Thesis Title

Low Budget Active Learning: Theory and Algorithms

Avihu Dekel

Thesis submitted in partial fulfillment of the requirements for the Master of Sciences degree in Computer Science

Under the supervision of Prof. Daphna Weinshall

October 2022
Abstract

Deep active learning aims to reduce the annotation cost for deep neural networks, which are notoriously data-hungry. In this work, we focus on the relation between the number of labeled examples (budget size), and suitable querying strategies. Our theoretical analysis shows a behavior reminiscent of phase transition: typical examples are best queried when the budget is low, while unrepresentative examples are best queried when the budget is large. Combined evidence shows that a similar phenomenon occurs in common classification models.

Accordingly, we propose TypiClust and ProbCover – two deep active learning strategies suited for low budgets. In a comparative empirical investigation of supervised learning, using a variety of architectures and image datasets, TypiClust and ProbCover outperform all other active learning strategies in the low-budget regime. In the semi-supervised framework, performance gets an even more significant boost. In particular, state-of-the-art semi-supervised methods trained on CIFAR-10 with 10 labeled examples selected by TypiClust, reach 93.2% accuracy – an improvement of 39.4% over random selection.

This thesis is based upon the findings of two published papers:

- Active Learning on a Budget: Opposite Strategies Suit High and Low Budgets [ICML 2022] (Avihu Dekel, Guy Hacohen, Daphna Weinshall)

Code is available at https://github.com/avihu111/TypiClust.
Acknowledgements

I would first like to thank my terrific advisor, Prof. Daphna Weinshall, who taught me how research is properly done and guided me in both fruitful and unfruitful times. Daphna knew when to encourage me to pursue my interests, and when it was time to focus on our main direction. I greatly enjoyed our talks, brainstorming and consulting sessions, which constantly kept me motivated.

I would also like to thank Guy Hacohen (my second wonderful mentor), Ofer Yehuda and all the lab members and all my friends from the Hebrew University. Finally, I with to thank My family and my life partner Inbal, who supported me through stressful deadlines and was always there for me.
Contents

1 Introduction 1

2 Previous Work 5
   2.1 Uncertainty Sampling ................................. 5
   2.2 Diversity Sampling ................................. 6
   2.3 Semi Supervised Active Learning ...................... 7
   2.4 Low Budget Active Learning .......................... 7

3 Theoretical Analysis 8
   3.1 Mixture Model Definition ............................. 8
   3.2 Deriving the Optimal Sampling Strategy ................. 10
   3.3 Error Scores Analyzed ................................ 11
   3.4 Undulating Error Scores .............................. 11
   3.5 SP-Undulating Error Scores ......................... 12
   3.6 Simple Classification Models ......................... 13
      3.6.1 Mixture of Two Linear Classifiers ............... 13
      3.6.2 KNN Classifier and High-Density Regions ......... 14

4 Methods 15
   4.1 Framework and Definitions ............................ 15
   4.2 Typical Clustering (TypiClust) ....................... 15
   4.3 Max Probability Coverage (ProbCover) ............... 17
      4.3.1 Greedy Algorithm ................................ 18

5 Empirical Results 19
   5.1 Methodology .......................................... 19
   5.2 Main Results .......................................... 20
   5.3 Ablation Study ........................................ 21
5.3.1 The Importance of Density and Diversity ........................................... 22
5.3.2 Uncertainty Delivered by an Oracle .................................................. 22
5.3.3 Imbalanced Data ............................................................................... 23
5.3.4 Random initial selection ..................................................................... 23
5.3.5 The interaction between δ and budget size ......................................... 24
5.3.6 Coreset vs. ProbCover ....................................................................... 24

6 Discussion ................................................................................................. 26

A Proofs ........................................................................................................ 33
   A.1 Mixture Model Lemmas and Proofs ...................................................... 33
      A.1.1 Undulating Error Score: Sufficient Conditions ............................... 33
      A.1.2 SP-undulating Error Score: Sufficient Conditions ......................... 35
   A.2 Error Function of Simple Mixture Models ............................................ 37
      A.2.1 Mixture of Two Linear Classifiers .................................................. 37
      A.2.2 1-NN Classifier ............................................................................ 42

B Visualizations ............................................................................................. 44
   B.1 Theoretical Analysis: Visualization ....................................................... 44
      B.1.1 Mixture of Two Linear Models ....................................................... 44
      B.1.2 Error Scores of Deep Neural Networks ........................................... 44
   B.2 Visualization of Query Selection .......................................................... 45

C Additional Empirical Results ..................................................................... 47
# List of Figures

1.1 Phase transition phenomenon ........................................... 2  
1.2 TypiClust selection example ........................................... 3  
1.3 *ProbCover* and *Coreset* duality .................................... 4  

4.1 Importance of diversity and typicality ................................ 16  

5.1 Fully supervised results .................................................. 20  
5.2 Self supervised embedding results .................................... 21  
5.3 Semi supervised results .................................................. 21  
5.4 TypiClust ablations ....................................................... 22  
5.5 Random initial pool ablation ............................................. 24  
5.6 RGB-space ablation ....................................................... 24  
5.7 Using different $\delta$ values ........................................... 24  
5.8 *ProbCover* vs *Coreset* ablation ..................................... 24  

6.1 TypiClust Results Summary ............................................. 26  

B.1 Mixture of Linear Models - Error Score ................................. 44  
B.2 Error Scores in Neural Networks ....................................... 45  
B.3 30 CIFAR-10 examples selected by *TypiClust* ...................... 45  
B.4 100 ImageNet-100 examples selected by *TypiClust* .............. 46  

C.1 ImageNet-50 Results ..................................................... 47
1 Introduction

Recent years have witnessed the emergence of deep learning as the dominant force in advancing machine learning and its applications. But this development is data-hungry – to a large extent, deep learning owes its success to the growing availability of annotated data. This is problematic even in our era of Big Data, as the annotation of data remains costly.

Active Learning (AL) aims to alleviate this problem [see surveys in 37, 39]. Given a large pool of unlabeled data, and possibly a small set of labeled examples, learning is done iteratively: the learner employs the labeled examples (if any), then queries an oracle by submitting examples to be annotated. This may be done repeatedly, often until a fixed budget is exhausted.

Many traditional active learning approaches are based on uncertainty sampling [e.g., 16, 28, 35, 42]. In uncertainty sampling, the learner queries examples about which it is least certain, presumably because such labels contain the most information about the problem. Another principle guiding deep AL approaches is diversity sampling [e.g., 11, 21, 38, 40], where queries are chosen to minimize the correlation between samples, to avoid redundancy in the annotations. Sensibly, diversity sampling is often combined with uncertainty sampling.

At present, effective deep active learning strategies are known to require a large initial set of labeled examples to work properly [34, 53]. We call this the high budget regime. In the low budget regime, where the initial labeled set is small or absent, it has been shown that random selection outperforms most deep AL strategies [see 2, 8, 31, 41, 57]. This “cold start” phenomenon is often explained by the poor ability of neural models to capture uncertainty, which is more severe with a small budget of labels [13, 33].

In this thesis, we suggest that the low and high budgets regimes are qualitatively different, and require opposite querying strategies. Furthermore, we claim that the uncertainty principle is only suited for the high-budget regime, while the opposite strategy – the selection of the least ambivalent points – is suitable for the low budget regime.

We begin, in Section 3, by establishing the theoretical foundations for this claim. We analyze a mixture model where two general learners, each limited to a distinct region of the input space, are independently learned. In this framework, we see a phase-transition-like phenomenon: in the low-budget regime, over-sampling the “easier” region, which can be learned from fewer examples, improves the outcome of learning. In the high-budget regime, over-sampling the alternative region is more beneficial. In other words, opposing querying strategies are suitable for the low-budget and high-
CHAPTER 1. INTRODUCTION

(a) Theoretical results
(b) Empirical results

Figure 1.1: Visualization of phase transition in deep active learning, as revealed by plotting the difference in accuracy between different AL strategies to a random baseline as a function of budget size (the number of labeled examples). We see similar behavior both theoretically and empirically: when the budget is low, oversampling typical examples is more beneficial, whereas when the budget is high, oversampling atypical examples is more beneficial. (a) The behavior of an idealized model (see Section 3). (b) The behavior of TypiClust, contrasted with 2 uncertainty-based strategies, as seen in deep neural models trained on CIFAR-10 (see Section 5).

We then identify a set of sufficient conditions, which guarantee that two independent learners display this phase-transition-like phenomenon. We then give a formal argument, showing that linear classifiers satisfy these conditions. We further provide empirical evidence to the effect that neural models may also satisfy these conditions.

Previous art established that in the high-budget regime, it is beneficial to preferentially sample uncertain examples. The phase transition result predicts that in the low-budget regime, a strategy that preferentially samples the most certain examples is beneficial. However, estimating prediction certainty is difficult, and cannot be reliably accomplished in the low-budget regime with access to very few labeled examples. We therefore adopt an alternative approach, replacing the notion of certainty with the notion of typicality (loosely defined): a point is typical if it lies in a high-density region of the input space, irrespective of labels.

In Section 4, guided by these observations, we propose two strategies for the low-budget regime:

- Typical Clustering (TypiClust)
- Probability Cover (ProbCover)

TypiClust aims to pick a diverse set of typical examples, which are likely to be representative of the entire dataset. To this end, TypiClust employs self-supervised representation learning and then estimates each point’s density in this representation. Diversity
is obtained by clustering the dataset and sampling the densest point from each cluster (see Fig. 1.2).

Figure 1.2: Visualizing the selection of 30 examples from CIFAR-10 by TypiClust. Left: the data is first clustered into 30 clusters, and the densest region within every cluster is sampled. We show t-SNE dimensionality reduction of the feature space, colored by cluster assignment, where selected examples are marked by ×. Right: the selected images, organized column-wise by class. Note that the ensuing labeled set is approximately class-balanced, even though the queries are chosen without access to class labels.

ProbCover also aims to select typical and diverse samples, but achieves that using a different formulation. ProbCover formulates active learning as a coverage problem – given some radius $\delta > 0$, we say that an example is covered if it is $\delta$-close to some labeled example. ProbCover attempts to maximize the number of covered samples. Unlike TypiClust, diversity is achieved implicitly (without clustering) – selecting similar samples would be redundant, as many examples are already covered. We note that ProbCover and Coreset are related – ProbCover enforces some $\delta$ and maximizes the coverage, while Coreset enforces complete coverage and minimizes $\delta$. This is visualized in Fig. 1.3.

In Section 5, we compare TypiClust and ProbCover to various AL strategies in the low-budget regime. Both methods consistently improve generalization by a large margin, across different datasets and architectures, reaching state-of-the-art (SOTA) results in many problems. In agreement with Zhu et al. [57], we also observe that the alternative AL strategies are not effective in this domain, and are even detrimental.

TypiClust and ProbCover are especially beneficial for semi-supervised learning. Although in the AL framework the learner has access to a big pool of unlabeled data by construction, most AL strategies do not exploit the unlabeled data for learning, beyond query selection. Recent studies report that the benefit of AL is marginal when incorporated into semi-supervised learning [4, 7], with little added value over the random selection of labels. Re-examining this observation, we note that semi-supervised learning is most beneficial in the low-budget regime, wherein the explored AL strategies
Figure 1.3: ProbCover selection (top) vs Coreset selection (bottom) of 5/20/50 samples (out of 600). Selected points are marked by x, which is color-coded by density (see color code bar to the right). Density is measured using Gaussian Kernel Density Estimation, and the covered area is marked in light blue. Coreset attempts to minimize ball size, constrained by complete coverage, while ProbCover attempts to maximize coverage, constrained by a fixed ball size. Note that especially in low budgets, such as 5 samples, Coreset only selects outliers of the distribution (yellow), while ProbCover selects from dense regions of the distribution (red).

are inherently not suitable. When incorporating these methods, which are designed for the low-budget regime, into semi-supervised learning, performance is indeed improved by a large margin.
2 Previous Work

The key idea behind active learning is that a learning model can possibly achieve greater accuracy with fewer labeled training examples if it is allowed to choose the data from which it learns (see surveys Ren et al. [36], Settles [39]). An active learner may ask queries in the form of unlabeled examples to be labeled by an oracle (e.g., a human annotator). In many machine learning problems, unlabeled data is abundant, but labels are time-consuming and expensive to obtain. Active learning is an important tool in extending machine learning algorithms into such problems.

At present, most active learning methods rely on uncertainty and diversity sampling. In the high-budget regime (when the initial pool of examples is relatively large), these methods indeed improve performance over random selection. However, in the low-budget regime (when the initial pool of examples is very small), these methods fail to improve over random selection, a phenomenon termed cold-start. As such, most works focused on the high-budget regime. Sections 2.1 and 2.2 focus on different approaches commonly used in the high-budget regime, while sections 2.3 and 2.4 focus on works in the low-budget regime.

2.1 Uncertainty Sampling

One classic querying strategy is uncertainty sampling [28], where the learner queries the examples it is least certain about. Tong and Koller [44] used uncertainty sampling for support vector machines, by querying the closest example to the decision boundary. Wang et al. [47] used uncertainty sampling in deep neural models, by measuring uncertainty using some function over the softmax probability vector. Examples with high confidence were pseudo-labeled, while examples with low confidence were sent for human annotation. Confidence was measured by:

1. The maximal probability in the softmax probability vector.
2. The margin between the max and second-max probabilities.
3. The entropy of the probability vector, where low entropy indicates high confidence.

The above methods rely on calibrated model uncertainty, yet neural models are notoriously known to produce overconfident predictions on out-of-distribution examples. In comparison, Bayesian models offer a mathematically grounded framework to capture model uncertainty, but usually come with a prohibitive computational cost. Gal and Ghahramani [13] showed an equivalence between dropout and approximate Bayesian
inference, enabling the application of Bayesian methods to deep learning. Gal et al. [14] the aforementioned Bayesian neural models to query examples, replacing the ill-calibrated softmax probability vector with the more-calibrated Bayesian posterior distribution. Beluch et al. [3] found that ensemble-based AL result in more calibrated model uncertainty, and thus better performance than Bayesian models.

Yoo and Kweon [52] attached a loss prediction module to the neural model and trained it to predict the loss examples, without seeing the label. Consequently, this module can suggest data that the model is likely to produce a high loss.

2.2 Diversity Sampling

Another guiding principle in querying strategies is diversity sampling. The motivation is to avoid redundancy in the annotations, and to represent all parts of the training distribution. Diversity sampling is orthogonal to uncertainty sampling, and can be added to almost any strategy (e.g [11, 20, 21, 40, 48, 49, 51, 56]). As training deep models is a computationally intensive process, deep active learning is practical only in batch settings. In batch settings, the selection of correlated examples leads to degraded performance, amplifying the importance of diversity.

Sener and Savarese [38] introduced the coreset approach, querying examples that cover the training distribution. Specifically, Coreset solves the k-Center problem (minimax facility location [12]) which can be defined as follows: choose $B$ center points such that the largest distance between a data point and its nearest center is minimized.

Sinha et al. [42] learns a latent space using a variational autoencoder (VAE), where a discriminator is trained to discriminate between unlabeled and labeled data, while the VAE tries to trick the discriminator into predicting that all data points are from the labeled pool. The examples the discriminator is most certain about are queried, as they are presumably different from the labeled samples.

Ash et al. [1] queries diverse examples with high-magnitude, represented in a hallucinated gradient space. It incorporates both uncertainty and diversity into every selected batch, without requiring any hand-tuned hyperparameters. Kirsch et al. [22] extended the formulation in Gal et al. [14] to also consider dependencies within a batch, consequently enforcing diversity.
2.3 Semi Supervised Active Learning

Semi-supervised learning leverages unlabeled data to improve the model’s performance. Common semi-supervised methods usually employ consistency regularization and pseudo-labeling \([43, 54]\). Recently, there has been progress in combining semi-supervised methods with active learning.

Gao et al. \([15]\) proposed to unify model training and sample selection with a semi-supervised active learning framework. Their selection metric is based on sample consistency and implicitly balances sample uncertainty and diversity during selection. Siméoni et al. \([41]\) and \([7]\) utilizes the unlabeled data by unsupervised pre-training at the beginning of the active learning pipeline and semi-supervised learning at every active learning cycle. These combinations are orthogonal to selection strategies.

In particular, Siméoni et al. \([41]\) and \([7]\) report that uncertainty sampling algorithms fail to improve over random selection in low-budgets when combined with semi-supervised learning.

2.4 Low Budget Active Learning

Recently, active learning in the low-budget regime received increased attention. Strategies designed for this regime usually employ self-supervised learning on the unlabeled pool and use the ensuing embedding space for selection, as it gives an informative distance measure \([55]\).

Yuan et al. \([53]\) aims to solve the cold-start problem by using the embedding of a pre-trained model on some unsupervised task. Their experiments use pre-trained language models, with a strategy that decreases the dependency on high budgets but is still faithful to uncertainty sampling. Mahmood et al. \([30]\) suggested querying a diverse set of examples with minimal Wasserstein distance from the unlabeled pool. They report a significant performance boost in the low-budget regime.
3 Theoretical Analysis

Given a large pool of $U$ unlabeled examples and a possibly small (or even empty) set of $L$ labeled examples, an Active Learning (AL) method selects a small subset of $B$ examples from $U$ to submit as label queries to an oracle. We call the number of labeled examples known to the learner **budget**, whose size is $m = B + L$. In this section, we aim to study the optimal query selection strategy as it depends on $m$.

To this end, we analyze a mixture model of two general learners. In §3.1, the model is defined by first splitting the support of the data distribution into two distinct regions, $R_1$ and $R_2$, further assuming that each region is independently learned by its own general learner. $R_1$ and $R_2$ are distinguished by the property that if they are learned independently, $R_1$ is *easier to learn* than $R_2$ (as formalized in Def. 2 below). We then define the error score $E_D(m)$ of this model, which measures the expected error of the model over all training samples of size $m$, as a function of $m$.

Within this framework, in §3.2 we derive a **threshold test** on the budget size $m$ and $E_D(m)$, which determines whether an optimal AL strategy should oversample $R_1$ or $R_2$. In §3.3 we obtain sufficient conditions on $E_D(m)$, which guarantee phase transition as illustrated in Fig. 1.1a. In accordance, an optimal AL strategy will oversample $R_1$ if $m$ is smaller than some threshold $m_0$, and oversample $R_2$ otherwise. We let the term **low budget regime** denote budgets with $m \leq m_0$, and **high budget regime** denote budgets with $m > m_0$. We may now conclude that for any learner model whose error score meets our sufficient conditions, opposite strategies suit the low and high budget regimes.

### 3.1 Mixture Model Definition

We analyze a mixture of two learners model, where each learner is independently trained on a distinct part of the domain of the data distribution. Formally, let $X = [x, y]$ denote a single data point, where $x \in \mathbb{R}^d$ denotes an input point and $y \in \mathcal{Y}$ its corresponding label or target value. For example, $\mathcal{Y}$ is $\mathbb{R}$ in regression problems, and $[k]$ in classification problems. Each point $X$ is drawn from a distribution $D$ with density $f_D(X)$. We denote an i.i.d. sample of $m$ points from $D$ as $X^m = \{X_1, ..., X_m\} \sim D^m$.

Let $R_1, R_2 \subseteq \mathbb{R}^d \times \mathcal{Y}$ denote a partition of the domain of $f_D$, where $R_1 \cup R_2 = \mathbb{R}^d \times \mathcal{Y}$ and $R_1 \cap R_2 = \emptyset$. Let $D_1, D_2$ denote the conditional distributions obtained when restricting $D$ to regions $R_1, R_2$ respectively. Note that $D$ can now be viewed as a mixture distribution, where points are sampled from $D_1$ with probability $p = \int_{X \in R_1} f_D(X) dX$, and $D_2$ with probability $(1 - p)$. Let $m_i, i \in [2]$ denote the number of points in $R_i$.
CHAPTER 3. THEORETICAL ANALYSIS

when sampling \( m \) points from \( \mathcal{D} \), and \( X^m_{R_i} \) denote the restriction of sample \( X^m \) to \( R_i \). We denote the hypothesis of a learner when trained independently on \( X^m_{R_i} \) as \( h(X^m_{R_i}) \).

Next, we define the error score of a learner, which is a function of \( m \) – the training set size. It measures the expected generalization error of the learner over all such training sets.

**Definition 1** (Error score). Assume training sample \( X^m \sim \mathcal{D}^m \), with the corresponding learned hypothesis \( h(X^m) \) – a random variable whose distribution is denoted \( \mathcal{D}_h \). Let \( Er(h(X^m)) \) denote the expected generalization error of this hypothesis. The expected error of the learner, over all training sets of size \( m \), is given by

\[
E_D(m) = \mathbb{E}_{X^m \sim \mathcal{D}^m} \left[ \mathbb{E}_{h(X^m) \sim \mathcal{D}_h} [Er(h(X^m))] \right].
\]

We adopt two common assumptions regarding \( E_D(m) \). (i) **Efficiency**: \( E_D(m) \) is strictly monotonically decreasing, namely, on average the learner benefits from additional examples. (ii) **Realizability**: \( \lim_{m \to \infty} E_D(m) = 0 \).

During training, we assume a mixture of independent learners in \( R_1, R_2 \), and a training set composed of \( m_1, m_2 \) examples from each region respectively. The error score of the mixture learner on \( \mathcal{D} \) for \( m = m_2 + m_1 \) is:

\[
E_D(m) = p \cdot E_{D_1}(m_1) + (1 - p) \cdot E_{D_2}(m_2).
\] (3.1)

As an important ingredient of the mixture model, we assume that one region requires fewer examples to be adequately learned, and call this region \( R_1 \). Essentially, we expect the error score to decrease faster at \( R_1 \), with \( E_{D_1}(m) < E_{D_2}(m) \) \( \forall m \). Other than this difference, we expect the error score to be similar in \( R_1 \) and \( R_2 \).

Making this more precise, we define an order relation on partition \( R_1, R_2 \) as follows:

**Definition 2** (Order \( R_1 \prec R_2 \)). Let \( R_1, R_2 \) denote a partition of the domain of \( f_D \). Assume that the error score in \( R_1 \) and \( R_2 \) can be written as \( E_{D_1}(m) = E(m) \) and \( E_{D_2}(m) = E(\alpha m) \) for a single function \( E(m) \) and \( \alpha > 0 \). We say that \( R_1 \) is easier to learn than \( R_2 \) and denote \( R_1 \prec R_2 \) if \( \alpha < \frac{p}{1-p} \), where \( p \) is the probability of \( R_1 \).

Note that \( \alpha < 1 \) if \( p = 0.5 \). We now assume that \( R_1 \prec R_2 \), and rewrite (3.1) as follows:

\[
E_D(m) = p \cdot E(m_1) + (1 - p) \cdot E(\alpha m_2).
\] (3.2)
Finally, we extend \( E(m) \) to domain \( \mathbb{R}_{\geq 0} \) with the continuation of \( E(m) \) denoted \( E(x) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \), which is in \( C^\infty \) and positive. Efficiency is extended to imply \( E'(x) < 0 \).

### 3.2 Deriving the Optimal Sampling Strategy

Considering the extended error score \( E(x) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \), we define a biased sampling strategy as follows:

\[
m_1 = p \cdot m + \Delta, \quad m_2 = (1 - p) \cdot m - \Delta.
\]

\( \Delta = 0 \) is essentially equivalent to random sampling from \( D \). \( \Delta > 0 \) implies that more training points are sampled from \( R_1 \) than \( R_2 \), and vice versa.

In Thm. 1 we show that to minimize the expected generalization error while considering a mixture model of two independent learners as defined above, choosing between the different sampling strategies can be done using a simple threshold test.

**Theorem 1.** Given partition \( R_1 < R_2 \) and error score \( E(x) \), let \( p = \text{Prob}(R_1) \) and \( 0 < \alpha < \frac{p}{1-p} \). The following threshold test decreases the error score for sample size \( m \):

\[
\begin{align*}
\frac{E'(pm)}{E'(\alpha(1-p)m)} &> \frac{\alpha(1-p)}{p} \quad \implies \text{over sample region } R_1 \\
\frac{E'(pm)}{E'(\alpha(1-p)m)} &< \frac{\alpha(1-p)}{p} \quad \implies \text{over sample region } R_2
\end{align*}
\]

**Proof.** Starting from (3.2), we obtain the test whereby the error score \( E_D(m) \) decreases when \( \Delta > 0 \)

\[
pE(pm + \Delta) + (1 - p) E(\alpha((1 - p)m - \Delta)) < pE(pm) + (1 - p) E(\alpha(1 - p)m)
\]

\[
\implies (1 - p) \left[ E(\alpha(1 - p)m) - E(\alpha((1 - p)m - \Delta)) \right] > p \left[ E(pm + \Delta) - E(pm) \right]
\]

\[
\implies \alpha(1 - p) \frac{E(\alpha(1 - p)m) - E(\alpha(1 - p)m - \alpha\Delta)}{\alpha\Delta} > p \frac{E(pm + \Delta) - E(pm)}{\Delta}.
\]

Since \( E(x) \) is differentiable and strictly monotonically decreasing with \( E' < 0 \), in the limit of infinitesimal \( \Delta \)

\[
\frac{E'(pm)}{E'(\alpha(1-p)m)} > \frac{\alpha(1-p)}{p}.
\]
The proof for $\Delta < 0$ is similar.

**Example: exponentially decreasing function.** Assume $E(m) = e^{-m}$, and a mixture model with $p = 0.8, \alpha = 0.1$. We simulate the error in (3.2) when biasing the train sample with $\Delta = \pm 0.01$. Fig. 1.1a shows the differences between the error score of biased sampling (in favor of either $R_1$ in blue or $R_2$ in orange) and random sampling, as a function of the number of examples $m$. For small $m$ it is beneficial to favorably bias region $R_1$, while for large $m$ it is beneficial to favorably bias $R_2$. This is the behavior often seen in our empirical investigation, see Fig. 1.1b and Section 5.

### 3.3 Error Scores Analyzed

We now report sufficient conditions on $E(m)$, which guarantee the phase-transition-like behavior illustrated in Fig. 1.1a, starting with some formal definitions.

Given partition $R_1 \prec R_2$ and Error score $E(x)$, we say that $E(x)$ is undulating if it displays the following behavior: in the beginning, when the number of training examples is small, the generalization error decreases faster when over-sampling region $R_1$. In the end, after seeing sufficiently many training examples, the generalization error decreases faster when over-sampling region $R_2$. Formally:

**Definition 3** (Undulating). An error score $E(m)$ is undulating if there exist $z_1, z_2 \in \mathbb{R}$ such that $\frac{E'(pm)}{E'(\alpha(1-p)m)} > \frac{\alpha(1-p)}{p} \quad \forall m < z_1$, and $\frac{E'(pm)}{E'(\alpha(1-p)m)} < \frac{\alpha(1-p)}{p} \quad \forall m > z_2$.

For undulating error scores, there could potentially be any number of transitions between the two conditions, switching the preference of $R_1$ to $R_2$, and vice versa. We extend the above definition to capture a case of particular interest, where this transition occurs only once, as follows:

**Definition 4** (SP-undulating). An error score $E(m)$ is Single-Phase undulating if it is undulating, and $z_1 = z_2$.

### 3.4 Undulating Error Scores

As motivated in Section 3.1, we define a proper error score as follows:

**Definition 5** (Proper error score). $E(x) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ is a proper error score if it is a positive twice differentiable function, which is strictly monotonically decreasing ($E > 0, E' < 0$), $E(0) = c_0 \in \mathbb{R}_{>0}$, and where $\lim_{x \to \infty} E(x) = 0$. 
CHAPTER 3. THEORETICAL ANALYSIS

Not all proper error scores will exhibit the phase undulating behavior. In Thm. 2 we state sufficient conditions that ensure this behavior (see proof in App A.1.1).

**Theorem 2** (Undulating error score: sufficient conditions). Given partition $R_1 \prec R_2$ and error score $E(x)$, let $p = \text{Prob}(R_1)$ and $0 < \alpha < \frac{p}{1-p}$. $E(x)$ is undulating if the following assumptions hold:

1. $E(x)$ is a proper error score (see Def. 5).
2. $\lim_{x \to \infty} E'(x) \cdot E(x)$, $\lim_{x \to \infty} \frac{E(x)}{E(ax)}$, $\lim_{x \to \infty} \frac{E'(x)}{E'(ax)}$ exist $\forall a \in (0, 1)$.
3. $-\log(E(x)) \in \omega(\log(x))$.

**Corollary 1** (Exponential error as a bound). Error score $E(x)$ is undulating if it satisfies assumptions (i) and (ii) of Thm. 2, and is bounded from above as follows

$$E(x) \leq ke^{-vx}, \quad \forall x \in \mathbb{R}_{\geq 0}, \quad (3.3)$$

for some constants $v, k \in \mathbb{R}_{> 0}$.

**Proof.** It can be readily verified that assumption (iii) of Thm. 2 follows from (3.3). □

### 3.5 SP-Udulating Error Scores

Thm. 3 extends the results of the previous section, by stating a set of sufficient conditions that ensure an SP-undulating error score. The proof can be found in App A.1.2.

**Theorem 3** (SP-undulating: sufficient conditions). Given partition $R_1 \prec R_2$ and error score $E(x)$, let $p = \text{Prob}(R_1)$ and $0 < \alpha < \frac{p}{1-p}$. $E(x)$ is SP-undulating if the following assumptions hold:

1. $E(x)$ is an undulating error score.
2. At least one of the following conditions holds:
   1. $E'(x) - E''(x)$ is monotonically increasing with $x$.
   2. $E'(x)$ is strictly monotonically decreasing and log-concave.

**Corollary 2** (Exponential error is SP-undulating). Consider error scores of the form $E(x) = ke^{-vx}$ for constants $v, k \in \mathbb{R}_{> 0}$. Such functions are SP-undulating.

Cor. 2 shows that classifiers with an exponentially decreasing error score are SP-undulating, with a single transition from favoring $R_1$ to favoring $R_2$.

In practice, we cannot assume that the error score of commonly used learners is exponentially decreasing. However, frequently we can bound the error score from above.
CHAPTER 3. THEORETICAL ANALYSIS

by an exponentially decreasing function, as we demonstrate theoretically (see Section 3.6.1) and empirically (see Fig. B.2 in App B.1.2). In such cases, it follows from Cor. 1 that these functions are undulating.

3.6 Simple Classification Models

To make the analysis above more concrete, we analyze a mixture model of two linear classifiers in Section 3.6.1, as an example of an actual undulating model in common use. Additionally, we analyze the nearest neighbors classification model in Section 3.6.2, to shed light on the rationale behind the partition of the support to regions $R_1$ and $R_2$. This case analysis demonstrates specific circumstances that make it possible to learn from fewer examples in certain regions.

3.6.1 Mixture of Two Linear Classifiers

Consider a binary classification problem and assume a learner that delivers a mixture of two linear classifiers in $\mathbb{R}^d$. The two classifiers are obtained by independently minimizing the $L_2$ loss on points in $R_1$ and $R_2$ respectively.

(i) Bounding the error of each mixture component. We first derive a bound on the error separately for $R_1$ and $R_2$, as it depends on the sample size $m_j$ for $j \in [2]$. Let $X \in \mathbb{R}^{d \times m_j}$ denote the matrix whose columns are the training vectors that lie in region $R_j$. Let $y \in\{-1, 1\}^{m_j}$ denote a row labels vector, where 1 marks positive examples and −1 negative examples. The learner seeks a separating row vector $\hat{w} \in \mathbb{R}^d$, where

$$\hat{w} = \arg \min_{w \in \mathbb{R}^d} \|wX - y\|^2 \implies \hat{w} = yX^T (XX^T)^{-1}. $$

In Thm. 4, we bound the error of a linear model by some exponential function of the number of training examples $m_j$. The proof and further details can be found in App A.2.1.

Theorem 4 (Error bound on a linear classifier). Assume: (i) a bounded sample $\|x_i\| \leq \beta$, where $XX^T$ is sufficiently far from singular so that its smallest eigenvalue is bounded from below by $\frac{1}{\Lambda}$; (ii) a realizable binary problem where the classes are separable by margin $\delta$; (iii) full rank data covariance, where $\frac{1}{\lambda}$ denotes its smallest singular value. Then there exist some positive constants $k, \nu > 0$, such that $\forall m_j \in \mathbb{N}$ and every sample $X^{m_j}$, the expected error of $\hat{w}$
obtained using $X^{m_j}$ is bounded by:

$$E_{X \sim D_j}[0 - 1 \text{ loss of } \tilde{w}] \leq ke^{-\nu m_j}.$$ 

(ii) A mixture classifier. Assume a mixture of two linear classifiers, and let $E(m) = p \cdot E_{D_1}(m_1) + (1 - p) \cdot E_{D_2}(m_2)$ denote its error score. The following theorem characterizes this function (the proof can be found in App A.2.1):

**Theorem 5 (Undulating error).** Retain the assumptions of Thm. 4, and assume that $\forall \alpha \in (0, 1)$ the following limits exist

$$\lim_{m \to \infty} E'(m) = \lim_{m \to \infty} E(\alpha m), \lim_{m \to \infty} E'(\alpha m).$$

Then the error score of a mixture of two linear classifiers is undulating.

### 3.6.2 KNN Classifier and High-Density Regions

Our analysis in Section 3.3 shows that given some partition of the data into $R_1$ and $R_2$, where $R_1 \prec R_2$ (see Def. 2), then oversampling from $R_1$ is preferable in the low budget regime, while oversampling from $R_2$ is preferable in the high budget regime.

To shed light on the nature of the assumed partition, we analyze below the discrete one-Nearest-Neighbor (1-NN) classification framework. Specifically, we show that selecting $R_1$ as the set of the most probable points in the dataset has the property that $R_1 \prec \{\Omega \setminus R_1\}$.

We further show in App A.2.2 that in this framework, the selection of an initial pool of size $m$ will benefit from the following heuristic:

- **Max density**: when selecting a point $X_i$, maximize its density $f_D(X_i)$.
- **Diversity**: select points that are far apart, so that their corresponding sets of nearest neighbors do not overlap.

While 1-NN is a rather simplistic model, we propose to use the derived heuristics to guide deep active learning. In the rest of this thesis, we show how these guiding principles benefit deep active learning in the low-budget regime.
4 Methods

In the low-budget regime, our theoretical analysis shows that it may be beneficial to bias the training sample in favor of certain regions in the data domain. It also establishes a connection between such regions and the principles of max density (or typicality) and diversity. Here, we incorporate these principles into two novel active learning strategies called TypiClust and ProbCover, which are designed for the low-budget regime.

4.1 Framework and Definitions

Let $L_0$ denote an initial labeled set of examples, and $U_0$ denote an initial unlabeled pool. Active learning is done iteratively: at each iteration $i$, a set of $B$ unlabeled examples is picked according to some strategy. These examples are annotated by an oracle, added to $L_{i-1}$, and removed from $U_{i-1}$. This process is repeated until the labeling budget is exhausted, or some predefined termination conditions are satisfied. In the low-budget regime, the total number of labeled examples $|L_{i-1}| + B$ is assumed to be small. The case where $L_0 = \emptyset$ is called “initial pool selection”.

4.2 Typical Clustering (TypiClust)

To capture the principle of max density, we define the Typicality of an example by its density in some semantically meaningful feature space. Formally, we measure an example’s Typicality by the inverse of the average Euclidean distance to its $K$ nearest neighbors$^1$, namely:

$$Typicality(x) = \left( \frac{1}{K} \sum_{x_i \in K\text{-NN}(x)} ||x - x_i||^2 \right)^{-1}. \quad (4.1)$$

In the low-budget regime, an active learning strategy based on typical examples needs to overcome several obstacles: (a) Networks trained on only a few examples are prone to overfit, making measures of typicality noisy and unreliable. (b) Typical examples tend to be very similar, amplifying the need for diversity. The importance of typicality and diversity is visualized in Fig. 4.1.

To overcome these obstacles, we propose a novel method called TypiClust, which attempts to select typical examples while probing different regions of the data distribu-

\footnote{We use $K = 20$, but other choices yield similar results.}
CHAPTER 4. METHODS

(a) Typical and diverse  (b) A-typical and diverse  (c) Typical and not diverse

Figure 4.1: Qualitative visualization of diversity and typicality in the low budget regime on CIFAR-10. (a) Diverse typical images chosen by TypiClust. (b) Picking the least typical example in each cluster. (c) Picking the most typical examples, without enforcing diversity.

In our method, self-supervised representation learning is used to overcome (a), while clustering is used to overcome (b). TypiClust is therefore composed of three steps:

**Step 1: Representation learning.** Utilize the large unlabeled pool $U_0$ to learn a semantically meaningful feature space: first train a deep self-supervised task on $U_0 \cup L_0$, then use the penultimate layer of the resulting model as feature space. Such methods are commonly used for semantic feature extraction [9, 17].

**Step 2: Clustering for diversity.** As typicality in (4.1) is evaluated by measuring distances to neighboring points, the most typical examples are usually close to each other, often resembling the same image (see Fig. 4.1c). To enforce diversity and thus better represent the underlying data distribution, we employ clustering. Specifically, at each AL iteration $i$, we partition the data into $|L_{i-1}| + B$ clusters. This choice guarantees that there are at least $B$ clusters that do not intersect with the existing labeled examples. We refer to such clusters as uncovered clusters.

**Step 3: Querying typical examples.** We select the most typical examples from the $B$ largest uncovered clusters. Selecting from uncovered clusters enforces diversity (also w.r.t $L_{i-1}$), while selecting the most typical example in each cluster favors the selection of representative examples.

As the steps above do not depend on any specific representation or clustering method, different variants of the TypiClust strategy can be constructed. Below, we evaluate two variants, both of which outperform by a large margin the uncertainty-based strategies in the low-budget regime:

1. **TPC$_{DC}$**: Using a deep clustering algorithm both for the self-supervised and clustering tasks. In our experiments, we used SCAN [46].

The pseudo-code of TypiClust for initial pool selection is given in Alg. 1. Note that, unlike traditional active learning strategies, TypiClust relies on self-supervised representation learning, and therefore can be used for initial pool selection.

**Algorithm 1 TypiClust initial pooling algorithm**

| Input: | Unlabeled pool $U$, Budget $B$ |
| Output: | $B$ typical and diverse examples to query |
| Embedding | $\leftarrow$ Representation_Learning($U$) |
| Clust | $\leftarrow$ Clustering_algorithm(Embedding, $B$) |
| Queries | $\leftarrow \emptyset$ |
| for all $i = 1, ..., B$ do |
| Add $\arg \max_{x \in \text{Clust}[i]} \{\text{Typicality}(x)\}$ to Queries |
| end for |
| return Queries |

### 4.3 Max Probability Coverage (ProbCover)

TypiClust relies on clustering to introduce diversity, and therefore depends on the quality of the clustering algorithm. In contrast, Max Probability Coverage (ProbCover) is a clustering-free algorithm, which introduces diversity implicitly. ProbCover attempts to maximize the number of covered samples, using fix sized balls. Formally, let $\delta > 0$ be the ball radius and let $X = \{x_i\}_{i=1}^n$. The coverage of a set $L \subseteq X$ is defined as

$$C_\delta(L) = \bigcup_{x \in L} B_\delta(x)$$  \hspace{1cm} (4.2)

where $B_\delta(x) = \{z : \|z - x\| \leq \delta\}$. ProbCover maximization objective is therefore

$$\arg \max_{L \subseteq X : |L| = b} |C_\delta(L) \cap X|$$  \hspace{1cm} (4.3)

Objective 4.3 can be formalized as the discrete optimization problem Max Coverage, where the set cover covered by $x$ is $B_\delta(x) \cap X$. As Max Coverage is NP-hard, its optimal solution cannot be found efficiently. Instead, as its objective is a submodular monotone function, we use the greedy approximate algorithm, which achieves $\left(1 - \frac{1}{e}\right)$-approximation [24]. Theoretical motivation and a method to select $\delta$ can be found in Yehuda et al. [50].
4.3.1 Greedy Algorithm

The algorithm (see Alg. 2 below for pseudo-code) goes as follows: First, construct a directed graph \( G = (V, E) \), with \( V = X \) the embedding of the data space, and \( (x, x') \in E \iff x' \in B_\delta(x) \iff d(x, x') \leq \delta \). In \( G \), each vertex represents a specific example, and there is an edge between two vertices \((x, x')\) if \( x' \) is covered by the \( \delta \)-ball centered at \( x \) (distances are measured in the embedding space). The algorithm then performs \( b \) iterations of the following two steps:

(i) Pick the vertex \( x_{\text{max}} \) with the highest out-degree for annotation;
(ii) Remove all incoming edges to \( x_{\text{max}} \) and its neighbors.

As \( \text{ProbCover} \) uses a sparse representation of the adjacency graph, it is able to scale to large datasets while requiring limited space resources.

Algorithm 2 \( \text{ProbCover} \)

\begin{verbatim}
Input: unlabeled pool \( U \), labeled pool \( L \), budget \( b \), ball-size \( \delta \),
Output: a set of points to query
X ← Embedding of representation learning algorithm on \( U \cup L \)
\( G = (V = X, E = \{ (x, x') : x' \in B_\delta(x) \}) \)
for all \( c \in L \) do
    Remove the incoming edges to covered vertices, \( \{ (x', x) \in E : (c, x) \in E \} \), from \( E \)
end for
Queries ← \( \emptyset \)
for all \( i=1, \ldots, b \) do
    Add \( c \in U \) with the highest out-degree in \( G \) to \( \text{Queries} \)
    Remove the incoming edges to covered vertices, \( \{ (x', x) \in E : (c, x) \in E \} \), from \( E \)
end for
return \( \text{Queries} \)
\end{verbatim}
5 Empirical Results

We report a set of empirical results, comparing TypiClust and ProbCover to other AL strategies in a variety of settings. We focus on the very low budget regime, with a budget size $b$ of a similar order of magnitude as the number of classes. Note that since the data is picked from an unlabeled pool, chances are that the initial labeled set is not going to be balanced across classes, and in the early stages of training some classes will almost always be missing.

5.1 Methodology

Three deep AL frameworks are evaluated:

(i) **Fully supervised**: train the deep learner only on the annotated data, as a fully supervised task.

(ii) **Semi-supervised by transfer learning**: create a representation of the data by training with a self-supervised task on the unlabeled data, then train a simple model on the ensuing representation in a supervised manner. This framework is intended to capture the basic benefits of semi-supervised learning, regardless of the added benefits provided by the sophistication of modern semi-supervised learning methods.

(iii) **Fully semi-supervised**: train a semi-supervised model on both the annotated and unlabeled data using FlexMatch [54].

In frameworks (i) and (ii) we adopt the evaluation framework created by Munjal et al. [32], in which we can compare multiple deep AL strategies in a principled way. In framework (iii), we adopt the code and hyper-parameters provided by FlexMatch.

When evaluating frameworks (i) and (ii), we compare TypiClust and ProbCover to 8 deep AL strategies as baselines. (1) **Random** query uniformly. (2)-(4) query examples with the lowest score, using the following basic scores: (2) **Uncertainty** – max softmax output, (3) **Margin** – margin between the two highest softmax outputs, (4) **Entropy** – inverse entropy of softmax outputs. (5) **BADGE** [1]. (6) **DBAL** [14]. (7) **BALD** [22]. (8) **Coreset** [38].

These AL methods are evaluated on the following classification datasets: CIFAR-10/100 [25], TinyImageNet [27] and ImageNet-50/100 (the latter group includes subsets of ImageNet [10] containing 50/100 classes respectively, following Van Gansbeke et al. [46]). When considering CIFAR-10/100, and TinyImageNet, we use as input the embedding of SimCLR [9]. When considering ImageNet-50/100 we use as input the embedding.
of DINO [6]. Results on ImageNet-50 are deferred to the appendix and can be seen in App. C. When evaluating frameworks (i) and (ii), we perform 5 active learning rounds, querying a fixed budget of $b$ examples in each round. In framework (iii), as FlexMatch is computationally demanding, we only evaluate methods on their initial pool selection capabilities.

## 5.2 Main Results

### (i) Fully supervised framework

We evaluate different AL methods based on the performance of a deep neural network trained directly on the raw queried data. On each round, we query $b$ samples (where $b$ is equal to the number of and train a ResNet-18 on the accumulated queried set. We repeat this for 5 active learning rounds. In Fig. 5.1, we plot the mean accuracy of 5 repetitions. Additional results can be seen in Hacohen et al. [18].

![Figure 5.1: Framework (i), fully supervised: comparing ProbCover with baseline AL strategies on image classification tasks in the low budget regime. The budget $b$ has an average of 1 sample per class (the initial sample may be imbalanced). The average test accuracy in each iteration is reported, using 5 repetitions. The shaded area reflects the standard error across repetitions.](image)

### (ii) Semi-supervised by transfer learning

In this framework, we make use of pretrained self-supervised features, and measure classification performance using the 1-NN classifier. Accordingly, each point is classified by the label of its nearest neighbor (within the selected labeled set $L$) in the self-supervised features space. In low budgets, this framework outperforms the fully-supervised framework (i), but is not as effective as fully semi-supervised learning (iii), demonstrating that these findings are not specific to any given semi-supervised method. Similarly to Fig. 5.1, in Fig. 5.2, we plot the mean accuracy of 5 repetitions for
the different tasks.

![Image](a) CIFAR-10  (b) CIFAR-100  (c) Tiny-ImageNet  (d) ImageNet-100

Figure 5.2: Same as caption of Fig. 5.1, for framework (ii) - semi-supervised by transfer learning.

(iii) Semi-supervised framework

We compare the performance of different AL strategies used prior to running FlexMatch, a state-of-the-art semi-supervised method. In Fig. 5.3 we show results with 3 repetitions of FlexMatch, using the labeled sets provided by different AL strategies and budget \( b \) equal to the number of classes. We see that our method outperforms random sampling and other AL baselines by a large margin. We note that in agreement with previous works [7, 18], AL strategies that are suited for high budgets do not improve the results of random sampling, while AL strategies that are suited for low budgets do.

![Image](a) CIFAR-10  (b) CIFAR-100  (c) Tiny-ImageNet

Figure 5.3: Framework (iii) Semi-supervised: comparison of AL strategies in a semi-supervised task. Each bar shows the mean test accuracy after 3 repetitions of FlexMatch trained using \( b \) labeled examples, where \( b \) is equal to the number of classes in each task. Error bars denote the standard error.

5.3 Ablation Study

We report a set of ablation studies, evaluating the added value of different steps of TypiClust and ProbCover.
5.3.1 The Importance of Density and Diversity

TypiClust clusters the dataset and selects the most typical examples from every cluster. To assess the added value of clustering and typicality selection, we consider the following alternative selection criteria: (a) Select a random example from each cluster (TPC\textsubscript{Rand}). (b) Select the most atypical example in every cluster (TPC\textsubscript{Inv}). (c) Select typical samples greedily, without clustering (TPC\textsubscript{NoClust}).

The results in Fig. 5.4a show that both clustering and high-density sampling are crucial for the success of TypiClust. The low performance of TPC\textsubscript{Rand} shows that representation learning and clustering alone cannot account for all the performance gain, while the low performance of TPC\textsubscript{NoClust} shows that typicality without diversity is not sufficient.

5.3.2 Uncertainty Delivered by an Oracle

When trained on only a few labeled examples, neural networks tend to overfit, which may result in the unreliable estimation of uncertainty. This offers an alternative explanation to our results – uncertain examples may be a good choice in the low-budget regime as well, if only we could compute uncertainty accurately.

To test this hypothesis we first train an “oracle” network (see Lowell et al. [29]) on the entire CIFAR-10 dataset and use its softmax margin to estimate uncertainty. This “oracle margin” is then used to choose the query examples. Subsequently, another network is trained similarly to the setup of Fig. 5.1, adding in each iteration the examples with either the highest or lowest softmax response margin according to the oracle.

Figure 5.4: (a) To isolate the added value of clustering for diversity and typical sample selection, we evaluate 3 additional selection heuristics on CIFAR-10 (see Section 5.3.1). (b) Certainty, as estimated by the margin of an oracle that knows the labels, is used for AL. We plot mean test accuracy of 100 models trained on CIFAR-10, $|L_0| = 10$, $B = 10$. (c) Result on an imbalanced subset of CIFAR-10. (See 5.3.3)

The results are shown in Fig. 5.4b. We see that even a reliable measure of uncertainty
leads to poor performance in the low-budget regime, even worse than the baseline uncertainty-based methods. This may be because these methods compute the uncertainty in an unreliable way, and thus behave more like the random selection strategy.

5.3.3 Imbalanced Data

Unsupervised representation learning algorithms often assume class-balanced datasets. As TypiClust is heavily based on representation learning, it could potentially fail in class-imbalance settings. We adopted the class-imbalanced subset of CIFAR-10 proposed by Munjal et al. [32], and compared TypiClust to the class-imbalanced AL strategies proposed by Kothawade et al. [23]. Similar to our main results, we found that TypiClust outperforms other methods in the low-budget regime, and under-performs in the high-budget regime. See low-budget results in Fig. 5.4c.

5.3.4 Random initial selection

When following the uncertainty sampling principle, as many AL methods do, a trained learner is needed. Any such method requires therefore a non-empty initial pool of labeled examples to train a rudimentary learner, from which uncertainty selection can be bootstrapped. In the set of methods evaluated here (see Section 5.1), only two - ProbCover and TypiClust - are not affected by this problem. This can be seen in Fig. 5.1, noting that only these two methods do better than random in the initial step. Is this the only reason they outperform other methods on low budgets?

To address this concern, we repeat the experiments reported in Fig. 5.1a-5.1b, using an initial random set of annotated examples across the board and by all methods. Results are reported in Fig. 5.5. When comparing Fig. 5.1a-5.1b and Fig. 5.5, we see that the advantage of ProbCover and TypiClust goes beyond the initial set selection, and can be seen even if this factor is eliminated.

RGB space distances As discussed in Section 4.3, our approach relies on the existence of a good embedding space, where distance is correlated with semantic similarity. We now verify this claim by repeating the basic fully-supervised experiments (Fig. 5.1) with one difference: ProbCover can only use the original RGB space representation to compute distances. Results are shown in Fig. 5.6. When comparing the original ProbCover with its variant using RGB space, a significant drop in performance is seen as expected, demonstrating the importance of the semantic embedding space.
CHAPTER 5. EMPIRICAL RESULTS

24

Figure 5.5: Random Initial pool in the supervised framework, an average of 1 sample per class.

5.3.5 The interaction between $\delta$ and budget size

To understand the interaction between the hyper-parameter $\delta$ and budget $b$, we repeat our basic experiments (Fig. 5.1) with different choices of $\delta$ and $b$ on CIFAR-10. For each pair $(\delta, b)$, we select an initial pool of $b$ examples using $\text{ProbCover}$ with $\delta$ balls, and report the difference in accuracy from the selection of $b$ random points. Average results across 3 repetitions are shown in Fig. 5.7 as a function of $b$. We see that as the budget $b$ increases, smaller $\delta$’s are preferred.

Figure 5.7: The accuracy difference between $\text{ProbCover}$ when using different $\delta$ values, and the outcome of $b$ random samples (average over 3 repetitions).

Figure 5.6: Comparison of $\text{ProbCover}$ when applied to the raw data vs the embedding space.

5.3.6 Coreset vs. $\text{ProbCover}$. 

In Fig. 1.3 we show that coreset selects samples with lower density than $\text{ProbCover}$. This might suggest that $\text{ProbCover}$ is suitable for low budgets, while Coreset is suitable for high budgets. To verify this claim, we compare their performance under the following 3 setups while using the same embedding space, and report results on CIFAR-10:

Figure 5.8: Comparing the performance under the supervised framework of $\text{ProbCover}$ and Coreset on different budget regimes. Low budget shows an initial pool selection of 100 samples. Mid/High budget start with 1K/5K samples and query additional 1K/5K samples (see text).
• Low budget - Select an initial pool of 100 samples using the SimCLR representation.
• High budget - Train a model on 5K randomly selected examples. Then select an additional set of 5K examples using the learner’s latent representation. This is the setup used in Sener and Savarese [38].
• Mid budget - Same as high budget, except the initial pool size and added budget are 1K.

Results are reported in Fig. 5.8. In the low budget, ProbCover outperforms Coreset as would be expected. In the mid budget, where the feature space of the learner is informative, only ProbCover achieves significant improvement over random selection. In the high budget, Coreset improves over random selection, while ProbCover does not.
6 Discussion

Figure 6.1: TypiClust achieves major accuracy gains as compared to the random selection baseline in the fully-supervised (3 repetitions on ImageNet, 10 otherwise), semi-supervised (3 reps) and self-supervised embedding (5 reps) frameworks. We use 10, 300, 1000, 50, 100, 200 examples in CIFAR-10, CIFAR-100, TinyImageNet, ImageNet-50, ImageNet-100 and ImageNet-200 respectively.

We show, theoretically and empirically, that strategies for active learning in the high and low-budget regimes should be based on opposite principles. Initially, in the low budget regime, the most typical examples, which the learner can learn most easily, are the most helpful to the learner. When reaching the high-budget regime, the best examples to query are those that the learner finds most confusing. This is the case both in the fully supervised and semi-supervised settings: we show that semi-supervised algorithms get a significant boost from seeing the labels of typical examples. Fig. 6.1 summarizes all our empirical results.

Our results are closely related to curriculum learning [5, 19], hard data mining, and self-paced learning [26], all of which reflect the added value of typical (‘easy’) examples when there is little information about the task, as against atypical (‘hard’) examples which are more beneficial later on. The point of transition – what makes the budget ‘small’ or ‘large’, depends on the task and corresponding data distribution. In complex real-life problems, the low budget regime may still contain a large number of examples, increasing the practicality of our method. Determining the range of training sizes with ‘low budget’ characteristics is a challenging problem, which we leave for future work.
Bibliography


A Proofs

A.1 Mixture Model Lemmas and Proofs

A.1.1 Undulating Error Score: Sufficient Conditions

Below we provide the proof for Thm. 2, which is stated in Section 3.3, and which lists sufficient conditions for error scores to be undulating (see Def. 3). We start with a few lemmas that will be used in this proof.

Lemma 1. Let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ denote a differentiable function with $f(0) \neq 0$. Then

$$\lim_{x \to 0^+} \frac{f(x)}{f(ax)} = 1 \quad \forall a \in (0, 1).$$

Proof. Omitted.

Lemma 2. Let $F : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ and $f = F'$ denote a positive differentiable strictly monotonically decreasing function ($F > 0, f < 0$). Assume that $\lim_{x \to \infty} F(x) = 0$, and that the limits $\lim_{x \to \infty} \frac{F(x)}{F(ax)}, \lim_{x \to \infty} \frac{f(x)}{f(ax)}$ exist $\forall a \in (0, 1)$. Denote $g(x) = -\ln(F(x))$. If

$$g'(x) \in \omega \left( \frac{1}{x} \right),$$

then:

$$\lim_{x \to \infty} \frac{f(x)}{f(ax)} = 0 \quad \forall a \in (0, 1).$$

Proof. We can write $F(x) = e^{-g(x)}$. It follows from the mean value theorem that $\exists t \ ax < t < x$ such that

$$\frac{F(x)}{F(ax)} = e^{-(g(x) - g(ax))} = e^{-(g'(t)x(1-a))} = e^{-(g'(t)x\cdot \frac{a}{1-a})}.$$

Since $g'(x) \in \omega \left( \frac{1}{x} \right)$ we get

$$\lim_{t \to \infty} t \cdot g'(t) = \infty.$$
As \((1 - a) < \frac{x}{a}(1 - a) < \frac{(1-a)}{a}\), it follows that
\[
\lim_{x \to \infty} \frac{F(x)}{F(ax)} = 0.
\]

From the assumption that the limits exist, and since \(\lim_{x \to \infty} F(x) = \lim_{x \to \infty} F(ax) = 0\), we can use L'Hôpital's rule and get
\[
\lim_{x \to \infty} \frac{f(x)}{af(ax)} = \lim_{x \to \infty} \frac{f(x)}{f(ax)} = 0.
\]

\[\square\]

**Lemma 3.** Let \(F: \mathbb{R}_{\geq 0} \to \mathbb{R}_{> 0}\) denote a positive differentiable function \((F > 0)\). Denote \(g(x) = -\ln(F(x))\). Assume that \(\lim_{x \to \infty} g'(x)\) exists, and
\[
g(x) \in \omega(\log(x)),
\]
then
\[
g'(x) \in \omega \left( \frac{1}{x} \right).
\]

**Proof.** \(g(x) \in \omega(\log(x))\) implies that
\[
\lim_{x \to \infty} \frac{g(x)}{\ln(x)} = \infty, \quad \lim_{x \to \infty} g(x) = \infty.
\]

We can now use L'Hôpital's rule and get
\[
\infty = \lim_{x \to \infty} \frac{g(x)}{\ln(x)} = \lim_{x \to \infty} \frac{g'(x)}{\frac{1}{x}} = \lim_{x \to \infty} xg'(x).
\]

\[\square\]

**Theorem 2.** Given partition \(R_1 \prec R_2\) and Error score \(E(x)\), let \(p = \text{Prob}(R_1)\) and \(0 < a < \frac{p}{1-p}\). \(E(x)\) is undulating if the following assumptions hold:

(i) \(E(x)\) is a proper error score (see Def. 5).

(ii) \(\lim_{x \to \infty} \frac{E'(x)}{E(x)}, \lim_{x \to \infty} \frac{E(x)}{E(ax)}, \lim_{x \to \infty} \frac{E'(ax)}{E'(ax)}\) exist \(\forall a \in (0, 1)\).
(iii) \( -\log(E(x)) \in \omega(\log(x)) \).

Proof. We define \( f(x) = -E'(px), a = \frac{\alpha(1-p)}{p} < 1 \). From assumption (i) and using Lemma 1, we get

\[
\lim_{x \to 0^+} \frac{E'(px)}{E'(\alpha(1-p)x)} = 1.
\]

Therefore there exists some \( z_1 \in \mathbb{R}_{\geq 0} \) such that \( \forall x < z_1 \)

\[
\frac{E'(px)}{E'(\alpha(1-p)x)} > \frac{\alpha(1-p)}{p}.
\]

From assumptions (i)-(iii) and using Lemmas 2-3, we get

\[
\lim_{x \to \infty} \frac{E'(x)}{E'(ax)} = \lim_{x \to \infty} \frac{E'(px)}{E'(\alpha(1-p)x)} = 0,
\]

and therefore there is some \( z_2 \in \mathbb{R}_{\geq 0} \) such that \( \forall x > z_2 \)

\[
\frac{E'(px)}{E'(\alpha(1-p)x)} < \frac{\alpha(1-p)}{p}.
\]

From Def. 3 we get that \( E(x) \) is undulating. \( \square \)

A.1.2 SP-undulating Error Score: Sufficient Conditions

We provide next the proof for Thm. 3, which is stated in Section 3.5, and which lists sufficient conditions for error scores to be SP-undulating (see Def. 3), extending Thm. 2. Once again, we start with a few lemmas.

Lemma 4. Let \( f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0} \) denote a positive differentiable function \( (f > 0) \). Let \( 0 < a < 1 \) denote some constant. If

\[
h(x) = \frac{-f'(x)x}{f(x)}
\]

is strictly monotonically increasing, then

\[
g(x) = \frac{f(x)}{f(ax)}
\]

is strictly monotonically decreasing.
Proof. $g(x)$ is monotonically decreasing iff $g'(x) < 0$, where

$$g'(x) = \frac{f'(x) f(ax) - af(x) f''(ax)}{f(ax)^2}.$$ 

This condition translates to

$$f'(x) f(ax) - af(x) f'(ax) < 0.$$ 

As by assumption $h(x)$ is monotonically increasing and $h(ax) < h(x)$, we get that $\forall x > 0$

$$\frac{-f'(ax) ax}{f(ax)} < \frac{-f'(x) x}{f(x)} \quad \Rightarrow \quad \frac{-f'(ax) a}{f(ax)} < \frac{-f'(x) x}{f(x)} \quad \Rightarrow \quad - f'(ax) f(x) a < -f'(x) f(ax).$$

Lemma 5. Let $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{>0}$ denote a positive differentiable log-concave function which is strictly monotonically decreasing ($f > 0, f' < 0, (\log(f))'' \leq 0$). Then the following function is strictly monotonically increasing

$$h(x) = \frac{-f'(x) x}{f(x)}.$$ 

Proof. $h(x)$ is strictly monotonically increasing iff $h'(x) > 0$, which holds iff

$$h'(x) = \frac{xf'(x)^2 - xf(x)f''(x) - f(x)f'(x)}{f(x)^2} > 0 \quad \Rightarrow \quad x \left[ f'(x)^2 - f(x)f''(x) \right] - f(x)f'(x) > 0.$$ 

Recall that $x > 0$ and $-f(x)f'(x) > 0 \forall x \in \mathbb{R}_{\geq 0}$. Since $f$ is log-concave, we also have that $f'(x)^2 - f(x)f''(x) \geq 0$, which concludes the proof. \qed
Theorem 3. Given partition $R_1 \prec R_2$ and Error score $E(x)$, let $p = \text{Prob}(R_1)$ and $0 < \alpha < \frac{p}{1-p}$. $E(x)$ is SP-undulating if the following assumptions hold:

1. $E(x)$ is an undulating proper error score.
2. At least one of the following conditions holds:
   (a) $-\frac{E''(x)}{E'(x)}$ is monotonically increasing with $x$.
   (b) $-E'(x)$ is strictly monotonic decreasing and log-concave.

Proof. Define the following positive continuous function

$$H(x) = \frac{E'(px)}{E'(\alpha(1-p)x)}.$$ 

Let $f(x) = -E'(x)$, $a = \frac{\alpha(1-p)}{p} < 1$. Note that assumption 2a follows from assumption 2b and Lemma 5. Assumption 2 therefore implies that $-\frac{f'(x)x}{f(x)}$ is strictly monotonically increasing, and by Lemma 4 we can conclude that $H(x)$ is monotonically decreasing. Together with assumption 1, $H(x) = \frac{\alpha(1-p)}{p}$ at a single point, and we may therefore conclude that $E(x)$ is SP-undulating.

Corollary 3. If $p$ – the probability of region $R_1$ – is sufficiently small so that $p < \frac{\alpha}{1+\alpha}$, then the conclusions are reversed: it is beneficial to initially over-sample $R_2$, and vice versa.

A.2 Error Function of Simple Mixture Models

A.2.1 Mixture of Two Linear Classifiers

We next prove Thm. 4 and Thm. 5, which are stated in Section 3.6.1. Thm. 4 provides a bound on the error score of a single linear classifier, showing that under mild conditions, this score is bounded by an exponentially decreasing function in the number of training examples $m$. Thm. 5 states conditions under which the error score of a mixture of two linear models $E(m)$ is undulating, and presents the phase-transition behavior.

Bounding the Error of Each Mixture Component

Henceforth we use the notations of Section 3.6.1, where for clarity, $m_j$ is replaced by $m$ while we are discussing the bound on a single component $j \in [2]$. Let $x_i$ denote the $i$-th
APPENDIX A. PROOFS

data point and $i$-th column of $X$. Let $\mu_1$ and $\mu_2$ denote the respective means of the two classes, and $\mu = \mu_1 - \mu_2$ denote the vector difference between the means.

Assuming that the data is normalized to 0 mean, the maximum likelihood estimators for the covariance matrix of the distribution $\Sigma$ and class means, denoted $\hat{\Sigma}$ and $\hat{\mu}_1, \hat{\mu}_2$ respectively, are the following

$$\hat{\Sigma} = \frac{1}{m} \sum_{i=1}^{m} x_i x_i^T = \frac{1}{m} XX^T \quad (A.1)$$

$$\hat{\mu}_j = \frac{1}{m_j} \sum_{x_i \in C_j} x_i \quad \Rightarrow \quad yX^T = m_1 \hat{\mu}_1 - m_2 \hat{\mu}_2, \quad (A.2)$$

where $C_j$ denotes the set of points in class $j \in [2]$. Thus, the ML linear separator can be written as

$$\hat{w} = yX^T (XX^T)^{-1} = [m_1 \hat{\mu}_1 - m_2 \hat{\mu}_2] (m\hat{\Sigma})^{-1} = \hat{\mu} \hat{\Sigma}^{-1},$$

where $\hat{\mu} = \frac{1}{m} yX^T$. Note that $\hat{\mu}$ is the sample mean of vectors $\{y_i x_i\}_{i=1}^{m}$, and $\hat{\Sigma}$ is the sample covariance of $\{x_i\}_{i=1}^{m}$.

When $d > m$ (fewer training points than the input space dimension), $\hat{\Sigma}$ is rank deficient and therefore $\hat{\Sigma}^{-1}$ is not defined. Moreover, the solution is not unique. Nevertheless, it can be shown that the minimal norm solution is the Penrose pseudo-inverse $\hat{\Sigma}^+$, where

$$\hat{\Sigma} = UDU^T \quad \Rightarrow \quad \hat{\Sigma}^+ = U\hat{\Sigma}^+ U^T,$$

and using the notations

$$D = \text{diag}(d_1, \ldots, d_m, 0, \ldots, 0)$$

$$D^+ = \text{diag}(d_1^{-1}, \ldots, d_m^{-1}, 0, \ldots, 0).$$

Ignoring the question of uniqueness, estimating $w$ is therefore reduced to evaluating the estimators in (A.1) and (A.2). These ML estimators have the following known upper bounds on their error:

1. Bounding $\hat{\Sigma}$: from known results on covariance estimation [45], using Bernstein
matrix inequality
$$P(\|\hat{\Sigma} - \Sigma\|_\text{op} \geq t) \leq 2de^{-\gamma^2mt^2}. \quad (A.3)$$

Constant $\gamma$ does not depend on $m$; it is determined by the assumed bound on the $L_2$ norm of vectors $x_i$, and the norm of the true covariance matrix $\Sigma$.

2. Bounding $\hat{\mu}$: starting from Hoeffding’s inequality in one dimension, we have that
$$P(\|\hat{\mu}^k - \mu^k\| \geq t) \leq 2e^{-\frac{2mt^2}{d^2}} \forall k \in [d],$$
where we assume a bounded distribution $\|x\| \leq \beta$. Thus
$$P(\|\hat{\mu} - \mu\| \geq t) = P(\|\hat{\mu} - \mu\|^2 \geq t^2)$$
$$= P\left(\sum_{k=1}^{d} |\hat{\mu}^k - \mu^k|^2 \geq t^2\right)$$
$$\leq \sum_{k=1}^{d} P\left(|\hat{\mu}^k - \mu^k|^2 \geq \frac{t^2}{d}\right)$$
$$\leq \sum_{k=1}^{d} P\left(|\hat{\mu}^k - \mu^k| \geq \frac{t}{\sqrt{d}}\right)$$
$$\leq 2de^{-\frac{2m^2\beta^2}{4d^2}} \quad (A.4)$$

The first inequality follows from the union-bound inequality.

**Lemma 6.**
$$\hat{\mu}^\top [\Sigma^{-1} - \hat{\Sigma}^+]x = \hat{\mu}^\top [\hat{\Sigma}^+ (\hat{\Sigma} - \Sigma)\Sigma^{-1}]x. \quad (A.5)$$

**Proof.** Because $\hat{\Sigma} \in \mathbb{R}^{d\times d}$ is of rank $m$, $Q = \hat{\Sigma}^+ \hat{\Sigma} = U\text{diag}(1, \ldots, 1, 0, \ldots, 0)U^\top$ is a projection matrix of rank $m$, projecting vectors to the subspace spanned by the training set $\{x_i\}_{i=1}^{m}$. Thus $Q\hat{\mu} = \hat{\mu}$. Additionally, by definition, $\hat{\Sigma}^+ \hat{\Sigma}^+ = \hat{\Sigma}^+$ and $Q$ is symmetric. It follows that
$$\hat{\mu}^\top (\Sigma^{-1} - \hat{\Sigma}^+)x = \hat{\mu}^\top (Q\Sigma^{-1} - \hat{\Sigma}^+)x,$$
while
$$\hat{\Sigma}^+ (\hat{\Sigma} - \Sigma)\Sigma^{-1} = Q\Sigma^{-1} - \hat{\Sigma}^+.$$ Together, we get (A.5). \qed
APPENDIX A. PROOFS

Theorem 4. Assume: (i) a bounded sample $\|x_i\| \leq \beta$, where $XX^\top$ is sufficiently far from singular so that its smallest eigenvalue is bounded from below by $\frac{1}{\Lambda}$; (ii) a realizable binary problem where the classes are separable by margin $\delta$; (iii) full rank data covariance, where $\frac{1}{\lambda}$ denotes its smallest singular value. Then there exist some positive constants $k, \nu > 0$, such that $\forall m_j \in \mathbb{N}$ and every sample $X^{mj}$, the expected error of $\hat{w}$ obtained using $X^{mj}$ is bounded by:

$$F(m_j) = \mathbb{E}_{X \sim D}[0 - 1 \text{ loss of } \hat{w}] \leq ke^{-\nu m_j}$$

Proof. It follows from our assumptions that $\|\hat{\Sigma}^+\|_{op} \leq \Lambda$. An error will occur at $x \in C_1$ if $w_{\text{opt}}x \geq \delta$ and $\hat{w}x < 0$, and vice versa for $x \in C_2$. In either case, the difference between the predictions of $w_{\text{opt}}$ and $\hat{w}$ deviate by more than $\delta$. Thus

$$F(m) = \mathbb{E}_{X \sim D}[0 - 1 \text{ loss}] = P(\text{error}) \leq P\left(\|\mu^\top \Sigma^{-1} x - \hat{\mu}^\top \hat{\Sigma}^+ x\| > \delta\right).$$

Invoking the triangular inequality

$$\Delta = \|\mu^\top \Sigma^{-1} x - \hat{\mu}^\top \hat{\Sigma}^+ x\| = |(\mu - \hat{\mu})^\top \Sigma^{-1} x + \hat{\mu}^\top (\Sigma^{-1} - \hat{\Sigma}^+)x| \leq |(\mu - \hat{\mu})^\top \Sigma^{-1} x| + |\hat{\mu}^\top (\Sigma^{-1} - \hat{\Sigma}^+)x|.$$  \hfill (A.6)

We now use Lemma 6 in order to shift $(\Sigma^{-1} - \hat{\Sigma}^+)$ to $(\Sigma - \hat{\Sigma})$. Specifically, we insert (A.5) into (A.6) to get

$$\Delta \leq \|\mu - \hat{\mu}\| \|\Sigma^{-1}\|_{op} \|x\| + \|\hat{\mu}\| \|\hat{\Sigma}^+\|_{op} \|\hat{\Sigma} - \Sigma\|_{op} \|\Sigma^{-1}\|_{op} \|x\| \leq \|\mu - \hat{\mu}\| \lambda \beta + \beta \Lambda \|\hat{\Sigma} - \Sigma\|_{op} \lambda \beta.$$
It follows that
\[
P\left(\left|\mu^\top \Sigma^{-1} x - \hat{\mu}^\top \hat{\Sigma} + x\right| > \delta\right)
\leq P\left(||\mu - \hat{\mu}|| \lambda \beta + ||\hat{\Sigma} - \Sigma||_{op} \lambda \Lambda \beta^2 > \delta\right)
\leq P\left(||\mu - \hat{\mu}|| \lambda \beta > \frac{\delta}{2}\right) + P\left(||\hat{\Sigma} - \Sigma||_{op} \lambda \Lambda \beta^2 > \frac{\delta}{2}\right)
\leq 2de^{-2m\left(\frac{\delta}{2\pi}\right)^2 \frac{1}{\sqrt{d}}} + 2de^{-am\left(\frac{\delta}{2\lambda \lambda \beta^2}\right)^2}
\leq 4de^{-kms^2}.
\]

The second transition follows from the union-bound inequality, and the third from (A.3)-(A.4) (where constant \(\gamma\) is defined). For the last transition we define
\[
k = \min\left\{2 \frac{1}{4\beta^2d} \left(\frac{1}{2\lambda \beta}\right)^2, \gamma \left(\frac{1}{2\lambda \Lambda \beta^2}\right)^2\right\}.
\]

\[\square\]

A Mixture Classifier

Assume a mixture of two linear classifiers, and let \(E(m) = p \cdot E_{D_1}(m_1) + (1 - p) \cdot E_{D_2}(m_2)\) denote its error score.

**Theorem 5.** Keep the assumptions stated in Thm. 4, and assume in addition that \(\forall a \in (0, 1), \exists \lim_{m \to \infty} E'(m), \lim_{m \to \infty} E(m), \lim_{m \to \infty} E'(am)\). Then the error score of a mixture of two linear classifiers is undulating.

**Proof.** In each region \(j\) of the mixture, Thm. 4 implies that its corresponding error score as defined in Def. 1,
\[
E_{D_j}(m_j) = \mathbb{E}_{x_{m_j} \sim D_j}[F(m_j)]
\]
is bounded by an exponentially decreasing function of \(m_j\). Since \(E_{D_j}(m_j)\) measures the expected error over all samples of size \(m_j\), it can also be shown that \(E_{D_j}(m_j)\) is mono-
Appendix A. Proofs

Tonically decreasing with \(m_j\). From the separability assumption, \(\lim_{m_j \to \infty} E_{D_j}(m_j) = 0\). Finally, since \(E(m)\) is a linear combination of two such functions, it also has these properties. We conclude from Cor. 1 that the error score of a mixture of two linear classifiers is undulating.

A.2.2 1-NN Classifier

If the training sample size \(m\) is small, our analysis shows that under certain circumstances, it is beneficial to prefer sampling from a region \(R\) where \(E_{D_R}(m) < E_{D_R \setminus [k]}(m)\). We now show that the set of densest points has this property.

To this end, we adopt the one-Nearest-Neighbor (1-NN) classification framework. This is a natural framework to address the aforementioned question for two reasons: (i) It involves a general classifier with desirable asymptotic properties. (ii) The computation of both class density and 1-NN is governed by local distances in the input feature space.

To begin with, assume a classification problem with \(k\) classes that are separated by at least \(\rho\). More specifically, assume that \(\forall x, x' \in \mathbb{R}^d, \text{if } \|x' - x\| \leq \rho \text{ then } y' = y\). Let \(B_v(x_i, r)\) denote a ball centered at \(x_i \in \mathbb{R}^d\), with radius smaller than \(r\) and volume \(v\). For \(X = (x, y)\), let \(f(x)\) denote the density function from which data is drawn when sampling is random (and specifically at test time).

Assume a 1-NN classifier whose training sample is \(T = \{x_i, y_i\}_{i=1}^m\), and whose prediction at \(X = (x, y)\) is

\[
y = \begin{cases} 
y_v, & v = \arg \min_{i \in [m]} \|x - x_i\| \quad x \in B_v(x_i, \rho) \\
y \sim U(1, k) & \text{otherwise}
\end{cases}
\]

The error probability of this classifier at \(x\) is

\[
P(x) = \begin{cases} 
0 & \exists i \text{ such that } x \in B_v(x_i, \rho) \\
k - 1 \over k & \text{otherwise}
\end{cases}
\]

The 0–1 loss of this classifier is

\[
\mathbb{E}_{x \sim D}[P(x)] = \frac{k - 1}{k} \text{Prob} \left[ x \notin \bigcup_{i=1}^m B_v(x_i, \rho) \right], \quad (A.7)
\]
where $B_v(x_i, r)$ denotes a ball centered at $x_i \in \mathbb{R}^d$, with radius smaller than $r$ and volume $v$.

The next theorem states properties of set $T$ which are beneficial to the minimization of this loss:

**Theorem 6.** Let $A_i$ denote the event $\{x \in B_v(x_i, r)\}$, and assume that these events are independent. Then we have

$$L(T) = \frac{k - 1}{k} \left[ 1 - \sum_{i=1}^{m} f_D(X_i)v + O(v^2) \right].$$

**Proof.** Using the independence assumption and (A.7), and assuming that $v$ is sufficiently small

$$L(T) = \frac{k - 1}{k} \left[ 1 - \sum_{i=1}^{m} f_D(X_i)v + O(v^2) \right].$$

In Thm. 6, we show that if $v$ is sufficiently small, the 0-1 loss is minimized when choosing a set of independent points $\{X_i\}_{i=1}^{m}$ that maximizes $\sum_{i=1}^{m} f_D(X_i)$. This suggests that the selection of an initial pool of size $m$ will benefit from the following heuristic:

- **Max density:** when selecting a point $X_i$, maximize its density $f_D(X_i)$.
- **Diversity:** select varied points, for which the events $\{x \in B_v(x_i, \rho)\}$ are approximately independent.
B Visualizations

B.1 Theoretical Analysis: Visualization

B.1.1 Mixture of Two Linear Models

We now empirically analyze the error score function of the mixture of two linear classifiers, defined in Section 3.6.1. Each linear classifier is trained on a different area of the support. The data is 100 dimensional, linearly separable in each region. The margin is used to determine the \( \alpha \) of the data. The data is chosen such that \( p = 0.9, \alpha = 0.2 \). The results are shown in Fig. B.1.

![Graphs](image)

Figure B.1: (a) The error score \( E(m) \) as a function of the number of examples, averaged over 10k repetitions. While the error score is not exponential, it could be upper bounded by an exponential function, as analytically shown in Section 3.6.1. (b) The differences in accuracy when over-sampling from either \( R_1 \) and \( R_2 \) over a random sampling from the data distribution. Although the error score is only proven to be undulating, we can see that in practice it is also SP-undulating.

B.1.2 Error Scores of Deep Neural Networks

Next, we plot the error scores of deep neural networks on image classification tasks. In all the datasets we evaluated, the error of deep networks as a function of the number of examples drops much faster than an exponential function and therefore can be shown to be undulating. In practice, such error functions are bounded from above by an exponent, and hence are also SP-undulating. To see some examples of error functions of neural networks trained on super-classes of CIFAR-100, refer to Fig. B.2.
Figure B.2: Log-error scores of image classification datasets as a function of the number of training examples. Each score is calculated as $1 - \text{accuracy}$, averaged on 100 VGG-16 networks trained on super-classes of CIFAR-100. Each line represents the error of a different super-class. As the log of the error is plotted, it can be seen that in all cases the error scores are monotonic decreasing and can be bounded above by some exponential function, suggesting that often the assumptions in Thm. 3 hold.

![Graphs showing log-error scores as a function of number of training examples for different super-classes.](image)

Figure B.3: (a)-(c) Visualizing the selection of 30 examples using the SCAN clustering algorithm – examples marked with $\times$ are selected for labeling. (d) The selected images, each column represents a different label.

![Visualizations of selected examples using different color schemes.](image)

### B.2 Visualization of Query Selection

Fig. B.3 demonstrates the selection of 30 examples from CIFAR-10 using TypiClust in greater detail. Recall that TypiClust first clusters the dataset to 30 clusters – using
Figure B.4: 100 ImageNet-100 examples selected by TypiClust.

SCAN clustering algorithm. We plot the tSNE dimensionality reduction of the model’s feature space, colored in various ways: Fig. B.3a shows the tSNE embedding colored by the GT labels. Fig. B.3b shows the tSNE embedding colored by the cluster assignment. Fig. B.3c shows the tSNE embedding colored by the log density (for better visualization). Examples marked with $\times$ are selected for labeling. Fig. B.3d shows the selected images. Fig. B.4 shows 100 examples selected by TypiClust from ImageNet-100.
C Additional Empirical Results

Figure C.1: A Comparative evaluation of our method using ImageNet-50. (a) Similar to Fig. 5.1, trained in the fully supervised framework. (b) Similar to Fig. 5.2, trained in the semi-supervised by transfer learning framework.

When evaluating our method on ImageNet-50, we report a similar qualitative behavior as seen in other datasets: *ProbCover* performs better than all baselines in the very low-budget regime, using 5 AL rounds with budget equal to $b = 50$ examples. More concretely, in Fig. C.1 we show results corresponding to Figs. 5.1-5.2 when using ImageNet-50.