Representation Alignment in Neural Networks

by

Ehsan Imani

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Classical wisdom in machine learning advises controlling the complexity of the hypothesis space for achieving good generalization. Despite this, modern overparametrized neural networks demonstrate remarkably high generalization performance, oftentimes with larger and more expressive architectures outperforming smaller ones. Motivated by these observations and other studies that produced similar phenomena in kernel regression, we study generalization in high-dimensional linear models through the lens of representation alignment, a measure of how much the labels vary in directions where the data is more spread out. Understanding when this relationship between the features and the labels holds and its potential for refining theoretical analyses and algorithms underlie the contributions in this thesis. We formally describe representation alignment and show how it connects to optimization and generalization. We then evaluate neural network hidden representations with this measure and find that training neural networks increase representation alignment in their hidden representations under a wide range of architectures and design choices. Based on these observation, we derive a regularization method for domain adaptation and find that enforcing alignment between the predictions and the given representation can help in domain adaptation. Finally, we extend the insights to policy evaluation and study generalization with temporal-difference learning.

Preface

The findings in Chapter 5 are based on our TMLR publication (Imani et al., 2022) although the experiments in this document have been modified to maintain a coherent setup through the thesis. The results in Chapter 6 have been accepted to JMLR (Imani et al., 2024).

In addition to the results in this document, over the past four years I have contributed to the following projects:

- **Imani, E.**, Luedemann, K., Scholnick-Hughes, S., Elelimy, E., & White, M. (2024). Investigating the Histogram Loss in Regression. *In Submission.*

- Jafferjee, T., Aminmansour, F., **Imani, E.**, Talvitie, E., White, M., & Bowling, M. (2024). Multistep Predecessor Models and Mitigating Errors due to Hallucinated Value in Dyna-Style Planning. *JAIR.*

- Masarczyk, W., Ostaszewski, M., **Imani, E.**, Pascanu, R., Miłoś, P., & Trzciński, T. (2023). The Tunnel Effect: Building Data Representations in Deep Neural Networks. *NeurIPS.*

- Graves, E., **Imani, E.**, Kumaraswamy, R., & White, M. (2023). Off-policy actor-critic using emphatic weightings. *JMLR.*

- Pan, Y., **Imani, E.**, Farahmand, A. M., & White, M. (2020). An implicit function learning approach for parametric modal regression. *NeurIPS.*

To the lovely three

Far across the sea

What feels better than being free?

– Sarina Esmaeilzadeh.

Acknowledgements

First of all, I am grateful to my supervisor Martha White for her effective management that helped this research progress seamlessly and made this PhD program an exciting adventure. In 2019, having spent a master's program with Martha, asking her for PhD supervision was an obvious decision, and I would make the same choice without hesitation if I were to go back in time. It has been a privilege to spend these years under her guidance and to learn such invaluable lessons in research and character.

My gratitude extends to my committee members Dale Schuurmans, Adam White, Marlos Machado, and Yaoliang Yu. Their feedback and questions helped me notice my mistakes and find how this research fits in the literature.

I was fortunate to collaborate and meet with many people from different organizations and I owe the breadth of this thesis to their diverse perspectives. Special thanks goes to my co-authors Wei Hu, Guojun Zhang, Runjia Li, Jun Luo, Pascal Poupart, Philip Torr, and Yangchen Pan. I am also indebted to Alireza Fallah, Roshan Shariff, Huizhen Yu, and Clare Lyle for helping me challenge my preconceptions and think more clearly about generalization. These collaborations were thanks to the support by the University of Alberta, Alberta Machine Intelligence Institute (Amii), and Huawei Noah's Ark Lab.

Finally, I take this chance to remember my classmates and dear friends Pouneh and Arash and all the victims of passenger flight PS752 which was shot down by two Islamic Revolutionary Guard Corps missiles fired around 30 seconds apart on January 8th, 2020.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Generalization, the ability to perform well on unseen data, is a central goal in machine learning (Bishop, 2006), and much of theoretical advances in this field have been dedicated to characterizing generalization (Mohri et al., 2018; T. Zhang, 2023). A good theoretical framework would provide guarantees and estimates for generalization and guide the design of new algorithms and models (Kawaguchi et al., 2017).

Classical frameworks have characterized the generalization gap, the deterioration of performance on unseen data, using a measure of capacity of the function class. Such theoretical results would guarantee smaller generalization gap for more restricted function classes (Vapnik & Chervonenkis, 1971; Shalev-Shwartz & Ben-David, 2014). The restriction can be inherent in the model design, implicit in the training algorithm, or explicitly enforced through a regularizer. These results have shown to be insightful in certain situations and often tight under their assumptions (Mukherjee et al., 2006; Mohri et al., 2018), reinforcing old rules of thumb for improving generalization by restricting the hypothesis class (Popper, 2005).

Modern deep learning practice challenges these guidelines. Large function classes of neural networks perform well in practice with more recent architectures such as Residual Networks and Transformer generalizing better despite growing in size and expressivity (Krizhevsky et al., 2009; He et al., 2016; Dosovitskiy et al., 2020; Kaplan et al., 2020). Empirical studies have shown that modern neural networks can fit large data with random labels, implying

that neither the architecture design, nor the training algorithm, nor moderate amounts of regularization as commonly applied heavily restrict the function class capacity (Neyshabur et al., 2014; C. Zhang et al., 2017). While this phenomenon does not contradict the classical theory—the theory does not imply that large function classes are guaranteed to generalize poorly, the observation motivates developments that answer when a model can generalize well regardless of its capacity.

A growing paradigm against this backdrop is benign overfitting whose premise is providing generalization bounds and estimates that remain under control as the model capacity grows (Belkin et al., 2019; Bartlett et al., 2020; Hastie et al., 2022). Since many empirical phenomena in deep learning can be reproduced with simpler linear models (Belkin et al., 2018; Jacot et al., 2018), most of benign overfitting results focus on linear models where capacity is controlled by the number of input dimensions. These results characterize properties in the input that guarantee good generalization even as the number of dimensions goes to infinity.

Approach: We are interested in the generalization performance of high-dimensional linear models. Rather than focusing on the input, we ask what type of relationship between the input and the target can ensure good generalization. One such relationship, *representation alignment*, is the central theme of this thesis. In short, representation alignment means that the targets vary mostly in directions where the data is more spread out. Through this document we will define representation alignment rigorously and show its role in generalization and then go beyond generalization and study how representation alignment affects optimization, when it emerges, and how it can be used as prior knowledge.

1.1 Contributions

The following list summarizes the key contributions with the thesis statements italicized:

- **Improved generalization bounds using representation alignment.**

Using a bias-variance decomposition in a regression model and margin theory in classification, we show that *estimators with high generalization performance can be obtained from a small sample if representation alignment is high.* We verify this insight on synthetic and practical experiments.

- **Empirical study on the connection between representation alignment and optimization.** Motivated by a simple characterization of the expected convergence rate of gradient descent, we conduct an empirical study on different optimizers in regression and classification and observe that *a variety of common optimizers tend to converge faster in practice when representation alignment is high.*

- **Empirical study on emergence of representation alignment.** We evaluate neural network hidden representations in a large range of training setups and find that *training neural networks increases representation alignment in their hidden representations on the training tasks and in common feature transfer scenarios.*

- **Label alignment regularization for distribution shift.** Using the observation in the previous contribution, we derive a regularization method for domain adaptation and find that *enforcing alignment between the predictions and the given representation can help in domain adaptation.* Instead of regularizing representation as done by popular domain adaptation methods, we regularize the classifier to align with the unsupervised target data based on our prior knowledge about the relationship between the representation and the labels in each of the source and target domains. In a linear regression setting, we characterize the relationship between our regularized solution and the optimal solution on the target domain. We show the efficacy of our method on problems where classic domain adaptation methods are known to fail. We also report improvement over domain adaptation baselines on cross-lingual sentiment analysis tasks.

- **Empirical study on generalization with temporal-difference learning.** Based on our earlier findings, we hypothesize that, in the context of policy evaluation, *temporal-difference learning can find generalizable solutions from a small sample of transitions if its expected convergence rate is high.* We verify this hypothesis on four small testbeds.

1.2 Roadmap

The next Chapter will give a precise definition of representation alignment and discuss its place in the literature. Then, Chapter 3 provides the theoretical results on generalization along with verification on synthetic experiments. Chapter 4 provides a simple expected convergence rate for gradient descent using representation alignment. In Chapter 5 we study when high representation alignment emerges. Then in the same chapter, having tasks with different degrees of representation alignment at hand, we empirically study our earlier insights on the connection between representation alignment and optimization and generalization. Chapter 6 develops and tests a regularizer to enforce prior knowledge about representation alignment in domain adaptation. Chapter 7 turns to policy evaluation and asks whether the connections between convergence rate and generalization extend to temporal-difference learning. Finally, Chapter 8 reviews the findings, discusses the limitations, and concludes the thesis.

Chapter 2

Representation Alignment

This chapter formalizes representation alignment, provides basic insights on the role of representation alignment on performance, and summarizes some previous work that introduced and benefited from similar concepts.

2.1 Definition

Within the scope of this document we define a task as a pair of random variables (ϕ, y) where $\phi \in \mathbb{R}^d, \mathbb{E}[\phi] = \mathbf{0}, \mathbb{E}[\|\phi\|^2] \neq \mathbf{0}$ and $y \in \mathbb{R}, \mathbb{E}[y] = 0, \mathbb{E}[y^2] \neq 0$, that is, both ϕ (the representation) and y (the target) are centered but they are not constantly zero. Define $w^* := \lim_{\lambda \to 0^+} \arg\min_{w \in \mathbb{R}^d} \mathbb{E}[(w^\top \phi - y)^2] + \lambda \|w\|^2$ as the min-norm Mean Squared Error (MSE) minimizer, and the random variable $\epsilon := {w^*}^\top \phi - y$. Note that while it is easy to show that $\mathbb{E}[\epsilon] = 0$, we do not have $\mathbb{E}[\epsilon|\phi] = 0$ in general, permitting model misspecification. The covariance matrix is $H := \mathbb{E}[\phi\phi^\top]$ which in eigendecomposition can be written as $H = \sum_{i=1}^d \sigma_i^2 v_i v_i^\top$ where the sequence $(\sigma_i)_{i=1}^d$ is non-negative and non-increasing. For a threshold $\tau \geq 0$ the truncated covariance matrix is $H_\tau := \sum_{\{i:\sigma_i \geq \tau\}} \sigma_i^2 v_i v_i^\top$. For a vector $x \in \mathbb{R}^d$ and a matrix $A \in \mathbb{R}^{d \times d}$ the matrix norm $\sqrt{x^\top A x}$ is denoted by $\|x\|_A$.

Definition 2.1.1. *For a task (ϕ, y) and threshold $\tau \geq 0$ we define representation alignment as*

$$\mathit{Alignment}(\phi, y, \tau) := \|w^*\|_{H_\tau}^2 / \mathbb{E}[y^2]$$

Figure 2.1: (Left) Consider the two tasks above that have identical two-dimensional Gaussian features shown by the scatter plots. The real-valued labels change from negative (blue) to positive (red) numbers. In Task 1 the labels change along the first principal component. (Right) The two curves show representation alignment as a function of the threshold for the two tasks. The curve for Task 1 drops later, and we informally say that representation alignment is higher for this task. This example is only a simple illustration. Interesting differences in generalization emerge in presence of noise and high dimensions.

The normalization ensures that the measure is between zero and one. This is because both the numerator and the denominator are positive, and $\mathbb{E}[y^2] = \mathbb{E}[(w^{*\top}\phi + \epsilon)^2] = \mathbb{E}[(w^{*\top}\phi)^2] + \mathbb{E}[\epsilon^2] + 0 \geq \|w^*\|_H^2 \geq \|w^*\|_{H_\tau}^2$.

To get a more intuitive understanding of representation alignment, consider the two tasks (ϕ, y_1) and (ϕ, y_2) shown in Figure 2.1. The two tasks have the same representation ϕ which is a two-dimensional anisotropic Gaussian with $H := \left(\begin{smallmatrix} 1.0 & 0.5 \\ 0.5 & 0.5 \end{smallmatrix}\right)$ as illustrated in the figure, and different targets $y_1 := \phi^\top v_1/\sigma_1$ and $y_2 := \phi^\top v_2/\sigma_2$, leading to MSE minimizers $w_1^* = v_1/\sigma_1$ and $w_2^* = v_2/\sigma_2$ and $\mathbb{E}[y_1^2] = \mathbb{E}[y_2^2] = 1$. We have plotted representation alignment for the two tasks as a function of the threshold in the same figure. As the threshold is raised, both curves drop to zero, while the first curve drops later than the second one. In such a clear case where the curves do not cross each other, we informally say that representation alignment is higher for the first task without referring to the threshold. The figure illustrates the intuitive interpretation that representation alignment is higher when targets mostly vary in directions where the representation is more elongated.

This relationship between the representation and the target has important consequences for generalization and optimization. Figure 2.2 shows the MSE landscape for the two tasks above and reveals the basic idea. For the task

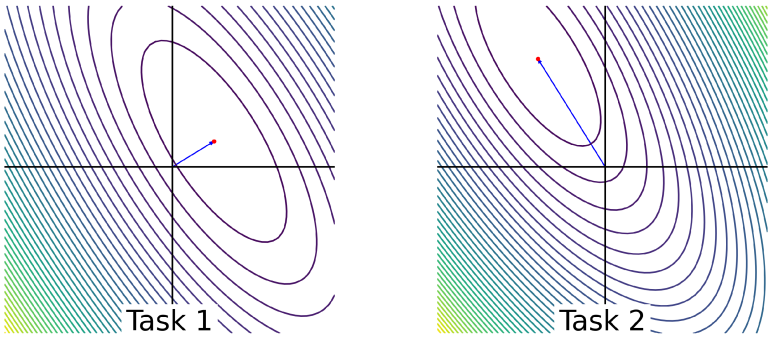


Figure 2.2: The MSE landscape for the two tasks in the introduction. The contours show $\mathbb{E}[(w^\top \phi - y)^2]$ for $w \in \mathbb{R}^2$. On the task on the left (whose labels vary along the first principal component) the MSE minimizer is closer to the origin and the trajectory from the origin to this point lies on the steep direction.

with higher representation alignment, compared to the other task, the MSE minimizer is closer to the origin and the path from the origin to this point is steeper. The first observation is consequential for generalization and the second for optimization and we will formulate these insights through the thesis.

2.2 Related Work

While the exact formulation in Definition 2.1.1 is new, multiple recent works have used related concepts to improve understanding of generalization and optimization. Arora et al. (2019) noted that in commonly encountered tasks the labels are such that the MSE minimizer for a sample has a small norm, and used this insight as an assumption in Rademacher complexity analysis to obtain improved generalization bounds for linear models and wide neural networks. Oymak et al. (2019) extended this idea to analysis of gradient descent with early stopping. The Rademacher complexity approach would require the assumption to hold simultaneously in any two independent samples from the task. In contrast, our notion of representation alignment is defined on the task itself rather than a sample and we will use alternative approaches of bias-variance decomposition and margin theory to analyze generalization.

The closest line of work to representation alignment is by Zou et al. (2021a,

2021b) and Wu et al. (2022). These analyses concern one-pass stochastic gradient descent (one-pass SGD) that performs one iteration on each item in the sample and, at the end, averages the parameters obtained along the way. Their assumption, similar to our definition, breaks the covariance matrix into two matrices with large and small eigenvalues and assumes that the MSE minimizer is mostly aligned with the first matrix. The theory then uses this assumption to characterize the performance of the obtained solution. Note that in one-pass SGD optimization and generalization are entangled since the algorithm performs one iteration per item in the sample. Our analysis is on gradient descent and will disentangle the role of representation alignment on optimization and generalization.

Another notion closely related to representation alignment is Task-Model Alignment. Canatar et al. (2021) introduced this measure and used it to obtain a generalization estimate with remarkable agreement with practice. Presenting the measure and the generalization estimate would require notation from kernel regression and statistical mechanics and we avoid it here and only highlight that the interpretation is again that when Task-Model Alignment is high the targets mostly vary in directions of elongation. Figure 2.3 shows how the obtained theoretical estimate captures the generalization performance in practice. Although such agreement in the experiment is striking and promising, the tool replica trick used to prove the generalization estimate has problems as discussed by Castellani and Cavagna (2005).

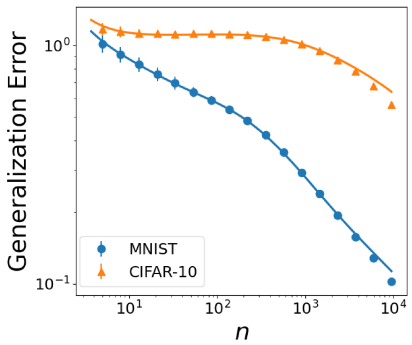


Figure 2.3: (Canatar et al., 2021) Generalization error of regression on two datasets. See the paper for details. Note the near-perfect match between the markers (showing performance in the experiments) and the curves (showing theoretical predictions).

Finally, Kernel-Target Alignment (KTA) is a common objective in kernel learning and maximizing it has been proven to improve linear separability and empirically associated with better classification performance (Cristianini et al., 2001; Cortes et al., 2012). KTA is a scalar measure, i.e. it does not

have an extra parameter like the threshold in representation alignment, and later we will see that neither KTA nor any other scalar measure can fully capture the trends in optimization. Regarding generalization, however, KTA plays an important role, and we will in fact connect representation alignment and generalization in classification through a connection with KTA.

Chapter 3

Representation Alignment and Generalization

This chapter provides theoretical results that connect representation alignment and generalization. Using a bias-variance decomposition in the first section we show that representation alignment helps in a certain regression setting by reducing the bias term. The next section uses margin theory to show the role of representation alignment on the performance of a classifier. Both results are applicable to high dimensions.

3.1 Generalization in Well-Specified Regression

Consider the task (ϕ, y) where $\phi := H^{1/2}z$, $H \in \mathbb{R}^{d\times d}$ is symmetric and $z \in \mathbb{R}^d$ is a random vector whose independent elements (not necessarily identically distributed) have zero mean and unit variance. Define the min-norm MSE minimizer w^* and target noise ϵ similar to Section 2.1. Assume $\mathbb{E}[\|y\|^2] = 1$ and that and the distribution of ϵ has mean zero and standard deviation σ_ϵ and is independent of ϕ, that is, there is no model misspecification and $\mathbb{E}[y|\phi] = {w^*}^\top\phi$.

The goal is to estimate w^* using only an independent and identically distributed (iid) sample in the form of $((\phi_i, y_i))_{i=1}^n$ arranged into a matrix $\Phi \in \mathbb{R}^{n\times d}$ and a vector $\mathbf{y} \in \mathbb{R}^n$. We will first study the performance of min-norm estimate $\hat{w} := (\Phi^\top\Phi)^\dagger\Phi^\top\mathbf{y}$. Equivalently, $\hat{w}$ is the estimate with the

smallest ℓ_2-norm among the set of estimates that minimize the MSE on $(\Phi, \mathbf{y})$ (Bartlett et al., 2020). Then we will give a result for the ridge regression estimate $\hat{w}^\lambda := (\Phi^\top \Phi + n\lambda I)^{-1}\Phi^\top \mathbf{y}$ for $\lambda > 0$. Given a new sample (ϕ_0, y_0), the risk of an estimate $w \in \mathbb{R}^d$ is defined and decomposed into a bias and a variance term in this way (Hastie et al., 2022).

$$
\begin{aligned}
R_\Phi(w, w^*) &:= \mathbb{E}[(w^\top \phi_0 - y_0)^2 | \Phi] = \mathbb{E}[\|w - w^*\|_H^2 \, | \Phi] \\
&= \underbrace{\|\mathbb{E}[w|\Phi] - w^*\|_H^2}_{B_\Phi(w,w^*)} + \underbrace{\mathrm{Tr}[\mathrm{Cov}[w|\Phi]H]}_{V_\Phi(w,w^*)}.
\end{aligned}
$$

To see the role of representation alignment in the analysis, note that H is the covariance matrix and can be written in eigendecomposition as $\sum_{i=1}^d \sigma_i^2 v_i v_i^\top$. In this setting, if w^* is in directions of eigenvectors of H with smaller eigenvalues it needs to be larger in ℓ_2 norm. This is because $\mathbb{E}[\|y\|^2] = \mathbb{E}[\left\|\phi^\top w^*\right\|^2] + \sigma_\epsilon^2 = w^* H^{1/2} \mathbb{E}[zz^\top] H^{1/2} w^* + \sigma_\epsilon^2 = \|w^*\|_H^2 + \sigma_\epsilon^2 = \sum_{i=1}^d (\sigma_i v_i^\top w^*)^2 + \sigma_\epsilon^2$. The term σ_ϵ^2 does not depend on w^* and therefore $\|w^*\|_H = \sqrt{1 - \sigma_\epsilon^2}$ regardless of the direction of w^*. The same does not hold for $\|w^*\|$. The later elements of the sum $\sum_{i=1}^d (\sigma_i v_i^\top w^*)^2$ are weighted by small eigenvalues and if w^* is mostly in these directions, then $\|w^*\|$ needs to be large.

3.1.1 A Numerical Result

The relationship between representation alignment and generalization is best introduced through an example. Therefore, we will first provide the numerical result in this section and then discuss the theory behind it in the next section. Consider the case where $H = \mathrm{diag}([1, 1/2^2, 1/3^2, \cdots, 1/d^2])$, $z_i \sim \mathcal{N}(0, 1)$ for all $i \in [d]$, $\epsilon \sim \mathcal{N}(0, 0.01)$. In other words, the elements of ϕ are mutually independent and the first elements have a larger scale compared to the later ones.

The tasks that we will compare have w^* set to $w_i := (1/\sqrt{1 - \sigma_\epsilon^2})(1/\sigma_i)v_i$ for different values of i. The normalization $(1/\sqrt{1 - \sigma_\epsilon^2})$ ensures that $\mathbb{E}[\|y\|^2] = 1$. As we discussed in the previous chapter, w_i for smaller values of i results in higher representation alignment. For each task we will estimate $\hat{w}$ using a sample with $n = 200$, and estimate its risk on an independent sample with

$n = 1000$ and then compare the risks across the tasks to get an idea of the role of representation alignment in generalization.

The method for obtaining $\hat{w}$ in this example is gradient descent

$$\hat{w}^0 := 0, \hat{w}^t := \hat{w}^{t-1} - \eta \nabla (\frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} (\phi_i^\top \hat{w}^{t-1} - y_i)^2).$$

We will cover gradient descent in more depth later in the thesis and here we only mention that it finds the min-norm estimate studied in the theorem above, that is, we have $\lim_{t \to \infty} \hat{w}^t = \hat{w}$ with appropriate value of η (Hastie et al., 2022).

Figure 3.1 (left) shows the risk for different tasks. All curves start at 1, since $\hat{w}^0 = 0$ and $\mathbb{E}[y^2] = 1$, and then converge to different risks. Tasks whose w^* are in directions with larger eigenvalues will result in lower risk.

Recall that in general there could be more than one estimate that minimizes the risk on the sample, and the analysis in this chapter will be on a certain estimate, the min-norm estimate. Other estimates may behave differently. To demonstrate this important caveat, we can use a preconditioned gradient descent approach

$$\hat{w}^0 := 0, \hat{w}^t := \hat{w}^{t-1} - \eta H^{-2} \nabla (\frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} (\phi_i^\top \hat{w}^{t-1} - y_i)^2)$$

which, under appropriate conditions on η, finds an estimate with the minimum risk on the sample but not necessarily the min-norm estimate (Amari et al., 2021). As shown in Figure 3.1 (right), this estimate shows the exact opposite pattern, with tasks whose w^* are in directions with larger eigenvalues resulting in higher risk. We do not pursue this direction further in this work and keep the analysis of generalization limited to the min-norm estimate and ridge regression estimate.

3.1.2 Basic Theoretical Insight

To understand the pattern in the previous section, we use a theorem by Hastie et al. (2022) to illustrate the role of representation alignment in generalization

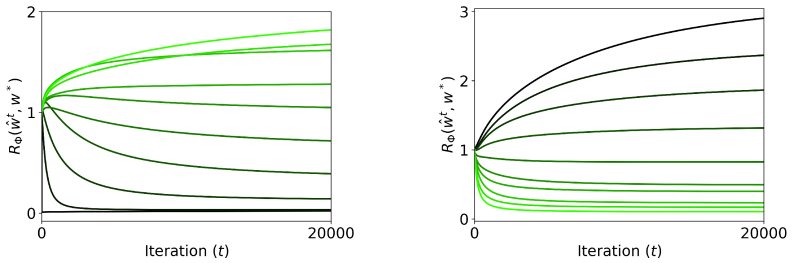


Figure 3.1: (Left) Performance of gradient descent. Note that the darkest curve immediately drops to near zero. (Right) Performance of preconditioned gradient descent. In both plots each curve corresponds to a task and the tasks with brighter curves have lower representation alignment. The results are averaged over 10 runs. Gradient descent solution (min-norm estimate) has lower risk when representation alignment is higher as will be shown by the theory. Preconditioned gradient descent shows the opposite trend.

through the norm of w^*. Define the following two distributions on $\mathbb{R}_{\geq 0}$:

$$\hat{S}(s) := \frac{1}{d}\sum_{i=1}^{d} \mathbb{1}\{\sigma_i^2 \leq s\}, \quad \hat{G}(s) := \frac{1}{\|w^*\|^2}\sum_{i=1}^{d} \langle w^*, v_i\rangle^2 \mathbb{1}\{\sigma_i^2 \leq s\}$$

and assume there are constants M and $\{C_k\}_{k\geq 2}$ that satisfy the following:

Assumption 1. *We have*

1. *For all $k \geq 2$ the magnitude of the k-th moment of each element of z is at most $C_k < \infty$.*
2. *$\sigma_1^2 = \|H\|_{op} \leq M, \int s^{-1} d\hat{S}(s) < M$.*
3. *$|1 - d/n| \geq 1/M, 1/M \leq d/n \leq M$.*

Define γ as d/n, $c_0 = c_0(\gamma, \hat{S}) \in \mathbb{R}_{>0}$ as the nonnegative solution of

$$1 - \frac{1}{\gamma} = \int \frac{1}{1 + c_0\gamma s} d\hat{S}(s)$$

which is unique (Hastie et al., 2022), and the predicted bias and the predicted variance as

$$\mathscr{B}(\hat{S}, \hat{G}, \gamma) := \|w^*\|^2 \left\{1 + \gamma c_0 \frac{\int \frac{s^2}{(1+c_0\gamma s)^2} d\hat{S}(s)}{\int \frac{s}{(1+c_0\gamma s)^2} d\hat{S}(s)}\right\} \cdot \int \frac{s}{(1 + c_0\gamma s)^2} d\hat{G}(s)$$

$$\mathscr{V}(\hat{S}, \gamma) := \sigma_\epsilon^2 \gamma \frac{\int \frac{s^2}{(1+c_0\gamma s)^2} d\hat{S}(s)}{\int \frac{s}{(1+c_0\gamma s)^2} d\hat{S}(s)}$$

The following theorem shows that $\mathscr{B}, \mathscr{V}$ characterize the bias and variance of the risk of the ridgeless estimate:

Theorem 3.1.1 (Hastie et al. (2022), Theorem 2)**.** *Assume $\sigma_d^2 > 1/M$ and Assumption 1 holds. Then for any constant $D > 0$ there exist $C = C(M, D)$ such that with probability at least $1 - Cn^{-D}$ we have*

$$|B_\Phi(\hat{w}, w^*) - \mathscr{B}(\hat{S}, \hat{G}, \gamma)| \leq \frac{C \|w^*\|^2}{n^{1/7}}$$
$$|V_\Phi(\hat{w}, w^*) - \mathscr{V}(\hat{S}, \gamma)| \leq \frac{C}{n^{1/7}}$$

We can set w^* to $w_i := (1/\sqrt{1-\sigma_\epsilon^2})(1/\sigma_i)v_i$ for different values of i. In the decomposition above, the predicted variance $\mathscr{V}$ does not depend on w^*. The predicted bias $\mathscr{B}$ depends on w^* only through $\|w^*\|^2$ and $\hat{G}$. Setting w^* to w_i we have $\mathscr{B}(\hat{S}, \hat{G}, \gamma) \propto 1/(1 + c_0\gamma\sigma_i^2)^2$. Since c_0 and γ are positive, the predicted bias will be smaller for smaller values of i that correspond to larger eigenvalues.

3.1.3 The Full Result

More generally the optimal weights do not have to be exactly in the direction of a certain eigenvector. We characterize this general case by incorporating representation alignment as an assumption to refine another theorem that considers ridge regression.

Hastie et al. (2022) characterized the risk of the ridge regression estimate in the following way. For $\zeta \in \mathbb{C}_+$ (a complex number with positive imaginary part) define $m(\zeta) = m(\zeta, \hat{S}, \gamma)$ as the unique solution of

$$m(\zeta) = \int \frac{1}{s[1 - \gamma - \gamma\zeta m(\zeta)] - \zeta} d\hat{S}(s)$$

and define $m_1(\zeta) = m_1(\zeta, \hat{S}, \gamma)$ as

$$m_1(\zeta) := \frac{\int \frac{s^2[1-\gamma-\gamma\zeta m(\zeta)]}{[s[1-\gamma-\gamma\zeta m(\zeta)]-\zeta]^2} d\hat{S}(s)}{1 - \gamma \int \frac{\zeta s}{[s[1-\gamma-\gamma\zeta m(\zeta)]-\zeta]^2} d\hat{S}(s)}$$

The functions are defined for $\mathrm{Im}(\zeta) = 0$ using limits when the limit exists. The predicted bias and variance are defined as

$$\mathscr{B}(\lambda, \hat{S}, \hat{G}, \gamma) := \lambda^2 \|w^*\|^2 (1 + \gamma m_1(-\lambda)) \int \frac{s}{[\lambda + (1 - \gamma + \gamma\lambda m(-\lambda))s]^2} d\hat{G}(s)$$

$$\mathscr{V}(\lambda, \hat{S}, \gamma) := \sigma_\epsilon^2 \gamma \int \frac{s^2(1-\gamma+\gamma\lambda^2 m'(-\lambda))}{[\lambda + s(1-\gamma+\gamma\lambda m(-\lambda))]^2} d\hat{S}(s)$$

Similar to the previous result, the following theorem characterizes the risk of the ridge regression estimate in terms of the predicted bias and variance:

Theorem 3.1.2 (Hastie et al. (2022), Theorem 5)**.** *Assume* $\max(\lambda, \sigma_d^2) > 1/M$ *and Assumption 1 holds. Then for any constants* $D > 0$ *and* $\varepsilon > 0$ *there exist* $C = C(M, D)$ *such that with probability at least* $1 - Cn^{-D}$ *we have*

$$|B_\Phi(\hat{w}^\lambda, w^*) - \mathscr{B}(\lambda, \hat{S}, \hat{G}, \gamma)| \leq \frac{C \left\|w^*\right\|^2}{\lambda n^{(1-\varepsilon)/2}}$$
$$|V_\Phi(\hat{w}^\lambda, w^*) - \mathscr{V}(\lambda, \hat{S}, \gamma)| \leq \frac{C}{\lambda^2 n^{(1-\varepsilon)/2}}$$

We can improve this result using representation alignment. For a threshold $\tau \geq 0$ the covariance matrix can be decomposed into the top subspace $H_\tau := \sum_{\{i:\sigma_i \geq \tau\}} \sigma_i^2 v_i v_i^\top$ and the bottom subspace $H_{\bar{\tau}} := \sum_{\{i:\sigma_i < \tau\}} \sigma_i^2 v_i v_i^\top$. Recall $\|w^*\|_H^2 = 1 - \sigma_\epsilon^2$ and then define $\delta := \|w^*\|_{H_\tau}^2$. A higher value of δ means that the optimal weights are mostly in the top subspace. By expanding the matrix norm we get that $\|w^*\|_{H_{\bar{\tau}}}^2 = 1 - \sigma_\epsilon^2 - \delta$.

The bias term in the theorem above grows with $\left\|w^*\right\|^2$ which in general can be large if $1/\sigma_d^2$ is large. Our goal is to give a result that removes this factor of $1/\sigma_d^2$ in the bias. That is, we want a bound on the bias to not grow as much when σ_d^2 goes to zero while other factors are kept the same. To do this we bound the bias in the top subspace and the bottom subspace in two different ways. The bias in the top subspace is controlled similar to the previous theorem and will depend on the smallest eigenvalue of this subspace which is at least as large as τ. The bias in the lower subspace is controlled with a different approach. This part of the bias will shrink to zero if the matrix norm of the weights in this bottom subspace is small and, importantly, it will not include the aforementioned factor of $1/\sigma_d^2$. For this step we add an extra assumption that the input noise is subgaussian. We only need this assumption to control the sample covariance matrix of the input noise. The full proof with the details is in Appendix A.

Define w_τ^* and $w_{\bar{\tau}}^*$ as the projection of w^* on H_τ and $H_{\bar{\tau}}$. Further define

$$\hat{G}_\tau(s) := \frac{1}{\|w_\tau^*\|^2} \sum_{i=1}^{d} \langle w_\tau^*, v_i \rangle^2 \mathbb{1}\{\sigma_i^2 \leq s\}$$

$$\mathscr{B}_\tau(\lambda, \hat{S}, \hat{G}, \gamma) := \lambda^2 \|w_\tau^*\|^2 (1 + \gamma m_1(-\lambda)) \int \frac{s}{[\lambda + (1 - \gamma + \gamma\lambda m(-\lambda))s]^2} d\hat{G}_\tau(s)$$

which are similar to $\hat{G}$ and $\mathscr{B}$ but depend on w_τ^* instead of w^*. The following theorem relates the risk to $\hat{G}_\tau, \mathscr{B}_\tau$ with an extra assumption on z.

Theorem 3.1.3. *Assume* $\max(\lambda, \sigma_d^2) > 1/M$ *and Assumption 1 holds and that the higher moments are such that* z *is* σ_z*-subgaussian. Then for any constants* $D > 0$ *and* $\varepsilon > 0$ *there exist* $C = C(M, D)$ *such that with probability at least* $1 - Cn^{-D}$ *we have*

$$\begin{aligned}
&R_\Phi(\hat{w}^\lambda, w^*) = B_{\Phi,\tau}(\hat{w}^\lambda, w^*) + V_\Phi(\hat{w}^\lambda, w^*) + \Delta \\
&|B_{\Phi,\tau}(\hat{w}^\lambda, w^*) - \mathscr{B}_\tau(\lambda, \hat{S}, \hat{G}, \gamma)| \leq \frac{C \|w_\tau^*\|^2}{\lambda n^{(1-\varepsilon)/2}} \\
&|V_\Phi(\hat{w}^\lambda, w^*) - \mathscr{V}(\lambda, \hat{S}, \gamma)| \leq \frac{C}{\lambda^2 n^{(1-\varepsilon)/2}} \\
&\Delta \leq \frac{C\sqrt{1 - \sigma_\epsilon^2 - \delta}}{\min(\lambda, \lambda^2)} (1 + \max(\sigma_z^2, \sigma_z^4) \max\{\sqrt{\frac{d + \log n}{n}}, (\frac{d + \log n}{n})^2\})
\end{aligned}$$

Compared to the previous theorem we have replaced $\|w^*\|^2$ in the bias term with $\|w_\tau^*\|^2$ by incorporating representation alignment as an assumption. This can be an improvement since in general $\|w^*\|^2$ can be $O(1/\sigma_d^2)$ while $\|w_\tau^*\|^2$ is $O(1/\tau^2)$. The extra term Δ will depend on δ and will be small if $\delta \approx 1 - \sigma_\epsilon^2$, that is when $\|w^*\|_{H_{\bar{\tau}}} \approx 0$, regardless of how small the smallest eigenvalue is.

3.2 Generalization in Classification

This section shows the role of representation alignment in generalization in classification through connections with margin theory. The first part shows a generalization bound using the fact that high representation alignment results in lower margin loss. Then we will discuss the connection between representation alignment and Kernel-Target Alignment. Finally, we provide numerical results similar to the previous section.

3.2.1 Generalization Bound

Consider the centered feature mapping $\phi : \mathcal{X} \to \mathbb{R}^d$, $\mathbb{E}[\phi] = \mathbf{0}$, and the associated kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined as $K(x, x') := \phi(x)^\top \phi(x')$ with $K(x, x') \le \Omega^2$ for $x, x' \in \mathcal{X}$. Similarly define a balanced labeling function $y : \mathcal{X} \to \{\pm 1\}$, $\mathbb{E}[y] = 0$ and the associated kernel function $K_y : \mathcal{X} \times \mathcal{X} \to \{\pm 1\}$ defined as $K_y(x, x') := y(x)y(x')$ for $x, x' \in \mathcal{X}$. Suppose we have a sample $((\phi_i, y_i))_{i=1}^n$ where, for $i \in [n]$, x_i are iid and ϕ_i and y_i are short for $\phi(x_i)$ and $y(x_i)$. The sample can be arranged into a matrix $\Phi \in \mathbb{R}^{n \times d}$ and $\mathbf{y} \in \mathbb{R}^n$. The kernels $\hat{K}$ and $\hat{K}_y$ can be defined similar to above but using Φ and $\mathbf{y}$. Our goal is to use the sample to find a linear classifier $w_{\text{clf}} \in \mathbb{R}^d$ such that the expected 0-1 risk $R(w_{\text{clf}}) := \Pr[y(x)(w_{\text{clf}}^\top \phi(x)) < 0]$ is small.

Step 1: The first step is to lower bound $\mathbb{E}[KK_y] := \mathbb{E}_{x,x'}[K(x, x')K_y(x, x')]$ using representation alignment.

Proposition 3.2.1. *Suppose Alignment*$(\phi, y, \tau) = \delta$ *for a* $\tau \ge 0$. *Then* $\mathbb{E}[KK_y] \ge \delta\tau^2$.

Proof. Since $y \in \{\pm 1\}$ we have $\mathbb{E}[y^2] = 1$. Recall the MSE minimizer in the definition of representation alignment and let us denote it as w_{MSE} in the context of classification for clarity, thus having $\|w_{\text{MSE}}\|_{H_\tau}^2 = \delta$. For an iid sample of size n arranged into Φ and $\mathbf{y}$, thin singular value decomposition (thin SVD) gives $\Phi = \hat{U}\hat{\Sigma}\hat{V}$ where, importantly, $\hat{\Sigma} \in \mathbb{R}^{d \times d}$ and the decomposition is chosen arbitrarily if it is not unique. Dependence of the matrices on n is left implicit for conciseness. Eigendecomposition on the covariance matrix $H := \mathbb{E}[\phi\phi^\top]$ gives $H = V\Sigma^2 V^\top$ where $\lim_{n\to\infty} \hat{\Sigma} = \Sigma$ and $\lim_{n\to\infty} \hat{V} = V$. Define $\Sigma_\tau \in \mathbb{R}^{d \times d}$ as a diagonal matrix with diagonal elements $\sigma_i \mathbb{1}\{\sigma_i \ge \tau\}$. We can write w_{MSE} as $w_{\text{MSE}} = H^\dagger \mathbb{E}[\Phi y] = \lim_{n\to\infty} H^\dagger \frac{1}{n} \Phi^\top \mathbf{y}$. Therefore

$$\begin{aligned}
\delta = \|w_{\text{MSE}}\|_{H_\tau}^2 &= w_{\text{MSE}}^\top H_\tau w_{\text{MSE}} = \lim_{n\to\infty} \frac{1}{n^2} \mathbf{y}^\top \Phi H^\dagger H_\tau H^\dagger \Phi^\top \mathbf{y} \\
&= \lim_{n\to\infty} \frac{1}{n^2} \mathbf{y}^\top (\hat{U}\hat{\Sigma}\hat{V}^\top)(V\Sigma^{2\dagger}V^\top)(V\Sigma_\tau^2 V^\top)(V\Sigma^{2\dagger}V^\top)(\hat{V}\hat{\Sigma}\hat{U}^\top)\mathbf{y} \\
&= \lim_{n\to\infty} \frac{1}{n^2} \mathbf{y}^\top \hat{U}\Sigma^\dagger \Sigma_\tau \hat{U}^\top \mathbf{y} = \lim_{n\to\infty} \frac{1}{n^2} \sum_{\sigma_i \ge \tau} (\hat{U}_{:i}^\top \mathbf{y})^2
\end{aligned}$$

We can write $\mathbb{E}[KK_y]$ similarly to obtain the final result.

$$\mathbb{E}[KK_y] = \lim_{n\to\infty} \frac{1}{n^2}\mathbf{y}^\top \Phi\Phi^\top \mathbf{y} = \lim_{n\to\infty} \frac{1}{n^2}\mathbf{y}^\top \hat{U}\Sigma^2\hat{U}^\top \mathbf{y} = \lim_{n\to\infty} \frac{1}{n^2}\sum_{i=1}^{d}(\sigma_i \hat{U}_{:i}^\top \mathbf{y})^2$$
$$\geq \lim_{n\to\infty} \frac{1}{n^2}\sum_{\sigma_i\geq\tau}(\sigma_i \hat{U}_{:i}^\top \mathbf{y})^2 \geq \lim_{n\to\infty} \frac{1}{n^2}\sum_{\sigma_i\geq\tau}\tau^2(\hat{U}_{:i}^\top \mathbf{y})^2 = \tau^2\delta$$

□

Step 2: Now we turn to the Frobenius dot product $\frac{1}{n^2}\langle\Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F$, the empirical analogue of $\mathbb{E}[KK_y]$, and show that these two are related with the following concentration inequality.

Proposition 3.2.2. *For any $D > 0$ with probability at least $1 - D$ we have*

$$\frac{1}{n^2}\langle\Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F \geq \mathbb{E}[KK_y] - 4\Omega^2\sqrt{\frac{\log\frac{1}{D}}{n^3}}$$

Proof. The kernels are centered and $\mathbb{E}[KK_y] = \mathbb{E}[\frac{1}{n^2}\langle\Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F]$ per Equation (11) by Cortes et al. (2012). Changing an element (ϕ_i, y_i) in a sample will change $\frac{1}{n^2}\langle\Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F$ by no greater than $(2\Omega)^2/n^2$. Therefore, McDiarmid's inequality (McDiarmid et al., 1989) gives that for any $\varepsilon > 0$

$$\Pr[\frac{1}{n^2}\langle\Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F < \mathbb{E}[KK_y] - \varepsilon] \leq \exp\left(\frac{-2\varepsilon^2}{n((2\Omega)^2/n^2)^2}\right)$$

Setting D equal to the right-hand side gives the result. □

Step 3: We will now show that high $\frac{1}{n^2}\langle\Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F$ implies that a weight vector with low empirical margin loss can be obtained using only the available sample. For a weight vector $w \in \mathbb{R}^d$ and margin $m > 0$ define $\Psi_m(x) := \min(1, \max(0, 1 - x/m))$ and define the empirical margin loss as $\hat{R}_m(w) := \frac{1}{n}\sum_{i=1}^{n}\Psi_m(y_i(w^\top\phi_i))$.

Proposition 3.2.3. *Set $w_{clf} := \frac{(1/n)\sum_{i=1}^{n} y_i\phi_i}{\sqrt{\frac{1}{n^2}\langle\Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F}}$. Then for any $0 < m < \Omega$*

$$\hat{R}_m(w_{clf}) \leq 1 - \frac{1}{\Omega}\sqrt{\frac{1}{n^2}\langle\Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F}$$

Proof. First note that $\|w_{\text{clf}}\| = \sqrt{\frac{\frac{1}{n^2}\langle \Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F}{\frac{1}{n^2}\langle \Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F}} = 1$, therefore $|y_i(w^\top\phi_i)| \leq \Omega$, and if the margin is set to Ω the max operator in Ψ_m will not come into effect. Secondly, the function Ψ and therefore $\hat{R}$ is nondecreasing in m, therefore for all $0 < m < \Omega$ we have $\hat{R}_m(w_{\text{clf}}) \leq \hat{R}_\Omega(w_{\text{clf}})$ which itself can be bounded as

$$\begin{aligned}\hat{R}_\Omega(w_{\text{clf}}) \leq & \frac{1}{n}\sum_{i=1}^{n} 1 - y_i(w_{\text{clf}}^\top\phi_i)/\Omega = 1 - \frac{1}{\Omega}w_{\text{clf}}^\top(\frac{1}{n}\sum_{i=1}^{n} y_i\phi_i) \\ = & 1 - \frac{1}{\Omega}\sqrt{\frac{1}{n^2}\langle \Phi\Phi^\top, \mathbf{y}\mathbf{y}^\top\rangle_F}\end{aligned}$$

□

Step 4: The following generalization bound relates the empirical margin loss and generalization performance.

Theorem 3.2.4 (Mohri et al. (2018), p98)**.** *For all $w \in \mathbb{R}^d, \|w\| \leq 1$ and $m \in (0, \Omega]$ and for any $D > 0$, with probability at least $1 - D$*

$$R(w) \leq \hat{R}_m(w) + 4\sqrt{\frac{\Omega^2/m^2}{n}} + \sqrt{\frac{\log\log_2\frac{2\Omega}{m}}{n}} + \sqrt{\frac{\log\frac{2}{D}}{2n}}$$

Final Result: Putting steps 1-4 together, setting the margin to Ω, and using a union bound on the stochastic events in steps 2 and 4 we get our final result that gives a generalization bound using representation alignment.

Corollary 3.2.5. *For any $D > 0$, if n is large enough such that $\tau^2\delta - 4\Omega^2\sqrt{\frac{\log(2/D)}{n^3}} > 0$, with probability at least $1 - D$*

$$R(w_{clf}) \leq 1 - \sqrt{(\frac{\tau}{\Omega})^2\delta - 4\sqrt{\frac{\log(2/D)}{n^3}}} + 4\sqrt{\frac{1}{n}} + \sqrt{\frac{\log\frac{2}{D}}{2n}}$$

Alternatively, we can skip step 1 to give a generalization bound using $\mathbb{E}[KK_y]$.

Corollary 3.2.6. *For any $D > 0$, if n is large enough such that $\mathbb{E}[KK_y] - 4\Omega^2\sqrt{\frac{\log(2/D)}{n^3}} > 0$, with probability at least $1 - D$*

$$R(w_{clf}) \leq 1 - \sqrt{\frac{\mathbb{E}[KK_y]}{\Omega^2} - 4\sqrt{\frac{\log(2/D)}{n^3}}} + 4\sqrt{\frac{1}{n}} + \sqrt{\frac{\log\frac{2}{D}}{2n}}$$

If the goal is obtaining a guarantee on generalization from a sample, the original theorem 3.2.4 is what one would use anyway since the empirical margin loss can be computed using only the available sample. Corollary 3.2.5 provides intuition on when the margin loss can be low using the representation alignment measure in the underlying task. Corollary 3.2.6 gives a similar insight and has the benefit of depending only on one quantity $\mathbb{E}[KK_y]$ rather than two quantities τ and δ simultaneously. The use of τ and δ in Corollary 3.2.5 is to show how the result connects with the rest of the thesis.

3.2.2 Connection to Kernel-Target Alignment

This section will briefly review the connection between representation alignment and a popular kernel learning objective. Let us drop the restriction that the two kernels in the previous section are centered by construction and instead define two centered kernels $K_c(x, x') := (\phi(x) - \mathbb{E}_x[\phi])^\top(\phi(x') - \mathbb{E}_{x'}[\phi])$ and $K_{yc}(x, x') := (y(x) - \mathbb{E}_x[y])^\top(y(x') - \mathbb{E}_{x'}[y])$. Suppose $0 < \mathbb{E}[K_c^2] < \infty$ and $0 < \mathbb{E}[K_{yc}^2] < \infty$. Centered Kernel-Target Alignment (CKTA) is defined as[1]

$$\text{CKTA}(K, K_y) := \frac{\mathbb{E}[K_c K_{yc}]}{\sqrt{\mathbb{E}[K_y^2]\mathbb{E}[K_{yc}^2]}}$$

An empirical analogue of this measure can be defined for kernel matrices. Recall Φ and $\mathbf{y}$ from the previous section. The kernels $\hat{K}$ and $\hat{K}_y$ and the associated centered kernels $\hat{K}_c$ and $\hat{K}_{yc}$ can be defined similar to above but using Φ and $\mathbf{y}$. Suppose $\left\|\hat{K}_c\right\|_F \neq 0$ and $\left\|\hat{K}_{yc}\right\|_F \neq 0$. Empirical Centered Kernel-Target Alignment (ECKTA) is defined as (Cortes et al., 2012; Kornblith et al., 2019a)

$$\text{ECKTA}(\hat{K}, \hat{K}_y) := \frac{1/(n-1)^2 \langle \hat{K}_c, \hat{K}_{yc} \rangle_F}{1/(n-1)^2 \left\|\hat{K}_c\right\|_F \left\|\hat{K}_{yc}\right\|_F}$$

CKTA and ECKTA are related by a concentration bound, showing that high CKTA in the task implies high ECKTA in a sample. Maximizing ECKTA

[1]The objective is generally called Centered Kernel Alignment when the second kernel does not necessarily correspond to targets (Cortes et al., 2012). We use the more explicit name Centered Kernel-Target Alignment for clarity.

is a popular approach to kernel learning, motivated by high linear separability of kernels with high CKTA (Cortes et al., 2012).

The normalization factor $1/(n-1)^2$ in ECKTA is arbitrary because its effect on the numerator and the denominator will be cancelled out. We used this certain normalization factor to highlight the connection to earlier work. The numerator of ECKTA is known as Hilbert-Schmidt Independence Criterion (HSIC) and is an estimator of $\mathbb{E}[K_c K_{yc}]$ (Gretton et al., 2005). HSIC is similar to the quantity that we earlier used to obtain low empirical margin loss. The main difference is that it uses the normalization factor $1/(n-1)^2$ rather than $1/n^2$. In the setting of our result the kernels K and K_y were centered by construction and we did not have to center the kernels $\Phi\Phi^\top$ and $\mathbf{y}\mathbf{y}^\top$, and the estimator with normalization $1/n^2$ was unbiased. Obtaining an unbiased estimator in general is more complicated and both $1/n^2$ and $1/(n-1)^2$ will result in biased estimators (L. Song et al., 2007).

3.2.3 Numerical Simulation

We define ϕ such that its i-th element is uniformly distributed on $[1/i^2]$ for $i \in [d]$. We compare 5 tasks where the label in each task is set as $(2\mathbb{1}\{\phi^\top v_j > 0\} - 1)$ for $j \in \{1, 2, 4, 8, 16\}$, i.e., the labels change in the direction of a certain eigenvector of the covariance matrix. Figure 3.2 (left) shows the representation alignment curves for the different tasks.

We used a sample with $n = 200$ (train sample), obtained estimators using this sample, and evaluated the estimates using an independent sample with $n = 1000$ (test sample). Figure 3.2 (right) shows the expected 0-1 risk of a common approach, gradient descent on logistic loss defined as

$$\hat{w}^0 := 0, \hat{w}^t := \hat{w}^{t-1} - \eta \nabla \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{\log(2)} \log(1 + \exp(\phi_i^\top \hat{w}^{t-1} y_i))\right)$$

in solid curves as well as the performance of w_{clf} in dashed horizontal lines. The number of iterations was large enough to ensure near-zero risk on the train sample for all the tasks. As predicted by the theory, w_{clf} achieves lower risk on tasks with higher representation alignment. The same trend also appears with the estimators obtained with gradient descent.

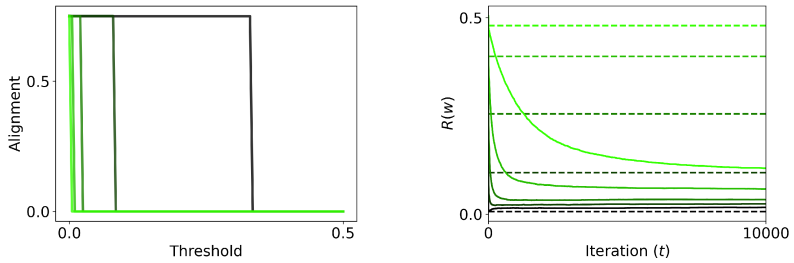


Figure 3.2: (Left) Representation alignment curves. Darker shades show tasks where the labels changed in directions with larger eigenvalues. (Right) Solid curves show the performance of gradient descent and dashed lines show the performance of w_{clf}. Each curve corresponds to a task and the tasks with brighter curves have lower representation alignment. The solid curves and the dashed lines on the right are both averaged over 10 runs where the randomness in the runs is in the draw of train and test samples. As predicted by the generalization bound, w_{clf} has lower risk when representation alignment is higher. The performance of gradient descent across tasks mirrors this trend.

3.3 Discussion

In this chapter we theoretically studied the role of representation alignment on generalization in regression and classification. The list of contributions in this chapter is as follows:

1. In a regression setup we used bias-variance decomposition to argue that if the MSE minimizer is in the direction of an eigenvector of the covariance matrix with high eigenvalue, min-norm estimator will have low bias and therefore low risk. We provided Numerical results on Gaussian data regarding this insight.

2. In a more general setting where the MSE minimizer is not perfectly aligned with an eigenvector and gave an risk estimate for ridge regression estimator that will be small if the MSE minimizer is mostly in a top subspace of the covariance matrix.

3. In a classification setup we used margin theory and provided a generalization bound that relates representation alignment and the expected 0-1 risk.

An interesting observation in the regression experiments was that estimators other than the min-norm estimator can behave differently and even show the opposite pattern. Amari et al. (2021) studied the role of preconditioning on the risk to understand which preconditioner is better for a certain task. The transpose of their question, which task is easier for a certain preconditioner, would be an interesting future direction.

Chapter 4

Representation Alignment and Optimization

This brief chapter discusses the connection between representation alignment and optimization through a well-known convergence rate of gradient descent. We will then provide a simple numerical simulation to demonstrate this connection and the need for a threshold in the definition of representation alignment. These basic theoretical insights will set the stage for an empirical study in the next chapter.

4.1 Convergence Rate

Consider the setup we introduced in Chapter 2 to define representation alignment and suppose $\mathbb{E}[y^2] = 1$ for simplicity. Gradient descent on expected MSE is defined as the sequence $\hat{w}^0 := 0$ and $\hat{w}^t := \hat{w}^{t-1} - \eta\nabla\mathbb{E}[\frac{1}{2}(\phi^\top\hat{w}^{t-1} - y)^2]$ for $t > 0$, where η is the step-size and t is the iteration. We are interested in minimizing the expected risk which for weights $w \in R^d$ is defined as $\mathbb{E}[(\phi^\top w - y)^2]$. Define $\tilde{w} \in \mathbb{R}^d$ as a vector composed of elements $\tilde{w}_i := {w^*}^\top v_i\sigma_i$. The following theorem shows the expected risk of $\hat{w}^t$:

Theorem 4.1.1. *If $0 < \eta < \sigma_1^{-2}$, the expected risk at each iteration is*

$$\mathbb{E}[(\phi^\top\hat{w}^t - y)^2] = \sum_{i=1}^{d}(1 - \eta\sigma_i^2)^{2t}\tilde{w}_i^2 + \mathbb{E}[\epsilon^2]$$

At the beginning the risk is simply $\mathbb{E}[y^2]$. Through the iterations, the terms in the loss with nonzero eigenvalues will shrink, reducing the overall risk. The

terms with zero eigenvalues will themselves remain zero throughout, because the corresponding element of $\tilde{w}$ is zero. In the limit of $t \to \infty$, the overall sum will become zero and the remaining risk will be $\mathbb{E}[\epsilon^2]$. The proof is in Appendix B.

To see how representation alignment comes into play note that representation alignment is high if the first elements of $\tilde{w}$ are large. This is because $\text{Alignment}(\phi, y, \tau) := \|w^*\|^2_{H_\tau} / \mathbb{E}[y^2] = \sum_{\{i:\sigma_i \geq \tau\}} \tilde{w}_i^2$. In the convergence rate above, the terms corresponding to the first elements of $\tilde{w}$ will shrink at a faster rate because the associated eigenvalue is larger. The following proposition captures this insight:

Proposition 4.1.2. *If $0 < \eta < \sigma_1^{-2}$ and $\text{Alignment}(\phi, y, \tau) = \delta$ for a threshold $0 < \tau < \sigma_1$, then gradient descent needs at most $O(1/\eta\tau^2)$ iterations to reduce the loss by 0.9δ.*

In other words, if representation alignment is large for a high threshold, then gradient descent will reduce a large amount of the risk at a high rate, regardless of how small the other eigenvalues are. The next section will illustrate these insights. See Appendix B for the proof.

4.2 Numerical Results

Let us return to the set of tasks in Section 3.1.1. Since we are now interested in the behavior of gradient descent on expected MSE, we will use a large sample of size 10000 and report the risk on the same sample. Once again we create different tasks by setting w^* to $w_i := (1/\sqrt{1-\sigma_\epsilon^2})(1/\sigma_i)v_i$ for different values of i. Each curve in the Figure 4.1 (left) shows the risk for one the tasks through the first 50000 iterations. All the curves start at 1 and shrink towards 0.01. As expected by the theorem above, darker curves drop at a faster rate.

Now we turn to the proposition above that takes representation alignment directly into account. By varying the threshold, the proposition describes the amount of risk that can be reduced at a certain rate. We can demonstrate this by comparing the two tasks with $\epsilon = 0$ and optimal weights v_2/σ_2 and

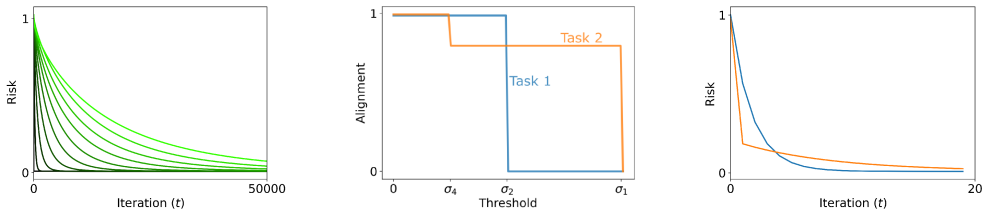


Figure 4.1: (Left) Convergence rate on 10 tasks with different levels of representation alignment. Each curve corresponds to a task and the tasks with brighter curves have lower representation alignment. As predicted by the theorem, higher representation alignment results in faster convergence. (Middle) Representation alignment curves for two tasks used to verify the proposition. (Right) Convergence rate for the same two tasks. The second tasks reduces a large amount of the loss at a fast rate. The rest of the loss is reduced faster on the first task.

$\sqrt{0.8}v_1/\sigma_1 + \sqrt{0.2}v_4/\sigma_4$. As seen in Figure 4.1 (middle), representation alignment can be higher for each of the two tasks depending on the threshold. The behavior of gradient descent on these two tasks in Figure 4.1 (right) reflects this pattern. In both tasks the risk shrinks from 1 to 0 through the iterations. Gradient descent on the second task reduces a large amount of the risk at a faster rate compared to the second task, since representation alignment for the second task is larger than the first task for high thresholds. The rest of the risk will be reduced slower in the second task compared to the first task, since it takes a lower threshold to make representation alignment 1 on the second task.

4.3 Discussion

We showed the role of representation alignment on the convergence rate of zero-initialized gradient descent on MSE in expectation. The theoretical result in this chapter is known (Arora et al., 2019; Jacot et al., 2018) and limited in scope. In the next chapter, after we discuss cases where high representation alignment emerges, we will empirically explore the connection between representation alignment and optimization in a wider range of settings.

Chapter 5

Representation Alignment in Feature Transfer

Previous chapters explored the role of representation alignment on optimization and generalization and verified the insights on synthetic tasks. This chapter turns to neural network hidden representations and considers a practical scenario where high representation alignment emerges. Our findings in this regard are that (1) across a wide range of architectures and hyperparameters, learned hidden representations of neural networks achieve higher representation alignment compared to the input representation and the hidden representation at initialization, (2) in fully-connected neural networks, hidden layers closer to the output layer tend to have higher representation alignment for a wide range of thresholds, and (3) in a typical object classification feature transfer scenario, neural network representations have higher representation alignment compared to handcrafted features for a wide range of thresholds.

Having tasks with different degrees of representation alignment at hand, we will then extend our previous synthetic experiments on optimization and generalization to more practical scenarios. We will verify that (4) we can find generalizable solutions from a small sample under high representation alignment and (5) representation alignment helps optimization with some common optimizers.

5.1 Feature Transfer

Let us first give a more rigorous definition of a feature transfer scenario. Suppose we have an upstream task (x^U, y^U) with $x^U \in \mathcal{X}^U, y^U \in \mathcal{Y}^U$ and a downstream task (x^D, y^D) with $x^D \in \mathcal{X}^D, y^D \in \mathcal{Y}^D$. A sample $S^U := ((x_i^U, y_i^U))_{i=1}^{n^U}$ is available from the upstream task, and $S^D := ((x_i^D, y_i^D))_{i=1}^{n^D}$ from the downstream task, the latter being much smaller, i.e. $n^D \ll n^U$. The ultimate goal is to learn an estimator from the small set S^D that performs well on the downstream task regardless of whether this estimator performs well on, or is even applicable to, the upstream task.

We can sometimes benefit from the large available sample S^U towards the aforementioned goal. Feature transfer paradigm proposes to learn a feature extractor $\phi : \mathcal{X}^U \cup \mathcal{X}^D \to \mathbb{R}^d$ using the large sample S^U. We can then use ϕ to extract a representation $\phi(x^D)$, abbreviated to ϕ^D. The premise is that, if the two tasks are in some sense related, obtaining high performance on (ϕ^D, y^D) would be easier than on (x^D, y^D).

A common example is to first learn a neural network on the upstream task with a large amount of data, then extract features from an intermediate layer of that network and finally train a subsequent model on the downstream task using those extracted features. The motivation is that neural networks adapt their intermediate representations—hidden representations—to the upstream task and, due to the commonalities between the two tasks, these learned representations help training on the downstream task (Y. Bengio et al., 2013). Availability of large datasets like ImageNet (Russakovsky et al., 2015) and the News Dataset for Word2Vec (Mikolov et al., 2013) provides suitable upstream tasks that facilitate using neural networks for feature construction for Computer Vision and Natural Language Processing (NLP) tasks (Kornblith et al., 2019b; Oquab et al., 2014; Devlin et al., 2018; Pennington et al., 2014).

There is as yet much more to understand about when and why feature transfer is successful. Understanding the properties of the learned hidden representations and their benefits for training on similar tasks has remained a longstanding challenge (Touretzky & Pomerleau, 1989; Zhou et al., 2015;

Marcus, 2018). One strategy has been to define properties of a good representation, and try to either measure or enforce those properties. Disentanglement and invariance are two such properties (Y. Bengio et al., 2013), where the idea is that disentangling the factors that explain the data and are invariant to most local changes of the input results in representations that generalize and transfer well. Although encoding properties for transfer is beneficial, it remains an important question exactly how to evaluate the representations that do emerge.

One challenge is that even hidden representations of two neural networks trained on identical tasks appear completely different, and studying the representations requires measures that separate recurring properties from irrelevant artifacts (Morcos et al., 2018). One direction has been to analyze what abstractions the network has learned, agnostic to exactly how it is represented. Shwartz-Ziv and Tishby (2017) studied neural networks through the lens of information theory and found that, during training, the network preserves the information necessary for predicting the output while throwing away unnecessary information successively in its intermediate layers. Using representational similarity matrices, Hermann and Lampinen (2020) found that on synthetic datasets where task-relevance of features can be controlled, learned hidden representations suppress task-irrelevant features and enhance task-relevant features. Neyshabur et al. (2020) showed that neural networks trained from pretrained weights stay in the same basin in the loss landscape. In reinforcement learning, Zahavy et al. (2016) explained the success of Deep Q-Networks by visualizing how the learned hidden representations break down the input space in a way that respects the temporal structure of the task. Analyses of NLP models have found linguistic information in the hidden representations after training (Belinkov et al., 2017; Shi et al., 2016; Adi et al., 2017; Qian et al., 2016).

Other works have focused on individual features in the learned representations. Saliency maps and Layer-Wise Relevance Propagation (Simonyan et al., 2014; Zeiler & Fergus, 2014; Bach et al., 2015) that show the sensitivity of the prediction to each unit in the model are popular in Computer Vision

and demonstrate the appearance of useful features like edge or face detectors in neural network. In NLP, Dalvi et al. (2019) studied the relevance of each unit to an external task or the model's own prediction.

5.2 Emergence of Representation Alignment in Hidden Representations

Our goal is to evaluate neural network hidden representations through the lens of representation alignment. Unless otherwise stated, by hidden representation we refer to the representation extracted from the hidden layer closest to the output layer. Throughout this section we will extract different representations and compare them by creating representation alignment curves similar to Figure 2.1.

5.2.1 Experiment Setup

Since our definition requires $\mathbb{E}[y] = 0$, in classification we will focus on balanced binary classification and in regression we will subtract the mean of the targets before the experiment. In regression we will also scale the targets to have $\mathbb{E}[y^2] = 1$. For each extracted representation we will subtract the mean, normalize by the square root of the trace of the covariance matrix, and report representation alignment for a range of thresholds. More rigorously, we will report Alignment$(\bar{\phi}, y, \tau)$ where $\bar{\phi} := (\phi - \mathbb{E}[\phi])/\sqrt{\mathrm{Tr}(\mathbb{E}[(\phi - \mathbb{E}[\phi])(\phi - \mathbb{E}[\phi])^\top])}$. The subtraction ensures $\mathbb{E}[\bar{\phi}] = 0$ and this particular normalization ensures all eigenvalues of $\mathbb{E}[\bar{\phi}^\top \bar{\phi}]$ are in $[0, 1]$. We will estimate all the expectations with large samples since the underlying distribution of (ϕ, y) is inaccessible.

Centering and scaling are common preprocessing steps in training linear models or neural networks (LeCun et al., 2002) so this methodology will not deviate us from the overarching goal of understanding the quality of representations. Meanwhile, as the following proposition shows, the resulting measure is invariant to isotropic scaling and rotation, a desirable property for a measure of representation quality (Kornblith et al., 2019a).

Proposition 5.2.1. *For a constant $c > 0$ and an orthogonal matrix $A \in \mathbb{R}^d$, if $\phi_2 = cA\phi_1$ then Alignment$(\bar{\phi}_2, y, \tau)$ = Alignment$(\bar{\phi}_1, y, \tau)$.*

Proof. Define $\tilde{\phi} := \phi - \mathbb{E}[\phi]$ for brevity. First note

$$\bar{\phi}_2 = \frac{\tilde{\phi}_2}{\sqrt{\mathrm{Tr}(\mathbb{E}[\tilde{\phi}_2\tilde{\phi}_2^\top])}} = \frac{cA\tilde{\phi}_1}{\sqrt{\mathrm{Tr}(c^2A\mathbb{E}[\tilde{\phi}_1\tilde{\phi}_1^\top]A^\top)}} = \frac{A\tilde{\phi}_1}{\sqrt{\mathrm{Tr}(\mathbb{E}[\tilde{\phi}_1\tilde{\phi}_1^\top])}} = A\bar{\phi}_1$$

If $\mathbb{E}[\phi\phi^\top]v = \sigma^2 v$ then $A\mathbb{E}[\phi\phi^\top]A^\top(Av) = A\mathbb{E}[\phi\phi^\top]v = \sigma^2 Av$ and thus Av is an eigenvector of $A\mathbb{E}[\phi\phi^\top]A^\top$ with eigenvalue σ^2. Therefore, $\mathbb{E}[\phi\phi^\top]_\tau$, defined as the truncated covariance matrix with threshold τ, is equal to $A\mathbb{E}[\phi\phi^\top]_\tau A^\top$. Recall the min-norm MSE minimizer for a task (ϕ, y) is equal to $w^* = \mathbb{E}[\phi\phi^\top]^\dagger\mathbb{E}[\phi y]$. Suppose $\mathbb{E}[y^2] = 1$ without loss of generality. Then

$$\begin{aligned}
&\text{Alignment}(\bar{\phi}_2, y, \tau) = \left\|\mathbb{E}[\bar{\phi}_2\bar{\phi}_2^\top]^\dagger\mathbb{E}[\bar{\phi}_2 y]\right\|_{\mathbb{E}[\bar{\phi}_2\bar{\phi}_2^\top]_\tau} \\
&= \mathbb{E}[\bar{\phi}_2 y]^\top\mathbb{E}[\bar{\phi}_2\bar{\phi}_2^\top]^\dagger\mathbb{E}[\bar{\phi}_2\bar{\phi}_2^\top]_\tau\mathbb{E}[\bar{\phi}_2\phi_2^\top]^\dagger\mathbb{E}[\bar{\phi}_2 y] \\
&= \mathbb{E}[\bar{\phi}_1 y]^\top A^\top A\mathbb{E}[\bar{\phi}_1\bar{\phi}_1^\top]^\dagger A^\top A\mathbb{E}[\bar{\phi}_1\bar{\phi}_1^\top]_\tau A^\top A\mathbb{E}[\bar{\phi}_1\phi_1^\top]^\dagger A^\top A\mathbb{E}[\bar{\phi}_1 y] \\
&= \mathbb{E}[\bar{\phi}_1 y]^\top\mathbb{E}[\bar{\phi}_1\bar{\phi}_1^\top]^\dagger\mathbb{E}[\bar{\phi}_1\bar{\phi}_1^\top]_\tau\mathbb{E}[\bar{\phi}_1\phi_1^\top]^\dagger\mathbb{E}[\bar{\phi}_1 y] \\
&= \left\|\mathbb{E}[\bar{\phi}_1\bar{\phi}_1^\top]^\dagger\mathbb{E}[\bar{\phi}_1 y]\right\|_{\mathbb{E}[\bar{\phi}_1\bar{\phi}_1^\top]_\tau} = \text{Alignment}(\bar{\phi}_1, y, \tau)
\end{aligned}$$

□

Using the methodology above we will test the following hypotheses.

H1 On the task used in training a neural network, learned hidden representations have higher representation alignment compared to the input and the initial hidden representation.

H2 In fully-connected networks (FCNs), layers closer to the output have higher representation alignment on the training task.

H3 In object classification and on the downstream task, representations obtained via common neural network feature transfer approaches have higher representation alignment than handcrafted features.

5.2.2 Results

H1: Define $\phi_{\text{init}} : \mathcal{X} \to \mathbb{R}^d$ and $\phi : \mathcal{X} \to \mathbb{R}^d$ as the neural network up to the last hidden layer at initialization and after training on the task (x, y) where $x \in \mathbb{R}^{d_x}$. Strictly, the hypothesis states that $\text{Alignment}(\phi(x), y, \tau) \geq \text{Alignment}(\phi_{\text{init}}(x), y, \tau)$ and that $\text{Alignment}(\phi(x), y, \tau) \geq \text{Alignment}(x, y, \tau)$ with the inequalities being strict for at least one value of τ.

We first tested the hypothesis on three large neural networks trained on a binary classification task from Cifar10. Each neural network is trained on a sample of size 5000 and representation alignment is computed on a separate sample of size 5000. Figure 5.1 shows the results. While there are many curves, often with intersections, in these figures, recall that we only want to answer whether the final representations alignments are larger than the initial values and larger than the input representation alignment. That is, we only need to compare whether (1) each colored solid curve is higher than the dashed curve with the same color, and (2) each colored solid curve is higher than the black curve. The experiment verifies H1 through the whole range of thresholds for two networks and over most of the thresholds for the other network.

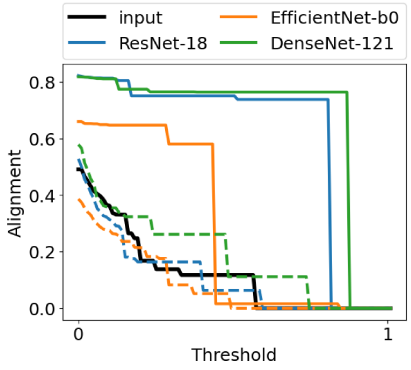
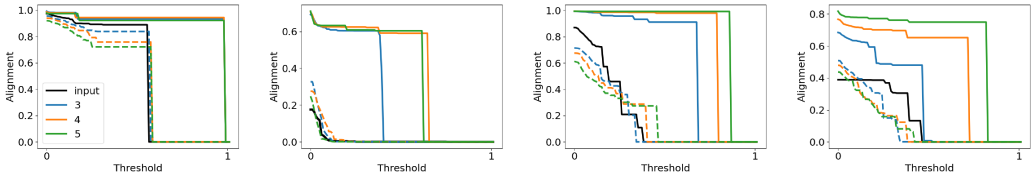


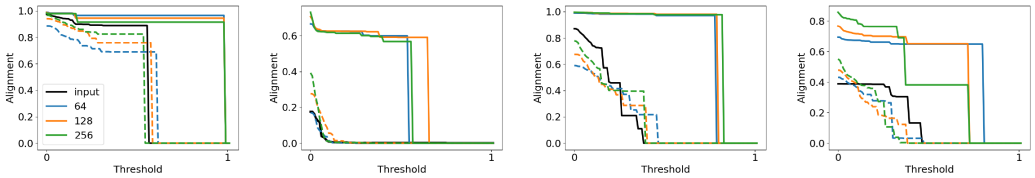
Figure 5.1: Representation alignment curves for CNNs on Cifar10. The curves show representation alignment for input (black curve), initial (dashed colored curves), and trained (solid colored curves) representation. Each solid colored curve is higher than the dashed curve with the same color and higher than the black curve.

We also tested this hypothesis on FCNs with a variety of hyperparameter and architecture choices trained on two binary classification datasets (sampled from multi-class datasets MNIST and SVHN) and two regression datasets (CT Position and Bike Sharing). We picked a default setting of depth: 4, width: 128, activation: ReLU, optimizer: Adam (step-size 1e-3), and mini-batch size: 128, and each time we varied one hyperparameter while keeping the rest to their default values. Figures 5.2 to 5.6 show the results, verifying the hy-



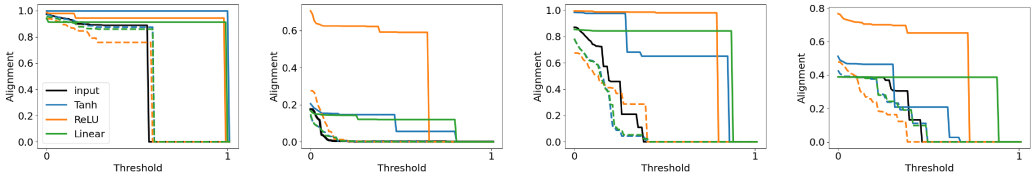
(a) MNIST (b) SVHN (c) CT Position (d) Bike Sharing

Figure 5.2: FCNs with different depths. The curves show representation alignment for input (black curve), initial (dashed colored curves), and trained (solid colored curves) representation. Each solid colored curve is higher than the dashed curve with the same color and higher than the black curve. Deeper networks tend to have lower initial and higher final representation alignment.



(a) MNIST (b) SVHN (c) CT Position (d) Bike Sharing

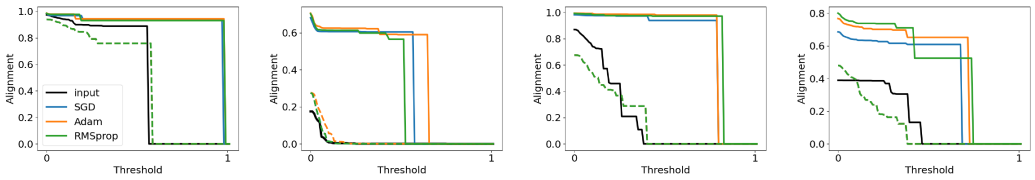
Figure 5.3: FCNs with different widths. The curves show representation alignment for input (black curve), initial (dashed colored curves), and trained (solid colored curves) representation. Each solid colored curve is higher than the dashed curve with the same color and higher than the black curve.



(a) MNIST (b) SVHN (c) CT Position (d) Bike Sharing

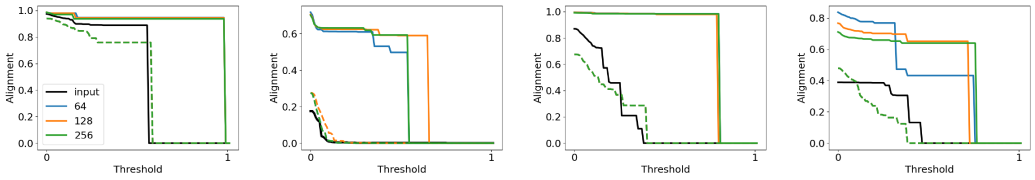
Figure 5.4: FCNs with different activations. The curves show representation alignment for input (black curve), initial (dashed colored curves), and trained (solid colored curves) representation. Each solid colored curve is higher than the dashed curve with the same color and in most cases and across nearly all thresholds, higher than the black curve. The increase in representation alignment even occurs with linear activations.

pothesis again. Each figure shows the effect of changing one hyperparameter on the four tasks. Interestingly, even linear activation increases representation alignment even though this activation cannot raise expressivity or linear separability.



(a) MNIST (b) SVHN (c) CT Position (d) Bike Sharing

Figure 5.5: FCNs with different optimizers. The curves show representation alignment for input (black curve), initial (dashed colored curves), and trained (solid colored curves) representation. The dashed curves fully overlap since the optimizers have no effect at initialization. Each solid colored curve is higher than the dashed curve with the same color and higher than the black curve.

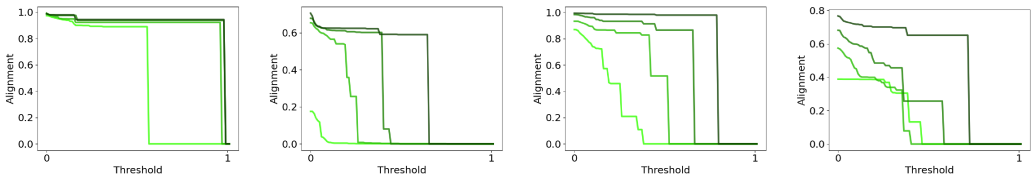


(a) MNIST (b) SVHN (c) CT Position (d) Bike Sharing

Figure 5.6: FCNs with different mini-batch sizes. The curves show representation alignment for input (black curve), initial (dashed colored curves), and trained (solid colored curves) representation. The dashed curves fully overlap since mini-batch size have no effect at initialization. Each solid colored curve is higher than the dashed curve with the same color and higher than the black curve.

H2: Define $\phi_i : \mathcal{X} \to \mathbb{R}^d$ as the neural network up to the i-th hidden layer (ordered from the input layer towards the output layer) after training on the task (x, y). The hypothesis states that $\text{Alignment}(\phi_1(x), y, \tau) \leq \text{Alignment}(\phi_2(x), y, \tau) \leq \text{Alignment}(\phi_3(x), y, \tau) \leq \cdots$ with the inequalities being strict for at least one value of τ.

We tested this hypothesis on FCNs on the four datasets used for the previous hypothesis and the mentioned default hyperparameter values. We have restricted the hypothesis to FCNs since presence of components like residual connections and pooling layers in larger networks would call for a more extensive experiment. As Figure 5.7 shows, in most cases FCNs do raise representation alignment through the layers. Although it is well-known that hidden layers closer to the output tend to learn more linearly separable representa-



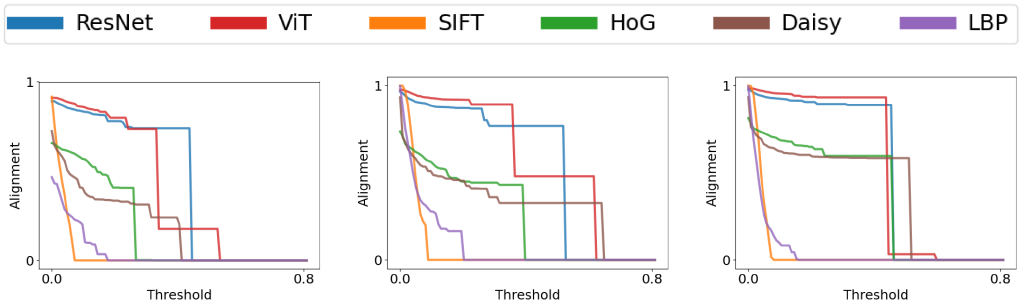
(a) MNIST (b) SVHN (c) CT Position (d) Bike Sharing

Figure 5.7: Representation alignment in different layers of trained FCNs. Each curve shows a hidden layer and darker shades show layers closer to the output. The layers closer to the output have higher representation alignment through most of the thresholds.

tions, the increase in representation alignment is still a new finding. This is because, as we saw in Chapter 3, representation alignment is not a necessary consequence of linear separability.

H3: We now turn to the feature transfer setup and ask if high representation alignment transfers to the related downstream task in a typical feature transfer experiment. Define $\phi : \mathcal{X} \to \mathbb{R}^d$ as the neural network up to the last hidden layer after training on the upstream task (x^U, y^U). We hypothesize that the transferred representation achieves higher representation alignment than common handcrafted features. In other words, Alignment$(\phi(x^D), y^D, \tau)$ as defined in the previous hypothesis is larger than Alignment$(\phi_H(x^D), y^D, \tau)$ for a common handcrafted feature extractor ϕ_H.

We compared the alignment curves for two neural networks against four different handcrafted feature extractors. The downstream task is a binary classification task from Cifar10, Cifar100, or STL10 and the upstream task for the neural networks is ImageNet. We use a sample from the downstream task to compute representation alignment with size 5000 for Cifar10 and 1000 for Cifar100 and STL10. The neural networks are ResNet-18 (He et al., 2016) (called **ResNet** in the results), the vision transformer architecture LeVit (Graham et al., 2021) (which we call **ViT** in the results), **SIFT** (Lowe, 1999) features with dictionary size 2048, Histogram of Gradients (**HoG**) descriptors (Dalal & Triggs, 2005) with 9 orientations, 64 pixels per cell, and 4 cells per block, **Daisy** descriptors (Tola et al., 2009) with 16 histograms and 16 orientations,



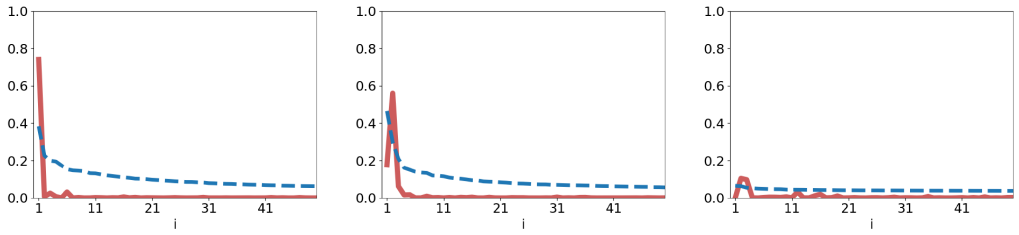
(a) Cifar10 (b) Cifar100 (c) STL10

Figure 5.8: Comparing representation alignment of pre-trained neural networks and handcrafted features. Neural networks have the highest representation alignment through most of the thresholds on all the three datasets.

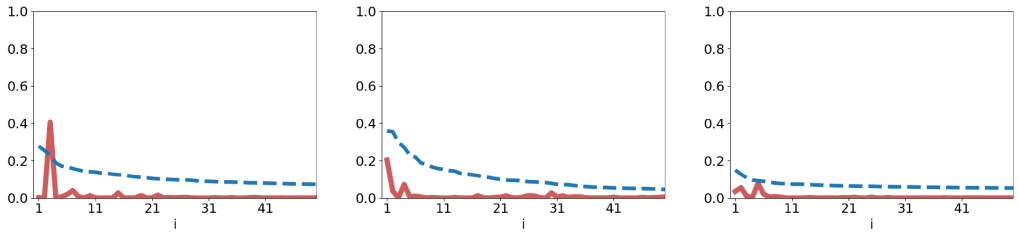
and Local Binary Pattern (**LBP**) (Ojala et al., 2002) with radius 3 and 84 points. These parameters are chosen to achieve high linear separability as we will discuss later.

Figure 5.8 shows the alignment curves. Although the hypothesis is not strictly verified since the curves intersect, we observe that the curves for the two neural networks remain the highest for a large range of thresholds and higher that some curves through all the thresholds. There is a trend through all the three tasks that for most of the thresholds the two neural network representations have the highest representation alignment, followed by the two handcrafted features Daisy and HoG, and then followed by the other two handcrafted features SIFT and LBP, whose curves drop quickly to zero at small thresholds.

Figures 5.9 to 5.11 show a fuller picture of representation alignment by plotting σ_i (dashed blue curve) and $\tilde{w}_i^2 := {w^*}^\top v_i \sigma_i$ (red curve) for $i \in \{1, \cdots, 50\}$. Recall from Chapter 4 that $\text{Alignment}(\phi, y, \tau) = \sum_{\{i:\sigma_i \geq \tau\}} \tilde{w}_i^2$. Therefore, if the red curve is large for small values of i, where σ_i is large, then representation alignment is high. In all the three datasets the red curve shows a large peak at the beginning for the neural network features. Visually comparing these peaks also reveals the same trend as the previous figure as among the handcrafted features, HoG and Daisy have higher initial peaks compared to SIFT and LBP.

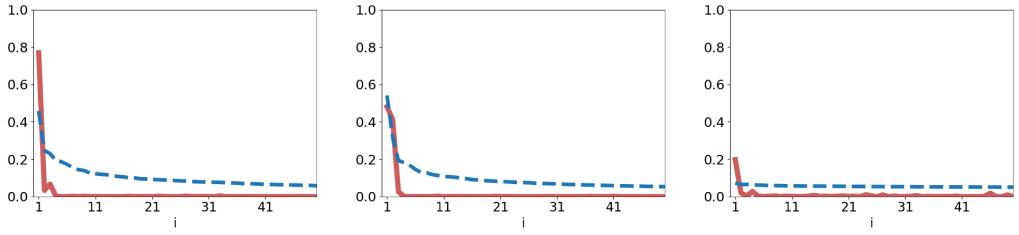


(a) ResNet (b) ViT (c) SIFT

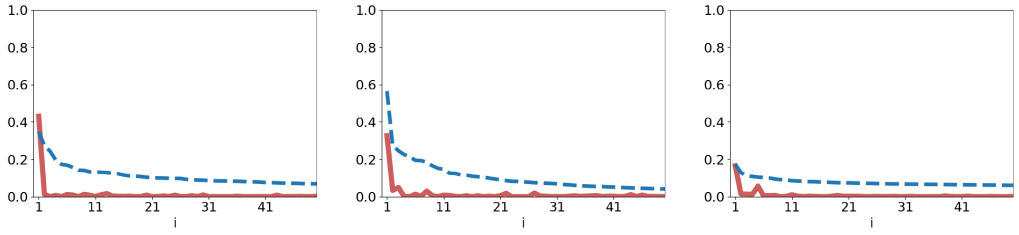


(d) HoG (e) Daisy (f) LBP

Figure 5.9: The first 50 elements of σ_i (blue) and $\tilde{w}_i^2$ (red) on Cifar10.



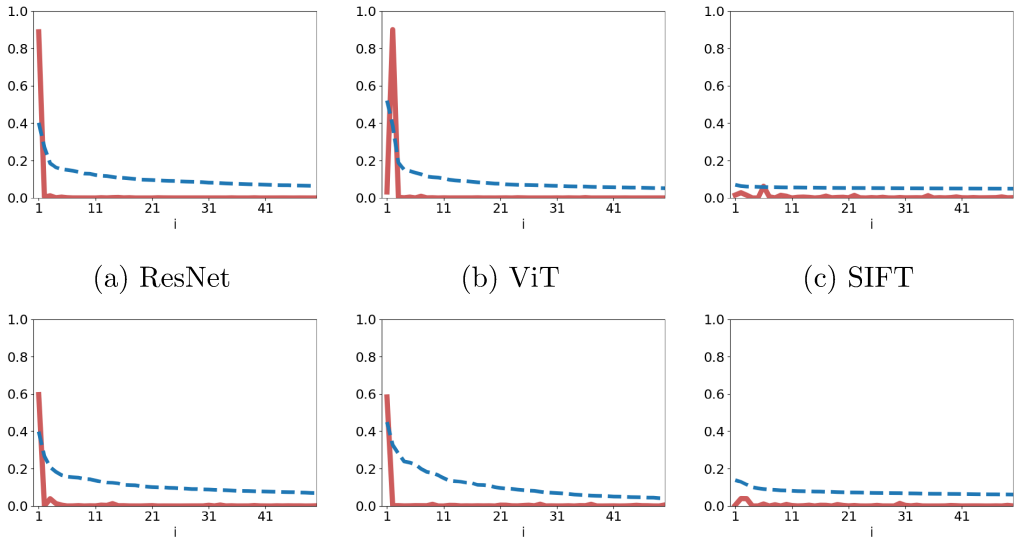
(a) ResNet (b) ViT (c) SIFT



(d) HoG (e) Daisy (f) LBP

Figure 5.10: The first 50 elements of σ_i (blue) and $\tilde{w}_i^2$ (red) on Cifar100.

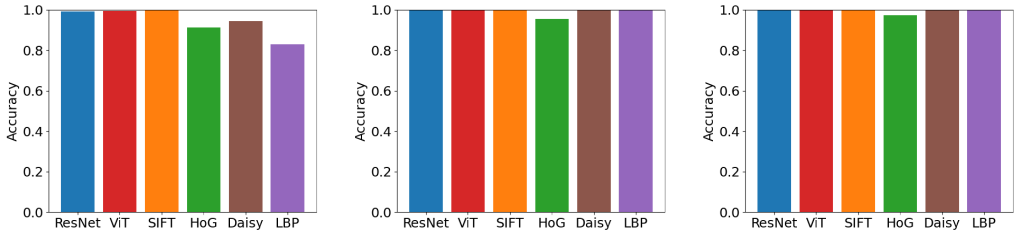
Despite the different degrees of representation alignment, the handcrafted representations are still highly linearly separable. We verified this by computing the 0-1 risk of the closed form min-norm MSE minimizer. The min-norm MSE minimizer achieved a risk of 0 in most cases, and a risk of less than 0.1 in all cases except lbp on Cifar10. This is shown by plotting accuracy (defined as



(a) ResNet (b) ViT (c) SIFT

(d) HoG (e) Daisy (f) LBP

Figure 5.11: The first 50 elements of σ_i (blue) and $\tilde{w}_i^2$ (red) on STL10.



(a) Cifar10 (b) Cifar100 (c) STL10

Figure 5.12: The accuracy of the min-norm MSE minimizer for different representations in object classification. All representations are highly linearly separable and in most representations zero risk is achievable.

one minus the risk) of the min-norm MSE minimizer in Figure 5.12. Although these weights may not be optimal for classification, the aim of these plots is to show that weights with near zero risk exist for most of the representations and to rule out linear separability as a confounding factor in our later results about generalization and optimization.

5.3 Consequences for Generalization and Optimization

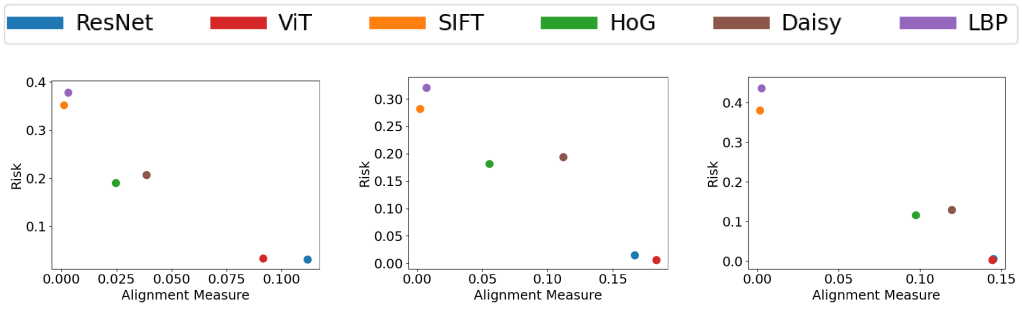
In Chapters 3 and 4 we verified the role of representation alignment on optimization and generalization on simple synthetic experiments. In this chapter, now that we have representations with different degrees of representation alignment, we can ask if this measure is correlated with better generalization (**H4**) and faster optimization (**H5**).

H4: We ask whether the relationship between representation alignment and generalization extends to a realistic scenario. The question is if a typical approach to classification can learn a better estimator using a small sample if representation alignment is high.

We use the same sample as the previous experiment. A subset of size 900 is put aside as the test sample. Then we pick a train sample of size 100, train a linear model on it, and report the risk on the test sample. We use Adam optimizer (Kingma & Ba, 2014) with ℓ_2 regularization weight 1e-4, 1000 iterations and step-size 1e-3. This setting ensures that all linear models achieve a final risk of less than 0.05 on the training sample and therefore possible differences in optimization rate are not confounding the results. Since performance with such small training samples can be prone to variance, we repeat the experiment 10 times, with the train and test subset are picked randomly each time, and report the average risk.

Since the representation alignment curves intersected in the previous experiment, we opt for the scalar measure $\mathbb{E}[KK_y]$ as defined in Chapter 3 and motivated by Corollary 3.2.6. Figure 5.13 shows the results. The horizontal axis is the measure of representation alignment and the vertical axis is the 0-1 risk on the test sample. There is a clear positive relationship between the measure of representation alignment and generalization performance. Recall that the differences can be neither explained by linear separability (since weights with near-zero risk exist) nor difficulties in optimization (since the risk on the train sample is near zero). Also, similar to the trend we saw in the repre-

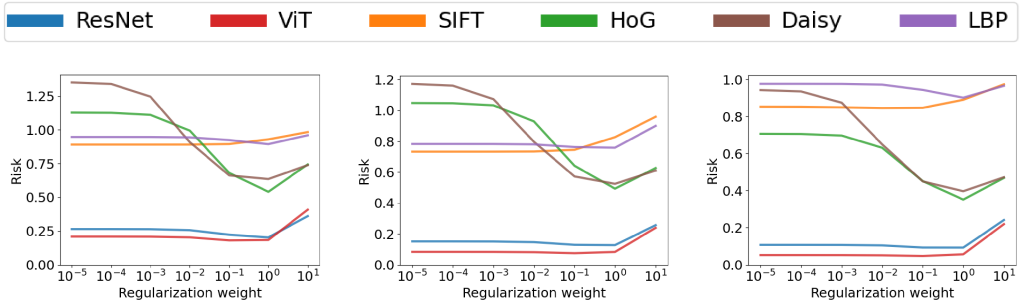
sentation alignment curves for most thresholds, the alignment measure and the generalization performance is generally highest for the two neural network features, followed by HoG and Daisy, followed by SIFT and LBP.



(a) Cifar10 (b) Cifar100 (c) STL10

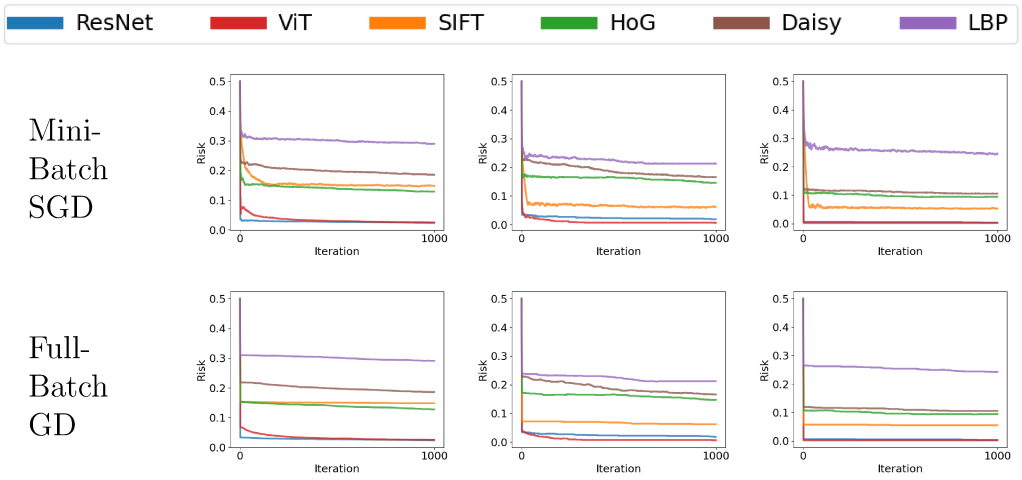
Figure 5.13: Test risk against a scalar measure of representation alignment. Neural network representations tend to have the highest representation alignment according to this measure and there is a clear negative correlation between the risk and representation alignment. Standard errors of the risk across the 10 runs are too small to be seen.

We will also treat these tasks as regression tasks and compute ridge estimators (defined in Chapter 3) and report the risk for different regularization weights. Note that in the context of regression the risk is MSE. Figure 5.14 shows the results. We see that the best performance across regularization weights is lowest for neural network representations, followed by HoG and Daisy, followed by SIFT and LBP.



(a) Cifar10 (b) Cifar100 (c) STL10

Figure 5.14: Regression test risk (MSE) for ridge estimator with different regularization weights. Neural network representation have the lowest risk and SIFT and LBP have the highest.



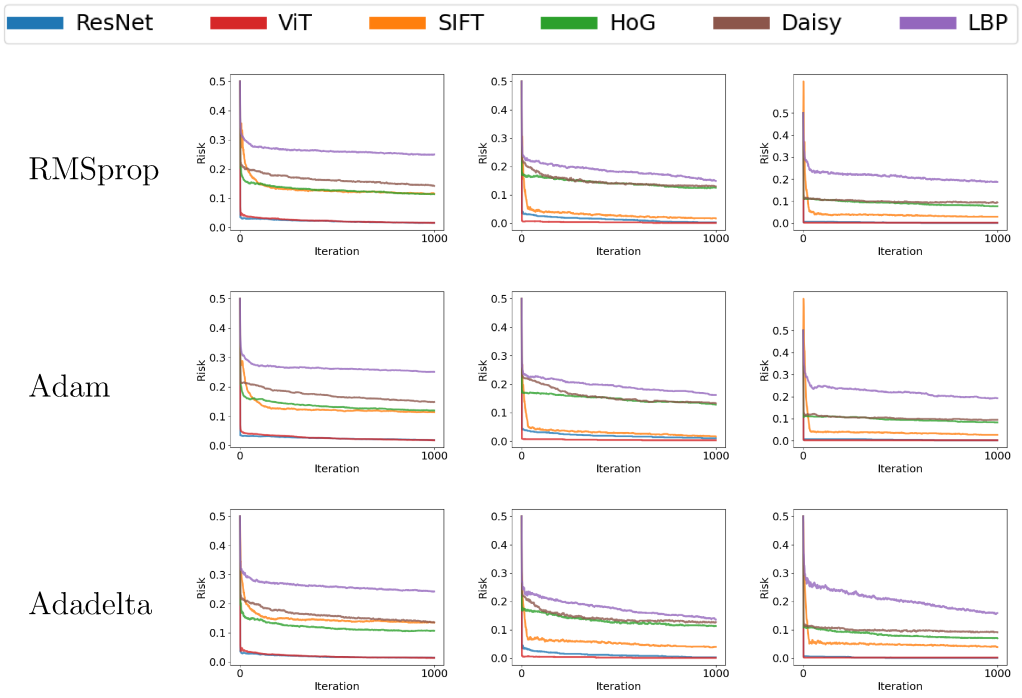
(a) Cifar10 (b) Cifar100 (c) STL10

Figure 5.15: Learning curves for 0-1 risk of mini-batch SGD (top row) and full-batch GD trained with logistic loss on different representations. The curves for neural networks drop to values near zero quickly while the other curves slow down at higher values.

H5: The final hypothesis is if the insight from Proposition 4.1.2 extends to a more practical setup. Specifically the question is whether a variety of common mini-batch optimization algorithms can reduce a large amount of risk in a short time if representation alignment is high for a high threshold.

Let us start with training a linear model with mini-batch SGD with step-size 0.1 and mini-batch size 32 and logistic loss and ask if it can reduce the 0-1 risk fast in case of high representation alignment. The mini-batch algorithm is the *shuffle* variant as described by Bottou (2009). We will compare optimization on the representations in the previous experiment but use the whole sample of 5000 items on Cifar10 and 1000 items on Cifar100 and STL10 and evaluate the risk on the same sample through the iterations.

Figure 5.15 (top row) shows the results on Cifar10, Cifar100, and STL10. The risk for neural network representations immediately drops to near zero while the other curves slow down at higher values with some curves appearing to have leveled off. The differences cannot be explained by linear separability. As we discussed before, in all the cases there exist weights with much lower risks. The high risk of some curves cannot be fully explained by the noise



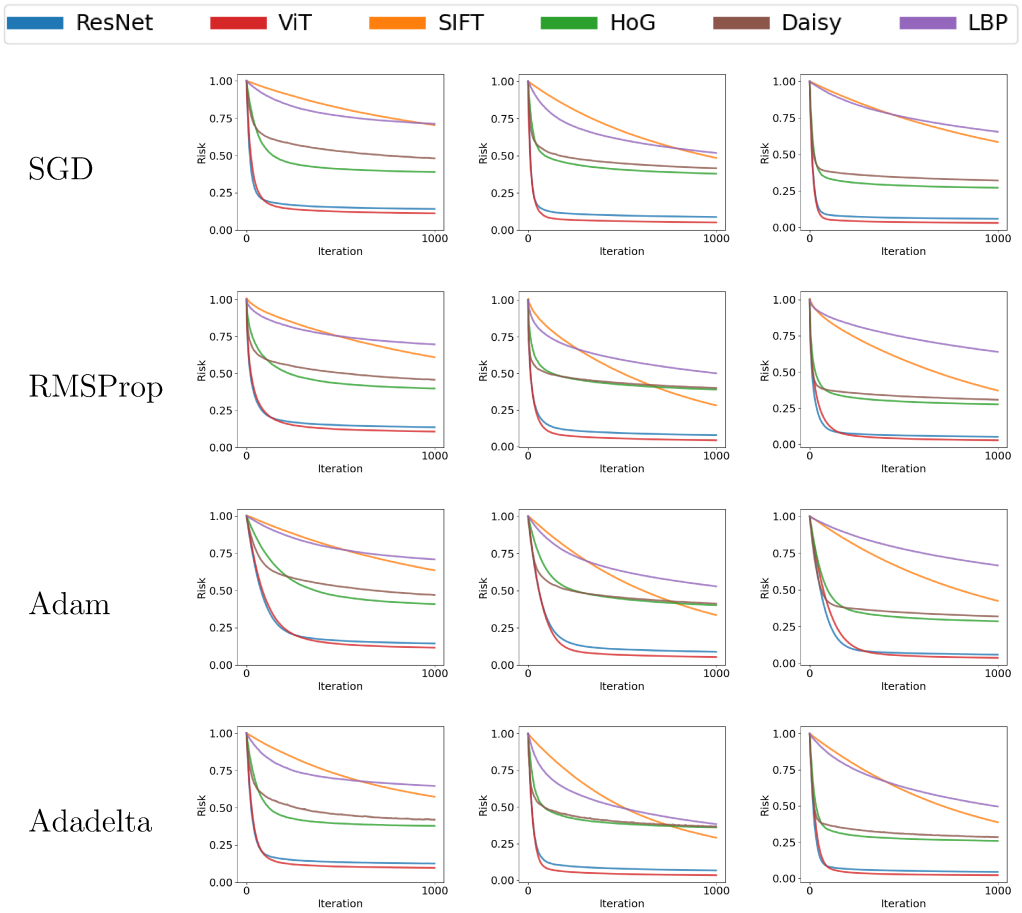
(a) Cifar10 (b) Cifar100 (c) STL10

Figure 5.16: Learning curves for 0-1 risk of mini-batch RMSprop (top row), Adam (middle row), and Adadelta (bottom row) trained with logistic loss on different representations. The curves for neural networks drop to values near zero quickly while the other curves, aside from SIFT on Cifar100, slow down at higher values.

in the mini-batches either. In Figure 5.15 (bottom row) we repeat the same experiment with (full-batch) GD and observe the same pattern.

We conduct a similar experiment with three other common optimizers, RMSprop (step-size 1e-3) (Hinton et al., 2012), Adam (step-size 1e-3) (Kingma & Ba, 2014), and Adadelta (step-size 1.0) (Zeiler, 2012), all with mini-batch size 32. The results are in Figure 5.16. Again we see that only the curves for neural network representations consistently and immediately drop to values near zero.

Similar to the previous hypothesis, we ask the same question in the context of regression by treating the tasks as regression tasks, training the linear models using the MSE loss, and reporting MSE as the risk. The optimizers and the step-sizes are similar to the classification experiment, all with mini-



(a) Cifar10 (b) Cifar100 (c) STL10

Figure 5.17: MSE Learning curves for mini-batch SGD (1st. row), RMSprop (2nd. row), Adam (3rd. row), and Adadelta (4th. row) trained with MSE loss on different representations. Neural network representations have the fastest optimization rate and SIFT and LBP have the lowest.

batch size 32. Figure 5.17 shows the learning curves. Once again, generally the neural network representations show the fastest optimization rates. The trend in optimization rate here mirrors the representation alignment curves where we saw neural network representations had the highest representation alignment followed by HoG and Daisy, followed by SIFT and LBP.

5.4 Discussion

This chapter studied neural network emergent representations using representation alignment. The observations can be summarized in this list:

1. With a wide variety of architectures and hyperparameters, neural networks learn hidden representation with higher representation alignment on the training task compared to the input or the initial hidden representation.

2. In FCNs, hidden layers close to the output tend to have higher representation alignment compared to the ones close to the input for a wide range of thresholds.

3. The transferred representation in object classification has higher representation alignment compared to common handcrafted features for a wide range of thresholds.

4. The generalization performance from a small sample in a common approach to classification and regression mirrors the degree of representation alignment.

5. The rate of convergence of several common optimizers mirrors the degree of representation alignment.

These observations prepare the ground the next two chapters. In the next chapter, we use the observation (3) above regarding high representation alignment of neural network features as prior knowledge for domain adaptation. In the chapter following that, we will use observations (5,6) and hypothesize that a different algorithm (Temporal-Difference learning) will also find generalizable solutions from a small sample when its convergence is fast in expectation.

Chapter 6

Label Alignment Regularization for Distribution Shift

Previously we observed that transferred neural network representations tend to have high representation alignment. In this chapter we use this prior knowledge and propose a regularizer for domain adaptation and find that enforcing alignment between the predictions and the given representation can help in domain adaptation. Unlike conventional domain adaptation approaches that focus on regularizing representations, we instead regularize the classifier to align with the unsupervised target data, guided by the label alignment property (closely related to representation alignment) in both the source and target domains. Theoretical analysis demonstrates that, under certain assumptions, our solution resides within the span of the top right singular vectors of the target domain data and aligns with the optimal solution. By removing the reliance on the commonly used optimal joint risk assumption found in classic domain adaptation theory, we showcase the effectiveness of our method on addressing problems where traditional domain adaptation methods often fall short due to high joint error. Additionally, we report improved performance over domain adaptation baselines in well-known tasks such as MNIST-USPS domain adaptation and cross-lingual sentiment analysis.

6.1 Background

Unsupervised domain adaptation studies knowledge transfer from a source domain with labeled data to a target domain with unlabeled data where the model will be deployed and evaluated (Ben-David et al., 2010; Mansour et al., 2009). This difference between the two domains, called domain shift, arises in many applications. A document classification or sentiment analysis model for an under-resourced language can benefit from a large corpus for a different language. A personal healthcare system is often trained on a group of users different from its target users. A real-world robot's predictions or decision-making can improve through safe and less costly interactions with a simulator (Pires et al., 2019; Ganin et al., 2016; Peng et al., 2018).

There are diverse settings to study domain adaptation problems. In classification problems, closed set domain adaptation assumes the same categories between the two domains while open-set domain adaptation assumes that the two domains only share a subset of their categories (Panareda Busto & Gall, 2017). Unsupervised, semi-supervised, and supervised domain adaptation assume that the data from the target domain is fully unlabeled, partly labeled, and fully labeled respectively (Ganin et al., 2016). Two related problems to domain adaptation are multi-target domain adaptation where there are multiple target domains (Gholami et al., 2020) and domain generalization where several source domains are sampled from a distribution over tasks and the goal is to generalize to a previously unseen domain from this distribution (Blanchard et al., 2011; Gulrajani & Lopez-Paz, 2021). Within these diverse settings, our work specifically addresses unsupervised domain shift problems.

The prevalence of domain shift in machine learning has inspired a large body of algorithmic and theoretical research on domain adaptation. Ben-David et al. (2010) and Y. Zhang et al. (2019) formulated the difference between the source and the target domain with the notion of $\mathcal{H}$-divergence and Margin Disparity Discrepancy and provided generalization bounds that relate performance on the two domains. Acuna et al. (2021) extended these results to a more general notion of f-divergence. Adversarial domain adaptation

algorithms are motivated by these theoretical findings and aim to learn representations that achieve high performance in the source domain while being invariant to the shift between the source and the target domain (Ganin et al., 2016; Y. Zhang et al., 2017; Conneau et al., 2018; Long et al., 2015; Pei et al., 2018).

The aforementioned representation-matching approach assumes that the optimal joint risk between the source and target is small. This assumption fails when the conditional distribution of the labels given input is different between source and target domains. An example occurs when labels in the source domain are much more imbalanced than in the target domain. For instance, Zhao et al. (2019) identified that under such label distribution shift, the optimal joint risk can be quite large and they empirically show the failure of domain adaptation methods on MNIST-USPS digit datasets. Johansson et al. (2019) also pointed out the limitation of matching feature representations by showing its inconsistency, and thus the tendency for high target errors.

In this work, we adopt a novel approach to domain adaptation that focuses on label alignment, defined as the alignment of labels with the top left singular vectors of the representation. Instead of striving for an invariant representation, our proposed algorithm fine-tunes the classifier for the target domain. It achieves this by removing the influence of label alignment in the source domain and applying this alignment principle to the target domain. A critical distinction of our approach from existing methodologies is that we adjust the classifier's weight rather than its representation. Consequently, our method can be applied in settings with linear function approximation and may complement existing approaches.

We describe the label alignment phenomenon in Section 6.2, and outline the proposed method in Section 6.3. Section 6.4 formally justifies our regularization method by showing that it projects the solution onto the span of the top right singular vectors of the target domain. Section 6.5 reviews related work. In Section 6.6, we first provide a synthetic example where the proposed regularizer shows a clear advantage. We then experiment with imbalanced MNIST-USPS binary classification tasks and find that our method, unlike the

domain-adversarial baseline, is robust to imbalance in one domain. Finally, we evaluate our algorithm on cross-lingual sentiment analysis tasks and observe improved F_1 score on training with our regularization, compared to adversarial domain adaptation baselines.

6.2 Label Alignment

In this section, we briefly review the standard linear regression problem and define relevant notations to explain the *label alignment property* (LAP, Imani et al., 2022).

6.2.1 Linear Regression and Notations

We consider a dataset with n samples, (possibly learned and nonlinear) representation matrix $\Phi \in \mathbb{R}^{n \times d}$ and label vector $y \in \mathbb{R}^n$ from a source domain. Denote the model's weights as $w \in \mathbb{R}^d$, we study the linear regression problem:

$$\min_w \|\Phi w - y\|^2 \tag{6.1}$$

Without loss of generality, we replace the bias unit with a constant feature in the representation matrix to avoid studying the unit separately. The model will be evaluated on a test set sampled from the target domain.

The singular value decomposition (SVD) of a representation matrix Φ is $\Phi = U\Sigma V^\top = \sum_{i=1}^{d} \sigma_i u_i v_i^\top$, where

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_d \\ & \mathbf{0} & \end{bmatrix} \in \mathbb{R}^{n \times d}$$

is a rectangular diagonal matrix whose main diagonal consists of singular values $\sigma_1, \cdots, \sigma_d$ in descending order with the remaining rows set to zero, and

$$U = [u_1, \ldots, u_n] \in \mathbb{R}^{n \times n} \text{ and } V = [v_1, \cdots, v_d] \in \mathbb{R}^{d \times d}$$

are orthogonal matrices whose columns $u_i \in \mathbb{R}^n$ and $v_j \in \mathbb{R}^d$ are the corresponding left and right singular vectors. In principal component analysis

(Pearson, 1901), $v_1, \cdots, v_k$ are also known as the first k principal components. For a vector a and orthonormal basis B, a^B is a shorthand for $B^\top a$, the representation of a in terms of the row vectors of B. We use $r(\cdot)$ to denote the rank of a matrix.

6.2.2 Definition of Label Alignment

Label alignment is specified in terms of the singular vectors of Φ and label vector $\mathbf{y}$. The left singular vectors of Φ, $\{u_1, \cdots, u_n\}$ form an orthonormal basis that spans the n-dimensional space. The label vector $y \in \mathbb{R}^n$ can be decomposed in this basis with:

$$y = Uy^U = y_1^U u_1 + \cdots + y_n^U u_n, \tag{6.2}$$

where y_i^U is the i^{th} component of vector $y^U \in \mathbb{R}^n$.

Label alignment (Imani et al., 2022) is a relationship between the labels and the representation where the variation in the labels are mostly along the top principal components of the representation. For our purpose we give the following definition and verify that it approximately holds in a number of real-world tasks. A dataset has *label alignment* with rank k if for $k \ll r(\Phi)$ we have $y_i^U = 0, \forall i \in \{k+1, ..., d\}$.

6.2.3 Emergence of Label Alignment in Realistic Tasks

We will investigate this property in binary classification tasks (with ± 1 labels) and regression tasks by reporting $k(\epsilon)$, defined as the smallest k where

$$\sqrt{\sum_{i=k+1}^{d} (y_i^U)^2} < \epsilon \sqrt{\sum_{i=1}^{d} (y_i^U)^2}.$$

If $k(\epsilon)$ is small for a small ϵ then the projection of the label vector on the span of Φ is mostly in the span of the first few singular vectors. The details of the tasks are as follows:

UCI CT Scan: A random subset of the CT Position dataset on UCI (Graf et al., 2011). The task is predicting a location of a CT Slice from histogram features.

Song Year: A random subset of the training portion of the Million Song dataset (Bertin-Mahieux et al., 2011). The task is predicting the release year of a song from audio features.

Bike Sharing: A random subset of the Bike Sharing dataset on UCI (Fanaee-T & Gama, 2014). The task is predicting the number of rented bikes in an hour based on information about weather, date, and time.

MNIST: The task is classifying digits 0 and 1 in MNIST. ($n = 12665$)

USPS: The task is classifying digits 0 and 1 in USPS. ($n = 2199$)

CIFAR-10: The task is classifying airplane and automobile in CIFAR-10 dataset using features from a ResNet-18 pretrained on ImageNet. ($n = 10000$)

CIFAR-100: The task is classifying beaver and dolphin in CIFAR-100 dataset using features from a ResNet-18 pretrained on ImageNet. ($n = 1000$)

STL-10: The task is classifying airplane and bird in STL-10 dataset using features from a ResNet-18 pretrained on ImageNet. ($n = 1000$)

XED (English): The English corpus from XED datsets whose details are discussed in the main paper. The features are sentence embeddings extracted from BERT. ($n = 6525$)

AG News: A random subset of the first two classes (World and Sports) in AG News document classification dataset. The features are obtained by feeding the document text to BERT. ($n = 10000$)

All datasets have an extra constant 1 feature to account for the bias unit. Rank is computed as the number of singular values larger than $\sigma_1 * \max(n, d) * 1.19209e-07$. This is the default numerical rank computation method in the Numpy package.

In Table 6.1 we see that in all the ten tasks less than half the singular vectors with nonzero singular values already span $\geq 90\%$ of the norm of the projection of $\mathbf{y}$ on the span of Φ. The number $k(0.1)$ is remarkably small, less than 10, in seven out of the ten tasks.

Similar patterns have been also observed in a deep learning setting. Recent work in the Neural Tangent Kernel (NTK) literature has observed that in com-

Task	d	$r(\Phi)$	$k(0.1)$	Task	d	$r(\Phi)$	$k(0.1)$
CT Scan	385	372	12	CIFAR-10	513	513	7
Song Year	91	91	6	CIFAR-100	513	513	7
Bike Sharing	13	13	4	STL-10	513	513	2
MNIST	785	580	2	XED (En)	769	769	231
USPS	257	257	2	AG News	769	769	40

Table 6.1: Label alignment in real-world tasks. The table on the left uses the original features in the dataset and the table on the right uses features extracted from neural networks. CT Scan, Song Year, and Bike Sharing are regression tasks and the rest are binary classification. We used the first two classes of multi-class classification datasets to create a binary classification task. In all of these tasks, a large portion of the label vector is in the span of a relatively small set of top singular vectors (compared to the rank).

mon datasets the label vector is largely within the span of the top eigenvectors of the NTK Gram matrix (Arora et al., 2019). In contrast, a randomized label vector would be more or less uniformly aligned with all eigenvectors. More recently, Baratin et al. (2021) and Ortiz-Jiménez et al. (2021) noted that training a finite-width NN makes the alignment between the network's kernel and the task even stronger. Imani et al. (2022) observed a similar behavior in NN hidden representations, indicating that training the NN aligns the top singular vectors of the hidden representations to the task.

6.2.4 Emergence of Label Alignment in a Controlled Setting

We can also easily show emergence of this property in a basic setting where a large number of features are correlated with the labels. The following lemma is needed for the proof.

Lemma 6.2.1. *If there are $k' < d$ orthonormal vectors $\{\nu_1, \cdots, \nu_{k'}\}$ such that $\|\Phi\nu_i\| < \epsilon$ for all $i \in [k']$ then $\Phi_{n\times d}$ has at most $d - k'$ singular values greater than or equal to $\sqrt{k'}\epsilon$.*

Proof. Suppose $\sigma_1, \cdots, \sigma_d$ are the singular values of Φ sorted in descending order. The matrix $N_{d\times k'}$ with orthonormal columns that minimizes $\|\Phi N\|_2$ is the matrix of the last k' right singular vectors, and $\|\Phi N\|_2 = \sqrt{\sum_{i=d-k'+1}^{d} \sigma_i^2} \geq$

$\sigma_{d-k'+1}$ (This easily follows from Section 12.1.2 by Bishop (2006)). If $\sigma_{d-k'+1} \geq \sqrt{k'}\epsilon$ then for any N with orthonormal columns we have $\|\Phi N\|_2 \geq \sqrt{k'}\epsilon \implies \|\Phi N\|_\infty \geq \epsilon$ which contradicts the assumption. □

The following proposition shows emergence of label alignment when a large number of features are highly correlated with the labels.

Proposition 6.2.2. *Suppose* $\|\mathbf{y}\| = 1$ *and that columns of* Φ *are normalized. If* $\Phi_{n\times d}$ *has* $\hat{k} \leq d$ *columns* $\{\phi_1, \cdots, \phi_{\hat{k}}\}$ *where* $|\phi_i^\top \mathbf{y}| > 1 - \delta$ *for all* $i \in [\hat{k}]$ *and*

- $0 < \delta < 0.2$
- $\hat{k} > 16\delta^2/(-15\delta^2 - 2\delta + 1)$
- $d > 16\delta^2(\hat{k} - 1)$

then the norm of the projection of y *on the span of the first* $k = d - \hat{k} + 1$ *left singular vectors of* Φ *is greater than*

$$\sqrt{\frac{\hat{k}(1-\delta)^2 - 16\delta^2(\hat{k}-1)}{d - 16\delta^2(\hat{k}-1)}}.$$

Proof. First suppose the dot products in the statement are positive.

Note that for all $i \in \hat{k}$ we have $\|\phi_i - \mathbf{y}\|_2^2 = (\phi_i - \mathbf{y})^\top(\phi_i - \mathbf{y}) = \phi_i^\top\phi_i + \mathbf{y}^\top\mathbf{y} - 2\phi_i^\top\mathbf{y} = 2 - 2\phi_i^\top\mathbf{y} < 2\delta$. Due to triangle inequality, $\|\phi_i - \phi_j\|_2^2 \leq \|\phi_i - y\|_2^2 + \|\phi_j - y\|_2^2 < 4\delta$.

The span of $\phi_{\hat{k}}, \phi_{\hat{k}+1}, \cdots \phi_d$ has at most $d - \hat{k} + 1$ dimensions. Choose $\hat{k} - 1$ orthonormal vectors $\nu_1, \cdots, \nu_{\hat{k}-1} \in \mathbb{R}^d$ that are perpendicular to this subspace. Then for any $i, j \in [\hat{k} - 1]$ we have $\phi_i^\top\nu_j = (\phi_i - \phi_{\hat{k}} + \phi_{\hat{k}})^\top\nu_j = (\phi_i - \phi_{\hat{k}})^\top\nu_j + 0 \leq \|\phi_i - \phi_{\hat{k}}\| < 4\delta$. Therefore $\|\Phi\nu_j\| < 4\delta\sqrt{\hat{k} - 1}$. Putting this orthonormal basis in the lemma above gives that Φ has at most $d - \hat{k} + 1$ singular values greater than or equal to $4\delta(\hat{k} - 1)$.

Now see that $\left\|\Phi^\top\mathbf{y}\right\|^2 = \left\|\sum_{i=1}^d \phi_i^\top\mathbf{y}\right\|^2$ and is also equal to $\sum_{i=1}^d(\sigma_i\mathbf{y}_i^U)^2$. Therefore $\sum_{i=1}^d(\sigma_i\mathbf{y}_i^U)^2 \geq \left\|\sum_{i=1}^{\hat{k}} \phi_i^\top\mathbf{y}\right\|^2 > \hat{k}(1-\delta)^2$. Since the columns are normalized, $\sum_{i=1}^d \sigma_i^2 = \|\Phi\|_F = d$. In addition, we have shown that the last $\hat{k} - 1$ singular values are smaller than $4\delta\sqrt{\hat{k} - 1}$. Define $\hat{\mathbf{y}}$ as the projection

of $\mathbf{y}$ on the first $d - \hat{k} + 1$ singular vectors of Φ. Then we have $\hat{k}(1-\delta)^2 < \sum_{i=1}^{d}(\sigma_i \mathbf{y}_i^U)^2 = \sum_{i=1}^{d-\hat{k}+1}(\sigma_i \mathbf{y}_i^U)^2 + \sum_{i=d-\hat{k}+2}^{d}(\sigma_i \mathbf{y}_i^U)^2 < d\,\|\hat{y}\|^2 + 16\delta^2(\hat{k}-1)(1 - \|\hat{y}\|^2)$. Rearranging the terms (with the extra conditions in the proposition statement) gives

$$\|\hat{y}\| > \sqrt{\frac{\hat{k}(1-\delta)^2 - 16\delta^2(\hat{k}-1)}{d - 16\delta^2(\hat{k}-1)}}$$

The inequality is tight in the extreme case where $\hat{k} = d$ and $\delta \to 0$ which results in the label vector being fully in the direction of the first left singular vector and all the other singular values tending to zero.

Now suppose some of the dot products in the statement are negative. We can multiply those columns with -1 and prove the result above for this modified matrix. The result holds for the original matrix since this operation only changes the right singular vectors of Φ and does not affect the left singular vectors or the singular values. □

Let us now demonstrate the emergence of alignment and the behavior of the bound above when multiple features are highly correlated with the output. In this toy experiment the label vector is sampled from a 1000-dimensional Gaussian distribution $\mathcal{N}(0_d, I)$ with mean zero and standard deviation 1 and then normalized to norm one. The matrix Φ has 10 columns. The first 9 columns are sampled from $\mathcal{N}(y, s^2 I)$ with mean y and a small standard deviation s and the other column is sampled from $\mathcal{N}(0_d, I)$. All columns are then normalized to norm one. Note that the proposition above does not assume Gaussian features. Figure 6.1 shows the norm of the projection of the label vector on the first two singular vectors at different levels of s and its relationship with δ.

6.3 Proposed Method

This section describes our approach to domain adaptation by enforcing the LAP.

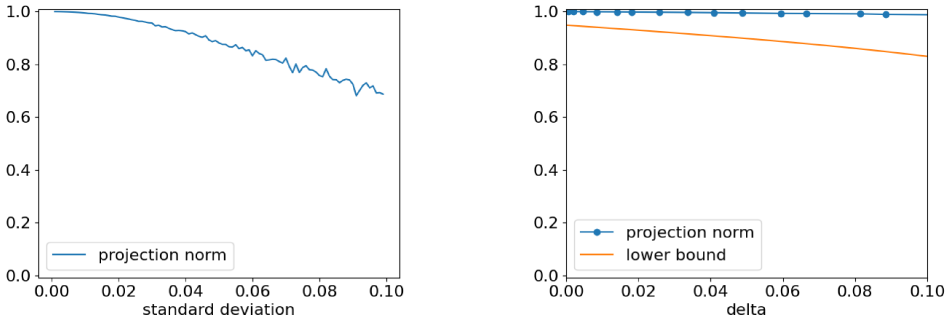


Figure 6.1: Projection of the label vector on the top two singular vectors in the Gaussian example. For small values of standard deviation (where the labels are highly correlated with the features) and small values of δ, the label vector is mostly in the direction of the top two singular vectors. The lower bound is applicable in this regime and is close to one.

6.3.1 Reformulating the Regression Objective

We describe how to reformulate the linear regression objective function with the label alignment property. This reformulation shows that the linear regression objective is implicitly enforcing the LAP on the source domain (i.e., the training data) and this encourages us to further derive our domain adaptation regularization on the target domain.

Objective (6.1) can be rewritten by the following steps.

$$\begin{aligned}
\min_w \|\Phi w - y\|^2 &= \min_w \|U\Sigma V^\top w - y\|^2 \\
&= \min_w \|\Sigma V^\top w - U^\top y\|^2 \\
&= \min_w \|\Sigma w^V - y^U\|^2, \text{shorthand notation} \\
&= \min_w \sum_{i=1}^{d} (\sigma_i w_i^V - y_i^U)^2 + \sum_{i=d+1}^{n} (y_i^U)^2. \qquad (6.3)
\end{aligned}$$

In the first line, since U is an orthogonal matrix, we have $UU^\top = I$ and $\|Ux\| = \|x\|$ for any vector x. Note that the last term $\sum_{i=d+1}^{n}(y_i^U)^2$ can be dropped as it is a constant and does not affect the optimization.

Assume the LAP holds for the first $k < d$ singular vectors. Then $y_i^U =$

$0, \forall i \in k+1, ..., d$. Hence the first term in (6.3) can be further decomposed to

$$\sum_{i=1}^{d}(\sigma_i w_i^V - y_i^U)^2 = \sum_{i=1}^{k}(\sigma_i w_i^V - y_i^U)^2 + \sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2.$$

Plugging this decomposition into the above objective (6.3) and dropping the last term, we get

$$\min_w \sum_{i=1}^{k}(\sigma_i w_i^V - y_i^U)^2 + \sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2. \tag{6.4}$$

We can interpret the first term in the rewritten objective (6.4) as linear regression on a smaller subspace and the second term as a regularization term implicitly enforcing label alignment property on the training data (Φ, y).

The latter is because minimizing the second term has the effect of regularizing the predictions so they likely align with the top singular vectors. This is because:

$$\mathbf{y} = \Phi w = U \Sigma V^\top w$$

and therefore $U^\top \mathbf{y} = \Sigma V^\top w$, which can be written as

$$y^U = \Sigma w^V$$

by using the shorthand notations. For the ith component in vector y^U, we have ${u_i}^\top \mathbf{y} = \sigma_i {v_i}^\top w$. Minimizing w_i^V for $i \in \{k+1, \cdots, d\}$ will reduce the corresponding y_i^U and leave y_i^U for those components $i < k+1$. We call the second term $\sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2$ from (6.4) *label alignment regularization*.

The derivation above shows that when minimizing the original mean squared error for linear regression, we implicitly use label alignment regularization on the training data (source domain data). In the next section, we introduce this regularization into the target domain.

6.3.2 Label Alignment Regularization for Domain Adaptation

In unsupervised domain adaptation, we have a labeled dataset (Φ, y) and an unlabeled dataset $\tilde{\Phi}$ with the corresponding label vector $\tilde{y}$ unknown. From

Equation (6.4), we know that enforcing the LAP does not require knowing the labels $\tilde{y}$. This inspires our key idea of improving the generalization on the target domain: we can use the unlabeled part to enforce the LAP.

Using tilde notation for the SVD of $\tilde{\Phi}$ and assuming $(\tilde{\Phi}, \tilde{y})$ satisfies the LAP with rank $\tilde{k}$, we can put together the supervised part of the source domain and unsupervised part of the target domain to form the objective:

$$\min_w \|\Phi w - y\|^2 + \sum_{i=\tilde{k}+1}^{d} \tilde{\sigma}_i^2 (w_i^{\tilde{V}})^2. \tag{6.5}$$

The second term $\sum_{i=\tilde{k}+1}^{d} \tilde{\sigma}_i^2 (w_i^{\tilde{V}})^2$ is the *label alignment regularization on the target domain*. As we explained in the previous section, the first term (i.e. the standard regression part) in the above objective implicitly enforces the LAP (with rank k) on the source domain. If we expand (6.5) by the reformulated linear regression objective (6.4), we have:

$$\min_w \sum_{i=1}^{k} (\sigma_i w_i^V - y_i^U)^2 + \sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2 + \sum_{i=\tilde{k}+1}^{d} \tilde{\sigma}_i^2 (w_i^{\tilde{V}})^2.$$

Therefore, we have actually done the regularization *twice*: one with the source domain and one with the target domain. We explicitly remove the label alignment regularization on the source domain and arrive at the final objective function:

$$\min_w \|\Phi w - y\|^2 - \sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2 + \lambda \sum_{i=\tilde{k}+1}^{d} \tilde{\sigma}_i^2 (w_i^{\tilde{V}})^2. \tag{6.6}$$

Algorithm 1 shows the pseudo-code. The objective to be minimized has three terms and the hyperparameter λ controls the relative importance of the regularizer. As we will show in § 6.4, under certain constraints this hyperparameter does not affect the final solution and only changes the convergence rate. The first term is the loss that uses the labeled data from the source domain. Following the recent evidence on the viability of the squared error loss for classification (Hui & Belkin, 2020), we use the squared error in both regression tasks and binary classification tasks. We use ± 1 labels in binary classification as these labels showed the label alignment property (LAP) in Table 6.1. The second term removes implicit regularization from the source domain. The third

term is the proposed regularizer that uses the unlabeled data from the target domain. The second and third terms serve as a projection onto the orthogonal complement of span($\tilde{v}_{k+1}, \ldots, \tilde{v}_d$), or namely, span($\tilde{v}_1, \ldots, \tilde{v}_k$), which we show in the next section.

Algorithm 1: Label Alignment Regression

Get data Φ, $\mathbf{y}$, $\tilde{\Phi}$, and hyperparameters t, α, k, $\tilde{k}$, λ
Compute covariance matrices $\Phi^\top\Phi$ and $\tilde{\Phi}^\top\tilde{\Phi}$
Perform eigendecomposition of $\Phi^\top\Phi$ and $\tilde{\Phi}^\top\tilde{\Phi}$ to get $\sigma_{k+1:d}$, $\tilde{\sigma}_{\tilde{k}+1:d}$, $\tilde{v}_{k+1:d}$ and $\tilde{v}_{\tilde{k}+1:d}$
Initialize w to zero
for t iterations **do**
  Perform gradient step with respect to $\|\Phi w - y\|^2 - \sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2 + \lambda \sum_{i=\tilde{k}+1}^{d} \tilde{\sigma}_i^2 (w_i^{\tilde{V}})^2$ with step-size α and update w
end for

6.4 Label Alignment Regularization as Projection

In this section, we provide theoretical insight into how the solution acquired by our regularization approach is related to the optimal solution on the target domain. First, we use a simple rotated Gaussian example to illustrate that our label alignment can exactly give the optimal target solution (see also Section 6.6). Second, we generalize our conclusion beyond the Gaussian example and present the main theorem, showing that when $k = \tilde{k}$ and under a weak additional assumption our solution lies in the span of the top few singular vectors of the target domain.

For convenience, we rewrite our objective (6.6) as:

$$\min_w \|\Phi w - y\|^2 - w^\top (S - S_k) w + \lambda w^\top (\tilde{S} - \tilde{S}_{\tilde{k}}) w,$$

where $S = \Phi^\top\Phi$ is the covariance matrix of Φ, S_k is the covariance matrix truncated to rank k and similar notations hold for $\tilde{S}$ and $\tilde{S}_{\tilde{k}}$. Then the optimal solution for this problem is:

$$\widehat{w^*} = (S_k + \lambda(\tilde{S} - \tilde{S}_{\tilde{k}}))^{-1} \Phi^\top y, \tag{6.7}$$

if the matrix $S_k + \lambda(\tilde{S} - \tilde{S}_{\tilde{k}})$ is full rank, which requires $k \geq \tilde{k}$. In practice, we can treat k and $\tilde{k}$ as hyper-parameters and choose them as we wish.

6.4.1 Rotated Gaussian Example

Consider a simple example where the source and target domain data are both two-dimensional Gaussians, but the target domain is acquired by rotating the source domain (Figure 1 provides a concrete example). Denote the following Gaussian distribution as:

$$\mathcal{N}(0, Q) = \frac{1}{2\pi\sqrt{|Q|}} \exp\left(-\frac{1}{2}x^\top Q^{-1}x\right), \tag{6.8}$$

where $Q = P\begin{bmatrix} s_1^2 & 0 \\ 0 & s_2^2 \end{bmatrix}P^\top$, and $P = \begin{bmatrix} p_1 & p_2 \end{bmatrix}$. Here we consider the spectral decomposition of the covariance matrix $Q \in \mathbb{R}^{2\times 2}$ with $s_1 > 0$, $s_2 > 0$. Here $P \in \mathbb{R}^{2\times 2}$ is an orthogonal matrix, and p_1, p_2 are its column vectors. Since $x = PP^\top x = x_1^P p_1 + x_2^P p_2$, we can rewrite the distribution as:

$$\mathcal{N}(0, Q) = \frac{1}{2\pi s_1 s_2} \exp\left(-\frac{1}{2s_1^2}(x_1^P)^2 - \frac{1}{2s_2^2}(x_2^P)^2\right).$$

We further define the conditional distributions as follows:

$$p_{\mathcal{S}}(x|y) = 2\mathcal{N}(0, Q)\mathbb{1}(yx_1^P > 0), \tag{6.9}$$

where $y \in \{1, -1\}$. Similarly, we can define the target distribution by replacing Q, P, s_i, p_i with $\tilde{Q}, \tilde{P}, \tilde{s}_i, \tilde{p}_i$. We now compute different solutions and then compare them. We assume that there is distribution shift and that p_1 is not parallel to $\tilde{p}_2$.

Recall the regression solution on the source domain:

$$w_{\mathcal{S}}^* = (\Phi^\top\Phi)^{-1}\Phi^\top y = S^{-1}\Phi^\top y. \tag{6.10}$$

Assuming that the sample size is large enough.

$$\frac{1}{n}\Phi^\top y \approx \mathbb{E}_{x,y}[xy] = \sqrt{\frac{2}{\pi}}s_1 p_1, \tag{6.11}$$

See C for the proof. Combining this equation with

$$\Phi^\top y = V\Sigma^\top y^U = \sigma_1 y_1^U v_1 + \sigma_2 y_2^U v_2, \tag{6.12}$$

we know that $y_2^U = 0$ if we identify $v_1 = p_1$. In other words, the label alignment property holds on the source domain with rank $k = 1$. The covariance matrix is:

$$\frac{1}{n}\Phi^\top\Phi \approx \mathbb{E}_x[xx^\top] = s_1^2 p_1 p_1^\top + s_2^2 p_2 p_2^\top, \tag{6.13}$$

We can identify $v_i = p_i$, $s_i^2 = \sigma_i^2/n$ from the SVD of Φ. Plugging (6.12) and (6.13) back into (6.10) we get the optimal solution on the source domain:

$$w_{\mathcal{S}}^* = \sqrt{\frac{2}{\pi}}\frac{1}{s_1}v_1, \tag{6.14}$$

which agrees with our intuition that $w_{\mathcal{S}}^*$ should be in the direction with the largest singular value. Similarly, the optimal solution on the target domain is

$$w_{\mathcal{T}}^* = \sqrt{\frac{2}{\pi}}\frac{1}{\tilde{s}_1}\tilde{v}_1, \tag{6.15}$$

where the tilde notations are the same type of variables used on the target domain. According to (6.7), the label alignment solution with the removal of implicit regularization (given that $\tilde{v}_2$ is not parallel to v_1) is:

$$\widehat{w^*} = (S_k + \lambda(\tilde{S} - \tilde{S}_{\tilde{k}}))^{-1}\Phi^\top y \tag{6.16}$$

$$= (s_1^2 v_1 v_1^\top + \lambda\tilde{s}_2^2\tilde{v}_2\tilde{v}_2^\top)^{-1}\sqrt{\frac{2}{\pi}}s_1 v_1, \tag{6.17}$$

To better understand the solution $\widehat{w^*}$, suppose $\lambda = 1$. Then if we replace $\tilde{s}_2$, $\tilde{v}_2$ by s_2, v_2, we obtain $w_{\mathcal{S}}^*$. If we replace s_1, v_1 by $\tilde{s}_1$, $\tilde{v}_1$, we obtain $w_{\mathcal{T}}^*$.

In fact in this example the effect of label alignment regularization is some kind of projection into the space of $\tilde{v}_1$. Regardless of the hyperparameter λ, we always have the following result:

Proposition 6.4.1. *In the example in this section suppose $v_1^\top\tilde{v}_1 \neq 0$. Then the label alignment solution is $\widehat{w^*} = cw_{\mathcal{T}}^*/v_1^\top\tilde{v}_1$ with $c > 0$.*

Proof. We rewrite (6.16) as:

$$(s_1^2 v_1 v_1^\top + \lambda\tilde{s}_2^2\tilde{v}_2\tilde{v}_2^\top)\widehat{w^*} = \sqrt{\frac{2}{\pi}}s_1 v_1. \tag{6.18}$$

Suppose $\widehat{w^*} = w_1\tilde{v}_1 + w_2\tilde{v}_2$ with $w_1 \in \mathbb{R}, w_2 \in \mathbb{R}$, then the equation above becomes:

$$s_1^2 v_1(v_1^\top\widehat{w^*}) + \lambda\tilde{s}_2^2 w_2\tilde{v}_2 = \sqrt{\frac{2}{\pi}}s_1 v_1. \tag{6.19}$$

Apply $v_2^\top$ on both sides we have:

$$\lambda \tilde{s}_2^2 w_2 \tilde{v}_2^\top v_2 = 0. \tag{6.20}$$

Since $\tilde{v}_2$ is not parallel to v_1, we must have $\tilde{v}_2^\top v_2 \neq 0$ and thus $w_2 = 0$. To obtain the exactly value of w_1, solve (6.19) by setting $w_2 = 0$, we have: $s_1^2(v_1^\top \tilde{v}_1)w_1 = \sqrt{\frac{2}{\pi}} s_1$, $w_1 = \sqrt{\frac{2}{\pi}} \frac{1}{s_1 v_1^\top \tilde{v}_1}$. □

The proposition shows that given $v_1^\top \tilde{v}_1 > 0$, our solution $\widehat{w^*}$ is exactly in the same direction as the optimal solution $w_{\mathcal{T}}^*$, which is verified in our experiments (Section 6.6.1). The above discussion also holds in a more generalized setting, as we show below.

6.4.2 Generalized Setting

This section derives the relation between the solutions $\widehat{w^*}$ and $w_{\mathcal{T}}^*$ in a more general setting, where x is high dimensional, and $k, \tilde{k}$ can be larger than one. We can rewrite

$$w_{\mathcal{S}}^* = \sum_{i \leq k} \sigma_i^{-1} y_i^U v_i,\ w_{\mathcal{T}}^* = \sum_{i \leq \tilde{k}} \tilde{\sigma}_i^{-1} \tilde{y}_i^{\tilde{U}} \tilde{v}_i. \tag{6.21}$$

Hence, $w_{\mathcal{S}}^* \in \text{span}(v_1, \ldots, v_k)$, $w_{\mathcal{T}}^* \in \text{span}(\tilde{v}_1, \ldots, \tilde{v}_k)$. We show that our solution is also in the span of the top right singular vectors of the target domain as $w_{\mathcal{T}}^*$:

Theorem 6.4.2 (Main Result)**.** *Assume $k = \tilde{k}$ and $(V'_{d-k})^\top \tilde{V}'_{d-k}$ is invertible with $V'_{d-k} = \begin{bmatrix} v_{k+1} & \ldots & v_d \end{bmatrix}$ and $\tilde{V}'_{d-k} = \begin{bmatrix} \tilde{v}_{\tilde{k}+1} & \ldots & \tilde{v}_d \end{bmatrix}$, then $\widehat{w^*} \in \text{span}(\tilde{v}_1, \ldots, \tilde{v}_k)$ holds and $\widehat{w^*}$ is independent of λ.*

Proof. From the definition of $\widehat{w^*}$, we see that:

$$(\sum_{i \leq k} \sigma_i^2 v_i v_i^\top + \lambda \sum_{j > \tilde{k}} \tilde{\sigma}_j^2 \tilde{v}_j \tilde{v}_j^\top) \widehat{w^*} = \sum_{i \leq k} \sigma_i y_i^U v_i. \tag{6.22}$$

Decompose $\widehat{w^*} = \sum_{i \leq d} w_i \tilde{v}_i$. From the equation above we find:

$$\sum_{i \leq k} \sigma_i^2 v_i v_i^\top \widehat{w^*} + \lambda \sum_{j > \tilde{k}} \tilde{\sigma}_j^2 \tilde{v}_j w_j = \sum_{i \leq k} \sigma_i y_i^U v_i. \tag{6.23}$$

Applying $v_m^\top$ only both sides with $m > k$, we have:

$$\sum_{j>\tilde{k}} \tilde{\sigma}_j^2 v_m^\top \tilde{v}_j w_j = 0,\ m > k, \tag{6.24}$$

which can be written as:

$$(V'_{d-k})^\top V'^{\tilde{}}_{d-k} \operatorname{diag}(\tilde{\sigma}^2_{\tilde{k}+1}, \ldots, \tilde{\sigma}^2_d) \begin{bmatrix} w_{\tilde{k}+1} \\ \cdots \\ w_d \end{bmatrix} = \mathbf{0}. \tag{6.25}$$

Note that multiplying by $\operatorname{diag}(\tilde{\sigma}^2_{\tilde{k}+1}, \ldots, \tilde{\sigma}^2_d)$ does not change the invertibility. By assumption we must have $w_{k+1} = \cdots = w_d = 0$, and (6.23) becomes independent of λ. □

This theorem tells us that after label alignment regularization, $\widehat{w^*}$ and $w^*_{\mathcal{T}}$ lie in the same subspace.

We now characterize when our solution can lie in exactly the same direction as the optimal target domain's solution. Denote $V_k = [v_1 \ldots v_k]$, $\widetilde{V}_{\tilde{k}} = [\tilde{v}_1 \ldots \tilde{v}_{\tilde{k}}]$, and

$$\mu_k = (y_1^U/\sigma_1, \ldots, y_k^U/\sigma_k),\ \tilde{\mu}_{\tilde{k}} = (\tilde{y}_1^{\widetilde{U}}/\tilde{\sigma}_1, \ldots, \tilde{y}_{\tilde{k}}^{\widetilde{U}}/\tilde{\sigma}_{\tilde{k}}).$$

We have the following theorem:

Theorem 6.4.3. *Given invertible $V_k^\top \widetilde{V}_k$, with $V_k = [v_1 \ldots v_k]$, $\widetilde{V}_{\tilde{k}} = [\tilde{v}_1 \ldots \tilde{v}_{\tilde{k}}]$ and with the same assumptions of Theorem 6.4.2, there exists $c > 0$ such that $\widehat{w^*} = cw^*_{\mathcal{T}}$ iff:*

$$\mu_k = cV_k^\top \widetilde{V}_k \tilde{\mu}_k = cV_k^\top w^*_{\mathcal{T}}. \tag{6.26}$$

Proof. With the assumption of Theorem 6.4.2, we have: $v_i^\top \widehat{w^*} = y_i^U/\sigma_i$, for $i \leq k$. Then we can write down the optimal solutions as $w^*_{\mathcal{S}} = V_k \mu_k$, $w^*_{\mathcal{T}} = \widetilde{V}_{\tilde{k}} \tilde{\mu}_{\tilde{k}}$. Since $V_k^\top \widetilde{V}_k$ is invertible (and under the assumptions of Theorem 6.4.2), then we obtain the label alignment regularized result

$$\widehat{w^*} = \widetilde{V}_k (V_k^\top \widetilde{V}_k)^{-1} \mu_k. \tag{6.27}$$

Note that the solution is independent of the hyperparameter λ.

From $\widehat{w^*} = \widetilde{V}_k (V_k^\top \widetilde{V}_k)^{-1} \mu_k$, and $w^*_{\mathcal{T}} = \widetilde{V}_{\tilde{k}} \tilde{\mu}_{\tilde{k}}$, the equation $\widehat{w^*} = cw^*_{\mathcal{T}}$ holds iff:

$$\widetilde{V}_k ((V_k^\top \widetilde{V}_k)^{-1} \mu_k - c\tilde{\mu}_k) = \mathbf{0}, \tag{6.28}$$

or in other words, $(V_k^\top \widetilde{V}_k)^{-1} \mu_k = c\tilde{\mu}_k + q$, where $q \in \operatorname{null}(\widetilde{V}_k) = \{\mathbf{0}\}$. □

In the special case of $k = \tilde{k} = 1$, we obtain the following:

Corollary 6.4.4. *Given $k = \tilde{k} = 1$ and $\tilde{y}_1^{\widetilde{U}} y_1^U v_1^\top \tilde{v}_1 > 0$, we have $\widehat{w^*} = c w_{\mathcal{T}}^*$.*

This corollary tells us that, in this special setting, if for both domains the labels can be determined by the principal component (or, in other words, the most significant feature), then our label alignment regularization finds the optimal target solution.

Next, we show a sufficient condition for our invertibility assumptions in Theorem 6.4.2 to hold: V and $\widetilde{V}$ are somehow similar to each other.

Proposition 6.4.5. *Suppose $\epsilon < \min\{\frac{1}{k}, \frac{1}{d-k}\}$, $|v_i^\top \tilde{v}_j| \leq \epsilon$ for any $i \neq j$, and $v_i^\top \tilde{v}_i \geq 1 - \epsilon$ for any i, then both $V_k^\top \widetilde{V}_k$ and $(V'_{d-k})^\top \tilde{V}'_{d-k}$ are invertible.*

Proof. $V_k^\top \widetilde{V}_k$ can be written as $[v_i^\top \tilde{v}_j]$ with $i, j \in [k]$. From the assumption, $V_k^\top \widetilde{V}_k = I + \epsilon\Delta$ where I is the identity matrix and Δ is a $k \times k$ matrix with each element $|\Delta_{ij}| \leq \epsilon$. Suppose $(I + \epsilon\Delta)x = \mathbf{0}$, then $x = -\epsilon\Delta x$, taking the norm on both sides we have:

$$\|x\| = \epsilon\|\Delta x\| \leq \epsilon\|\Delta\| \cdot \|x\| \tag{6.29}$$

$$\leq \epsilon\|\Delta\|_F \cdot \|x\| \leq \epsilon k \cdot \|x\| < \|x\|, \tag{6.30}$$

which gives $x = \mathbf{0}$. Therefore, $I + \epsilon\Delta$ is invertible. Similarly, $(V'_{d-k})^\top \tilde{V}'_{d-k}$ is also invertible. □

We can give a stronger guarantee for the assumption that $V_k^\top \tilde{V}_k$ is invertible. Note that $\mathbb{S}^{d-1}$ denotes the $(d-1)$-dimensional unit hypersphere in $\mathbb{R}^d$.

Proposition 6.4.6. *Suppose the target singular vectors $\tilde{v}_1, \ldots, \tilde{v}_d$ satisfies the following probability distribution:*

$$p(\tilde{v}_1, \ldots, \tilde{v}_d) = p(\tilde{v}_1) p(\tilde{v}_2|\tilde{v}_1) \ldots p(\tilde{v}_d|\tilde{v}_1, \ldots, \tilde{v}_{d-1}), \tag{6.31}$$

where $p(\tilde{v}_1)$ is a continuous distribution on $\mathbb{S}^{d-1}$ and each $p(\tilde{v}_i|\tilde{v}_1, \ldots, \tilde{v}_{i-1})$ is a continuous distribution on the manifold $\mathbb{S}^{d-1}$ for $2 \leq i \leq d$. Then $V_k^\top \widetilde{V}_k$ is invertible almost surely.

Proof. It suffices to show that $P(\det(V_k^\top \widetilde{V}_k) = 0) = 0$. Note that $\det(V_k^\top \widetilde{V}_k) = 0$ can be rewritten as:

$$\det(\tilde{v}_1^{V_k} \dots \tilde{v}_k^{V_k}) = 0, \tag{6.32}$$

and thus

$$P(\det(V_k^\top \widetilde{V}_k) = 0) \leq p(\tilde{v}_1^{V_k} = 0) + p(\tilde{v}_2^{V_k} \in \text{span}(\tilde{v}_1^{V_k}) | \tilde{v}_1) + \dots + \tag{6.33}$$

$$p(\tilde{v}_k^{V_k} \in \text{span}(\tilde{v}_1^{V_k}, \dots, \tilde{v}_{k-1}^{V_k}) | \tilde{v}_1, \dots, \tilde{v}_{k-1}). \tag{6.34}$$

Since each condition gives a sub-manifold with a smaller dimension and the probability distributions are continuous, from Sard's theorem (e.g. Guillemin & Pollack, 2010), each probability is zero. Therefore, $P(\det(V_k^\top \widetilde{V}_k) = 0) = 0$. □

Our result does not depend on any assumption about the optimal joint error, as is commonly required in the domain adaptation literature (e.g. Ben-David et al., 2010; Acuna et al., 2021). Moreover, as pointed out by Zhao et al. (2019), the usual generalization bound would fail in the presence of heavy shift of label distributions, under which our method is still robust (see Section 6.6).

6.5 Related Work

The result by Ben-David et al. (2010) provides a general theoretical guidance regarding how to learn the domain-invariant representations. The basic idea is to make the joint error of the best hypothesis on the two domains on the invariant representation small. Low joint error in the domain-adversarial model is crucial to the model's performance on the target domain.

The dominant approach to domain adaptation is learning domain-invariant representations that are "similar" in some sense between source and target domains (Tzeng et al., 2014; Zhuang et al., 2015; Ghifary et al., 2016; Long et al., 2016, 2017; Benaim & Wolf, 2017; Bousmalis et al., 2017; Courty et al., 2017; Motiian et al., 2017; Rebuffi et al., 2017; Saito et al., 2017; Y. Zhang et al., 2019). Different methods differ in how the invariance property is enforced, which typically includes how the similarity is defined and implemented. Recent

work in deep learning encourages this invariance in one or multiple hidden representations of a neural network.

The popular domain-adversarial methods achieve domain-invariant representations based on the idea of adversarial models (Long et al., 2015; Zhuang et al., 2017; Lee et al., 2019; Damodaran et al., 2018; Acuna et al., 2021). Specifically, Long et al. (2015) adversarially learn representations to distinguish the data points from the source and target domain while minimizing the supervised loss. Conneau et al. (2018) use a domain-adversarial approach to align representations of the source and target domains in a shared space. They transform the source embeddings with a linear mapping that is encouraged to be orthogonal. The domain-adversarial model then generates pseudo-labels on the target domain for additional refinement. The shared representation, which is learned without a parallel corpus, outperforms previous supervised methods in several cross-lingual tasks.

There are various similarity or distance measures to define a loss function for enforcing invariant representations. For example, Zhuang et al. (2017) and Meng et al. (2018) minimize the KL-divergence and Lee et al. (2019) and Damodaran et al. (2018) minimize the Wasserstein distance. Sun and Saenko (2016) minimize the ℓ_2 distance between the covariance matrices of the source and target domain representations. Long et al. (2015) minimize Maximum Mean Discrepancy between source and target domain hidden representations embedded with a kernel.

Despite the flourishing literature on representation-based domain adaptation methods, they have critical limitations. Zhao et al. (2019) and Johansson et al. (2019) have presented synthetic examples in which a domain-adversarial model that minimizes the supervised loss in the source domain, while aiming for an invariant representation, still fails in the target domain. We will demonstrate this failure through our experiments in the next section and show that our proposed algorithm remains robust in such situations.

Domain-adaptation methods that do not rely on representation learning are less studied and can be applied in highly restricted settings. For example, importance sampling (Shimodaira, 2000) assumes label conditional distribu-

tion must be the same and target domain is within the support of the source domain. Our work supplements this direction.

6.6 Experiments

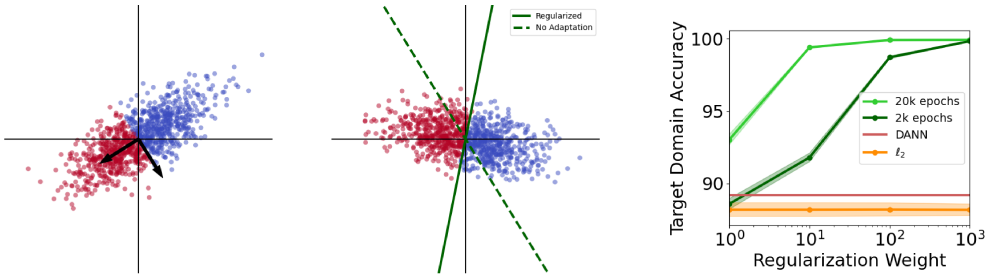
In this section, we first design a synthetic dataset to verify that our regularizer is indeed beneficial in a distribution shift setting by adjusting the classifier and to perform an ablation study on the role of removing implicit regularization. Then, we demonstrate the effectiveness of our method on a well-known benchmark where classic domain-adversarial methods are known to fail (Zhao et al., 2019). Last, we show our algorithm's practical utility in a cross-lingual sentiment classification task.

6.6.1 Synthetic Data

We create a distribution shift scenario where the alignment property is present in the labeled data distribution (Figure 6.2), as theoretically discussed in Section 6.4.1. For the source domain (a), the input is sampled from a two-dimensional Gaussian distribution. The distribution is more spread out in the direction of the first principal component (see the black arrows) which corresponds to a larger singular value. In this task, the two classes are separated along this direction as shown in the figure. The resulting vector of all labels is mostly in the direction of the first singular vector of the representation matrix. We rotate the input by 45° to create the target distribution in (b).

We then run the proposed algorithm with hyperparameters $k = \tilde{k} = 1$ and different values of λ and compared it with the ℓ_2 regularizer and a domain-adversarial baseline DANN (Ganin et al., 2016) with one hidden layer of width 64. Note that the optimal solution should be independent of λ (Proposition 6.4.1), but λ may affect the convergence rate. Figures 6.2 (b) to (c) show the results. In Figure 6.2 (b) we see that the solution without regularization separates the classes as they are separated in the source domain. The proposed algorithm finds a separating hyperplane that matches how the classes are separated in the target domain. Finally, (c) shows that our regularizer surpasses

both the ℓ_2 regularizer and the domain-adversarial baseline and achieves a near perfect classification in the target domain in this example. The dark green line in this figure uses 2k epochs and its accuracy is sensitive to λ. Increasing the number of epochs to 20k (green line) reduces this sensitivity, indicating that, as the theory predicted, the final solution is robust to λ and the sensitivity is due to slow optimization.



(a) Source Domain (b) Target Domain (c) Performance

Figure 6.2: (a) Source domain. The black arrows show principal components. (b) Target domain. The green lines show separating hyperplanes found without using any regularization (dashed) and with our regularizer with $\lambda = 10^3$ (solid). (c) Performance on the target domain. The red line shows the performance of DANN. The x axis is the regularization coefficient for ℓ_2 regularization (orange curve) and λ for the proposed regularizer (green curves). The proposed regularizer achieves near-perfect accuracy on this domain. Shaded areas are standard errors over 10 runs. Variations in target accuracy of DANN are near zero.

We also want to evaluate the effectiveness of removing the implicit regularization term $\sum_{i=k+1}^{d} \sigma_i^2 (w_i^V)^2$ as described in Equation 6.6. More specifically, we are interested in when removing the implicit regularization would be effective.

Recall from Section 6.4 that the vanilla closed-form solution without any regularization is:

$$w = S^{-1}\Phi^T y \tag{6.35}$$

where $S = \Phi^T\Phi$, and $\Phi \in \mathbb{R}^{n\times d}$ is the feature matrix.

To utilize more specific chracteristics of a dataset, we want to first explore the synthetic data case. Then the closed-form solution with label alignment

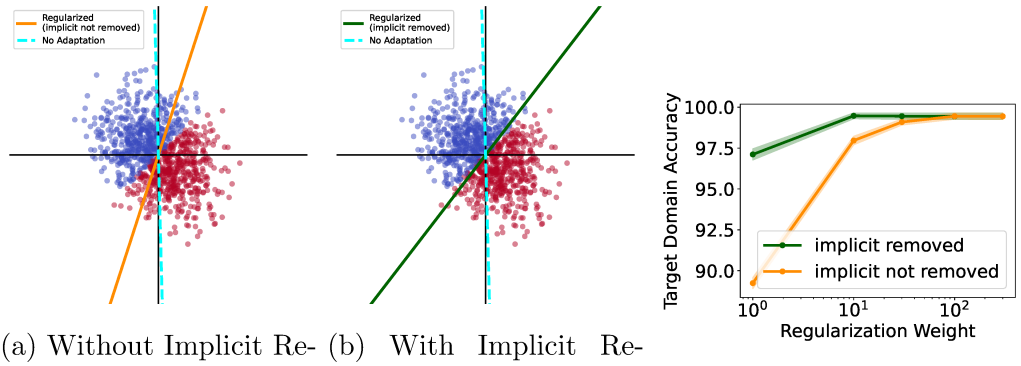
regularization with removal of the implicit regularization term is:

$$\begin{aligned} w &= (S_k + \lambda(\tilde{S} - \tilde{S}_k))^{-1}\Phi^T y \\ &= (s_1^2 p_1 p_1^T + \lambda \tilde{s}_2^2 \tilde{p}_2 \tilde{p}_2^T)^{-1}\sqrt{\frac{2}{\pi}} s_1 p_1 \end{aligned} \tag{6.36}$$

And the closed-form solution with label alignment regularization without implicit regularization removal is:

$$\begin{aligned} w &= (S + \lambda(\tilde{S} - \tilde{S}_k))^{-1}\Phi^T y \\ &= (s_1^2 p_1 p_1^T + s_2^2 p_2 p_2^T + \lambda \tilde{s}_2^2 \tilde{p}_2 \tilde{p}_2^T)^{-1}\sqrt{\frac{2}{\pi}} s_1 p_1 \end{aligned} \tag{6.37}$$

The only difference is $s_2^2 p_2 p_2^T$, and therefore the relative magnitude of s_2 and λ should be the deciding factor of the solution w. Experiments conducted on synthetic data corroborated the theoretical conclusion, as illustrated in Figure 6.3. As seen in the figure, s_2 is larger compared to the previous experiment. The target distribution is a rotation of this new distribution. Removing implicit regularization results in a different performance especially with smaller values of λ.

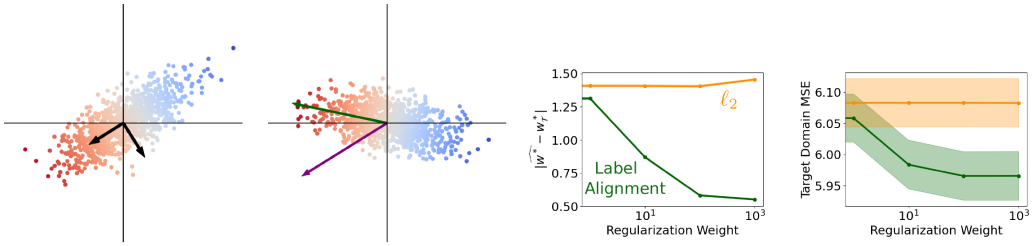


(a) Without Implicit Removal (b) With Implicit Removal (c) Performance

Figure 6.3: (a) Without Implicit Removal. The cyan dashed line is the decision boundary without any adaptation. The orange line shows the decision boundary when λ is set to 1 for our proposed regularizer without implicit removal. (b) With Implicit Removal. The green line shows the decision boundary when λ is set to 1 with implicit removal. (d) Performance on the target domain. The horizontal axis is λ for the proposed regularizer. Before λ dominates, the benefits of removing implicit regularization are significant. Shaded areas are standard errors over 10 runs.

6.6.2 Regression

We create a regression task similar to the synthetic experiment above. The aim is to understand if results similar to the synthetic experiment hold in a setting where the labels are not restricted to ± 1 and where mean squared error is used for evaluation. All the details are the same as in the classification experiment except that the label vector is simply set to the first left singular vector. Figure 6.4 shows the results and corroborates the previous findings. Note that DANN is not directly applicable here.



(a) Source domain (b) Target Domain (c) Parameter Error (d) Performance

Figure 6.4: (a) Source domain. The black arrows show principal components. (b) Target domain. The arrows show weights found without using any regularization (purple) and with our regularizer with $\lambda = 10^3$ (green). (c) Distance between the estimated and the optimal weights. The proposed regularizer reduces this distance. (d) Performance on the target domain. The x axis is the regularization coefficient for ℓ_2 regularization and λ for the proposed regularizer. The proposed regularizer achieves lower error on this domain. Shaded areas are standard errors over 10 runs.

6.6.3 MNIST-USPS

The experiments in this section consider binary classification tasks from the MNIST-USPS domain adaptation benchmark with linear and shallow models. Both MNIST and USPS are digit classification datasets with 10 classes and therefore 45 binary classification tasks between two digits. In MNIST, the input is a 28×28 grayscale image flattened to a 784-dimensional vector. USPS images are 16×16 and we resize them to 28×28 and flatten each input to a 784 vector.

Each column of the table is the average accuracy over 45 domain adaptation

tasks. In the first column, the source domain (fully labeled) is a pair of digits from MNIST and the target domain (fully unlabeled) is the same pair, but from USPS. The datasets for the source and target domains are reversed in the second column. The last three columns are like the second column except that, in binary classification between two digits, only a certain ratio of the lower digit in the source domain, as indicated in the header, is used. This subsampling creates a large degree of imbalance that, as Zhao et al. (2019) observed, poses a challenge to domain-adversarial methods.

We use the train split of the dataset for the source domain and the test split of the other dataset for the target domain. A small set of 100 labeled points from the target domain is used for hyperparameter selection as we have not developed a fully unsupervised hyperparameter selection strategy. However, we give the baseline the same validation set to keep the experiment fair.

The domain-adversarial baseline DANN uses a one-layer ReLU neural network. This is the Shallow DANN architecture suggested by the original authors (Ganin et al., 2016). We swept over values of 128, 256, 512, and 1024 for the depth of the hidden layer. The neural network is trained for 10 epochs using SGD with mini-batch size 32, learning rate 0.01, and momentum 0.9. This model already achieves near perfect accuracy on the source domain. Candidate hyperparameter values for Label Alignment Regularizer were $\{1e-1, 1e+1, 1e+3\}$ for λ and $\{8, 16, 32, \cdots\}$ up to the rank of Φ or $\tilde{\Phi}$ for k and $\tilde{k}$. Although the number of hyperparameter configurations is greater for our method, this experiment is in favor of DANN if we take runtime into account. The linear model is trained using full-batch gradient descent for 5000 epochs with learning rate $1/(2\sigma_1)$.

The first two rows of Table 6.2 show the performance of the domain-adversarial method DANN (Ganin et al., 2016) with one hidden layer on these tasks. (Deeper NNs performed worse on the highly imbalanced tasks in our preliminary experiments.) The first row is the average target domain accuracy of a two-layer ReLU NN trained purely on the source domain. In the second row, the domain-adversarial objective is added to reduce domain shift in the hidden representation. DANN improves accuracy in both U→M and

M→U. In the cases with subsampling, however, DANN consistently hurts performance. The third and fourth row show the performance of a linear method without and with our regularizer. Using our regularizer improves performance in all columns and outperforms the models in the other rows in the cases with subsampling.

	U → M	M → U	.3 → U	.2 → U	.1 → U
No Adaptation (NN)	77.85	84.88	**83.36**	**72.84**	**53.58**
DANN	**83.93**	**86.69**	78.05	64.2	47.27
No Adaptation (Linear)	78.68	83.84	80.99	79.47	75.41
LAR	**81.97**	**88.96**	**86.99**	**84.84**	**82.71**

Table 6.2: Accuracies on MNIST-USPS benchmark. LAR is Label Alignment Regression. Each column is averaged over the 45 binary classification tasks. M and U indicate MNIST and USPS. Ratios indicate MNIST tasks where one digit is subsampled. In tasks with severe subsampling the proposed algorithm improves the accuracy and achieves the highest performance. DANN performs worse than a regular neural network under subsampling.

We then investigate why DANN hurts performance under subsampling. A domain-adversarial network like DANN has three components: a domain classifier (discriminator) that predicts whether a data point is from the source or the target domain, a generator that learns a shared embedding between the two domains, and a label predictor that performs classification on the task of interest using the generator's embedding. The label predictor uses the labeled source data to increase source accuracy, i.e. the label predictor's accuracy on the source domain. The ultimate goal is to have the label predictor achieve high accuracy on the target domain. The discriminator's accuracy on the other hand shows how successful the discriminator is in recognizing whether a point is from the source or the target domain. In an ideal case this accuracy should be close to that of a random classifier since the data points from the two domains are mapped close to each other in the shared embedding.

Table 6.3 shows the average source domain accuracy and domain classifier accuracy of DANN. Average source accuracy remains $\geq 95\%$ and average domain classifier remains $\approx 50\%$, indicating that DANN has managed to learn a representation that is suitable for the source domain and maps the points from

the source and target domain close to each other. The large drop in DANN's performance can be attributed to the fact that the representation maps positive points in the source domain close to negative points in the target domain and vice-versa and therefore the joint error of the best hypothesis on the two domains (as described in Section 6.5) is large. We verify this by training a nearest neighbour (1-NN) classifier on the learned representation in the subsampled settings. The 1-NN classifier uses the source domain representations as the training data and the target domain representations as the test data. The accuracy of this classifier will suffer if in the learned representation the source domain points from one class are mapped close to the target domain points from the other class. The third row in the table, which is also averaged over the 45 tasks, shows a noticeable drop in the performance of the 1-NN classifier and indicates that this problem is present in the learned embeddings.

	U $\rightarrow$ M	M $\rightarrow$ U	.3 $\rightarrow$ U	0.2 $\rightarrow$ U	.1 $\rightarrow$ U
Source Accuracy	98.06	98.83	98.3	97.56	95.3
Discriminator Accuracy	46.4	50.63	50.42	50.48	50.44
1-NN Accuracy	-	-	77.89	73.22	69.75

Table 6.3: Source accuracy and domain classifier accuracy of DANN on MNIST-USPS. The drop in source accuracy under severe subsampling is minimal compared to the drop in target accuracy in the previous table. The domain classifier accuracy is near random regardless of the amount of subsampling. The performance of a nearest neighbour classifier trained on the mapped source data points and evaluated on the mapped target data points degrades to a large extent with more subsampling.

6.6.4 Multi-Class Classification

Although our overall focus is on binary classification, we also try to generalize the formulation of the label alignment regression to a multiclass setting following the derivation in Equation 6.3. Given a dataset comprising n samples, each characterized by d features, we denote the feature matrix by $\Phi \in \mathbb{R}^{n \times d}$. In a classification context involving c distinct classes and employing a one-versus-all strategy, the target matrix Y, which adopts a ± 1 style one-hot encoding scheme, is of dimension $\mathbb{R}^{n \times c}$. Consequently, the weight matrix W, which

maps the feature space to the class labels, is represented as $\mathbb{R}^{d\times c}$. Then the learning objective can be formulated as:

$$\begin{aligned}\min_W \|\Phi W - Y\|^2 &= \min_W \|U\Sigma V^T W - Y\|^2 \\ &= \min_W \|\Sigma V^T W - U^T y\|^2 \\ &= \min_W \|\Sigma W^V - Y^U\|^2 \\ &= \min_W \sum_{j=1}^{c}\sum_{i=1}^{d}(\sigma_i W_{ij}^Y - Y_{ij}^U)^2 + \sum_{j=1}^{c}\sum_{i=d+1}^{n}(Y_{ij}^U)^2\end{aligned} \tag{6.38}$$

The notation $\|A\|^2$ signifies the 2-norm of matrix A, encapsulating the square root of the sum of the squares of its elements. In this setup, W^V corresponds to a weight matrix with dimensions $\mathbb{R}^{d\times c}$, while Y^U denotes a modified target matrix also of dimension $\mathbb{R}^{n\times c}$. Thus, the expression $(\Sigma W^V - Y^U)$, representing the discrepancy between the projected feature space and the modified target matrix, retains the dimensionality of $\mathbb{R}^{n\times c}$, highlighting the alignment or misalignment of the model predictions with the modified targets in the given multidimensional space.

Assume k is the same for every one vs all setting and $k < d$, we can obtain:

$$\begin{aligned}\min_W \|\Phi W - Y\|^2 &= \min_W \sum_{j=1}^{c}\sum_{i=1}^{d}(\sigma_i W_{ij}^Y - Y_{ij}^U)^2 + \sum_{j=1}^{c}\sum_{i=d+1}^{n}(Y_{ij}^U)^2 \\ &= \min_W \sum_{j=1}^{c}\sum_{i=1}^{d}(\sigma_i W_{ij}^V - Y_{ij}^U)^2 \\ &= \min_W \sum_{j=1}^{c}\sum_{i=1}^{k}(\sigma_i W_{ij}^V - Y_{ij}^U)^2 + \sum_{j=1}^{c}\sum_{i=k+1}^{d}(\sigma_i W_{ij}^V)^2\end{aligned} \tag{6.39}$$

Therefore the final objective function looks like:

$$\min_W \|\Phi W - Y\|^2 - \sum_{j=1}^{c}\sum_{i=k+1}^{d}(\sigma_i W_{ij}^V)^2 + \lambda\sum_{j=1}^{c}\sum_{i=\tilde{k}+1}^{d}(\tilde{\sigma}_i W_{ij}^{\tilde{V}})^2 \tag{6.40}$$

We also validated the label alignment property by computing $k(0.1)$ for the one-versus-all label vector corresponding to each digit similar to the binary classification case in Table 6.1. The value of $k(0.1)$ for all the digits was 1.

Then, we compare the classification performance of our label alignment regression to DANN in the multiclass MNIST-USPS classification setting. Our

method outperforms DANN by a large margin as shown by the evaluation results in Table 6.4.

	U → M	M → U	.3 → U	.2 → U	.1 → U
No Adaptation (NN)	35.66	52.46	47.24	45.06	19.48
DANN	41.32	53.46	49.88	43.09	32.01
No Adaptation (Linear)	37.53	54.41	48.71	46.21	41.54
LAR	**42.47**	**63.90**	**54.69**	**51.70**	**47.49**

Table 6.4: Accuracies on MNIST-USPS multiclass benchmark. M and U indicate MNIST and USPS. Ratios (0.3, 0.2, 0.1) indicate MNIST tasks where 9 out of the 10 digits are subsampled. For the subsampling setting (last three columns), each column is averaged over the 10 subsampling classification tasks. In all tasks, the proposed algorithm improves the accuracy and achieves the highest performance.

6.6.5 Parameter Sensitivity

The parameter λ indicates the ratio of the loss obtained from the unsupervised information of the target domain. We want to quantitatively evaluate how this ratio influences the performance of our proposed method on the target domain. The sensitivity visualization is shown in Figure 6.5.

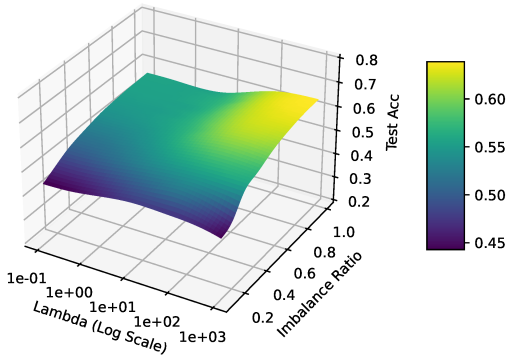


Figure 6.5: λ sensitivity curves of accuracies on MNIST USPS multiclass benchmark. The performance of the proposed method is relatively invariant over different λ under various imbalance (subsampling) ratios. Generally greater λ comes with better performance in the target domain because more weight and emphasis of loss is put on the information of the target domain.

6.6.6 Cross-Lingual Sentiment Classification

This section includes cross-lingual sentiment analysis experiments on deep features. XED (Öhman et al., 2020) is a sentence-level sentiment analysis dataset consisting of 32 languages. We use English as the source domain and another language as the target domain and create 9 binary classification domain adaptation tasks.

There are a total of 1984 language pairs from each of the 32 languages to another. We chose 9 language pairs before running the experiment. The sentences in the dataset are labeled with one or more emotions *anger*, *anticipation*, *disgust*, *fear*, *joy*, *sadness*, *surprise*, and *trust*. Following the authors' guidelines we turn these multi-label classification tasks to binary classification by labeling data points positive if their original labels only include *anticipation*, *joy*, and *trust*, and negative if the original labels only include *anger*, *disgust*, *fear*, and *sadness*. (*Surprise* is discarded.)

We perform 5 runs and in each one 100 points are randomly sampled from the target domain for validation and the rest are used for evaluation. Similar to the previous experiment, this validation set is used for all algorithms with hyperparameter configurations discussed in Appendix B to have a fair comparison. The representations for the source and target domain are 768-dimensional sentence embeddings obtained with BERT (Devlin et al., 2019) models pre-trained on the corresponding languages. The experiment compares Label Alignment Regression with the following baselines.

Source: This baseline trains a linear regression model with squared error (MSE) or a logistic regression model with crossentropy loss (CE) directly on the source domain and evaluates it on the target domain.

Adversarial-Refine (Conneau et al., 2018): This baseline uses a domain-adversarial approach to learn a linear transformation that maps the source and target domain into a shared space. A refinement step then encourages the transformation to be orthogonal. This approach has shown promising results in several cross-lingual NLP tasks with word embeddings. We train

a linear regression model with squared error (MSE) or a logistic regression model with crossentropy loss (CE) on the source data using the learned shared space and then evaluate it on the target domain. We sweep over values of $\{1e-3, 1e-2, 1e-1\}$ for β. The parameter controls the degree of orhtogonality of the transformation that maps the source and target embeddings into a common space.

CDAN (Long et al., 2017): Recall that a domain-adversarial method consists of a domain classifier (discriminator) that predicts whether a data point is from the source or the target domain, a generator that learns a shared embedding between the two domains, and a label predictor that performs classification on the task of interest using the generator's embedding. CDAN improves on DANN by conditioning the domain classifier on the label predictor's prediction. The motivation is the improvements observed by incorporating this modification to generative adversarial networks (Goodfellow et al., 2020; Mirza & Osindero, 2014).

f-DAL (Acuna et al., 2021): This approach modifies DANN to use a separate domain classifier for each class which allows minimizing a family of divergence measures between the source and domain embeddings. We use f-DAL to minimize Pearson χ^2 divergence as the authors had observed superior performance with this divergence in previous vision and NLP benchmarks.

IWDAN and IWCDAN (Tachet et al., 2020): These two methods modify DANN and CDAN by incorporating importance sampling to reduce deterioration in performance due to class imbalance. Computing the importance sampling ratios requires access to target domain labels. The authors propose to estimate the ratios and provide the theoretical requirements for the estimation to be accurate.

The models used in Source, Adversarial-Refine, and Label Alignment Regression are linear regression or logistic regression (on the nonlinear extracted representations). These models are trained with learning rate $1/(2\sigma_1)$ (MSE

loss) and $1e-2$ (CE loss) and momentum 0.9. Candidate hyperparameter values for Label Alignment Regularizer were $\{1e-1, 1e+1, 1e+3\}$ for λ and $\{8, 16, 32, \cdots\}$ up to the rank of Φ or $\tilde{\Phi}$ for k and $\tilde{k}$. The other methods are neural network based and we sweep over regularization coefficients $\{1e-4, 1e-2, 1\}$ with a one-hidden-layer ReLU network. This is the architecture suggested by (Ganin et al., 2016) for domain adaptation with a shallow network.

Table 6.5 shows F_1 scores for the nine tasks and the average score over the tasks. The first 8 rows are the baselines above and in the last row (LAR) we employ the Label Alignment Regression algorithm. The proposed algorithm achieves the highest F_1 score on seven out of the nine tasks as well as on average over the tasks. Adv - Refine, CDAN, and f-DAL do not provide a consistent benefit over No Adaptation. The two methods with importance weightings, IWDAN and IWCDAN, find better solutions than No Adaptation as well as the other domain-adversarial methods, suggesting that the reweighting in this algorithm, even if it is an estimate of the true importance weighting, is beneficial.

6.7 Discussion

In this work, we proposed a domain adaptation regularization method based on the observation of label alignment property—the label vector of a dataset usually lies in the top left singular vectors of the feature matrix. We show that a regression algorithm in a standard supervised learning task actually contains an implicit regularization method to enforce such a property. Then we demonstrate how we can adapt such a regularization method in a domain adaptation setting. A critical difference between our algorithm and the conventional domain adaptation method is that we do not use regularization to adjust the representation learning. We observe that our algorithm does work well under high imbalance, where the conventional representation-based domain adaptation method fails. We also report improvement over baselines on cross-lingual sentiment analysis tasks.

	en → bg	en → br	en → cn	en → da	en → de
Source (MSE)	55.22 (0.23)	53.51 (0.53)	4.48 (0.27)	64.75 (0.14)	47.95 (0.52)
Source (CE)	51.55 (0.12)	56.94 (0.04)	0.37 (0.00)	64.00 (0.18)	46.55 (0.22)
Adv-R (MSE)	46.88 (0.61)	46.20 (1.56)	53.54 (1.74)	51.98 (1.87)	50.43 (1.07)
Adv-R (CE)	45.12 (0.82)	36.96 (0.95)	49.87 (1.37)	50.99 (1.31)	43.62 (0.66)
CDAN	49.99 (5.43)	31.97 (8.43)	21.93 (12.76)	55.80 (5.12)	33.52 (9.60)
f-DAL	51.95 (0.88)	**57.79 (0.50)**	15.04 (3.65)	64.23 (0.14)	45.74 (0.82)
IWDAN	56.16 (1.05)	55.95 (0.50)	46.78 (3.55)	63.41 (1.20)	46.30 (5.69)
IWCDAN	57.70 (1.32)	54.96 (1.44)	42.08 (5.85)	63.64 (1.49)	41.86 (6.48)
LAR	**59.85 (0.08)**	53.42 (0.33)	**65.10 (0.24)**	**65.58 (0.12)**	**60.46 (0.05)**
	en → es	en → fr	en → he	en → hu	Average
Source (MSE)	39.17 (0.93)	49.89 (0.57)	58.19 (0.23)	59.66 (0.12)	48.09 (0.37)
Source (CE)	47.09 (0.20)	40.90 (0.36)	58.23 (0.10)	55.82 (0.06)	46.83 (0.10)
Adv-R (MSE)	**48.37 (1.38)**	46.45 (0.88)	48.93 (1.03)	47.15 (1.18)	48.88 (0.69)
Adv-R (CE)	41.30 (1.40)	44.18 (2.16)	46.95 (1.39)	44.06 (1.44)	44.78 (0.40)
CDAN	21.26 (6.97)	36.30 (14.82)	41.29 (9.49)	34.95 (6.14)	36.33 (3.73)
f-DAL	48.25 (12.06)	**58.23 (0.62)**	60.10 (0.31)	47.42 (1.28)	49.86 (1.53)
IWDAN	36.21 (2.93)	54.04 (1.72)	58.18 (0.97)	56.00 (1.54)	52.56 (0.84)
IWCDAN	37.63 (2.45)	52.14 (2.39)	61.08 (0.56)	55.82 (1.54)	51.88 (1.24)
LAR	43.47 (0.90)	58.11 (0.24)	**61.24 (0.09)**	**59.68 (0.12)**	**58.55 (0.17)**

Table 6.5: F_1 score in percents on different XED source-language pairs. The numbers in parentheses are standard errors. Adv-R refers to Adversarial-Refine. MSE and CE denote Mean Squared Error and Cross Entropy loss. LAR (Label Alignment Regression) outperforms the baselines on average and on most of the tasks. For adversarial baselines we verified that the discriminator accuracy is near random in this experiment similar to the MNIST-USPS experiment.

Immediate next steps are providing an unsupervised hyperparameter selection strategy and extension to multi-class classification. The current method uses a validation set for choosing the hyperparameters. This validation set is remarkably small and on the NLP tasks we found little benefit from involving this set to train a semi-supervised method.

A better hyperparameter selection strategy can also help with applying the proposed method to multi-class classification problems. In a small experiment in Section 6.6 we briefly discussed how the regularizer can be extended to multi-class problems using multiple outputs and one-hot labels. In general, the multi-class version would require tuning the hyperparameters separately for each output and the current grid search method would become expensive with large number of classes or fine grids. Using a fixed set of hyperparameter values for all the outputs, we showed promising results on the MNIST-USPS benchmark in the same section and we leave further exploration to future work.

Other future directions are to investigate the combination of our method and the conventional representation-based domain adaptation method, with

the hope that the hybrid method has the advantage of both—it can provide a significant advantage in a broad range of domain-shift settings. It would also be interesting to have a more rigorous theoretical characterization regarding when the label alignment property holds and to what extent the label vector can align with the top singular vectors.

Chapter 7

Generalization in Temporal Difference Learning

In the previous chapters we created synthetic tasks that resulted in different expected convergence rates and different generalization errors in regression. In particular, we observed that tasks that resulted in faster expected convergence also resulted in better generalization from a small sample. In this chapter we turn to the problem of policy evaluation with the aim of better understanding generalization with Temporal Difference learning. We will introduce a similar procedure for conducting small synthetic experiments and observe the same trend in optimization and generalization in this context.

7.1 Background

The goal in Reinforcement Learning (RL) is finding a high performing policy through interaction with an environment and a common intermediate step is evaluating a given policy (Sutton & Barto, 2018). Neural networks are often used in this context either to parametrize the policy itself or to predict the value of a policy. A popular algorithm for policy evaluation is Temporal Difference learning (TD) that, instead of fitting a fixed target, bootstraps from current predictions to obtain better predictions of the value of the policy.

The generalization and transfer abilities of neural networks that we discussed in Chapters 1 and 5 do not extend to the RL problem. Neural networks trained with common algorithms are known to largely capture superfluous pat-

terns in the environment and the resulting networks are hardly usable on a slightly different problem, even as a starting point for further training (Farebrother et al., 2018; X. Song et al., 2020; Packer et al., 2019). In contrast to a regression problem, in RL the neural network has to learn a sequence of tasks as the policy is improved, and this non-stationarity is conjectured as a reason behind the poor generalization and transfer (Lyle et al., 2022a; Nikishin et al., 2022; Dohare et al., 2021).

Even in the simpler problem of evaluating a fixed policy in a fixed environment from a given sample, TD is known to often result in undergeneralizing solutions in practice and requirements for obtaining good generalization in policy evaluation with TD can be different from regression with gradient descent (E. Bengio et al., 2020; Lyle et al., 2022b). Such differences have motivated regularization techniques and theoretical frameworks focused on RL algorithms (Farahmand, 2011; Amit et al., 2020; Le Lan et al., 2022; François-Lavet et al., 2019). Since TD bootstraps from current predictions, one approach to studying policy evaluation with TD is to look at it as solving a sequence of regression problems with different targets, even when the evaluated policy is fixed. Using target networks to reduce this form of non-stationarity or enforcing robustness to it has been helpful in policy evaluation and the broader RL problem (Mnih et al., 2015; Lyle et al., 2023).

The chasm between the behavior of gradient descent and TD is also manifest in the requirements for fast optimization. The difference has led to developing dedicated acceleration methods for TD and its related algorithms along with theoretical characterizations of rate of convergence (Meyer et al., 2014; Pan et al., 2017; Gupta et al., 2019; E. Bengio et al., 2021; Bhandari et al., 2018; Chen et al., 2020; Patil et al., 2023).

Properties of the representation are consequential to both optimization and generalization in policy evaluation and RL (Bellemare et al., 2019; Le Lan et al., 2022; Patil et al., 2023) and learning or constructing a good representation for RL has been a longstanding area of research (Dayan, 1993; Menache et al., 2005; Keller et al., 2006; Yu & Bertsekas, 2009; Mahadevan & Maggioni, 2007; Sutton, 1995; Jaderberg et al., 2017). Despite this, the theoretical formulation

of the role of representation in generalization in policy evaluation and RL is still underdeveloped (Le Lan et al., 2022).

In this chapter we focus on policy evaluation with TD and linear models and ask what relationship between the representation and the environment makes generalization from a small sample easier. Rather than looking at TD as solving a sequence of regression problems, we will look at it as a single dynamical system and juxtapose it with a dynamical system corresponding to gradient descent. Earlier in Chapter 4 we designed a sequence of tasks such that gradient descent showed different expected convergence rates. In Chapter 3 we observed that generalization performance of gradient descent across these different tasks reflected the trend in expected convergence rates. In this chapter we will design an analogous sequence of policy evaluation tasks for TD and empirically find that, across these tasks, TD generalizes well from a small sample when it converges fast in expectation.

7.2 Policy Evaluation and TD

The environment in an RL problem is modeled as a discounted MDP, a tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu)$ where $\mathcal{S}$ and $\mathcal{A}$ denote the state space and action space, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the transition probability kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the immediate reward, $\gamma : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1)$ is the transition-dependent discount factor, and $\mu \in \mathcal{P}(\mathcal{S})$ is the start state distribution. A policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ induces a probability distribution over actions at each state. We define $r_\pi(s) := \sum_{a \in \mathcal{A}} \pi(s, a) r(s, a)$ and $P_{\pi,\gamma}(s, s') := \sum_{a \in \mathcal{A}} \pi(s, a) P(s, a, s') \gamma(s, a, s')$. We assume both the state space and the action state are finite. The interaction of a policy and the MDP induce a trajectory of random variables $S_0, A_0, R_0, \gamma_0, S_1, \cdots$ where $S_0 \sim \mu, A_0 \sim \pi(S_0), R_0 = r(S_0, A_0), S_1 \sim P(S_0, A_0), \gamma_0 = \gamma(S_0, A_0, S_1), \cdots$. The return is $G := \sum_{t \geq 1} (\Pi_{1 \leq i < t} \gamma_i) R_t$.

In policy evaluation we are interested in evaluating the state-value function of a policy defined as $v_\pi(s) := \mathbb{E}_{\mathcal{M},\pi}[G | S_0 = s]$ where $\mathcal{M}$ and π must satisfy the constraint $\lim_{t \to \infty} \mathbb{E}_{\mathcal{M},\pi}[\Pi_{1 \leq i < t} \gamma_i | S_0 = s] = 0$ for all $s \in \mathcal{S}$. It can be shown that $v_\pi = (I - P_{\pi,\gamma})^{-1} r_\pi$. Given a representation function $\phi : \mathcal{S} \to \mathbb{R}^d$ we

can arrange the obtained representations into $\Phi \in \mathbb{R}^{|\mathcal{S}| \times d}$ and, using weights $w \in \mathbb{R}^d$, state values can be estimated as $v_w := \Phi w$. We can use gradient descent (GD) or temporal-difference learning (TD) to predict state values.

GD: Suppose a sample in the form of $((\phi_i, g_i))_{i=1}^n$ of the representation and return is available where a state s_i is sampled independently from a sampling distribution d_S, $\phi_i := \phi(s_i)$, and g_i is a sample of return following from s_i. Similar to a regression problem, gradient descent (GD) can be applied on this sample to find a predictor. In this context, GD updates are $\hat{w}_{GD}^0 := 0$, $\hat{w}_{GD}^t := \hat{w}_{GD}^{t-1} - \eta(\frac{1}{n}\sum_{i=1}^n(\phi_i^\top \hat{w}_{GD}^{t-1} - g_i)\phi_i)$.

TD: Now suppose a sample is available in the form of $((\phi_i, r_i, \phi_i', \gamma_i))_{i=1}^n$ where each tuple in the sample is called a transition and $s_i \sim d_S, \phi_i := \phi(s_i), a_i \sim \pi(s_i), r_i := r(s_i, a_i), s_i' \sim P(s_i, a_i), \phi'(i) := \phi(s_i'), \gamma_i := \gamma(s_i, a_i, s_i')$. TD updates are $\hat{w}_{TD}^0 := 0$, $\hat{w}_{TD}^t := \hat{w}_{TD}^{t-1} - \eta(\frac{1}{n}\sum_{i=1}^n(\phi_i^\top \hat{w}_{TD}^{t-1} - (r_i + \gamma_i {\phi_i'}^\top \hat{w}_{TD}^{t-1}))\phi_i)$.

7.3 Convergence Rate of TD

In this section we use a standard approach similar to Chapter 4 to develop a convergence rate applicable to both GD and TD. We will then describe an approach for creating synthetic reward vectors that will result in different convergence rates for GD and TD.

Define the expected weights $w_{GD}^t := \mathbb{E}[\hat{w}_{GD}^t]$ and $w_{TD}^t := \mathbb{E}[\hat{w}_{TD}^t]$ where the expectations are over the draw of the samples. The behavior of the expected weights is as follows (Sutton & Barto, 2018).

$$
\begin{aligned}
w_{GD}^{t+1} &:= w_{GD}^t - \eta\Phi^\top D(\Phi w_{GD}^t - v_\pi) \\
&= (I - \eta\Phi^\top D\Phi)w_{GD}^t + \eta\Phi^\top D v_\pi \qquad (7.1) \\
w_{TD}^{t+1} &:= w_{TD}^t - \eta\Phi^\top D(\Phi w_{TD}^t - (r_\pi + P_{\pi,\gamma}\Phi w_{TD}^t)) \\
&= (I - \eta\Phi^\top D(I - P_{\pi,\gamma})\Phi)w_{TD}^t + \eta\Phi^\top D r_\pi \qquad (7.2)
\end{aligned}
$$

where $D := \text{diag}(d_S)$. The right-hand side of both Equations (7.1) and (7.2) can be written as $(I - \eta A)w^t + \eta b$ for a matrix A and a vector b. Assume

A is diagonalizable. We can then write $A = Q\Sigma^2 Q^{-1}$ where Q and Σ can be complex and columns of Q, denoted as q_i, represent the normalized eigenvectors of A and diagonal elements of Σ^2, denoted as σ_i^2 and sorted by magnitude in non-increasing order, represent the eigenvalues. We also define $\Sigma^\dagger$ as the pseudo-inverse of the diagonal matrix Σ and C as the operator norm of Q. An important vector for the analysis is $b' := \Sigma^\dagger Q^{-1} b$. For the GD update, A is the covariance matrix of the representation, b is the correlation vector between the values and the features, and $C = 1$.

Theorem 7.3.1. *(Adapted from Pan et al. (2017). Proof in Appendix D.) If A is diagonalizable and $0 < \eta < 1/|\sigma_1^2|$ the updates $w^{t+1} := (I - \eta A)w^t + \eta b$ starting from $w^0 := 0$ converge to $w^* := \lim_{t\to\infty} w^t$ at the following rate:*

$$\left\|w^{t+1} - w^*\right\| \leq \sqrt{C \sum_{j=1}^{d} \left|\frac{(1-\eta\sigma_j^2)^{(t+1)}}{\sigma_j}\right|^2 \left|b'_j\right|^2} \tag{7.3}$$

$$\left\|(b - Aw^{t+1}) - (b - Aw^*)\right\| \leq \sqrt{C \sum_{j=1}^{d} |(1-\eta\sigma_j^2)^{(t+1)}|^2 \left|b'_j\right|^2} \tag{7.4}$$

The main takeaway of the theorem is that convergence is fast if the first elements of b' are large. This vector plays a similar role as $\tilde{w}$ in the Chapter 4. In GD b' is the projection of v_π on the left singular vectors of Φ and the first elements are the projections of v_π on singular vectors with large singular values. A large value of these elements means that state values change mostly along directions of top principal components of the representation. TD results in a different matrix A and vector b and therefore a different b' compared to GD.

Non-diagonalizable matrices: Similar insights on convergence in different directions are known for non-diagonalizable matrices using Jordan form. While Jordan form is a useful theoretical tool, its fundamental numerical instability precludes its use in experiments. In this case a non-diagonlizable matrix can be approximated arbitrarily closely with a diagonlizable matrix that can be then used in the analysis (Horn & Johnson, 2012). Although this approximation, however small, can drastically change the asymptotic behavior

of an optimization algorithm (Hirsch & Smale, 1974) (Chapter 16), the linear appearance of A in updates (7.1) and (7.2) along with zero initialization of w suggests that a good approximation of A can closely reflect the trajectory of w during practical training times. We have not encountered non-diagonalizable A in our experiments.

The following proposition allows us to create real-valued reward vectors that result in convergence rates dictated by certain eigenvalues. Using this proposition we can create MDPs where the policy evaluation algorithm (either GD or TD) has a certain convergence rate.

Proposition 7.3.2 (Proof in Appendix D)**.** *Assume D is invertible. For all i such that $|\sigma_i| > 0$, the convergence rate is*

$$\left\|(b - Aw^{t+1}) - (b - Aw^*)\right\| \leq C' |(1 - \eta\sigma_i^2)^{(t+1)}|$$

for a C' whose value is independent of the reward vector if either

1. *the updates have the form of Equation* (7.1) *and the reward vector is set to $r^i_{GD} := (I - P_{\pi,\gamma})D^{-1}\Phi(\Phi^\top\Phi)^\dagger(Re(q_i)Re(\sigma_i) - Im(q_i)Im(\sigma_i))$, or*

2. *the updates have the form of Equation* (7.2) *and the reward vector is set to $r^i_{TD} := D^{-1}\Phi(\Phi^\top\Phi)^\dagger(Re(q_i)Re(\sigma_i) - Im(q_i)Im(\sigma_i))$ and $(Re(q_i)Re(\sigma_i) - Im(q_i)Im(\sigma_i))$ is in the column space of Φ.*

Reward vectors $r^1_{GD}, r^2_{GD}, \cdots$ above result in different MDPs with different convergence rates for GD and reward vectors $r^1_{TD}, r^2_{TD}, \cdots$ above result in a different MDPs with different convergence rates for TD. A low index i corresponds to a larger eigenvalue and thus faster convergence. In the next section we will verify these convergence rates and study the generalization performance of GD and TD on these MDPs.

7.4 Experiments on Optimization and Generalization

The previous section introduced a set of reward vectors to determine the convergence rate of GD and TD. The first goal of the current section is to verify

this numerically. These reward vectors are also analogous to the sequence of tasks in Chapter 4 that resulted in different convergence rates with GD in regression. In Chapters 3 and 5 we saw that generalization from a small sample was easier on tasks where the expected convergence rate was faster. Our second goal in this section is to see if this trend occurs in policy evaluation with GD and TD.

7.4.1 Experiment Setup

This section describes the testbed, evaluation criteria, and the hypotheses in the experiments. Note that we will use synthetic reward vectors through this chapter to discuss convergence and generalization. Since creating the reward vector for every experiment requires perfect knowledge of the environment transitions in matrix form and the computationally expensive eigendecomposition of a possibly asymmetric matrix A, we will restrict the experiments, along with the claims, to small domains. We will use tile-coding and RBF features. Our goal is not to reduce the dimensionality of the state space as the environments are already small but to encode a notion of closeness in the state space into the representation and to allow updates in one state to affect predictions in other states.

The first environment is a directional gridworld portrayed in Figure 7.1. Each position in the 6 by 6 gridworld corresponds to 4 states based on the agent's direction except the bottom right corner which is the terminal position. Thus, excluding the terminal position, we have a $(6 \times 6 - 1) \times 4 = 140$ states, each of which can be presented with a triplet (i, j, k) to include the current row, column, and direction where $i, j \in \{0, \cdots, 5\}$ and $k \in \{0, \cdots, 3\}$. The starting distribution is uniform. There are three actions for turning left, turning right, and going forward. The first two actions change the direction and the third the position. If the agent attempts to move past the edge of the gridworld it will remain in place and if it moves to the terminal position it will transition to a state sampled from the uniform distribution. The discount factor is 0 for transition into the terminal position and 0.99 otherwise. The evaluated policy gives equal probability of $0.\bar{3}$ to every action in every state. We will create

one testbed (Grid-Tile) by applying tile-coder on the aforementioned triplets with 8 tilings, each of which has $3 \times 3 \times 2$ tiles. The other testbed (Grid-RBF) has RBF features with bandwidth 1.0 and 140 centers set on each triplet and applied to the triplets. We do not center or normalize the representations in this chapter.

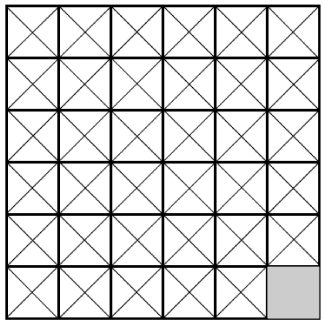


Figure 7.1: The directional gridworld environment. There are four possible directions at each location. Each triangle depicts a state. Left and right actions will move the agent to the next triangle within the same grid. Forward action moves the agent to the same direction at next grid.

The second environment is a random walk chain. There are a total of 200 states presented with an integer in $\{0, \cdots, 199\}$. The starting distribution is uniform. There are two action, left and right. At state i, a number i' is sampled uniformly randomly from $\{i-20, \cdots, i-1\}$ if the left action is taken and from $\{i+1, \cdots, i+20\}$ if the right action is taken. Then, if $i' \in \{0, \cdots, 199\}$, the agent will transition to state i' with a discount factor of 0.99. Otherwise, the agent will transition to state sampled uniformly from $\{0, \cdots, 199\}$ with a discount factor of 0. The evaluated policy at every state takes the left and right actions with probabilities 0.8 and 0.2 respectively. We will create a tile-coding testbed (Chain-Tile) with 10 tilings, each with 20 tiles, and an RBF testbed (Chain-RBF) with bandwidth 2.0 and 200 centers set on $\{0, \cdots, 199\}$. The encodings are applied to the state index.

There are two evaluation criteria in the experiments. The first one is Mean Squared Value Error (MSVE), defined as $\text{MSVE}(w) := \sum_{s \in \mathcal{S}} d_S(s)(v_w(s) - v_\pi(s))^2$ which is a common evaluation criterion for policy evaluation. The second one, which we report for convergence rate results, is Norm of the Expected

Update NEU$(w) = \|(b - Aw^{t+1})\|^2$, the error that the theoretical convergence rate studies.[1]

We will study the following hypotheses:

H1 The sequence of rewards in Proposition 7.3.2 results in non-increasing order of expected convergence rates for TD.

H2 The sequence of rewards in Proposition 7.3.2 results in non-increasing order of generalization performance from a small sample for GD and TD.

H3 The reward sequence for GD will not necessarily create the same trends in convergence and generalization of TD and vice-versa.

H4 The trend in generalization with TD also occurs if the states are sampled from the policy's trajectory rather than independently.

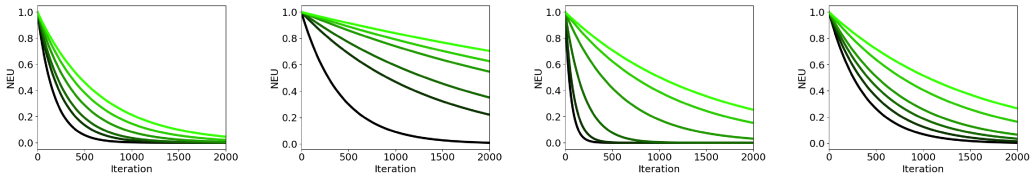
7.4.2 Results

We will now present the results for the four hypotheses in the previous section on the four testbeds and using the two evaluation criteria. For each evaluation criterion, we scale the reward vectors such that all the compared curves start at one. Therefore the plots for MSVE and NEU do not correspond to the same training process. The sampling distribution d_S is uniform unless stated otherwise.

H1: We ask if the sequences of rewards in this chapter show the same pattern in convergence as we saw in Chapter 4. Specifically, we want to know if convergence for r^i_{TD} is faster than for r^j_{TD} if $\sigma_i > \sigma_j$. Note that, unlike Theorem 4.1.1, the theorem in this chapter is not an equality but a bound in the case of TD and so needs numerical verification. We trained weight vectors with the expected updates in Equation (7.2) on the sequence of reward vectors r^i_{TD} with different values of i to create the error curves. There is no random sampling in this experiment. The plots in Figures 7.2 and 7.3 show the NEU and MSVE

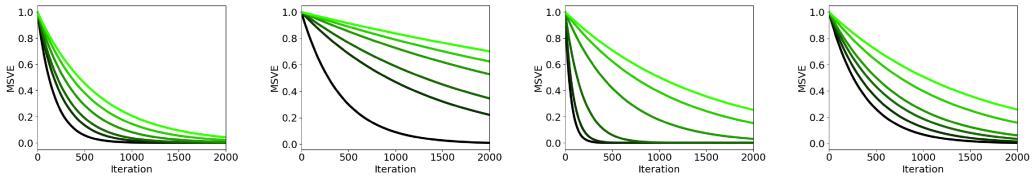
[1]The abbreviation NEU typically refers to the Norm of the Expected *TD* Update (Sutton et al., 2009) where b and A are the ones used in Equation (7.2). We use the more general measure Norm of the Expected Update where the update can be either GD or TD.

error curves for the sequences of reward vectors for TD. Comparing the curves within each set verifies the hypothesis.



(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

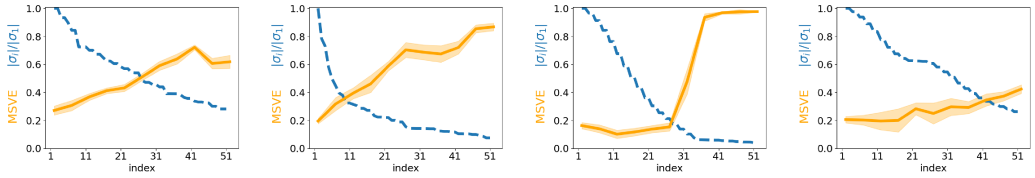
Figure 7.2: NEU of TD expected updates on r^i_{TD} for $i \in \{1, 11, 21, 31, 41, 51\}$. Darker shades correspond to smaller values of i and show faster convergence.



(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

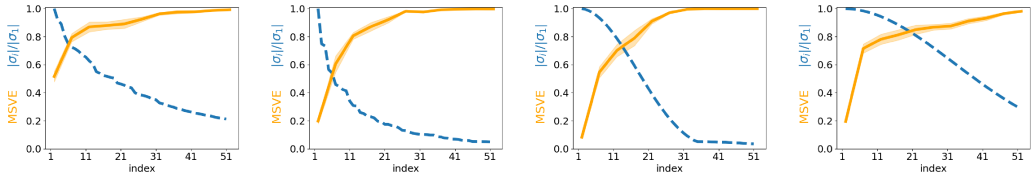
Figure 7.3: MSVE of TD expected updates on r^i_{TD} for $i \in \{1, 11, 21, 31, 41, 51\}$. Darker shades correspond to smaller values of i and show faster convergence.

H2: Now we ask if the pattern in generalization performance across these sequences of rewards is similar to the pattern in Chapter 3. The hypothesis is that generalization from a small sample with TD is easier for r^i_{TD} compared to r^j_{TD} if $\sigma_i > \sigma_j$ and similarly for GD. For each reward vector in the sequence, we created a sample with 100 items, where the items are pairs of state representation and sample return for GD and transition for TD, then trained weight vectors on the sample with step-size 0.01, and logged the error through the iterations. The errors that we report are measured on the overall state space and not the sample. Figures 7.4 and 7.5 show the lowest MSVE through the iterations for TD and GD and for each reward vector. There is a clear increasing pattern in the generalization error as the reward vector corresponds to smaller eigenvalues, verifying the hypothesis. Figures 7.6 and 7.7 show full error curves for some of the reward vectors in the sequence.



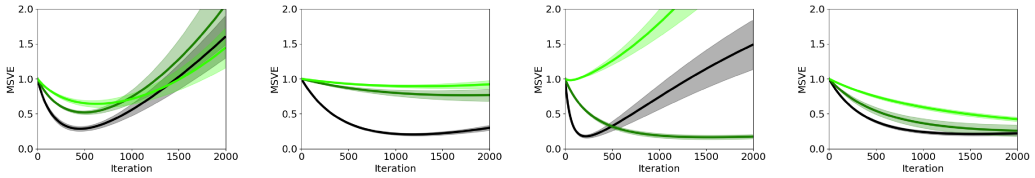
(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

Figure 7.4: Generalization error of TD updates on a batch of 100 items using r^i_{TD} for $i \in \{1, 6, 11, \cdots, 51\}$. The point on the orange curves at horizontal position i shows the minimum MSVE through the iterations, averaged over 10 runs, with the sample drawn independently in each run, when using r^i_{TD}. The shade shows standard errors. The dashed blue curve shows the eigenvalue spectrum. The orange curve rises, showing that generalization is harder for reward vectors corresponding to smaller eigenvalues.



(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

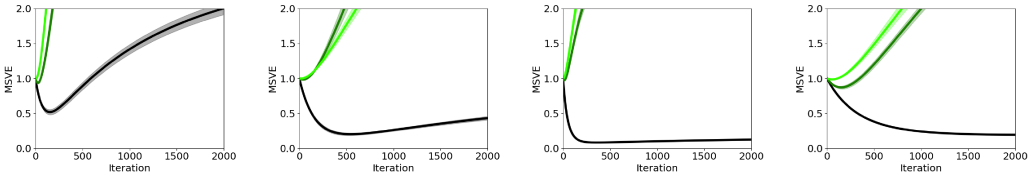
Figure 7.5: Generalization error of GD updates on a batch of 100 items using r^i_{GD} for $i \in \{1, 6, 11, \cdots, 51\}$. The orange curve rises, showing that generalization is harder for reward vectors corresponding to smaller eigenvalues.



(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

Figure 7.6: Generalization error curves of TD updates on a batch of 100 items using r^i_{TD} for $i \in \{1, 26, 51\}$. Darker shades correspond to smaller values of i and show better generalization.

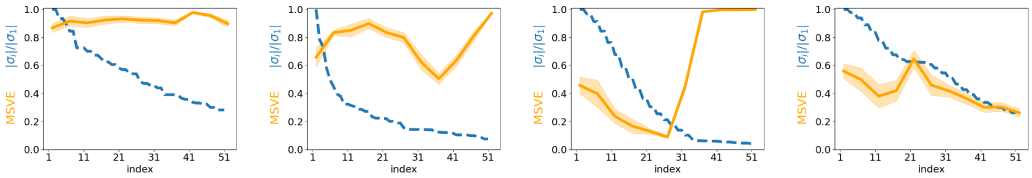
H3: Now we ask if the trend in generalization performance is contingent on the training algorithm. The hypothesis is that generalization error of GD does not necessarily follow an increasing pattern across the sequence of reward vectors for TD and vice-versa. The experiment here is similar to the one for the second hypothesis except we use GD as training algorithm for the reward vector



(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

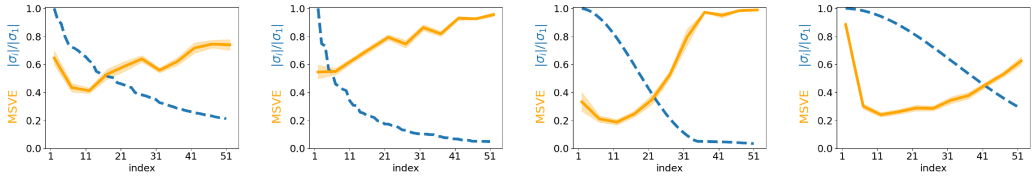
Figure 7.7: Generalization error curves of GD updates on a batch of 100 items using r^i_{GD} for $i \in \{1, 26, 51\}$. Darker shades correspond to smaller values of i and show better generalization.

sequence r^i_{TD} and vice-versa. The results in Figures 7.8 and 7.9 verify this hypothesis. The ascending pattern in the orange curve is absent in some cases and sometimes even the reward vector corresponding to the largest eigenvalue results in poor generalization. An example is GD on Grid-Tile in Figure 7.8 (a). Figures 7.10 and 7.11 show the full curves for some of the reward vectors. The curves for GD on Grid-Tile in 7.10 (a) show that the generalization error of GD on all these reward vectors, even the one corresponding to the largest eigenvalue, immediately grows beyond 1.0 since GD has only memorized the sample and has not learned generalizable patterns. Compare this with Figure 7.6 (a) where TD could find generalizable patterns on the same reward vectors.



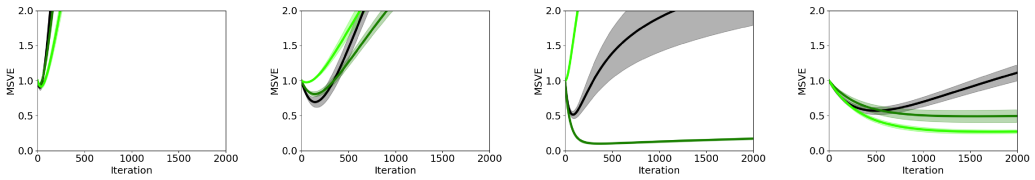
(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

Figure 7.8: Generalization error of **GD** updates on a batch of 100 items using r^i_{TD} for $i \in \{1, 6, 11, \cdots, 51\}$. The trend in H2 does not consistently repeat here.



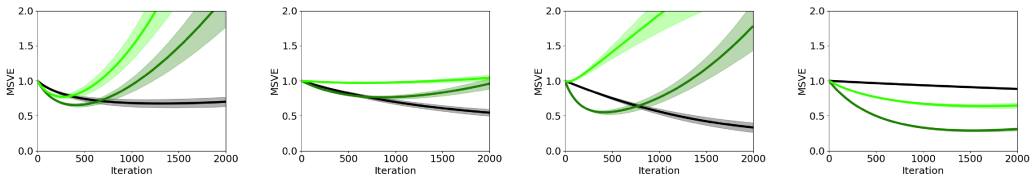
(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

Figure 7.9: Generalization error of **TD** updates on a batch of 100 items using r^i_{GD} for $i \in \{1, 6, 11, \cdots, 51\}$. The trend in H2 does not consistently repeat here.



(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

Figure 7.10: Generalization error curves of **GD** updates on a batch of 100 items using r^i_{TD} for $i \in \{1, 26, 51\}$. Darker shades correspond to smaller values of i and not necessarily better generalization.

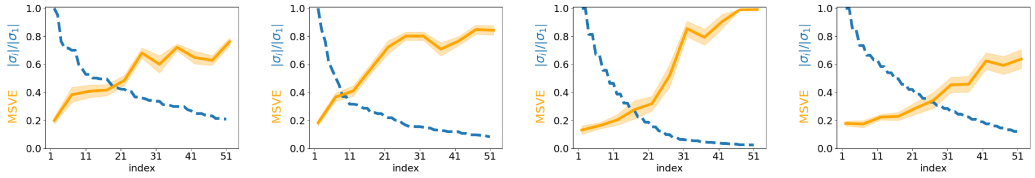


(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

Figure 7.11: Generalization error curves of **TD** updates on a batch of 100 items using r^i_{GD} for $i \in \{1, 26, 51\}$. Darker shades correspond to smaller values of i and not necessarily better generalization.

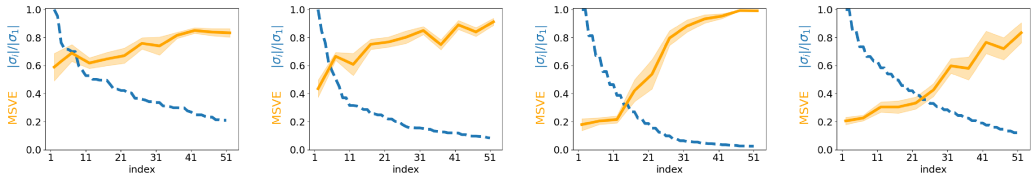
H4: The experiment for H2 had the states sampled independently from the uniform distribution. We ask if the same pattern in generalization of TD also occurs if the states in the sample used for training are gathered in a trajectory by following the evaluated policy. The evaluation is still on the whole state space. There are two points of difference here. One is the change in the sampling distribution from uniform to the state distribution induced by the policy (the policy's stationary distribution) and the second is the temporal dependence of the states through a trajectory. Thus, we use the policy's

stationary distribution in the process of creating the synthetic reward vectors. Then to separate the points of difference we run a first experiment by sampling the states independently from this distribution and in the second experiment we sample the states by following a trajectory. Figures 7.12 and 7.13 show the results for these two experiments. The increasing trend in generalization error for rewards corresponding to smaller eigenvalues extends to both experiments. However, the differences are sometimes smaller when the states in the sample are temporally dependent, suggesting that these cases would require a special theoretical analysis to take temporal dependence into account.



(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

Figure 7.12: Generalization error of TD updates on a batch of 100 items using r^i_{TD} for $i \in \{1, 6, 11, \cdots, 51\}$ and sampled independently from the policy's stationary distribution.



(a) Grid-Tile (b) Grid-RBF (c) Chain-Tile (d) Chain-RBF

Figure 7.13: Generalization error of TD updates on a batch of 100 items using r^i_{TD} for $i \in \{1, 6, 11, \cdots, 51\}$ and sampled by following the policy's trajectory.

7.5 Discussion

This chapter empirically explored generalization in the context of policy evaluation. We extended the previous chapters' convergence rates on GD so that it applies to both GD and TD. Then we described a process for creating a reward vector that results in a certain expected convergence rate. In experiments on four testbeds we saw that

1. The theoretical pattern in expected convergence rates happens in practice as well.

2. Generalization of GD and TD from a small sample is better when the expected convergence rate is faster.

3. The trend in generalization is contingent in the training algorithm and a training algorithm (GD or TD) does not consistently generalize better if the expected convergence rate of the other algorithm is fast.

4. The trend in generalization of TD also occurs when the states used for training are not indepdendent but are sampled from a trajectory by following the evaluated policy.

We reiterate here that our testbeds were small and making claims about larger environments requires a separate study. Also, the connection between fast convergence and good generalization in this chapter is a correlation, and we do not claim that the fast expected convergence causes good generalization. Theoretical results on the reason behind the observed trends in generalization error can answer whether there is a causal relationship between the two. Finally, note that in this chapter we considered the best generalization performance through the training rather than the final performance. The learning curves sometimes cross each other, suggesting that the observed trends in generalization do not appear if we consider the final performance.

Chapter 8

Conclusion

Motivated by recent challenges in the study of generalization we asked

> What relationship between the representation and the target makes generalization from a small sample easier?

Representation alignment was our partial answer to this question. In this final chapter we will review our findings in this regard and as well as the limitations and open questions.

8.1 List of Findings

This section recaps the findings on representation alignment throughout this document.

In regression, gradient descent generalizes well from a small batch if representation alignment is high. In Chapter 3 we showed this theoretically on a well-specified regression model and demonstrated it on synthetic tasks. Then, in Chapter 5 we verified this in a more practical scenario by comparing generalization on representations with different degrees of representation alignment.

In binary classification, a solution with low generalization error can be obtained from a small batch if representation alignment is high. We first showed this in Chapter 3 by first providing a generalization bound

using margin theory and then through synthetic experiments. Then we compared generalization with different representations in a practical scenario in Chapter 5 and observed a similar pattern.

In practice, common optimizers tend to converge faster if representation alignment is high. We showed this for a variety of optimizers by comparing representations with different level of representation alignment in Chapter 5.

Neural networks find hidden representations with high representation alignment on the training task. We showed in Chapter 5 that, across a wide range of training setups, trained neural networks learn hidden representations that has higher representation alignment on the training task compared to the input features and the hidden representation at initialization.

In fully-connected networks, hidden layers closer to the output have higher representation alignment on the training task. This was shown in Chapter 5 on multiple regression and binary classification tasks.

In a typical feature transfer scenario in object classification, pretrained neural networks have high representation alignment on the downstream task. We compared pretrained neural network hidden representations and several handcrafted features in object classification in Chapter 5 and observed that neural network representations have higher representation alignment for a wide range of thresholds.

The prior knowledge of high representation alignment can be used to improve performance in domain adaptation. In Chapter 6 we developed a regularizer to enforce a prior knowledge related to representation alignment. We then verified the efficacy of this regularizer in synthetic and practical experiments.

In policy evaluation, gradient descent and temporal-difference learning generalize well from a small sample when their expected convergence rate is high. We verified this in Chapter 7 on experiments on four small testbeds and with two evaluation criteria.

8.2 Limitations and Future Work

The findings presented in this thesis and their limitations open up several new directions for further exploration.

The definition of representation alignment is restricted to centered representation and targets. Removing this restriction would make both the definition and the theory for regression and classification more involved and we did not pursue it here. The theory for regression was based on a model by Hastie et al. (2022) with centered representation and targets. For the classification generalization bound we added this restriction to ensure that an unbiased estimate of a quantity of interest can be obtained from the empirical kernel. Without this assumption, one would have to either consider the effect of using a non-centered kernel or the possible bias introduced by centering an empirical kernel (Kornblith et al., 2019a).

Representation alignment is defined for vector inputs and the results are for linear models (on possibly non-linear representation functions). Some directions for extensions are providing theoretical or empirical results for nonlinear models or data that is not available in vector form, such as sequential or graph data. First-order approximations are widely used to approximate generalization bounds and estimates and convergence rates to nonlinear models such as wide neural networks (Jacot et al., 2018).

The generalization estimate in regression requires a well-specified model and sub-gaussian input noise. Hastie et al. (2022) relaxed the well-specified assumption in their results with a more elaborate analysis. The sub-gaussian assumption in our result is only added as a convenient way to ensure the operator norm of the covariance matrix of input noise is bounded with high probability. Depending on the context, one can explore whether assumptions

other that sub-gaussian input noise may guarantee this property. The regression generalization result is also for ridge estimator while most of the empirical results in the document are obtained with gradient descent. Ali et al. (2019) theoretically explored the connection between the performance of ridge estimator with a certain ridge parameter and gradient descent with a certain number of iterations. A next step is to see if such connection holds in our setup.

We observed that the fast expected convergence of gradient descent in the case of high representation alignment extends to the practical case of mini-batch training and some other common optimizers in our experiments. A theoretical characterization of this behavior is a future direction. Aside from the expected convergence rate that we already reviewed, we are aware of the improved convergence rate by Zou et al. (2021b) for tail-averaged stochastic gradient descent that relies on an assumption similar to representation alignment.

A study on when and why representation alignment emerges in neural network hidden layers and when it transfers to a different task is another venue for exploration. Starting points for this direction are characterizing possible connections between representation alignment and the well-studied weight alignment phenomenon in classification tasks with linear activation (Ji & Telgarsky, 2019) and the neural collapse phenomenon (Papyan et al., 2020) and the conditions for emergence and transfer of neural collapse (Zhu et al., 2021; Galanti et al., 2021).

The chapter on policy evaluation is much limited in scope as there is no theoretical characterization on the trends in generalization and the experiments are on small testbeds due to computational complexity and requirements on perfect knowledge about environment transitions in matrix form. One possible approach to study the pattern in generalization on larger environments is to design a meta-objective to improve the convergence rate of TD in terms of NEU. One can then test if pre-training with such meta-objective helps with generalization from a small sample.

8.3 Summary

This thesis revolved around *representation alignment*, which we defined as a measure of variation of the target in directions where the representation is more elongated. Throughout the document we gave a rigorous definition of representation alignment and studied its role in optimization and generalization in certain classification, regression, and policy evaluation models. We also studied when high representation alignment emerges and how prior knowledge of high representation alignment can be used in domain adaptation.

References

Acuna, D., Zhang, G., Law, M. T., & Fidler, S. (2021). F-domain adversarial learning: Theory and algorithms. *International Conference on Machine Learning*. 46, 63, 64, 75

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., & Goldberg, Y. (2017). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *International Conference on Learning Representations*. 29

Ali, A., Kolter, J. Z., & Tibshirani, R. J. (2019). A continuous-time view of early stopping for least squares regression. *International Conference on Artificial Intelligence and Statistics*. 97

Amari, S.-i., Ba, J., Grosse, R., Li, X., Nitanda, A., Suzuki, T., Wu, D., & Xu, J. (2021). When does preconditioning help or hurt generalization? *International Conference on Learning Representations*. 12, 23

Amit, R., Meir, R., & Ciosek, K. (2020). Discount factor as a regularizer in reinforcement learning. *International Conference on Machine Learning*. 80

Arora, S., Du, S., Hu, W., Li, Z., & Wang, R. (2019). Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *International Conference on Machine Learning*. 7, 26, 51

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, *10*(7). 29

Baratin, A., George, T., Laurent, C., Hjelm, R. D., Lajoie, G., Vincent, P., & Lacoste-Julien, S. (2021). Implicit regularization via neural feature alignment. *International Conference on Artificial Intelligence and Statistics*. 51

Bartlett, P. L., Long, P. M., Lugosi, G., & Tsigler, A. (2020). Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*. 2, 11

Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., & Glass, J. (2017). What do neural machine translation models learn about morphology? *Annual Meeting of the Association for Computational Linguistics*. 29

Belkin, M., Hsu, D., Ma, S., & Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*. 2

Belkin, M., Ma, S., & Mandal, S. (2018). To understand deep learning we need to understand kernel learning. *International Conference on Machine Learning*. 2

Bellemare, M., Dabney, W., Dadashi, R., Ali Taiga, A., Castro, P. S., Le Roux, N., Schuurmans, D., Lattimore, T., & Lyle, C. (2019). A geometric perspective on optimal representations for reinforcement learning. *Advances in Neural Information Processing Systems*. 80

Benaim, S., & Wolf, L. (2017). One-sided unsupervised domain mapping. *Advances in neural information processing systems*. 63

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine learning*. 46, 63

Bengio, E., Pineau, J., & Precup, D. (2020). Interference and generalization in temporal difference learning. *International Conference on Machine Learning*. 80

Bengio, E., Pineau, J., & Precup, D. (2021). Correcting momentum in temporal difference learning. *arXiv*. 80

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8). 28, 29

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., & Lamere, P. (2011). The million song dataset. *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*. 50

Bhandari, J., Russo, D., & Singal, R. (2018). A finite time analysis of temporal difference learning with linear function approximation. *Conference on learning theory*. 80

Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer. 1, 52

Blanchard, G., Lee, G., & Scott, C. (2011). Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*. 46

Bottou, L. (2009). Curiously fast convergence of some stochastic gradient descent algorithms. *Proceedings of the symposium on learning and data science, Paris*. 41

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., & Krishnan, D. (2017). Unsupervised pixel-level domain adaptation with generative adversarial networks. *IEEE conference on computer vision and pattern recognition*. 63

Bunea, F., & Xiao, L. (2015). On the sample covariance matrix estimator of reduced effective rank population matrices, with applications to fpca. 110

Canatar, A., Bordelon, B., & Pehlevan, C. (2021). Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications*, *12*. x, 8

Castellani, T., & Cavagna, A. (2005). Spin-glass theory for pedestrians. *Journal of Statistical Mechanics: Theory and Experiment*. 8

Chen, S., Devraj, A., Busic, A., & Meyn, S. (2020). Explicit mean-square error bounds for monte-carlo and linear stochastic approximation. *International Conference on Artificial Intelligence and Statistics*. 80

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., & Jégou, H. (2018). Word translation without parallel data. *International Conference on Learning Representations*. 47, 64, 74

Cortes, C., Mohri, M., & Rostamizadeh, A. (2012). Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, *13*. 8, 18, 20, 21

Courty, N., Flamary, R., Habrard, A., & Rakotomamonjy, A. (2017). Joint distribution optimal transportation for domain adaptation. *Advances in Neural Information Processing Systems*. 63

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., & Kandola, J. (2001). On kernel-target alignment. *Advances in neural information processing systems*. 8

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Conference on Computer Vision and Pattern Recognition*. 35

Dalvi, F., Durrani, N., Sajjad, H., Belinkov, Y., Bau, A., & Glass, J. (2019). What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. *AAAI Conference on Artificial Intelligence*. 30

Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., & Courty, N. (2018). Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. *European Conference on Computer Vision*. 64

Dayan, P. (1993). Improving generalization for temporal difference learning: The successor representation. *Neural computation*. 80

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 28

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 74

Dohare, S., Sutton, R. S., & Mahmood, A. R. (2021). Continual backprop: Stochastic gradient descent with persistent randomness. *arXiv*. 80

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*. 1

Fanaee-T, H., & Gama, J. (2014). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*. 50

Farahmand, A.-m. (2011). Regularization in reinforcement learning. 80

Farebrother, J., Machado, M. C., & Bowling, M. (2018). Generalization and regularization in dqn. *arXiv*. 80

François-Lavet, V., Rabusseau, G., Pineau, J., Ernst, D., & Fonteneau, R. (2019). On overfitting and asymptotic bias in batch reinforcement learn-

ing with partial observability. *Journal of Artificial Intelligence Research*. 80

Galanti, T., György, A., & Hutter, M. (2021). On the role of neural collapse in transfer learning. *International Conference on Learning Representations*. 97

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*. 46, 47, 65, 69, 76

Ghifary, M., Kleijn, W. B., Zhang, M., Balduzzi, D., & Li, W. (2016). Deep reconstruction-classification networks for unsupervised domain adaptation. *European conference on computer vision*. 63

Gholami, B., Sahu, P., Rudovic, O., Bousmalis, K., & Pavlovic, V. (2020). Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*. 46

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*. 75

Graf, F., Kriegel, H.-P., Schubert, M., Pölsterl, S., & Cavallaro, A. (2011). 2d image registration in ct images using radial image descriptors. *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 49

Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., & Douze, M. (2021). Levit: A vision transformer in convnet's clothing for faster inference. *Proceedings of the IEEE/CVF international conference on computer vision*, 12259–12269. 35

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. *International conference on algorithmic learning theory*. 21

Guillemin, V., & Pollack, A. (2010). *Differential topology* (Vol. 370). American Mathematical Soc. 63

Gulrajani, I., & Lopez-Paz, D. (2021). In search of lost domain generalization. *International Conference on Learning Representations*. 46

Gupta, H., Srikant, R., & Ying, L. (2019). Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning. *Advances in Neural Information Processing Systems*. 80

Hastie, T., Montanari, A., Rosset, S., & Tibshirani, R. J. (2022). Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*. 2, 11–15, 96, 112

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *IEEE conference on computer vision and pattern recognition*. 1, 35

Hermann, K., & Lampinen, A. (2020). What shapes feature representations? exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems*. 29

Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Coursera*. 42

Hirsch, M., & Smale, S. (1974). Differential equations, dynamical systems, and linear algebra. *Pure and Appl. Math.* 84

Horn, R. A., & Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press. 83

Hui, L., & Belkin, M. (2020). Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *International Conference on Learning Representations*. 56

Imani, E., Hu, W., & White, M. (2022). Representation alignment in neural networks. *Transactions on Machine Learning Research*. iii, 48, 49, 51

Imani, E., Zhang, G., Li, R., Luo, J., Poupart, P., Torr, P. H., & Pan, Y. (2024). Label alignment regularization for distribution shift. *Journal of Machine Learning Research*. iii

Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*. 2, 26, 96

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., & Kavukcuoglu, K. (2017). Reinforcement learning with unsupervised auxiliary tasks. *International Conference on Learning Representations*. 80

Ji, Z., & Telgarsky, M. (2019). Gradient descent aligns the layers of deep linear networks. *International Conference on Learning Representations*. 97

Johansson, F. D., Sontag, D., & Ranganath, R. (2019). Support and invertibility in domain-invariant representations. *The 22nd International Conference on Artificial Intelligence and Statistics*. 47, 64

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models. *arXiv*. 1

Kawaguchi, K., Kaelbling, L. P., & Bengio, Y. (2017). Generalization in deep learning. *arXiv*. 1

Keller, P. W., Mannor, S., & Precup, D. (2006). Automatic basis function construction for approximate dynamic programming and reinforcement learning. *International Conference on Machine Learning*. 80

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 39, 42

Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019a). Similarity of neural network representations revisited. *International Conference on Machine Learning*. 20, 30, 96

Kornblith, S., Shlens, J., & Le, Q. V. (2019b). Do better imagenet models transfer better? *Conference on Computer Vision and Pattern Recognition*. 28

Krizhevsky, A., et al. (2009). Learning multiple layers of features from tiny images. 1

Le Lan, C., Tu, S., Oberman, A., Agarwal, R., & Bellemare, M. G. (2022). On the generalization of representations in reinforcement learning. *International Conference on Artificial Intelligence and Statistics*. 80, 81

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (2002). Efficient backprop. In *Neural networks: Tricks of the trade*. 30

Lee, C.-Y., Batra, T., Baig, M. H., & Ulbricht, D. (2019). Sliced wasserstein discrepancy for unsupervised domain adaptation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 64

Long, M., Cao, Y., Wang, J., & Jordan, M. (2015). Learning transferable features with deep adaptation networks. *International conference on machine learning*. 47, 64

Long, M., Zhu, H., Wang, J., & Jordan, M. I. (2016). Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*. 63

Long, M., Zhu, H., Wang, J., & Jordan, M. I. (2017). Deep transfer learning with joint adaptation networks. *International conference on machine learning*. 63, 75

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *International Conference on Computer Vision*. 35

Lyle, C., Rowland, M., & Dabney, W. (2022a). Understanding and preventing capacity loss in reinforcement learning. *International Conference on Learning Representations*. 80

Lyle, C., Rowland, M., Dabney, W., Kwiatkowska, M., & Gal, Y. (2022b). Learning dynamics and generalization in deep reinforcement learning. *International Conference on Machine Learning*. 80

Lyle, C., Zheng, Z., Nikishin, E., Pires, B. A., Pascanu, R., & Dabney, W. (2023). Understanding plasticity in neural networks. *International Conference on Machine Learning*. 80

Mahadevan, S., & Maggioni, M. (2007). Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*. 80

Mansour, Y., Mohri, M., & Rostamizadeh, A. (2009). Domain adaptation: Learning bounds and algorithms. *Annual Conference on Learning Theory*. 46

Marcus, G. (2018). Deep learning: A critical appraisal. 28

McDiarmid, C., et al. (1989). On the method of bounded differences. *Surveys in combinatorics*. 18

Menache, I., Mannor, S., & Shimkin, N. (2005). Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*. 80

Meng, Z., Li, J., Gong, Y., & Juang, B.-H. (2018). Adversarial teacher-student learning for unsupervised domain adaptation. *IEEE International Conference on Acoustics, Speech and Signal Processing*. 64

Meyer, D., Degenne, R., Omrane, A., & Shen, H. (2014). Accelerated gradient temporal difference learning algorithms. *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. 80

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*. 28

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv*. 75

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*. 80

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT press. 1, 19

Morcos, A. S., Raghu, M., & Bengio, S. (2018). Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*. 29

Motiian, S., Jones, Q., Iranmanesh, S., & Doretto, G. (2017). Few-shot adversarial domain adaptation. *Advances in neural information processing systems*. 63

Mukherjee, S., Niyogi, P., Poggio, T., & Rifkin, R. (2006). Learning theory: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*. 1

Neyshabur, B., Sedghi, H., & Zhang, C. (2020). What is being transferred in transfer learning? *Advances in Neural Information Processing Systems*. 29

Neyshabur, B., Tomioka, R., & Srebro, N. (2014). In search of the real inductive bias: On the role of implicit regularization in deep learning. *International Conference on Learning Representations*. 2

Nikishin, E., Schwarzer, M., D'Oro, P., Bacon, P.-L., & Courville, A. (2022). The primacy bias in deep reinforcement learning. *International Conference on Machine Learning*. 80

Öhman, E., Pàmies, M., Kajava, K., & Tiedemann, J. (2020). XED: A multilingual dataset for sentiment analysis and emotion detection. *Proceedings of the 28th International Conference on Computational Linguistics*. 74

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*. 36

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. *Conference on Computer Vision and Pattern Recognition*. 28

Ortiz-Jiménez, G., Moosavi-Dezfooli, S.-M., & Frossard, P. (2021). What can linearized neural networks actually say about generalization? 51

Oymak, S., Fabian, Z., Li, M., & Soltanolkotabi, M. (2019). Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *arXiv preprint arXiv:1906.05392*. 7

Packer, C., Gao, K., Kos, J., Krahenbuhl, P., Koltun, V., & Song, D. (2019). Assessing generalization in deep reinforcement learning. *arXiv*. 80

Pan, Y., White, A., & White, M. (2017). Accelerated gradient temporal difference learning. *AAAI Conference on Artificial Intelligence*. 80, 83

Panareda Busto, P., & Gall, J. (2017). Open set domain adaptation. *IEEE international conference on computer vision*. 46

Papyan, V., Han, X., & Donoho, D. L. (2020). Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*. 97

Patil, G., Prashanth, L., Nagaraj, D., & Precup, D. (2023). Finite time analysis of temporal difference learning with linear function approximation: Tail averaging and regularisation. *International Conference on Artificial Intelligence and Statistics*. 80

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, *2*(11), 559–572. 49

Pei, Z., Cao, Z., Long, M., & Wang, J. (2018). Multi-adversarial domain adaptation. *Thirty-second AAAI conference on artificial intelligence*. 47

Peng, X. B., Andrychowicz, M., Zaremba, W., & Abbeel, P. (2018). Sim-to-real transfer of robotic control with dynamics randomization. *IEEE international conference on robotics and automation*. 46

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Conference on Empirical Methods in Natural Language Processing*. 28

Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is multilingual BERT? *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 46

Popper, K. (2005). *The logic of scientific discovery*. Routledge. 1

Qian, P., Qiu, X., & Huang, X.-J. (2016). Investigating language universal and specific properties in word embeddings. *Annual Meeting of the Association for Computational Linguistics*. 29

Rebuffi, S.-A., Bilen, H., & Vedaldi, A. (2017). Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*. 63

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*. 28

Saito, K., Ushiku, Y., & Harada, T. (2017). Asymmetric tri-training for unsupervised domain adaptation. *International Conference on Machine Learning*. 63

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press. 1

Shi, X., Padhi, I., & Knight, K. (2016). Does string-based neural mt learn source syntax? *Conference on Empirical Methods in Natural Language Processing*. 29

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*. 64

Shwartz-Ziv, R., & Tishby, N. (2017). Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*. 29

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. *International Conference on Learning Representations*. 29

Song, L., Smola, A., Gretton, A., Borgwardt, K. M., & Bedo, J. (2007). Supervised feature selection via dependence estimation. *International Conference on Machine Learning*. 21

Song, X., Jiang, Y., Tu, S., Du, Y., & Neyshabur, B. (2020). Observational overfitting in reinforcement learning. *International Conference on Learning Representations*. 80

Sun, B., & Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. *European conference on computer vision*. 64

Sutton, R. S. (1995). Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems*. 80

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press. 79, 82

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., & Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. *International Conference on Machine Learning*. 87

Tachet, R., Zhao, H., Wang, Y.-X., & Gordon, G. J. (2020). Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*. 75

Tola, E., Lepetit, V., & Fua, P. (2009). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*. 35

Touretzky, D. S., & Pomerleau, D. A. (1989). What's hidden in the hidden layers? *BYTE*, *14*(8). 28

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., & Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *arXiv*. 63

Vapnik, V., & Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*. 1

Wu, J., Zou, D., Braverman, V., Gu, Q., & Kakade, S. (2022). The power and limitation of pretraining-finetuning for linear regression under covariate shift. *Advances in Neural Information Processing Systems*. 7, 8

Yu, H., & Bertsekas, D. P. (2009). Basis function adaptation methods for cost approximation in mdp. *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*. 80

Zahavy, T., Ben-Zrihem, N., & Mannor, S. (2016). Graying the black box: Understanding dqns. *International Conference on Machine Learning*. 29

Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. *arXiv*. 42

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *European Conference on Computer Vision*. 29

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations*. 2

Zhang, T. (2023). *Mathematical analysis of machine learning algorithms*. Cambridge University Press. 1

Zhang, Y., Barzilay, R., & Jaakkola, T. (2017). Aspect-augmented adversarial networks for domain adaptation. *Transactions of the Association for Computational Linguistics*. 47

Zhang, Y., Liu, T., Long, M., & Jordan, M. (2019). Bridging theory and algorithm for domain adaptation. *International Conference on Machine Learning*. 46, 63

Zhao, H., Des Combes, R. T., Zhang, K., & Gordon, G. (2019). On learning invariant representations for domain adaptation. *International Conference on Machine Learning*. 47, 63–65, 69

Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., & Torralba, A. (2015). Object detectors emerge in deep scene cnns. *International Conference on Learning Representations*. 28

Zhu, Z., Ding, T., Zhou, J., Li, X., You, C., Sulam, J., & Qu, Q. (2021). A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*. 97

Zhuang, F., Cheng, X., Luo, P., Pan, S. J., & He, Q. (2015). Supervised representation learning: Transfer learning with deep autoencoders. *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 63

Zhuang, F., Cheng, X., Luo, P., Pan, S. J., & He, Q. (2017). Supervised representation learning with double encoding-layer autoencoder for transfer learning. *ACM Transactions on Intelligent Systems and Technology*. 64

Zou, D., Wu, J., Braverman, V., Gu, Q., Foster, D., & Kakade, S. (2021a). The benefits of implicit regularization from sgd in least squares problems. *Advances in neural information processing systems*. 7

Zou, D., Wu, J., Braverman, V., Gu, Q., & Kakade, S. (2021b). Benign overfitting of constant-stepsize sgd for linear regression. *Conference on Learning Theory*. 7, 97

Appendix A

Representation Alignment and Generalization (Supplementary)

A.1 Proof for Theorem 3.1.3

Proof. It is easy to show that $w_\tau^* \perp w_{\bar{\tau}}^*$ and $w^* = w_\tau^* + w_{\bar{\tau}}^*$. Define $\hat{w}_\tau^\lambda$ and $\hat{w}_{\bar{\tau}}^\lambda$ as the ridge regression estimator if the labels were generated using w_τ^* and $w_{\bar{\tau}}^*$ instead of w^*. Then we have $\mathbb{E}[\hat{w}^\lambda|\Phi] = (\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top\Phi w^*$, $\mathbb{E}[\hat{w}_\tau^\lambda|\Phi] = (\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top\Phi w_\tau^*$, $\mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi] = (\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top\Phi w_{\bar{\tau}}^*$, $\mathbb{E}[\hat{w}^\lambda|\Phi] = \mathbb{E}[\hat{w}_\tau^\lambda|\Phi] + \mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi]$. Also $\|w_\tau^*\|_H = \|w^*\|_{H_\tau}$ and similarly for $w_{\bar{\tau}}^*$. Now let us decompose the risk

$$\begin{aligned} R_\Phi(\hat{w}^\lambda, w^*) &= B_\Phi(\hat{w}^\lambda, w^*) + V_\Phi(\hat{w}^\lambda, w^*) \\ &= \underbrace{B_\Phi(\hat{w}_\tau^\lambda, w_\tau^*)}_{=:B_{\Phi,\tau}(\hat{w}^\lambda, w^*)} + V_\Phi(\hat{w}^\lambda, w^*) + \underbrace{B_\Phi(\hat{w}^\lambda, w^*) - B_\Phi(\hat{w}_\tau^\lambda, w_\tau^*)}_{=:\Delta} \end{aligned}$$

The first term is the bias of an alternate problem whose optimal weights are w_τ^*. The previous theorem controls bias and variance independently and the two terms do not have to correspond to the same problem, so by applying that theorem we get

$$\begin{aligned} &R_\Phi(\hat{w}^\lambda, w^*) = B_{\Phi,\tau}(\hat{w}^\lambda, w^*) + V_\Phi(\hat{w}^\lambda, w^*) + \Delta \\ &|B_{\Phi,\tau}(\hat{w}^\lambda, w^*) - \mathscr{B}_\tau(\lambda, \hat{S}, \hat{G}, \gamma)| \leq \frac{C\,\|w_\tau^*\|^2}{\lambda n^{(1-\varepsilon)/2}} \\ &|V_\Phi(\hat{w}^\lambda, w^*) - \mathscr{V}(\lambda, \hat{S}, \gamma)| \leq \frac{C}{\lambda^2 n^{(1-\varepsilon)/2}} \end{aligned}$$

It only remains to control Δ.

$$\Delta = \left\|\mathbb{E}[\hat{w}^\lambda|\Phi] - w^*\right\|_H^2 - \left\|\mathbb{E}[\hat{w}_\tau^\lambda|\Phi] - w_\tau^*\right\|_H^2$$

$$= \left\|\mathbb{E}[\hat{w}_\tau^\lambda|\Phi] - w_\tau^* + \mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi] - w_{\bar{\tau}}^*\right\|_H^2 - \left\|\mathbb{E}[\hat{w}_\tau^\lambda|\Phi] - w_\tau^*\right\|_H^2$$
$$= 2(\mathbb{E}[\hat{w}_\tau^\lambda|\Phi] - w_\tau^*)^\top H(\mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi] - w_{\bar{\tau}}^*) + \left\|\mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi] - w_{\bar{\tau}}^*\right\|_H^2$$

We bound the two terms separately for clarity. Define $Z \in \mathbb{R}^{n\times d}$ as $(z_i)_{i=1}^n$ in matrix form. Note that $\Phi^\top\Phi = {H^{1/2}}^\top Z^\top Z H^{1/2}$. Define $A := (\Phi^\top\Phi + n\lambda I)^{-1}{H^{1/2}}^\top Z^\top Z$, so that $(\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top\Phi = AH^{1/2}$. The first term is

$$\begin{aligned}
&2(\mathbb{E}[\hat{w}_\tau^\lambda|\Phi] - w_\tau^*)^\top H(\mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi] - w_{\bar{\tau}}^*) \\
&= 2\mathbb{E}[\hat{w}_\tau^\lambda|\Phi]^\top H\mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi] - 2\mathbb{E}[\hat{w}_\tau^\lambda|\Phi]^\top H w_{\bar{\tau}}^* - 2{w_\tau^*}^\top H\mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi] + 0 \\
&2{w^*}_\tau^\top H^{1/2}A^\top HAH^{1/2}{w^*}_{\bar{\tau}} - 2({w^*}_\tau)^\top H^{1/2}A^\top H{w^*}_{\bar{\tau}} - 2({w^*}_\tau)^\top HAH^{1/2}{w^*}_{\bar{\tau}} \\
&\le 2\left\|w_\tau^*\right\|_H \left\|w_{\bar{\tau}}^*\right\|_H \|A\|_{op}^2 \|H\|_{op} + 4\left\|w_\tau^*\right\|_H \left\|w_{\bar{\tau}}^*\right\|_H \|A\|_{op} \|H\|_{op}^{1/2} \qquad \text{(A.1)}
\end{aligned}$$

The operator norm of H is bounded by M according to the assumption, and $\|A\|_{op} = \left\|(\Phi^\top\Phi + n\lambda I)^{-1}{H^{1/2}}^\top Z^\top Z\right\|_{op} \le (1/\lambda)M^{1/2}\left\|\frac{1}{n}Z^\top Z\right\|_{op}$. Therefore

$$\text{(A.1)} \le 2\sqrt{\delta}\sqrt{1-\sigma_\epsilon^2-\delta}((1/\lambda^2)M^{3/2}\left\|\frac{1}{n}Z^\top Z\right\|_{op}^2 + 2(1/\lambda)M\left\|\frac{1}{n}Z^\top Z\right\|_{op}) \qquad \text{(A.2)}$$

The second term becomes

$$\begin{aligned}
&\left\|\mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi] - w_{\bar{\tau}}^*\right\|_H^2 \le 2\left\|\mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi]\right\|_H^2 + 2\left\|w_{\bar{\tau}}^*\right\|_H^2 \\
&= 2\left\|\mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi]\right\|_H^2 + 2(1-\sigma_\epsilon^2-\delta) \\
&\le 2\|H\|_{op}\left\|\mathbb{E}[\hat{w}_{\bar{\tau}}^\lambda|\Phi]\right\|^2 + 2(1-\sigma_\epsilon^2-\delta) \\
&= 2\|H\|_{op}\left\|AH^{1/2}w_{\bar{\tau}}^*\right\|^2 + 2(1-\sigma_\epsilon^2-\delta) \\
&\le 2/(1/\lambda)M^{3/2}\left\|\frac{1}{n}Z^\top Z\right\|_{op}(1-\sigma_\epsilon^2-\delta) + 2(1-\sigma_\epsilon^2-\delta) \\
&\le 2((1/\lambda)M^{3/2}\left\|\frac{1}{n}Z^\top Z\right\|_{op} + 1)(1-\sigma_\epsilon^2-\delta) \qquad \text{(A.3)}
\end{aligned}$$

Both the first and the second term in Δ will be bounded once $\left\|\frac{1}{n}Z^\top Z\right\|_{op}$ is controlled. We use the extra sub-gaussianity assumption here. Since z is zero mean and unit variance, its covariance matrix is I. Triangle inequality gives $\left\|\frac{1}{n}Z^\top Z\right\|_{op} \le 1 + \left\|\frac{1}{n}Z^\top Z - I\right\|_{op}$. Also, $\left\|\frac{1}{n}Z^\top Z\right\|_{op}^2 \le 2 + 2\left\|\frac{1}{n}Z^\top Z - I\right\|_{op}^2$. Due to a result by Bunea and Xiao (2015), there exists a universal constant

$C_z > 0$ such that with probability at least $1 - 2n^{-1}$ we have

$$\left\| \frac{1}{n} Z^\top Z - I \right\|_{op} \leq C_z \sigma_z^2 \max\{\sqrt{\frac{d + \log n}{n}}, \frac{d + \log n}{n}\} =: \Delta_z$$

Using this high probability bound along with the bounds on the two terms of Δ from Eqs. (A.2) and (A.3), we get that with probability at least $1 - 2n^{-1}$

$$\begin{aligned}
\Delta \leq & 2\sqrt{1 - \sigma_\epsilon^2 - \delta}\sqrt{\delta}(M^{3/2}/\lambda^2)(2 + 2\Delta_z^2) + \\
& 4\sqrt{1 - \sigma_\epsilon^2 - \delta}\sqrt{\delta}(M/\lambda)(1 + \Delta_z) + \\
& 2(1 - \sigma_\epsilon^2 - \delta)(M^{3/2}/\lambda)(1 + \Delta_z) + \\
& 2(1 - \sigma_\epsilon^2 - \delta)(M^{3/2}/\lambda)
\end{aligned}$$

To simplify the expression note that $\sqrt{\delta} \leq 1$, $1 - \sigma_\epsilon^2 - \delta \leq \sqrt{1 - \sigma_\epsilon^2 - \delta}$, $\max\{\sqrt{\frac{d+\log n}{n}}, \frac{d+\log n}{n}, (\frac{d+\log n}{n})^2\} = \max\{\sqrt{\frac{d+\log n}{n}}, (\frac{d+\log n}{n})^2\}$. Then with probability at least $1 - 2n^{-1}$ and for a constant $C' = C'(M)$ the expression above can be bounded with the simpler expression

$$\begin{aligned}
\Delta \leq & C' \sqrt{1 - \sigma_\epsilon^2 - \delta} \max(\frac{1}{\lambda}, \frac{1}{\lambda^2})(1 + \max(\sigma_z^2, \sigma_z^4) \\
& \max\{\sqrt{\frac{d + \log n}{n}}, (\frac{d + \log n}{n})^2\})
\end{aligned}$$

□

Appendix B

Representation Alignment and Optimization (Supplementary)

B.1 Proof for Theorem 4.1.1

Proof. The min-norm MSE minimizer can be easily obtained algebraically and it satisfies $\mathbb{E}[\phi\phi^\top]w^* = \mathbb{E}[\phi y]$ and $w^* = \mathbb{E}[\phi\phi^\top]^\dagger\mathbb{E}[\phi y]$ and $w^* = \lim_{t\to\infty}\hat{w}^t$ (Hastie et al., 2022). Then the error can be decomposed as

$$\begin{aligned}\mathbb{E}[(\phi^\top\hat{w}^t - y)^2] =& \mathbb{E}[(\phi^\top\hat{w}^t - \phi^\top w^*)^2 + \underbrace{(\phi^\top w^* - y)^2}_{\epsilon^2} \\ &+ 2(\phi^\top\hat{w}^t - \phi^\top w^*)(\phi^\top w^* - y)] \\ =& \mathbb{E}[(\phi^\top\hat{w}^t - \phi^\top w^*)^2] + \mathbb{E}[\epsilon^2] \\ &+ 2(\hat{w}^t - w^*)^\top \underbrace{\mathbb{E}[\phi(\phi^\top w^* - y)]}_{0} \\ =& \mathbb{E}[(\phi^\top\hat{w}^t - \phi^\top w^*)^2] + \mathbb{E}[\epsilon^2] \end{aligned} \tag{B.1}$$

Define $b := \mathbb{E}[\phi^\top y]$ for brevity and note that b is in the span of H, therefore for any $\{j : \sigma_j = 0\}$ we have $v_j^\top b = 0$. Now recall $\tilde{w} = \Sigma V^\top w^* = \Sigma V^\top H^\dagger b = \Sigma V^\top V\Sigma^{2\dagger}V^\top b = \Sigma^\dagger V^\top b \implies b = V\Sigma\tilde{w}$. Gradient descent iterates $\hat{w}^t$ for $t > 0$ can be written and unfolded as

$$\begin{aligned}\hat{w}^t =& (I - \eta H)\hat{w}^{t-1} + \eta b = \eta\sum_{i=0}^{t-1}(I - \eta H)^i b \\ =& \eta\sum_{i=0}^{t-1}(VV^\top - \eta V\Sigma^2 V^\top)^i b = \eta V(\sum_{i=0}^{t-1}(I - \eta\Sigma^2)^i)V^\top b\end{aligned}$$

$$=\eta V(\sum_{i=0}^{t-1}(I-\eta\Sigma^2)^i)V^\top V\Sigma\tilde{w}=\eta V(\sum_{i=0}^{t-1}(I-\eta\Sigma^2)^i)\Sigma\tilde{w}$$

Define $\tilde{\Lambda}_{n_1\to n_2}:=\sum_{i=n_1}^{n_2}(I-\eta\Sigma^2)^i\Sigma$ for $n_1\in\mathbb{N}_0$ and $n_2\in\mathbb{N}_\infty$ and $\tilde{\lambda}_{n_1\to n_2,j}$ as the j-th diagonal element of this diagonal matrix. With this notation we have $\hat{w}^t=\eta V\tilde{\Lambda}_{0\to t-1}V^\top b$. Geometric sum under the condition $0<\eta<1/\sigma_1^2$ gives that for each index $\{j:\sigma_j>0\}$

$$\begin{aligned}\tilde{\lambda}_{0\to t-1,j}=&\sum_{i=0}^{t-1}(1-\eta\sigma_j^2)^i\sigma_j=\frac{1-(1-\eta\sigma_j^2)^t}{\eta\sigma_j}\\ \tilde{\lambda}_{0\to\infty,j}=&\sum_{i=0}^{\infty}(1-\eta\sigma_j^2)^i\sigma_j=\frac{1}{\eta\sigma_j}\\ \tilde{\lambda}_{t\to\infty,j}=&\sum_{i=t}^{\infty}(1-\eta\sigma_j^2)^i\sigma_j=\frac{-(1-\eta\sigma_j^2)^t}{\eta\sigma_j}\end{aligned}$$

And for elements $\{j:\sigma_j>0\}$ we have $\tilde{\lambda}_{0\to t-1,j}=\tilde{\lambda}_{0\to\infty,j}=\tilde{\lambda}_{0\to t,j}=0$. With this notation we have $\hat{w}^t-w^*=\eta V\tilde{\Lambda}_{t,\infty}\tilde{w}$. Therefore

$$\begin{aligned}\mathbb{E}[(\phi^\top\hat{w}^t-\phi^\top w^*)^2]=&(\hat{w}^t-w^*)^\top H(\hat{w}^t-w^*)=\left\|\Sigma V^\top(\hat{w}^t-w^*)\right\|^2\\ =&\left\|\eta\tilde{\Lambda}_{t,\infty}\Sigma\tilde{w}\right\|^2=\sum_{i=1}^{d}(1-\eta\sigma_i^2)^{2t}\tilde{w}_i^2\end{aligned}$$

In the last equality holds because for each $\{j:\sigma_j=0\}$ we have $\tilde{w}_j=0$. Putting this in Equation (B.1) proves the result for $t>0$. The case of $t=0$ is trivial. □

B.2 Proof for Proposition 4.1.2

Proof. Define the reduction in loss as $0\le\omega<\delta$ which is

$$\begin{aligned}\omega:=&\mathbb{E}[(\phi^\top\hat{w}^0-y)^2]-\mathbb{E}[(\phi^\top\hat{w}^t-y)^2]=\sum_{i=1}^{d}\tilde{w}_i^2-\sum_{i=1}^{d}(1-\eta\sigma_i^2)^{2t}\tilde{w}_i^2\\ =&\sum_{i=1}^{d}\tilde{w}_i^2-(1-\eta\sigma_i^2)^{2t}\tilde{w}_i^2\ge\sum_{\{i:\sigma_j\ge\tau\}}\tilde{w}_i^2-(1-\eta\sigma_i^2)^{2t}\tilde{w}_i^2\ge\delta-(1-\eta\tau^2)^{2t}\delta\end{aligned}$$

Now due to the condition on η we have $0<\eta\tau^2<1$ and thus $(1-\eta\tau^2)^{2t}<\exp(-2t\eta\tau^2)$. Therefore

$$\omega\ge\delta(1-\exp(-2t\eta\tau^2))\implies\exp(-2t\eta\tau^2)\ge1-\omega/\delta$$

$$\implies -2t\eta\tau^2 \log(1-\omega/\delta)$$
$$\implies t \leq -\log(1-\omega/\delta)/(2\eta\tau^2)$$

□

Appendix C

Label Alignment Regularization for Distribution Shift (Supplementary)

Lemma C.0.1. *In the rotated Gaussian example in Section 6.4,* $\mathbb{E}_{x,y}[xy] = \sqrt{\frac{2}{\pi}} s_1 p_1$.

Proof.

$$\frac{1}{n}\Phi^\top y \approx \mathbb{E}_{x,y}[xy] \tag{C.1}$$

$$= \int_x \int_y xy p_{\mathcal{S}}(x|y)p(y)dydx \tag{C.2}$$

$$= \int_x x p_{\mathcal{S}}(x|y=1)p(y=1) - x p_{\mathcal{S}}(x|y=-1)p(y=-1)dx \tag{C.3}$$

$$= \int_x x \cdot 2\mathcal{N}(0,Q)(\mathbb{1}(x_1^P > 0)p(y=1) - \mathbb{1}(x_1^P < 0)p(y=-1))dx, \tag{C.4}$$

where we plug into the definition (6.9) to get the last equality. Further note that $\mathbb{1}(x_1^P < 0) = 1 - \mathbb{1}(x_1^P > 0)$ and plug this into above,

$$\text{(C.4)} = \int_x x \cdot 2\mathcal{N}(0,Q)(\mathbb{1}(x_1^P > 0) - p(y=-1))dx \tag{C.5}$$

$$= \int_x x \cdot 2\mathcal{N}(0,Q)\mathbb{1}(x_1^P > 0)dx \tag{C.6}$$

$$= 2P^\top \int_z z \cdot \frac{1}{2\pi s_1 s_2} \exp\left(-\frac{1}{2s_1^2}z_1^2 - \frac{1}{2s_2^2}z_2^2\right)\mathbb{1}(z_1 > 0)dz \tag{C.7}$$

where in the last equality we let $z = Px$, then $x_1^P = x^\top p_1 = z^\top P p_1 = z_1$. The integral above is a vector with two elements because it includes z. The first

element is

$$\int_{z_1}\int_{z_2} z_1 \cdot \frac{1}{2\pi s_1 s_2} \exp\left(-\frac{1}{2s_1^2}z_1^2 - \frac{1}{2s_2^2}z_2^2\right) \mathbb{1}(z_1 > 0) dz_1 dz_2 \tag{C.8}$$

$$= \int_{z_1} z_1 \cdot \frac{1}{\sqrt{2\pi} s_1} \exp\left(-\frac{1}{2s_1^2}z_1^2\right) \mathbb{1}(z_1 > 0) dz_1 \tag{C.9}$$

$$= \frac{1}{2}\int_{0}^{+\infty} z_1 \cdot \frac{\sqrt{2}}{\sqrt{\pi} s_1} \exp\left(-\frac{1}{2s_1^2}z_1^2\right) dz_1 \tag{C.10}$$

$$= \frac{1}{2} s_1 \sqrt{\frac{2}{\pi}} \tag{C.11}$$

The last equality is because the integration is the mean of half-normal distribution. The second element would become zero as written below and noting that $\mathbb{E}[z_2]$ is the mean of a zero-mean Gaussian random variable:

$$\int_{z_1}\int_{z_2} z_2 \cdot \frac{1}{2\pi s_1 s_2} \exp\left(-\frac{1}{2s_1^2}z_1^2 - \frac{1}{2s_2^2}z_2^2\right) \mathbb{1}(z_1 > 0) dz_1 dz_2 \tag{C.12}$$

$$= \int_{z_1} \frac{1}{\sqrt{2\pi} s_1} \exp\left(-\frac{1}{2s_1^2}z_1^2\right) \mathbb{1}(z_1 > 0) dz_1 \mathbb{E}[z_2] = 0 \tag{C.13}$$

Then

$$\mathbb{E}_{x,y}[xy] = 2P^\top \begin{bmatrix} \frac{1}{2} s_1 \sqrt{\frac{2}{\pi}} \\ 0 \end{bmatrix} = \sqrt{\frac{2}{\pi}} s_1 p_1. \tag{C.14}$$

□

Appendix D

Generalization in Temporal-Difference Learning (Supplementary)

D.1 Proof for Theorem 7.3.1

Proof. First note

$$(I - \eta A)^i = (QQ^{-1} - \eta Q\Sigma^2 Q^{-1})^i = (Q(I - \eta\Sigma^2)Q^{-1})^i = Q(I - \eta\Sigma^2)^i Q^{-1}$$

Now unfolding the recursive update gives

$$\begin{aligned} w_{t+1} &= (I - \eta A)w_t + \eta b = \eta \sum_{i=0}^{t} (I - \eta A)^i b = \eta \sum_{i=0}^{t} Q(I - \eta\Sigma^2)^i Q^{-1} b \\ &= \eta \sum_{i=0}^{t} Q(I - \eta\Sigma^2)^i \Sigma b' = \eta Q \tilde{\Lambda}_t b' \end{aligned}$$

where $\tilde{\Lambda}_t \doteq \sum_{i=0}^{t}(I - \eta\Sigma^2)^i\Sigma$. Similarly $w_* = \eta Q\tilde{\Lambda}_* b'$ for $\tilde{\Lambda}_* \doteq \sum_{i=0}^{\infty}(I - \eta\Sigma^2)^i\Sigma$. Then

$$\|w_{t+1} - w_*\| = \left\|\eta Q\tilde{\Lambda}_t b' - \eta Q\tilde{\Lambda}_* b'\right\| = \left\|\eta Q(\tilde{\Lambda}_t - \tilde{\Lambda}_*)b'\right\| \leq C \left\|\eta(\tilde{\Lambda}_t - \tilde{\Lambda}_*)b'\right\| \tag{D.1}$$

where C is the operator norm of Q. If A is symmetric (as in GD) then Q is orthonormal and the last step is an equality with $C = 1$.

Both $\tilde{\Lambda}_*$, $\tilde{\Lambda}_t$ are diagonal matrices and the elements corresponding to zero eigenvalues are zero since all terms of the series are multiplied by zero. For other diagonal elements, the geometric sum formula and the condition on

step-size and eigenvalues in the proposition statement gives each element j of $\eta(\tilde{\Lambda}_t - \tilde{\Lambda}_*)$

$$(\sum_{i=0}^{\infty}(I-\eta\Sigma^2)^i)_j = \frac{1}{\eta\sigma_j^2}, \quad (\sum_{i=0}^{t}(I-\eta\Sigma^2)^i)_j = \frac{1-(1-\eta\sigma_j^2)^{t+1}}{\eta\sigma_j^2},$$
$$(\sum_{i=0}^{t}(I-\eta\Sigma^2)^i - \sum_{i=0}^{\infty}(I-\eta\Sigma^2)^i)_j = \frac{(1-\eta\sigma_j^2)^{t+1}}{\eta\sigma_j^2} \implies (\eta(\tilde{\Lambda}_t - \tilde{\Lambda}_*))_j = \frac{(1-\eta\sigma_j^2)^{t+1}}{\sigma_j}$$

and we can continue with Eq (D.1) to get

$$\|w_{t+1} - w_*\| \leq \sqrt{C\sum_{j=1}^{d}|\frac{(1-\eta\sigma_j^2)^{(t+1)}}{\sigma_j}|^2 \ |b'_j|^2}$$

and prove the first statement. For the second statement we have

$$\begin{aligned}\|(b - Aw_{t+1}) - (b - Aw_*)\| = \|A(w_{t+1} - w_*)\| &= \left\|\eta Q\Sigma^2 Q^{-1} Q(\tilde{\Lambda}_t - \tilde{\Lambda}_*)b'\right\| \\ &= \left\|\eta Q\Sigma^2(\tilde{\Lambda}_t - \tilde{\Lambda}_*)b'\right\| \leq C\left\|\eta\Sigma^2(\tilde{\Lambda}_t - \tilde{\Lambda}_*)b'\right\| \\ &= \sqrt{C\sum_{j=1}^{d}|(1-\eta\sigma_j^2)^{(t+1)}|^2 \ |b'_j|^2}\end{aligned}$$

Again the inequality becomes an equality with $C = 1$ for the GD update. □

D.2 Proof for Proposition 7.3.2

Proof. First note that complex eigenvalues and eigenvectors come in conjugate pairs, i.e., $\bar{q}_i$ is also an eigenvector of A with eigenvalue $\bar{\sigma_i^2}$ with the same magnitude and also note that $\bar{\sigma_i^2} = \bar{\sigma_i}^2$. By writing the complex form it is easy to show that $(q_i\sigma_i + \bar{q}_i\bar{\sigma}_i)/2 = (\mathrm{Re}(q_i)\mathrm{Re}(\sigma_i) - \mathrm{Im}(q_i)\mathrm{Im}(\sigma_i))$.

With TD updates and for the reward vector r_{TD}^i we have $b = \Phi^\top D r_{TD}^i = (\Phi^\top\Phi)(\Phi^\top\Phi)^\dagger(\mathrm{Re}(q_i)\mathrm{Re}(\sigma_i) - \mathrm{Im}(q_i)\mathrm{Im}(\sigma_i)) = (q_i\sigma_i + \bar{q}_i\bar{\sigma}_i)/2$. This is because $(\Phi^\top\Phi)(\Phi^\top\Phi)^\dagger$ is the projection operator on the column space of Φ and has no effect if the operand is already in this subspace, which we have assumed to be the case. In the case of GD we have $b = \Phi^\top D(I - P_{\pi,\gamma})^{-1} r_{GD}^i = (\Phi^\top\Phi)(\Phi^\top\Phi)^\dagger(\mathrm{Re}(q_i)\mathrm{Re}(\sigma_i) - \mathrm{Im}(q_i)\mathrm{Im}(\sigma_i)) = (q_i\sigma_i + \bar{q}_i\bar{\sigma}_i)/2$. The requirement regarding the column space of Φ holds by construction because in this case q_i are the also the right singular vectors of Φ and also we have assumed $\sigma_i > 0$.

Construct the d-dimensional vector s_i such that if q_i above is real-valued then the i-th element of s_i is set to σ_i and if q_i above is complex-valued then the elements of s_i corresponding to q_i and $\bar{q_i}$ are set to $\sigma_i/2$ and $\bar{\sigma_i}/2$. All other elements of this vector are zero. Then b can be written as Qs_i, and $b' = \Sigma^\dagger Q^{-1} b = \Sigma^\dagger s_i$ has only an element of 1 at index i if q_i is real-valued and only has elements of $1/2$ at indices corresponding to q_i and $\bar{q_i}$ if q_i is complex-valued. Putting this b' in Theorem 7.3.1 and noting that $(1 - \eta\sigma_i^2)^{(t+1)}$ and $(1 - \eta\bar{\sigma_i^2})^{(t+1)}$ have the same magnitude proves the result. $\square$