Generalization Effects of Transferring Across High-dimensional Data in Multi-Task Learning

24/05/2020 at 1:30pm

1 Introduction

Multi-task learning has recently become a powerful paradigm to solve complex prediction tasks in computer vision [? ?], natural language processing [? ?] and numerous other areas [?]. By combining multiple information sources, multi-task learning allows new information to be shared across different sources in a model [?]. At the same time, it is well-known that the performance of multi-task learning methods depends on the relationship between the information sources. *Negative transfer*, the phenomenon where for a particular prediction task, multi-task learning performs worse than single-task learning, is prevalent when the information sources are heterogeneous [? ?]. While numerous studies have sought to alleviate negative transfer when it occurs [?], a rigorous understanding to the contributing causes of negative transfer has remained elusive in the literature [?]. In this work, we consider learning multiple high-dimensional linear regression tasks to better understand when and why negative transfer happens. We provide theoretical and practical insights to show how task data affects the transfer of information.

Identifying when negative transfer occurs requires comparing the performance of multi-task learning directly to single-task learning. The technical challenge is to develop generalization bounds that are able to scale tightly with the qualities of task data, such as the number of datapoints. In classical Rademacher or VC based theory of multi-task learning [? ? ?], the generalization bounds are usually presented in a way so that the error goes down as more labeled data is added. On the other hand, we have observed that adding more labeled data does not always improve performance in multi-task learning. More recent work has shown the benefit of learning multi-task representations for certain half-spaces [?] and multipl sparse regressions [19, 20].

In this work, we consider multiple high-dimensional linear regression tasks as input and focus on predicting a particular task that only has limited amount of labeled data. Concretely, each task consists of n_i datapoints in space $\mathbb{R}^p$ organized as matrix $X_i \in \mathbb{R}^{n_i \times p}$, for $1 \leqslant i \leqslant p$. The labels of X_i are given by $Y_i = X_i\beta_i + \varepsilon_i$, where β_i denotes the ground truth parameters for task i and ε_i denotes i.i.d. random noise with mean zero and variance σ^2. Importantly, we assume that the number of datapoints n_i is a small constant ρ_i times p and that p is large. [Todo: rational] We use a hard parameter sharing architecture that contains a shared body $B \in \mathbb{R}^{p\times r}$ for all tasks and a separate prediction head $\{W_i \in \mathbb{R}^r\}_{i=1}^k$ for each task [? ?]. This corresponds to minimizing the following optimization objective.

$$f(B; W_1, \dots, W_k) = \sum_{i=1}^{k} \|X_i B W_i - Y_i\|^2. \tag{1.1}$$

Let $\hat{\beta}_t^{\text{MTL}} = BW_t$ denote the optimal predictor obtained from solving equation (1.1Introductionequation.1.1) for task t. Our goal is to compare the test error of $\hat{\beta}_t^{\text{MTL}}$, denoted as $te(\hat{\beta}_t^{\text{MTL}})$, to $te(\hat{\beta}_t^{\text{STL}})$, where $\hat{\beta}_t^{\text{STL}}$ is likewise obtained from equation (1.1Introductionequation.1.1) but with task t in isolation.

Main results. We begin by observing that $te(\hat{\beta}_t^{\text{MTL}})$ can be decomposed into two parts, a variance part that is reduced from $te(\hat{\beta}_t^{\text{STL}})$, and a bias part that captures the difference between β_t and the rest of β's.

We term the bias part as *model shift bias*. Intuitively, whether $te(\hat{\beta}_t^{\mathrm{MTL}})$ is lower than $te(\hat{\beta}_t^{\mathrm{STL}})$ depends on the trade-off between the amount of variance reduced and the model shift bias part. For the high-dimensional regime when p goes to infinity, we derive the asymptotic limit of $te(\hat{\beta}_t^{\mathrm{MTL}}) - te(\hat{\beta}_t^{\mathrm{STL}})$ as a function of the number of datapoints $\{n_1, n_2\}$, the covariance matrices $\{\Sigma_1, \Sigma_2\}$, the ground truth parameters $\{\beta_1, \beta_2\}$, and a certain fixed value derived from solving equation (1.1Introductionequation.1.1) (see Theorem **??** for the statement). Then, we show a similar result for any number of tasks that have the same covariates, i.e. the X_i's are equal to each other in Theorem **??**. This setting is prevalent in applications of multi-task learning to image classification, where there are multiple prediction labels/tasks for every image [**?** 12].

Next, we use our technical tools to show how to determine negative transfer based on task data. Our first theoretical insight is we provide a sharp transition to show that task model similarities can determine whether there is positive transfer. For settings where tasks are similar, we further show that the transfer effect depends on the single-task accuracy of the source task.

Our second theoretical insight is we show how source task data size can also determine transfer by providing a sharp transition. We use our tools to explain the result of taskonomy [**?**], regarding the data efficiency of multi-task learning.

Our third theoretical insight is has implications on the following question. Is it better for two tasks to have the same covariance matrix or complementary covariance matrices? For our setting, we show that when the data ratio is large, having the same covariance matrix provably yields the lowest test performance on the target task. On the other hand, when data ratio is small, we find that there are cases when having complementary covariance matrices is better.

Finally, we extend our reuslts to study the transfer functions used in taskonomy [**?**].

A crucial technical tool that we develop is the asymptotic limit of the trace of the inverse of the sum of two independent sample covariance matrices. Our result not only extends the well-known Marchenko–Pastur law in random matrix theory to the sum of two independent sample covariance matrices with general covariance matrices, but also gives an almost optimal error bound, which may be of independent interest.

Experimental results. We provide practical implications to validate our theory. First, we validate on text and image classification tasks that comparing single-task accuracies can help determine whether multi-task learning performs better than single-task learning. [Todo: Second] Finally, We show that when the number of source task datapoints is large compared to the target task, then aligning task covariances always improves performance. On the other hand, if the number of source task datapoints is comparable to the target task, aligning task covariances may hurt performance.

2 Preliminaries

We assume that for every row $x^\top$ of X_i, we have $\mathbb{E}\left[xx^\top\right] = \Sigma_i$. We also write $x = \Sigma_i^{1/2} z_i$, where z_i is a random vector that has i.i.d. entries with mean 0 and variance 1. We will designate the k-th task as the target. Our goal is to come up with an estimator $\hat{\beta}$ to provide accurate predictions for the target task, provided with the other auxiliary task data. Concretely, we focus on the test error for the target task:

$$\begin{aligned} te_k(\hat{\beta}) &:= \mathop{\mathbb{E}}_{x\sim\Sigma_k}\left[\mathop{\mathbb{E}}_{\varepsilon_i, \forall 1\leqslant i\leqslant k}\left[(x^\top\hat{\beta} - x^\top\beta_t)^2\right]\right] \\ &= \mathop{\mathbb{E}}_{\varepsilon_i, \forall 1\leqslant i\leqslant k}\left[(\hat{\beta} - \beta_t)^\top\Sigma_k(\hat{\beta} - \beta_t)\right]. \end{aligned}$$

[Todo: show that $te_k(\hat{\beta}_t^{\text{TL-FT}})$ is less than both $te_k(\hat{\beta}_t^{\mathrm{MTL}})$ and $te_k(\hat{\beta}_t^{\mathrm{STL}})$.]

The High-Dimensional Setting. We would like to get insight on how covariate and model shifts affect the rate of transfer. We will consider the high-dimensional setting where for the target task, its number of data points is a small constant times p. This setting captures a wide range of applications of multi-task learning where we would like to use auxiliary task data to help train tasks with limited labeled data. Furthermore, this setting is particularly suited to our study since there is need for adding more data to help learn the target task.

For the case of two tasks, we can get precise rates using random matrix theory. For the sake of clarity, we call task 1 the source task and task 2 the target task, i.e. $\beta_1 = \beta_s$ and $\beta_2 = \beta_t$. We introduce the following notations for the high-dimensional setting

$$c_{n_1} := \frac{n_1}{p} \to c_1, \quad c_{n_2} := \frac{n_2}{p} \to c_2, \quad \text{as} \quad n_1, n_2 \to \infty,$$

for some constants $c_1, c_2 \in (1, \infty)$. A crucial quantity is what we call the *covariate shift* matrix $M = \Sigma_1^{1/2}\Sigma_2^{-1/2}$. Let $\lambda_1, \lambda_2, \dots, \lambda_p$ denote the singular values of M.

3 Dissecting the Effects of Different Task Data in Multi-Task learning

We illustrate our main results (to be presented in Section **??**) by considering a few special cases, namely special settings of the task models $\{\beta_i\}_{i=1}^k$, covariance matrices $\{\Sigma_i\}_{i=1}^k$, and number of data points $\{n_i\}_{i=1}^k$. We show that our results explain several phenomenon that cannot be explained before. [Todo: list those here]

3.1 Model Shift Bias versus Variance Trade-off

We assume that for every row $x^\top$ of X_i, we have $\mathbb{E}\left[xx^\top\right] = \Sigma_i$. We also write $x = \Sigma_i^{1/2} z_i$, where z_i is a random vector that has i.i.d. entries with mean 0 and variance 1. We will designate the k-th task as the target. Our goal is to come up with an estimator $\hat{\beta}$ to provide accurate predictions for the target task, provided with the other auxiliary task data. Concretely, we focus on the test error for the target task:

$$\begin{aligned} te_k(\hat{\beta}) &:= \mathop{\mathbb{E}}_{x\sim\Sigma_k}\left[\mathop{\mathbb{E}}_{\varepsilon_i, \forall 1\leqslant i\leqslant k}\left[(x^\top\hat{\beta} - x^\top\beta_t)^2\right]\right] \\ &= \mathop{\mathbb{E}}_{\varepsilon_i, \forall 1\leqslant i\leqslant k}\left[(\hat{\beta} - \beta_t)^\top\Sigma_k(\hat{\beta} - \beta_t)\right]. \end{aligned}$$

[Todo: show that $te_k(\hat{\beta}_t^{\text{TL-FT}})$ is less than both $te_k(\hat{\beta}_t^{\text{MTL}})$ and $te_k(\hat{\beta}_t^{\text{STL}})$.]

The High-Dimensional Setting. We would like to get insight on how covariate and model shifts affect the rate of transfer. We will consider the high-dimensional setting where for the target task, its number of data points is a small constant times p. This setting captures a wide range of applications of multi-task learning where we would like to use auxiliary task data to help train tasks with limited labeled data. Furthermore, this setting is particularly suited to our study since there is need for adding more data to help learn the target task.

For the case of two tasks, we can get precise rates using random matrix theory. For the sake of clarity, we call task 1 the source task and task 2 the target task, i.e. $\beta_1 = \beta_s$ and $\beta_2 = \beta_t$. We introduce the following notations for the high-dimensional setting

$$c_{n_1} := \frac{n_1}{p} \to c_1, \quad c_{n_2} := \frac{n_2}{p} \to c_2, \quad \text{as} \quad n_1, n_2 \to \infty,$$

for some constants $c_1, c_2 \in (1, \infty)$. A crucial quantity is what we call the *covariate shift* matrix $M = \Sigma_1^{1/2}\Sigma_2^{-1/2}$. Let $\lambda_1, \lambda_2, \dots, \lambda_p$ denote the singular values of M.

We begin by observing that the test error of $\hat{\beta}_t^{\text{MTL}}$ consists of two parts. One part captures how similar the task models are and the other part captures the variance of $\hat{\beta}_t^{\text{MTL}}$. Compared with $\hat{\beta}_t^{\text{STL}}$, we observe that the variance part of $\hat{\beta}_t^{\text{MTL}}$ gets reduced, since more data is added from source tasks. The bias part of $\hat{\beta}_t^{\text{MTL}}$, which we term as *model shift bias*, affects performance negatively. We derive the asympotic limit of $te(\hat{\beta}_t^{\text{MTL}})$ as p approaches infinity. We compare it with the asympotic limit of $te(\hat{\beta}_t^{\text{STL}})$, for settings where the target data size is limited. We show sharp generalization bounds for two settings: i) two tasks with general covaraites; ii) many tasks with the same covariates.

Two Tasks with General Covariance Matrices. For the case of two tasks, we decompose the test error of $\hat{\beta}_t^{\mathrm{MTL}}$ on the targe task into two parts

$$\begin{aligned} te(\hat{\beta}_t^{\mathrm{MTL}}) =& \hat{v}^2 \left\| \Sigma_2^{1/2} (\hat{v}^2 X_1^\top X_1 + X_2^\top X_2)^{-1} X_1^\top X_1 (\beta_s - \hat{v}\beta_t) \right\|^2 \\ &+ \sigma^2 \cdot \mathrm{Tr}\left[(\hat{v}^2 X_1^\top X_1 + X_2^\top X_2)^{-1} \Sigma_2 \right], \end{aligned} \tag{3.1}$$

where $\hat{v}$ denotes the ratio of the output layer weights (to be defined more precisely in Appendix **??**). It is not hard to see that the variance of $\hat{\beta}_t^{\mathrm{MTL}}$ is reduced compared to $\hat{\beta}_t^{\mathrm{STL}}$, i.e.

$$\mathrm{Tr}\left[(\hat{v}^2 X_1^\top X_1 + X_2^\top X_2)^{-1} \Sigma_2 \right] \leqslant \mathrm{Tr}\left[(X_2^\top X_2)^{-1} \Sigma_2 \right].$$

Because of model shift bias, we can no longer guarantee that $te(\hat{\beta}_t^{\mathrm{MTL}}) \leqslant te(\hat{\beta}_t^{\mathrm{STL}})$. The technical crux of our approach is to derive the asymptotic limit of $te(\hat{\beta}_t^{\mathrm{MTL}})$ in the high-dimensional setting, when p approaches infinity. A key highlight of our approach implies a precise limit on $\mathrm{Tr}\left[(X_1^\top X_1 + X_2^\top X_2)^{-1} \right]$, which only depends on Σ_1, Σ_2 and n_1, n_2 (see Lemma A.1theorem.A.1 in Appendix **??** for the result).

Theorem 3.1. *Let $X_i \in \mathbb{R}^{n_i \times p}$ and $Y_i = X_i\beta_i + \varepsilon_i$, for $i = 1, 2$. Suppose that $n_1 = c_1 p$ and $n_2 = c_2 p$, where $c_1 > 1$ and $c_2 > 1$ are fixed constants. There exists two deterministic functions Δ_β and Δ_{var} and a small deterministic error δ that only depend on $\{\hat{v}, M, n_1, n_2, \beta_1, \beta_2\}$ such that*

- *If $\Delta_{var} - \Delta_\beta \geqslant \delta$, then whp $te(\hat{\beta}_t^{MTL}) < te(\hat{\beta}_t^{STL})$.*
- *If $\Delta_{var} - \Delta_\beta \leqslant \delta$, then whp $te(\hat{\beta}_t^{MTL}) > te(\hat{\beta}_t^{STL})$.*

Theorem **??** shows upper and lower bounds that guarantee positive transfer, which is determined by the change of variance Δ_{var} and a certain model shift bias parameter Δ_β determined by the covariate shift matrix and the model shift. The bounds get tighter and tighter as n_1/p increases.

Many Tasks with the Same Covariates. We extend the above result to any number of tasks that have the same covariates. Since the tasks all have the same number of datapoints and covariance matrix, the trade-off between model shift bias and variance will be captured by their task models $\{\beta_i\}_{i=1}^k$. Let $B^\star = [\beta_1, \beta_2, \dots, \beta_k] \in \mathbb{R}^{p \times k}$ denote the underlying task model parameters. We derive the model shift bias and variance in the following result.

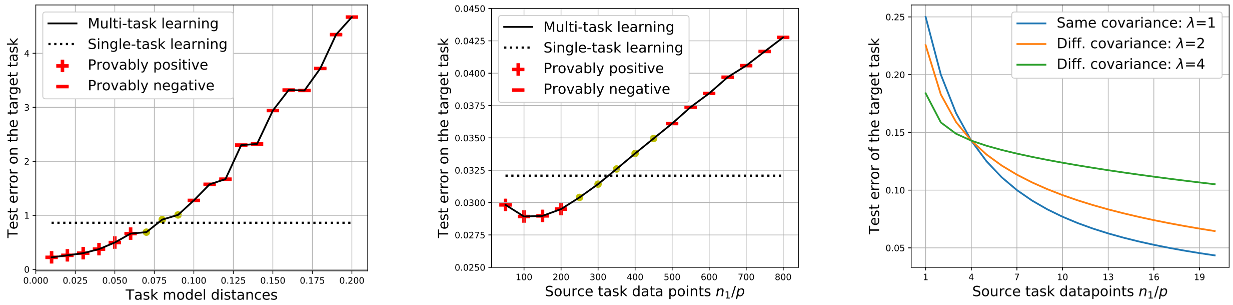
3.2 Task Model Similarity

We compare the test error of $\hat{\beta}_t^{\mathrm{MTL}}$ to that of $\hat{\beta}_t^{\mathrm{STL}}$. For a simple example, we show that whether $\hat{\beta}_t^{\mathrm{MTL}}$ performs better than $\hat{\beta}_t^{\mathrm{STL}}$ is determined by the distance of the task models. We derive a sharp threshold when positive transfer transitions to negative transfer, as a ratio between the model distance and the noise level.

Example 3.2. Consider a setting where $\Sigma_1 = \Sigma_2 = \mathrm{Id}$, in other words there is no covariate shift between the two tasks. For the noises, we assume that ε_i has i.i.d. entries with mean zero and variances σ_i^2, $i = 1, 2$. For the task models, suppose that β_2 has i.i.d. entries with mean zero and variance κ^2 and $\beta_1 - \beta_2$ also has i.i.d. entries with mean 0 and variance d^2. We have $n_i = c_i \cdot p$ data points from each task, for $i = 1, 2$. For simplicity, we assume that all the random variables have subexponential decay, while keeping in mind that our results can be applied under weaker moments assumptions as shown in the supplementary material.

We illustrate the example in a synthetic setting. We demonstrate our result with a simulation. ([Todo: uses the tighter bound Proposition A.3theorem.A.3?]) We consider a setting where $p = 200$, $n_1 = 90p$, $n_2 = 30p$. [Todo: Fill in other params.] We fix the target task and vary the source task, by varying the task model distance parameter d. We show that Theorem 3.1theorem.3.1 predicts whether we can get positive or negative transfer. Figure 2Positive vs negative transfer as a parameter of the task model distances.figure.caption.4 shows the result.

Specifically, the transition threshold is derived in the following proposition.



(a) Task model similarity (b) Data size (c) Covariate shift

Figure 1: Comparing the test error of multi-task learning to single-task learning: we observe transitions from positive to negative transfer. (a) Section **??**: Task model similarities; (b) Section **??**: Source task noise level; (c) Section **??**: [Todo: same vs. different covariance matrices].

Proposition 3.3. *In the setting of Example* **??** *with $\sigma_1 = \sigma_2 = \sigma$, assume that $c_1 > 110$ is a fixed constant. Whether $te(\hat{\beta}_t^{MTL})$ is lower than $te(\hat{\beta}_t^{STL})$ is determined by the ratio between the model distance and the noise level:*

- *If $d^2 < \frac{2\sigma^2}{3p} \frac{(c_1+c_2-1)^2}{c_1(c_1+c_2)(c_2-1)}$, then whp we have that $te(\hat{\beta}_t^{MTL}) < te(\hat{\beta}_t^{STL})$.*
- *If $d^2 \geqslant \frac{3\sigma^2}{2p} \frac{(c_1+c_2-1)^2}{c_1(c_1+c_2)(c_2-1)}$, then whp we have that $te(\hat{\beta}_t^{MTL}) \geqslant te(\hat{\beta}_t^{STL})$.*

Here the constants $\frac{2}{3}$ and $\frac{3}{2}$ in this proposition (and the following ones) can be replaced with $(1 + c_1^{-1/2})^{-4}$ and $(1 - c_1^{-1/2})^{-4}$. In particular, these bounds becomes tighter as c_1 increases.

The proof of Proposition **??** involves two parts. First, adding the source task has a positive effect of reducing the variance of the estimator, which scales with $n_1 = c_1 p$, the number of source task data points. Second, the difference between task models β_1 and β_2 introduces an additional bias term, which scales with pd^2, the distance between β_1 and β_2. Hence, the type of transfer is determined by the trade-off between model shift bias and the reduction of variance. The proof can be found in Appendix **??**, which is based on our main result described later in Theorem 3.1theorem.3.1.

Consider a more general setting where the noise level σ_1 of task 1 differs from the noise level σ_2 of task 2. We derive a sharp transition similar to Proposition **??**.

Proposition 3.4. *In the setting of Example* **??** *with d being fixed but σ_1 varies, assume that $c_1 > 110$ is a fixed constant and $d^2 < \frac{2\sigma_2^2}{3p} \frac{(c_1+c_2-1)^2}{c_1(c_1+c_2)(c_2-1)}$, i.e. we have positive transfer when $\sigma_1 = \sigma_2$. Then we derive the following transition as a parameter of σ_1:*

- *If $\sigma_1^2 \leqslant pd^2 \cdot c_1 + \left[1 + \frac{2}{3} \frac{(c_1+c_2-1)^2}{(c_1+c_2)(c_2-1)}\right] \cdot \sigma_2^2$, then whp $te(\hat{\beta}_t^{MTL}) < te(\hat{\beta}_t^{STL})$.*
- *If $\sigma_1^2 > pd^2 \cdot c_1 + \left[1 + \frac{3}{2} \frac{(c_1+c_2-1)^2}{(c_1+c_2)(c_2-1)}\right] \cdot \sigma_2^2$, then whp $te(\hat{\beta}_t^{MTL}) > te(\hat{\beta}_t^{STL})$.*

Implications. [Todo: add connection to tasksonomy]

3.3 Data Size and Efficiency

In classical Rademacher or VC based theory of multi-task learning, adding more labeled data improves the generalization performance of a model. On the other hand, we have observed that adding more labeled data does not always improve performance in multi-task learning. Using Example **??**, we analyze the effect of varying the source task data size.

Proposition 3.5. *In the setting of Example ??, assume that $c_2 \geqslant 3$ is fixed and $c_1 > a$ for some fixed integer a. We have the following conditions to determine whether $te(\hat{\beta}_t^{MTL})$ is lower than $te(\hat{\beta}_t^{STL})$:*

- *If $d^2 \leqslant (1 + a^{-1/2})^{-4} \frac{\sigma^2}{p(c_2-1)}$, then whp $te(\hat{\beta}_t^{MTL}) < te(\hat{\beta}_t^{STL})$.*
- *If $d^2 > (1 - a^{-1/2})^{-4}(1 - (a + c_2 - 2)^{-2})^{-1} \frac{\sigma^2}{p(c_2-1)}$, then we have the following transition depending on c_1:*
  - *If $c_1 < \frac{(c_2-2)\sigma^2}{(1+a^{-1/2})^4(c_2-1)pd^2-\sigma^2}$, then whp $te(\hat{\beta}_t^{MTL}) < te(\hat{\beta}_t^{STL})$.*
  - *If $c_1 > \frac{(c_2-2)\sigma^2}{(1-a^{-1/2})^4(1-(a+c_2-2)^{-2})(c_2-1)pd^2-\sigma^2}$, then whp $te(\hat{\beta}_t^{MTL}) > te(\hat{\beta}_t^{STL})$.*

The proof of Proposition ?? is similar to Proposition ??. We compare the model shift bias and the amount of reduced variance of $\hat{\beta}_t^{\text{MTL}}$. An intuitive interpretation of Proposition ?? is that: i) If the two task models are sufficiently similar (as specified under the first bullet), adding the source task always provides positive transfer; ii) Otherwise, as we increase the number of source task data points, the transfer is positive initially, but transitions to negative eventually. We leave the proof of Proposition ?? to Appendix ??.

Implications. We use our tools to explain a key result of taskonomy [?], which shows that by learning from multiple related tasks, one can reduce the amount of labeled data from each task. This is formalized by a metric called the data efficiency ratio as follows. Given several tasks, let $\alpha^\star$ be the largest factors such that the total number of labeled datapoints needed for solving all the tasks can be reduced by an $\alpha^\star$ factor (compared to training independently) while keeping the performance nearly the same. More precisely, suppose we have n_i datapoints for each task, for $i = 1, 2$. If we only use αn_i datapoints from every task to train the multi-task learning estimator $\hat{\beta}(\alpha)$, then $\alpha \in (0, 1)$ will be the smallest number such that

$$\alpha^\star := \underset{\alpha \in (0,1)}{\arg\min} \quad te_1(\hat{\beta}(\alpha)) + te_2(\hat{\beta}(\alpha)) \leqslant te_1(\hat{\beta}_t^{\text{STL}}) + te_2(\hat{\beta}_t^{\text{STL}}).$$

We quantify the data efficiency ratio of $\hat{\beta}_t^{\text{MTL}}$ for Example ?? as follows.

Proposition 3.6. *In the setting of Example ??, assume that $c_1 = c_2 = c \geqslant 200$ and $d^2 < 8\sigma^2/(3pc)$. Then the data efficiency ratio is at most $\frac{1}{2c} + \frac{\sigma^2}{2\sigma^2 - 3pd^2c/4}$.*

Note that we have stated the result assuming that $c_1 = c_2$. Similar results can also be obtained when they are different. We omit the details. The proof of Proposition ?? can be found in Appendix ??.

3.4 Covariate Shift

So far we have considered settings where $\Sigma_1 = \Sigma_2$. This setting is relevant for multi-class image classification settings, where different tasks share the same input features. In general, the covariance matrices of the two tasks may be different, e.g. in text classification. In this part, we use our tools to provide a case study on the effect of applying multi-task learning for two tasks when $\Sigma_1 \neq \Sigma_2$.

For this setting, covariate shift is captured by the matrix $M = \Sigma_1^{1/2}\Sigma_2^{-1/2}$. We ask: is it better to have M as being close to identity, or should M involve varying levels of singular values? Understanding this question has implications for applying normalization methods in multi-task learning [? ? ?]. Our result shows that if n_1 is much larger than n_2, then the optimal M matrix should be proportional to identity, under certain assumptions on its range of singular values (to be formulated in Proposition ??). On the other hand, if n_1 is comparable or even smaller than n_2, we show an example where having "complementary" covariance matrices is better performing than having the same covariance matrices.

Example 3.7. To compare different choices of M on the performance of $\hat{\beta}_t^{\text{MTL}}$, we assume an upper bound on the scale of M. Consider the following family of matrices

$$\mathcal{S}_\mu := \left\{ M \mid \det\left(M^\top M\right) \leqslant \mu^p, \lambda(M) \in [\mu_{\min}, \mu_{\max}] \right\},$$