{d}{n}}\right) \tag{18}$$

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}_{1-1/t}}\left[\mathop{\mathrm{err}}_{\mathcal{D}_p}[h_S^{MDA}]\right] \le \frac{1+p}{2}\Phi\left(-\frac{\|\boldsymbol{\mu}\|}{\sigma}\right) + \frac{1-p}{2}\underbrace{\Phi\left(-\frac{\|\boldsymbol{\mu}\|}{\sigma} + \frac{\sigma\ln t}{2\|\boldsymbol{\mu}\|}\right)}_{\text{(B)}} + \Theta\left(\frac{1}{t}\right) + \widetilde{O}\left(\sqrt{\frac{d}{n}}\right) \tag{19}$$

Proof. See the full version [3]. □

As we can see, the difference between the bounds (18) and (19) is mainly in the terms marked as (A) and (B). It is clear that (A) > (B) as long as $t < \exp(8\|\boldsymbol{\mu}\|^2/\sigma^2)$. For example, when $\|\boldsymbol{\mu}\| = 2$, and $\sigma = 1$, we have $\exp(8\|\boldsymbol{\mu}\|^2/\sigma^2) \approx 8 \cdot 10^{13}$. Thus, the gap between LDA error and the upper bound on MDA error remains feasible even for large values of t (i.e. when q is close to 1).

5 Experiments

In this section, we empirically validate the predictions from our theory for synthetic data (generated from our model $\mathcal{D}$) as well as for real data, the distribution of which is not necessarily the same as $\mathcal{D}$, but shares its main characteristics, such as the presence of minority subpopulations in at least one of the classes.[4]

5.1 Synthetic Data

First of all, we verify our error bounds from Lemmas 1 and 2 experimentally. To do this, we generate training and test sets from our data model $\mathcal{D}$, fit LDA and MDA to the training set, compute the misclassification errors on the test set, and compare with theoretical bounds (10) and (13) modulo the asymptotic terms $\widetilde{O}(\sqrt{d/n})$. Since the bounds depend on several parameters, we vary each of these parameters while keeping the others fixed. Unless otherwise specified, the default values of the parameters are: $d = 50$, $p = 0.9$, $\|\boldsymbol{\mu}\| = 2$, $\sigma = 1$, $n = 7000$. Test samples are of size $n_{\text{test}} = 3000$. For each variable parameter value, we generate 10 training and test samples and estimate the generalization errors with 95% confidence intervals across test samples.

[4] The code for reproducing the experiments is at https://github.com/armanbolatov/long_tail. The random seeds we used are indicated in the code.

Dependence on $\|\boldsymbol{\mu}\|$. To test the dependence of error bounds on $\|\boldsymbol{\mu}\|$, we vary $\|\boldsymbol{\mu}\|$ in the interval $[2, 6]$ with a step 0.08. For each value of $\|\boldsymbol{\mu}\|$, we take a random direction in $\mathbb{R}^d$, and place $\boldsymbol{\mu}$, $-\boldsymbol{\mu}$, and $3\boldsymbol{\mu}$ along that direction. The results of the experiments are shown in Figure 6. As we can see, the empirical errors are generally consistent with our bounds. The LDA test error is statistically close to our LDA error bound (10), as the latter is mainly within the 95% confidence band. Meanwhile, the MDA test error is significantly lower than our MDA error bound (13). This is not surprising because, as we already mentioned in Section 3, for LDA we derived the exact misclassification error modulo $\widetilde{O}(\sqrt{d/n})$, while for MDA we got the *upper* bound for the error.

Dependence on p. To test the dependence of error bounds on p, we vary p in the interval $[0.5, 1]$ with a step 0.01. The results are shown in Figure 5. As we can see, the situation is similar to the previous one: for LDA, the empirical error agrees well with our formula (10), especially for p closer to 1; while for MDA, in most cases, it is significantly below our bound (13).

Dependence on n. To check the correctness of the order $\widetilde{O}(\sqrt{d/n})$ of the estimation error, we consider the expression

$$|\text{Test Error} - \text{Error Bound}| \cdot \sqrt{\frac{n}{d\ln n}} \tag{20}$$

for increasing n. The results are shown in Figure 7. Here we observe the boundedness of the expression (20), which confirms that the order $\widetilde{O}(\sqrt{d/n})$ of the estimation error is correct.

Training on $\mathcal{D}_q$, but Testing on $\mathcal{D}_p$. Finally, for experimental verification of the conclusions from Theorem 2, we generate training samples from $D_{1-1/t}$, and test samples from D_p. We vary t in the interval $[10, 2000]$ and observe the behavior of the LDA and MDA test errors. The results are shown in Figure 8. As we can see, the empirical error curves agree with the predictions of our Theorem 2. Namely, the LDA error exceeds the MDA error, and the gap between the two remains feasible with the increase of t.

5.2 Real Data

We conduct our experiments[5] on SST-2 [16], which is a dataset for sentence-level binary (positive vs. negative) sentiment classification. It has 6920 training, 872 validation, and 1821 test examples. We use the pre-trained Distill-BERT model [14] that consists of 6 transformer layers, where each layer is composed of 12 attention heads. We take the representation of the `[CLS]` token from the 6th layer for classification. We train the network with Adam, setting the learning rate and batch size to 10^{-6} and 100, respectively.

Calculating the memorization score according to (17) requires retraining the network for every training example (that is being removed) which is not computationally feasible. Thus we approximate the memorization score using the method of Zheng and Jiang [21].

We finetune two versions of the pre-trained Distill-BERT on SST-2: in one we freeze all layers except the top classification layer, in the second we finetune all layers. The first model—called Linear—is essentially a linear classifier (logistic regression), which receives

[5] Our computing infrastructure for these experiments is as follows. CPU: Intel Core i9-10900X CPU @ 3.70GHz, GPU: 2× Nvidia RTX 3090, RAM: 128 Gb, Operating System: Ubuntu 22.04.1 LTS, torch: 1.13.1, cuda: 11.7, pandas: 1.5.3.

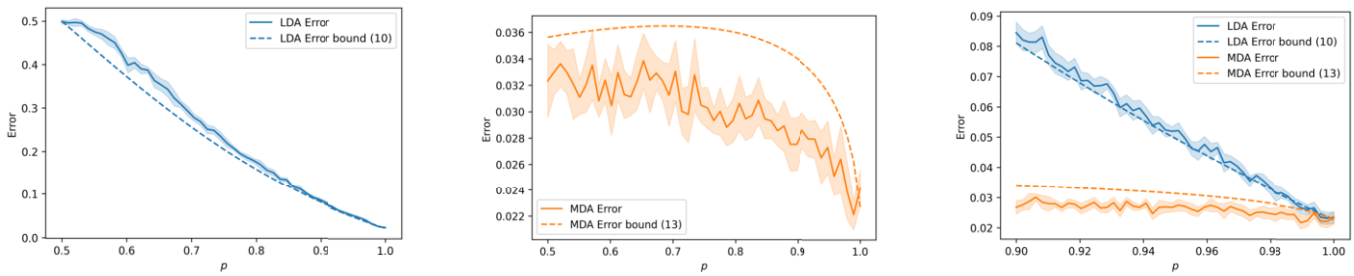

Figure 5. Comparison of empirical errors (solid) and theoretical error bounds (dashed) when p varies. The shaded areas are 95% confidence bands around the average across 10 runs. The values of the remaining parameters are fixed as follows: $d = 50$, $\|\boldsymbol{\mu}\| = 2$, $\sigma = 1$, $n = 7000$, $n_{\text{test}} = 3000$.

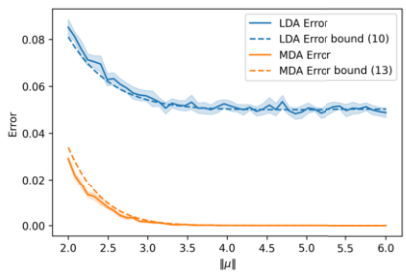

Figure 6. Comparison of empirical errors (solid) and theoretical error bounds (dashed) when $\|\boldsymbol{\mu}\|$ varies. The shaded areas are 95% confidence bands around the average across 10 runs. The values of the remaining parameters are fixed as follows: $d = 50$, $p = 0.9$, $\sigma = 1$, $n = 7000$, $n_{\text{test}} = 3000$.

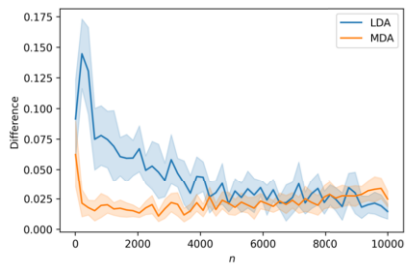

Figure 7. Verifying the order $\widetilde{O}\left(\sqrt{\frac{d}{n}}\right)$ of the estimation error by plotting $|\text{Test Error} - \text{Error Bound}| \cdot \sqrt{\frac{n}{d \ln n}}$ for increasing values of n. The values of the remaining parameters are fixed as follows: $d = 50$, $\|\boldsymbol{\mu}\| = 2$, $p = 0.9$, $\sigma = 1$, $n_{\text{test}} = 3000$. The shaded areas are 95% confidence bands around the average across 10 runs.

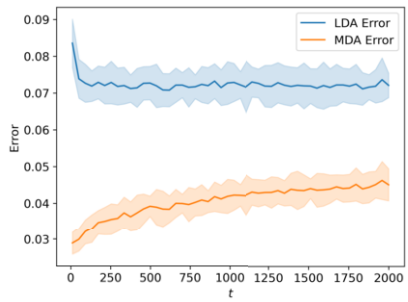

Figure 8. Comparison of the empirical LDA and MDA errors when training on $\mathcal{D}_{1-1/t}$ but testing on $\mathcal{D}_p$, as t grows. The values of the remaining parameters are fixed as follows: $d = 50$, $p = 0.9$, $\|\boldsymbol{\mu}\| = 2$, $\sigma = 1$, $n = 10000$, $n_{\text{test}} = 10000$. The shaded areas are 95% confidence bands around the average across 20 runs.

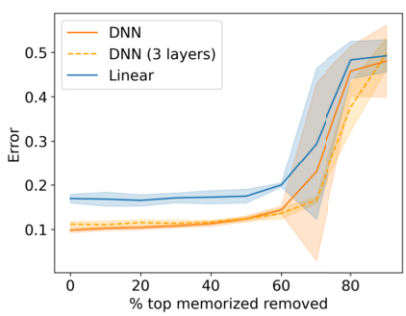

Figure 9. Evaluation of a linear classifier and deep neural networks (Distill-BERT) on a dataset of real movie reviews (SST-2). The shaded areas are 95% confidence bands around the average across 9 runs per each % top memorized examples removed.

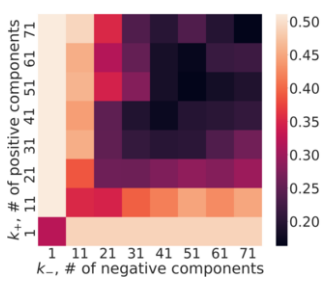

Figure 10. Heatmap of errors when fitting overparameterized MDA classifiers to data generated from our model $\mathcal{D}$, $d = 50$, $\boldsymbol{\mu} = \frac{1}{\sqrt{2}}(1, \ldots, 1)$, $n = 300$, $\sigma = 1$, $p = 0.9$.

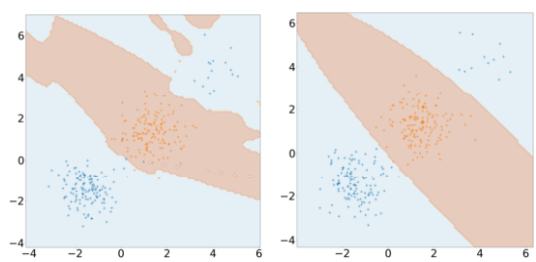

Figure 11. Decision boundaries of overparameterized and properly parametrized MDA classifiers. Left: $k_+ = k_- = 30$, Right: $k_+ = 1, k_- = 2$. The rest parameters are as follows: $\boldsymbol{\mu} = \frac{1}{\sqrt{2}}(1, 1)$, $n = 300$, $\sigma = 1$, $p = 0.9$.

text representations from the 6th layer of Distill-BERT as input, and the representations are not trained. It is clear that such a model is not capable of memorizing rare atypical examples. The second model—called DNN—is a full-fledged deep neural network that has enough capacity (66 million parameters) to memorize atypical examples from rare subpopulations. We also consider an intermediate option—called DNN (3 layers)—when the three lower layers are frozen, and the remaining layers are finetuned.

The experiment is as follows: (1) we compute the memorization score for each training example through DNN,[6] (2) remove $m\%$ of the top memorized examples from the training set, (3) train the Linear and DNN models on such a set with hyperparameters tuned on the validation set, (4) and finally evaluate both models on the test set. The results of this experiment for different m are shown in Figure 9. As we can see, the error of the Linear classifier is always greater than the DNN error. Further, when the tail is shortened (i.e., when a larger number of top memorized examples from training are discarded), the gap between the errors of Linear and DNN slightly decreases. This is consistent with the predictions of our theory, albeit built with simpler assumptions on the distribution of data and for more interpretable classifiers.

At certain point, the difference between the errors is not statistically significant. This happens because at a high percentage of removal, examples are removed not only from the tail, but also from the main subpopulations, which sharply worsens the performance of both the Linear and DNN classifiers. Note that this regime is not considered in our theory.

A careful reader may notice that the DNN error is not close to zero, even when no examples are removed from the training set. This is because there are examples in the test set that the DNN cannot classify correctly, even though it fits the training set perfectly. In principle, such difficult examples can be simulated in our data-generating model by introducting label flipping noise, and we defer such modification to our future work.

6 Discussion on Benign Overfitting

As was already mentioned in the Introduction, Feldman's long-tail theory explains the *need* for overfitting, but does not explain how exactly modern overparameterized learning algorithms manage to overfit without harming generalization [20]. Despite the availability of the answer for some model classes [5, 15, 11, 4, 19], our main concern with the current trend in theoretical studies of benign overfitting is in the assumptions about the data generating process, namely that only one subpopulation is allowed in each class, and linear inseparability is achieved by introducing random label-flipping noise. We repeat once again that such a setup does not fit in with Feldman's long-tail theory, in which the presence of rare subpopulations in classes is a prerequisite. Without this condition, there is no need for overfitting. Therefore, we would like to draw the attention of the learning theory community to the existing gap between theoretical setups and reality regarding the distribution of data.

In the theoretical part of our work, we do not deal with overparameterized models like deep neural networks. Our "complex" classifier (MDA) is actually only as complex as the data requires. Therefore, we cannot claim to have shown benign overfitting under the conditions of our data-generating model. We have shown the underfitting of the linear classifier (LDA) and the *proper* fitting of the MDA classifier with the right number of components.

However, we are curious about what happens if we give the MDA classifier the ability to fit many more Gaussians than necessary. To do this, we conduct the following experiment. The data is generated from the same model $\mathcal{D}$ that we used earlier (we fixed $d = 50$, $\boldsymbol{\mu} = \frac{2}{\sqrt{d}}(1, \ldots, 1)$, $n = 300$, $\sigma = 1$, $p = 0.9$). For data points from the positive class, we fit k_+ Gaussians, and for points from the negative class, we fit k_- Gaussians. Since in this case, we have no simple way to estimate the parameters by the method of moments, all parameters are estimated by the approximate maximum likelihood method through the EM algorithm. Let $f_+(\mathbf{x})$ and $f_-(\mathbf{x})$ be the resulting estimated p.d.f.'s for the positive and negative classes, respectively. Then the MDA classifier can be written as

$$h^{\text{MDA}}(\mathbf{x}; k_+, k_-) = \begin{cases} +1, & \text{if } f_+(\mathbf{x}) \geq f_-(\mathbf{x}) \\ -1 & \text{otherwise} \end{cases}$$

We vary k_+ and k_- in the interval $[1, 71]$ with a step 10 and calculate the classifier error on the test sample. The results of this experiment are shown in Figure 10, which is a heatmap of test errors for different pairs (k_+, k_-). The training error is zero for all pairs (k_+, k_-), except for $k_+ = k_- = 1$. Notably, there is a clear pattern: when k_+ and k_- are close to each other, the performance can be better than when there is a heavy imbalance between k_+ and k_-.

To understand how an overparametrized MDA classifier manages to overfit benignly, we plot a decision curve for the case $d = 2$, $k_+ = k_- = 30$ (Figure 11, left). As we can see, despite the potential to overfit malignantly with a complex decision curve, the EM algorithm chooses a fairly simple classifier that is not so different from the optimal one (Figure 11, right), that uses $k_+ = 1$, $k_- = 2$.

It is noteworthy that an overparameterized MDA classifier is able to overfit beningly on data generated from our model, because such a framework is more interpretable and amenable to analysis than overparameterized deep neural networks trained on real data. Accordingly, it becomes possible to study the phenomenon of benign overfitting in a simplified setting without linking it to deep learning, in which it is usually considered.

7 Conclusion

In this work we have focused on building an *interpretable* mathematical framework for the analysis of learning algorithms capable of memorizing rare/atypical examples that usually occur in natural data, such as texts and images. The key point in our work is the data-generating model based on Gaussian mixtures, which demonstrates the inability of a simple classifier without sufficient memory to correctly label rare and atypical test examples. At the same time, for a more complex (but not too complex) classifier with sufficient memory, the near-to-optimal generalization ability is shown. Moreover, the dynamics of the performance of these classifiers with tail shortening has been studied both theoretically and experimentally, and the experiments were carried out both on synthetic and real data.

The last but not least property of our framework is that it allows for benign overfitting, and this is what we plan to study in the near future. In this regard, it will be interesting to analyze the behavior of overparameterized learning algorithms (such as MDA with a redundant number of components, deep neural networks, and nearest-neighbor classifiers) on data generated from our model. This will require obtaining new results in terms of sufficient conditions for benign overfitting to happen under the assumptions of our model.

[6] This means that in the definition of the memorization score (17), we use a DNN trained by gradient-based method as a learning algorithm A. However, we *approximate* Eq. 17 via the method of Zheng and Jiang [21] to avoid repeated retraining for each example.

Acknowledgements

This research has been funded by Nazarbayev University under Faculty-development competitive research grants program for 2023-2025 Grant #20122022FD4131, PI R. Takhanov. Igor Melnykov's work on this project was supported by a Fulbright US Scholar Grant administered by the US Department of State Bureau of Educational and Cultural Affairs (grant ID: PS00334837). The authors would like to thank Christopher Dance for a thorough review of our work (including mathematical proofs), Matthias Gallé for his constructive feedback, including the suggestion to add a discussion on benign overfitting. We would like to thank the reviewers for their valuable feedback, in particular Reviewer 1 for a deep reading of our work, Reviewers 2 and 3 for carefully reading our response, Reviewer 6 for good questions that helped improve the presentation of the material.

References

[1] Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin, 'Deep learning: a statistical viewpoint', *Acta Numerica*, **30**, 87–201, (2021).

[2] Mikhail Belkin, 'Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation', *Acta Numerica*, **30**, 203–248, (2021).

[3] Arman Bolatov, Maxat Tezekbayev, Igor Melnykov, Artur Pak, Vassilina Nikoulina, and Zhenisbek Assylbekov, 'Long-tail theory under gaussian mixtures', *arXiv preprint arXiv:2307.10736*, (2023).

[4] Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu, 'Benign overfitting in two-layer convolutional neural networks', in *NeurIPS*, (2022).

[5] Niladri S. Chatterji and Philip M. Long, 'Finite-sample analysis of interpolating linear classifiers in the overparameterized regime', *J. Mach. Learn. Res.*, **22**, 129:1–129:30, (2021).

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms, 3rd Edition*, MIT Press, 2009.

[7] Li Deng, 'The MNIST database of handwritten digit images for machine learning research [best of the web]', *IEEE Signal Process. Mag.*, **29**(6), 141–142, (2012).

[8] Vitaly Feldman, 'Does learning require memorization? a short tale about a long tail', in *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, eds., Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, pp. 954–959. ACM, (2020).

[9] Vitaly Feldman and Chiyuan Zhang, 'What neural networks memorize and why: Discovering the long tail via influence estimation', in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, eds., Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, (2020).

[10] R. A. Fisher, 'The use of multiple measurements in taxonomic problems', *Annals of Eugenics*, **7**(2), 179–188, (1936).

[11] Spencer Frei, Niladri S. Chatterji, and Peter L. Bartlett, 'Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data', in *Conference on Learning Theory, 2-5 July 2022, London, UK*, eds., Po-Ling Loh and Maxim Raginsky, volume 178 of *Proceedings of Machine Learning Research*, pp. 2668–2703. PMLR, (2022).

[12] Trevor Hastie and Robert Tibshirani, 'Discriminant analysis by gaussian mixtures', *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**(1), 155–176, (1996).

[13] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*, Springer Series in Statistics, Springer, 2009.

[14] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf, 'Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter', *CoRR*, **abs/1910.01108**, (2019).

[15] Ohad Shamir, 'The implicit bias of benign overfitting', in *Conference on Learning Theory, 2-5 July 2022, London, UK*, eds., Po-Ling Loh and Maxim Raginsky, volume 178 of *Proceedings of Machine Learning Research*, pp. 448–478. PMLR, (2022).

[16] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts, 'Recursive deep models for semantic compositionality over a sentiment treebank', in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, (October 2013). Association for Computational Linguistics.

[17] Leslie G Valiant, 'A theory of the learnable', *Communications of the ACM*, **27**(11), 1134–1142, (1984).

[18] Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition, 1974.

[19] Ke Wang and Christos Thrampoulidis, 'Benign overfitting in binary classification of gaussian mixtures', in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pp. 4030–4034. IEEE, (2021).

[20] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, 'Understanding deep learning requires rethinking generalization', in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, (2017).

[21] Xiaosen Zheng and Jing Jiang, 'An empirical study of memorization in NLP', in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, eds., Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, pp. 6265–6278. Association for Computational Linguistics, (2022).