On the Accuracy of Influence Functions for Measuring Group Effects

Pang Wei Koh∗  Kai-Siang Ang∗  Hubert H. K. Teo∗  Percy Liang
Department of Computer Science
Stanford University
{pangwei@cs, kaiang@, hteo@, pliang@cs}.stanford.edu

Abstract

Influence functions estimate the effect of removing particular training points on a model without needing to retrain it. They are based on a first-order approximation that is accurate for small changes in the model, and so are commonly used for studying the effect of individual points in large datasets. However, we often want to study the effects of large groups of training points, e.g., to diagnose batch effect or apportion credit between different data sources. Removing such large groups can result in significant changes to the model. Are influence functions still accurate in this setting? In this paper, we find that across many different types of groups and in a range of real-world datasets, the influence of a group correlates surprisingly well with its actual effect, even if the absolute and relative error can be large. Our theoretical analysis shows that such correlation arises under certain settings but need not hold in general, indicating that real-world datasets have particular properties that keep the influence approximation well-behaved.

1 Introduction

Influence functions can be used to estimate the effect that removing an individual training point has on a model’s predictions, without the computationally-prohibitive cost of repeatedly retraining the model. This ability to trace a model’s output back to its training data is valuable: in recent years, influence functions have been used to explain predictions (Koh and Liang, 2017), produce confidence intervals (Schulam and Saria, 2019), increase model fairness (Wang et al., 2019), improve human trust (Zhou et al., 2019), and even craft data poisoning attacks (Koh et al., 2019).

Influence functions are based on first-order approximations that are accurate for asymptotically small perturbations to the training data, which makes them suitable for predicting the effects of removing individual training points on the model. However, we often want to study the effects of removing groups of points, which represent large perturbations to the data. For example, we might wish to analyze the effect of data collected from different experimental batches (Leek et al., 2010) or demographic groups (Chen et al., 2018); apportion credit between crowdworkers, each of whom generated part of the data (Arrieta-Ibarra et al., 2018); or, in a multi-party learning setting, ensure that no individual user has too much influence on the joint model (Hayes and Ohrimenko, 2018). Are influence functions still accurate when predicting the effects of (removing) these groups?

In this paper, we show that on real datasets and across a broad variety of groups of data, the predicted and actual effects are strikingly correlated (Spearman $\rho$ of 0.8 to 1.0), such that the groups with the largest actual effect also tend to have the largest predicted effect. Moreover, the predicted effect tends to underestimate the actual effect, suggesting that it could be an approximate lower bound. Using influence functions to predict the effect of large, coherent groups of data can therefore still be a useful
What explains these phenomena of correlation and underestimation? Prior theoretical work focused on establishing the conditions under which this first-order influence approximation is accurate, i.e., the error between the actual and predicted effects is small \cite{giordano2019black,rad2018understanding}. However, measuring the influence of groups is in a qualitatively different regime: one in which this error can be quite large. We characterize the relationship between influence and actual effects in this regime via the one-step Newton approximation \cite{pregibon1981assessing} and show that correlation and underestimation arise under certain settings. However, our theoretical analysis shows that neither phenomena need to hold in general, which opens up the intriguing question of why we observe these phenomena across a wide range of empirical settings.

Finally, we exploit the correlation of influence with actual effects to study the effects of different data points (from different sources) can reveal insights about the nature of the data and application. In both cases, we show that the influences of the different groups of data points (from different sources) can reveal insights about the nature of the data and application.

2 Background and problem setup

Consider the task of learning a predictive model with parameters \( \theta \in \Theta \) that maps from an input space \( X \) to an output space \( Y \). We are given \( n \) training points \( \{(x_1, y_1), \ldots, (x_n, y_n)\} \) and a loss function \( \ell(x, y, \theta) \) that is twice-differentiable and convex in \( \theta \). To train the model, we select the model parameters

\[
\hat{\theta}(1) = \arg\min_{\theta \in \Theta} \left[ \sum_{i=1}^{n} \ell(x_i, y_i; \theta) + \frac{\lambda}{2} \|\theta\|_2^2 \right] \tag{1}
\]

that minimize the \( L_2 \)-regularized empirical risk, where \( \lambda > 0 \) is a hyperparameter that controls regularization strength. The all-ones vector \( 1 \) in \( \hat{\theta}(1) \) denotes that the initial training points all have uniform sample weights of one.

Our goal is to measure the effects that different groups of training data have on the model: if we removed a given subset of training points \( W \), how much would the model \( \hat{\theta} \) change? Concretely, we define a vector \( w \in \{0, 1\}^n \) of sample weights with \( w_i = 1 \) if \( (x_i, y_i) \in W \) and consider the modified parameters

\[
\hat{\theta}(1 - w) = \arg\min_{\theta \in \Theta} \left[ \sum_{i=1}^{n} (1 - w_i) \ell(x_i, y_i; \theta) + \frac{\lambda}{2} \|\theta\|_2^2 \right], \tag{2}
\]

which correspond to retraining the model after excluding \( W \). We will refer to \( w \) directly as the subset (corresponding to \( W \)); the number of removed points as \( \|w\|_1 \); and the fraction of removed points as \( \alpha = \|w\|_1/n \).

The actual effect \( I^*_f : [0, 1]^n \rightarrow \mathbb{R} \) of the subset \( w \) is

\[
I^*_f(w) = f(\hat{\theta}(1 - w)) - f(\hat{\theta}(1)), \tag{3}
\]

where the evaluation function \( f : \Theta \rightarrow \mathbb{R} \) measures a quantity of interest. Specifically, we study:

- The change in test prediction, with \( f(\theta) = \theta^\top x_{test} \). Linear models (for regression or binary classification) make predictions that are functions of \( \theta^\top x_{test} \), so this measures the effect that removing a subset will have on the model’s prediction for some test point \( x_{test} \).
- The change in test loss, with \( f(\theta) = \ell(x_{test}, y_{test}; \theta) \). This is similar to the test prediction.
- The change in self-loss, with \( f(\theta) = \sum_{i=1}^{n} w_i \ell(x_i, y_i; \theta) \), measures the increase in loss on the removed points \( w \). Its average over all subsets of size \( \|w\|_1 \) is the estimated extra loss that leave-\( \|w\|_1 \)-out cross-validation (CV) measures over the training loss, so measuring this allows us to approximate CV with influence.
2.1 Influence functions

The issue with computing the actual effect $\mathcal{I}_f^*(w)$ is that retraining the model to compute $\hat{\theta}(1 - w)$ for each different subset $w$ can be prohibitively expensive. Influence functions provide a relatively efficient first-order approximation to $\mathcal{I}_f(w)$ that avoids retraining \cite{Hampel1986}.

Consider the function $q_w : [0, 1] \to \mathbb{R}$ with $q_w(t) = f(\hat{\theta}(1 - tw))$, such that the actual effect $\mathcal{I}_f^*(w)$ can be written as $q_w(1) - q_w(0)$. We define the predicted effect of the subset $w$ to be its influence $\mathcal{I}_f(w) = q_w(0) \approx q_w(1) - q_w(0)$. Intuitively, this is the effect of removing an infinitesimal weight from each point in $w$ and then linearly extrapolating to removing all of $w$.

It can be computed as

$$\mathcal{I}_f(w) \overset{\text{def}}{=} q_w'(0)$$

$$\quad = \nabla_\theta f(\hat{\theta}(1)) \top \left[ \frac{d}{dt} \hat{\theta}(1 - tw) \right]_{t=0}$$

$$\quad = \nabla_\theta f(\hat{\theta}(1)) \top H_{\lambda,1}^{-1} g_1(w),$$

where $g_1(w) = \sum_{i=1}^n w_i \nabla_\theta \ell(x_i, y_i; \hat{\theta}(1))$, $H_1 = \sum_{i=1}^n \nabla_\theta^2 \ell(x_i, y_i; \hat{\theta}(1))$, and $H_{\lambda,1} = H_1 + \lambda I$. When measuring the change in test prediction or test loss, influence is additive: if $w = w_1 + w_2$, then $\mathcal{I}_f(w) = \mathcal{I}_f(w_1) + \mathcal{I}_f(w_2)$, i.e., the influence of a subset is the sum of influences of its constituent points, so we can efficiently compute the influence of any subset without any model retraining by pre-computing the influence of each individual point. \cite{Koh2017} describe how to compute the influence of all individual points on a test point with a single inverse Hessian-vector product.

2.2 Relation to prior work

Influence functions—introduced in the seminal work of \cite{Hampel1974} and in \cite{Jaeckel1972}, where it was called the infinitesimal jackknife—have a rich history in robust statistics. The use of influence functions in the ML community is more recent, though growing; in Section 1 we provide references for several recent applications of influence functions in ML.

Removing a single training point, especially when the total number of points $n$ is large, represents a small perturbation to the training distribution, so we expect the first-order influence approximation to be accurate. Indeed, prior work on the accuracy of influence has focused on this regime: e.g., \cite{Debruyne2008, Liu2014, Rad2018, Giordano2019} give evidence that the influence on self-loss can approximate LOOCV, and \cite{Koh2017} similarly examined the accuracy of estimating the change in test loss after removing single training points.

However, removing a constant fraction $\alpha$ of the training data represents a large perturbation to the training distribution. To the best of our knowledge, this setting has not been empirically studied; perhaps the closest work is \cite{Khanna2019}'s use of Bayesian quadrature to estimate a maximally influential subset. Instead, older references have alluded to the phenomena we observe: Pregibon et al. (1981) note that influence tends to be conservative, while \cite{Hampel1986} say that “bold extrapolations” (i.e., large perturbations) are often still useful. On the theoretical front, Giordano et al. (2019) established finite-sample error bounds that apply to groups, e.g., showing that the leave-$k$-out approximation is consistent as $k/n \to 0$. Our focus is instead on the relationship of the actual effect $\mathcal{I}_f^*(w)$ and influence $\mathcal{I}_f(w)$ in the regime where $\alpha$ is constant and the error $|\mathcal{I}_f^*(w) - \mathcal{I}_f(w)|$ is large.

3 Empirical accuracy of influence functions on constructed groups

How well do influence functions estimate the effect of (removing) a group of training points? If $n$ is large and we remove a subset $w$ uniformly at random, the new parameters $\hat{\theta}(1 - w)$ should remain close to $\hat{\theta}(1)$ even when fraction of removed points $\alpha$ is non-negligible, so the influence error $|\mathcal{I}_f^*(w) - \mathcal{I}_f(w)|$ should be small. However, we are usually interested in removing coherent, non-random groups, e.g., all points from a data source, or that share some features. In such settings, the parameters $\hat{\theta}(1 - w)$ and $\hat{\theta}(1)$ might differ substantially, and the error $|\mathcal{I}_f^*(w) - \mathcal{I}_f(w)|$ could be

\footnote{In the statistics literature, it is typically defined as the effect of adding weight, so the sign is flipped.}
we picked 3 random test points and the 3 test points with the highest loss (the latter as they seemed
Top), test loss (Mid), and self-loss (Bot) are highly correlated (Spearman ρ
Table 1: Dataset characteristics and the test accuracies that logistic regression achieves (with regular-
variety of subsets, from small (MultiNLI language inference dataset (Williams et al., 2018); we omitted it from the experiments here because
is a chemical-disease relationship (CDR) dataset from Hancock et al. (2018). In Section 5, we will also study the
In Section 5, we will use the CDR dataset and the MultiNLI (Williams et al., 2018) dataset to show
Moreover, the influence of a group tends to underestimate its actual effect in all settings except for
plots), even though the absolute and relative error of the influence approximation can be quite large.
Figure 1 shows that the influences and actual effects of all of these subsets on test prediction
constructed subsets of training points with large positive (or, separately, negative) influence on the
likely candidates for model developers to want to investigate). For each of these 6 test points, we
random points from the same class; and, for comparison, random points from any class. In addition,
we grouped points that shared a similar
Group construction. For each dataset, we grouped points that shared a similar i-th feature value, for random i; points that clustered on their features (or, separately, on their gradients ∇θℓ(𝐱, 𝘺, 𝜃(1))); random points from the same class; and, for comparison, random points from any class. In addition, we picked 3 random test points and the 3 test points with the highest loss (the latter as they seemed likely candidates for model developers to want to investigate). For each of these 6 test points, we constructed subsets of training points with large positive (or, separately, negative) influence on the test loss ℓ(𝐱test, 𝘺test, 𝜃(1)). Intuitively, training points that all have high influence on a test point should act together to change the model substantially. Overall, for each dataset, we constructed 1, 700 subsets ranging in size from 0.25% to 25% of the training points; more details are in Appendix A.2.
Results. Figure 1 shows that the influences and actual effects of all of these subsets on test prediction (Top), test loss (Mid), and self-loss (Bot) are highly correlated (Spearman ρ of 0.89 to 0.99 across all plots), even though the absolute and relative error of the influence approximation can be quite large. Moreover, the influence of a group tends to underestimate its actual effect in all settings except for groups with negative influence on test loss (the left side of each plot in Figure 1 Mid). These trends held across a wide range of regularizations λ, though correlation increased with λ (Appendix C.2).
In Section 5 we will use the CDR dataset and the MultiNLI (Williams et al., 2018) dataset to show that these phenomena of correlation and underestimation also apply to groups of data that arise naturally, and that influence functions can therefore be used to derive insights about real datasets and applications. Before that, we first attempt to develop some theoretical insight into the results above.

1 Except when ℓ(𝐱test, 𝘺test; 𝜃) measures test loss and the actual effect I_f(𝐰) is negative, in which case I_f(𝐰) and I_f(𝐰) still tend to share the same sign, but |I_f(𝐰)| < |I_f(𝐰)| need not hold (Figure 1 Mid).
2 The first 4 datasets involve hospital readmission prediction, spam classification, and object recognition, and were used in Koh and Liang (2017) to study the accuracy of the influence of individual points. The fifth dataset is a chemical-disease relationship (CDR) dataset from Hancock et al. (2018). In Section 5 we will also study the MultiNLI language inference dataset (Williams et al., 2018); we omitted it from the experiments here because its size makes repeated retraining to compute the actual effect too expensive. See Appendix D for dataset details.
Figure 1: Influences vs. actual effects of coherent groups ranging from 0.25% to 25% in size. In Top and Mid, we show results for the test point with highest loss; other test points are similar (Appendix C.1), though with more curvature for test loss (see Appendix C.3). The grey line indicates $y = x$. We omit the top row for MNIST, as $\theta^\top x_{test}$ is not meaningful in the multi-class setting.

4 Analysis

The experimental results above show that there is high correlation and consistent underestimation (as defined in Section 3) between the influences and actual effects of groups of data across a broad variety of datasets, despite the influence approximation incurring large absolute and relative error. As we discussed in Section 2.2, this is far outside the regime of existing theory. Here, we present initial results characterizing the conditions under which the influence $I_f(w)$ and actual effect $I^*_f(w)$ correlate despite high error. We give counterexamples demonstrating that high correlation and underestimation need not always hold. In restricted settings, however, we show that the approximate cone constraint $I_f(w) \lesssim I^*_f(w) \lesssim C_{\text{max}} I_f(w)$ holds for some $C_{\text{max}} \in \mathbb{R}$ that decreases with increasing regularization $\lambda$, which implies both underestimation and correlation when $\lambda$ is high.

Our analysis centers on the one-step Newton approximation, which estimates the change in parameters

$$\hat{\theta}(1 - w) - \hat{\theta}(1) \approx \Delta \theta_{\text{Nt}}(w)$$

$$\overset{\text{def}}{=} (H_{\lambda,1}(1 - w))^{-1} g_1(w),$$

where $H_{\lambda,1}(1 - w) = \sum_{i=1}^{n}(1 - w_i)\nabla^2 \ell(x_i, y_i; \hat{\theta}(1))$ is the empirical Hessian at $\hat{\theta}(1)$ but reweighted after removing the subset $w$. This change in parameters gives the Newton approximation of the effect $I^*_f(w) = f(\hat{\theta}(1) + \Delta \theta_{\text{Nt}}(w)) - f(\hat{\theta}(1))$ and the corresponding Newton error $Err_{\text{Nt}}(w) = I^*_f(w) - I^*_f(w)$, which measures its gap from the actual effect. Specifically, we decompose the difference between the influence and actual effects, $I^*_f(w) - I_f(w)$, into:

$$I^*_f(w) - I_f(w) = I^*_f(w) - I^*_{\text{Nt}}(w) + I^*_{\text{Nt}}(w) - I_f(w).$$
Newton approx. on test prediction

We next characterize the difference between the Newton approximation and influence, \( I \). The Newton approximation is computationally expensive because it computes the Newton error be

\[
\text{Err}_N(w) = \| \nabla^2 f(w) - \nabla^2 f(w) \|_F^2. 
\]

Let the Newton error be

\[
\text{Err}_N(w) = \| \nabla^2 f(w) - \nabla^2 f(w) \|_F^2. 
\]

Proposition 1. Let the Newton error be \( \text{Err}_N(w) = \| \nabla^2 f(w) - \nabla^2 f(w) \|_F^2. \) Then

\[
|\text{Err}_N(w)| \leq \frac{n \| \nabla^2 f(w) \|^2 C_f C_H C_i^2}{\sigma_{\min} + \lambda)^3}. 
\]

\( \text{Err}_N(w) \) only involves third-order or higher derivatives of the loss, so it is 0 for quadratic losses.

Proposition 1 tells us that the Newton approximation is accurate when \( \lambda \) is large or the third derivative of \( \ell(x, y; \cdot) \) (controlled by \( C_H \)) is small. Empirically, the Newton error \( \text{Err}_N(w) \) is small in most of our settings (Figure 2), even though the overall error of the influence approximation \( I_f(w) - I_f(w) \) is still large; this implies the difference must be in between the Newton and influence approximations.

4.2 Characterizing the difference between the Newton approximation and influence

We next characterize the difference between the Newton approximation and influence, \( I_f^N(w) - I_f(w) \), and apply it to the overall decomposition in (6).
We turn to the change in test prediction

\[ I_f^*(w) = I_f(w) - I_f(w) = \nabla_\theta f(\hat{\theta}(1))^\top H_{\lambda,1}^{-1/2} D(w) H_{\lambda,1}^{-1/2} g_1(w) + \frac{1}{2} \Delta \theta_N(w) \nabla^2 f(\hat{\theta}(1)) \Delta \theta_N(w) + \text{Err}_f(w) + \text{Err}_N(w) \]

where we define the error matrix \( D(w) \) as \( D(w) = (I - H_{\lambda,1}^{-1} H_1(w) H_{\lambda,1}^{-1})^{-1} - I \), with \( H_1(w) \) as \( \sum_{i=1}^n w_i \nabla^2 \ell(x_i, y_i; \hat{\theta}(1)) \). \( \text{Err}_f(w) \) measures the error caused by third-order derivatives of \( f(\cdot) \) and is defined in Appendix [4]. Furthermore, \( D(w) \) has eigenvalues between 0 and \( \frac{2}{\lambda} \), and

\[
|\text{Err}_f(w)| \leq \frac{\|w\| C_f C_H^2}{6(\sigma_{\text{min}} + \lambda)^3} \quad \text{and} \quad |\text{Err}_N(w)| \leq \frac{n\|w\| C_f C_H C_f^2}{(\sigma_{\text{min}} + \lambda)^3}.
\]

We can interpret Theorem [1] as a formalization of Hampel et al. [1986]'s observation that influence approximations are accurate when the model is robust and the curvature of the loss is low. In general, the error decreases as \( \lambda \) increases and \( q(\cdot) \) gets less curved; in Figure [C.2], we show that increasing \( \lambda \) reduces error and increases correlation in our experiments. Note that if we hold the per-example regularization \( \lambda/n \) constant, then \( \|I_f^*(w) - I_f(w)\| \to 0 \) as the fraction of removed points \( \alpha = \|w\|_1/n \to 0 \). However, we are interested in the setting where \( \alpha \) is a constant.

### 4.3 The relationship between influence and actual effect on self-loss

Let us now apply Theorem [1] to analyze the behavior of influence under different choices of evaluation function \( f(\cdot) \). We start with the self-loss \( f(\theta) = \sum_{i=1}^n w_i \ell(x_i, y_i; \theta) \), as it is the clearest to characterize the conditions under which the phenomena of correlation and underestimation arise:

**Proposition 2.** Suppose that \( f(\theta) = \sum_{i=1}^n w_i \ell(x_i, y_i; \theta) \). Then

\[ I_f(w) + \text{Err}_f(w) + \text{Err}_N(w) \leq I_f^*(w) \leq (1 + \frac{3\sigma_{\text{max}}}{2\lambda} + \frac{\sigma_{\text{max}}}{2\lambda^2}) I_f(w) + \text{Err}_f(w) + \text{Err}_N(w). \]

The cone constraint in Proposition [2] shows that influence underestimates the actual effect (up to \( O(1/\lambda^3) \) terms, and by an amount that decreases with \( \lambda \)). This explains the previously-unexplained downward bias observed when using influence to approximate LOOCV (Debruyne et al. 2008; Giordano et al. 2019). Moreover, on the graph of influences vs. actual effects, all points lie within the cone bounded by the \( y = x \) and \( y = \frac{1}{\lambda + 3\sigma_{\text{max}}/\lambda^2} x \) lines (up to \( O(1/\lambda^3) \) terms). As \( \lambda \) grows, these lines will converge, and the error terms \( \text{Err}_f(w) \) and \( \text{Err}_N(w) \) will decay at a rate of \( O(1/\lambda^3) \), forcing the influences and actual effects to be tightly correlated.

However, \( \lambda/\sigma_{\text{max}} \) is quite small in our experiments in Section [3], so the actual correlation of influence is better than predicted by this theory: in Figure [1] Bot, the sizes of the theoretically-permissible cones can be quite large, but the points in the graphs nevertheless trace a tight line through the cone.

### 4.4 The relationship between influence and actual effect on a test point

We turn to the change in test prediction \( f(\theta) = \theta^\top x_{\text{test}} \). Here, counterexamples show we cannot obtain a similar cone constraint except in a restricted setting, and that correlation and underestimation do not always hold for this \( f(\cdot) \). Define \( v_{\text{test}} = H_{\lambda,1}^{-1/2} x_{\text{test}} \) and \( v_w = H_{\lambda,1}^{-1/2} g_1(w) \). Theorem [1] gives:

**Corollary 1.** Suppose \( f(\theta) = \theta^\top x_{\text{test}} \). Then \( I_f^*(w) = I_f(w) + v_{\text{test}}^\top D(w)v_w + \text{Err}_N(w) \), where

\[ D(w) = (I - H_{\lambda,1}^{-1} H_1(w) H_{\lambda,1}^{-1})^{-1} - I \]

is the error matrix from Theorem [1].

Unfortunately, Corollary [1] implies that no cone constraint applies: if the error matrix \( D(w) \) rotates \( v_w \) relative to \( v_{\text{test}} \), then we can find \( x_{\text{test}} \) such that the influence \( I_f(w) = v_{\text{test}}^\top v_w = 0 \) but the Newton approximation \( I_f^*(w) = I_f(w) + v_{\text{test}}^\top D(w)v_w \) is large. Figure [3] left shows that on synthetic
We wrap up our analysis with a brief note on measuring the change in test loss.

The analysis in Section 4 shows that the influences of groups vs. actual effects, all points lie within the cone bounded by the $y$-correlation between influence and actual effects is much stronger than the theory predicts. We close

Figure 3: Influence $I_f(w)$ vs. Newton approximation $I_f^{\text{Nt}}(w)$ on the test prediction on the two synthetic datasets detailed in Appendix D.1. Left: We adversarially choose a set of $w$'s such that $I_f(w)$ and $I_f^{\text{Nt}}(w)$ can have different signs and need not correlate. Right: Even if we only remove (repeats of) single distinct points, we can control the scaling factor $d(w)$ between $I_f(w)$ and $I_f^{\text{Nt}}(w)$ on different groups, so low $I_f(w)$ need not imply low $I_f^{\text{Nt}}(w)$.

data, $I_f(w)$ and $I_f^{\text{Nt}}(w)$ can even have opposite signs on some subsets $w$; thus, underestimation does not always hold even if the Newton error $\text{Err}_{\text{Nt}}(w)$ is small.

We can recover a cone constraint similar to Proposition 2 if we restrict our attention to the special case where we use a margin-based model and remove (possibly multiple copies) of a single point:

**Proposition 3.** Consider a binary classification setting with $y \in \{-1, +1\}$ and a margin-based model with loss $\ell(x, y; \theta) = \phi(y\theta^\top x)$ for some $\phi : \mathbb{R} \to \mathbb{R}_+$. Suppose $f(\theta) = \theta^\top x_{\text{test}}$ and that the subset $w$ comprises $||w||_1$ identical training points. Let $(x_k, y_k)$ be a representative training point from the subset. Then the actual effect $I_f^A(w)$ is related to the influence $I_f(w)$ according to

$$I_f^A(w) = \frac{I_f(w)}{1-||w||_1 \cdot \phi'_k x_k^\top H_{\lambda, 1}^{-1} x_k} + \text{Err}_{\text{Nt}}(w),$$

where $\phi_k \text{ def } = \phi(y_k \hat{\theta}(1)^\top x_k)$. This implies that $I_f^A(w)$ and $I_f(w)$ follow the cone constraint

$$I_f(w) + \text{Err}_{\text{Nt}}(w) \leq I_f^A(w) \leq \left(1 + \frac{\sigma_{\text{max}}}{\lambda}\right)I_f(w) + \text{Err}_{\text{Nt}}(w).$$

Similar to Proposition 2, Proposition 3 shows that when removing copies of a single point, the influence underestimates the actual effect (up to $\text{Err}_{\text{Nt}}(w)$ error). Moreover, on the graph of influences vs. actual effects, all points lie within the cone bounded by the $y = x$ and $y = \lambda x/(\lambda + \sigma_{\text{max}})$ lines, plus $\text{Err}_{\text{Nt}}(w)$ noise. As $\lambda/\sigma_{\text{max}}$ grows, the cone shrinks, and correlation increases.

However, if $\lambda/\sigma_{\text{max}}$ is small (as in our experiments in Section 5), the cone is wide, and the scaling factor $d(w) = 1/(1-||w||_1 \cdot \phi'_k x_k^\top H_{\lambda, 1}^{-1} x_k)$ in Proposition 3 can be quite large for some subsets $w$ but not for others. In particular, $d(w)$ is large when there are few remaining points in the direction of the removed points. In Figure 3 Right, we exploit this fact to show that the influence $I_f(w)$ and Newton approximation $I_f^{\text{Nt}}(w)$ can exhibit low correlation (e.g., low $I_f(w)$ need not mean low $I_f^{\text{Nt}}(w)$), even in the simplified setting of removing copies of single points. We comment on the analogue of $d(w)$ in the general multiple-point setting in Appendix D.2.

### 4.5 Linking test prediction and test loss

We wrap up our analysis with a brief note on measuring the change in test loss $f(\theta) = \ell(x_{\text{test}}, y_{\text{test}}; \theta)$. In the margin-based setting, the loss $\ell(x_{\text{test}}, y_{\text{test}}; \theta)$ is a monotone function of the linear prediction $\theta^\top x_{\text{test}}$. Thus, measuring $f(\theta) = \ell(x_{\text{test}}, y_{\text{test}}; \theta)$ will display the same rank correlation as measuring $f(\theta) = \theta^\top x_{\text{test}}$ above, so the same results about correlation (or lack thereof) carry over.

However, the second-order $f$-curvature term $\frac{1}{2} \Delta \theta_{\text{Nt}}(w)^\top \nabla^2 f(\hat{\theta}(1)) \Delta \theta_{\text{Nt}}(w)$ from Theorem 1 is always non-negative, even if the influence is negative. Under the assumption that $\text{Err}_f(w)$ and $\text{Err}_{\text{Nt}}(w)$ are both small because they decay as $O(1/\lambda^3)$, this implies that underestimation is only preserved when the influence is positive, as we observed empirically in Figure 1 Mid.

### 5 Applications of influence functions on natural groups of data

The analysis in Section 4 shows that the influences of groups $w$ need not always be correlated with their actual effects. Nonetheless, our experiments in Section 3 demonstrate that on real datasets, the correlation between influence and actual effects is much stronger than the theory predicts. We close
by showing that we can exploit this correlation to glean insights in two case studies on the CDR and MultiNLI datasets, where groups of data arise naturally.

**CDR.** The CDR dataset tackles the following task: given text about the relationship between a chemical and a disease, predict if the chemical causes the disease. It was collected via data programming, where users provide labeling functions (LFs)—instead of labels—that take in an unlabeled point and either abstain or output a heuristic label (Ratner et al., 2016). Specifically, Hancock et al. (2018) asked annotators for natural language explanations of provided classifications; parsed those explanations into LFs; and used those LFs to label a large pool of data (Appendix B.1). We studied the effect of each LF by computing the influence of the groups of points that each LF labeled; these correlated with their actual effects on overall test loss (Spearman $\rho = 1$; Figure C.5).

We used influence functions to study two important properties of LFs: *coverage*, the fraction of unlabeled training points they output a label on; and *precision*, the proportion of labels they output that are correct. We found that high coverage was essential for LFs to help overall test performance (Figure 4-Left), though precision was not predictive of influence (Figure 4-Mid). This suggests that in data programming, annotators should craft LFs that are broadly applicable, even if imprecise.

**MultiNLI.** The MultiNLI dataset deals with natural language inference: determining if a pair of sentences agree, contradict, or are neutral. Williams et al. (2018) presented crowdworkers with initial sentences from five genres and asked them to generate follow-on sentences that were neutral or in agreement/contradiction (Appendix B.2). We studied the effect that each crowdworker had on the model’s test set performance by computing the influence of the examples they created on overall test loss (Spearman $\rho$ of 0.77 to 0.86 with actual effects across different genres; see Figure C.8).

Studying the influence of each crowdworker revealed that the number of examples they created was not predictive of influence on test performance: e.g., the most prolific crowdworker contributed 35,000 examples but ended up hurting overall test performance (Figure 4-Right). Curiously, this effect was genre-specific: crowdworkers who improved performance on some genres would lower performance on others (Figure C.10), even though the number of examples a crowdworker contributed to a genre did not track their influence on it (Figure C.11). Identifying precisely what makes a crowdworker’s contributions useful could help us improve dataset collection and credit attribution as well as better understand how models transfer across genres.

**6 Discussion**

In this paper, we showed empirically that the influences of groups of points are highly correlated with, and consistently underestimate, their actual effects across a range of datasets, types of groups, and sizes ranging from 0.25% to 25% of the training data. Our analysis of this surprising observation reveals that while these phenomena are provably true in some restricted settings, they need not always hold in a more general and realistic setting. This gap between theory and experiments opens up important directions for future work: why do we observe such striking, even linear, correlation between predicted and actual effects on real data? To what extent is this due to the specific model, datasets, or subsets used? Our work suggests that there could be distributional assumptions that hold for real data and give rise to the broad phenomena of correlation and underestimation.

The correlation of the influence of groups with their actual effects lets us use influence functions to better understand the “different stories that different parts of the data tell,” in the words of Hampel et al. (1986). As applications, we showed that we can gain insight into the effects of a labeling
function in data programming, or a crowdworker in a crowdsourced dataset, by computing the influence of their corresponding group effects. While these applications involved predefined groups, influence functions could potentially also discover coherent, semantically-relevant groups in the data. We also note that influence functions can be used to approximate Shapley values, which are a different but related way of characterizing the effect of data points; see Jia et al. (2019) and Ghorbani and Zou (2019) for discussion.

Finally, the framework of influence functions can also be applied to studying the effects of adding training points. In this context, the phenomenon of underestimation turns into overestimation, i.e., the influence of adding a group of training points tends to overestimate the actual effect of adding that group. This raises the possibility of using influence functions to evaluate the vulnerability of a given dataset and model to data poisoning attacks (Steinhardt et al., 2017).

Reproducibility

The code and data for replicating our experiments will be available on Github [http://bit.ly/gt-group-influence] and Codalab [http://bit.ly/cl-group-influence].

Acknowledgments

We are grateful to Zhenghao Chen, Brad Efron, Jean Feng, Tatsunori Hashimoto, Robin Jia, Stephen Mussmann, Aditi Raghunathan, Marco Túlio Ribeiro, Noah Simon, Jacob Steinhardt, and Jian Zhang for helpful discussions and comments. We are further indebted to Ryan Giordano, Ruoxi Jia, and Will Stephenson for discussion about prior work, and Samuel Bowman, Braden Hancock, Emma Pierson, and Pranav Rajpurkar for their assistance with applications and datasets. This work was funded by an Open Philanthropy Project Award. PWK was supported by the Facebook Fellowship Program.

References

I. Arrieta-Ibarra, L. Goff, D. Jiménez-Hernández, J. Lanier, and E. G. Weyl. Should we treat data as labor? Moving beyond “free”. In American Economic Association Papers and Proceedings, volume 108, pages 38–42, 2018.

S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.

I. Chen, F. D. Johansson, and D. Sontag. Why is my classifier discriminatory? In Advances in Neural Information Processing Systems (NeurIPS), pages 3539–3550, 2018.

M. Debruyne, M. Hubert, and J. A. Suykens. Model selection in kernel based regression using the influence function. Journal of Machine Learning Research (JMLR), 9(0):2377–2400, 2008.

A. Ghorbani and J. Zou. Data shapley: Equitable valuation of data for machine learning. arXiv preprint arXiv:1904.02868, 2019.

R. Giordano, W. Stephenson, R. Liu, M. Jordan, and T. Broderick. A Swiss Army infinitesimal jackknife. In Artificial Intelligence and Statistics (AISTATS), pages 1139–1147, 2019.

F. R. Hampel. The influence curve and its role in robust estimation. Journal of the American Statistical Association, 69(346):383–393, 1974.

F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. Robust Statistics: The Approach Based on Influence Functions. Wiley, 1986.

B. Hancock, P. Varma, S. Wang, M. Bringmann, P. Liang, and C. Ré. Training classifiers with natural language explanations. In Association for Computational Linguistics (ACL), 2018.

J. Hayes and O. Ohrimenko. Contamination attacks and mitigation in multi-party machine learning. In Advances in Neural Information Processing Systems (NeurIPS), pages 6604–6615, 2018.

L. A. Jaeckel. The infinitesimal jackknife. Unpublished memorandum, Bell Telephone Laboratories, Murray Hill, NJ, 1972.
R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gurel, B. Li, C. Zhang, D. Song, and C. Spanos. Towards efficient data valuation based on the shapley value. *arXiv preprint arXiv:1902.10275*, 2019.

R. Khanna, B. Kim, J. Ghosh, and O. Koyejo. Interpreting black box predictions using Fisher kernels. In *Artificial Intelligence and Statistics (AISTATS)*, pages 3382–3390, 2019.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, 2017.

P. W. Koh, J. Steinhardt, and P. Liang. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2019.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

J. T. Leek, R. B. Scharpf, H. C. Bravo, D. Simcha, B. Langmead, W. E. Johnson, D. Geman, K. Baggerly, and R. A. Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10), 2010.

Y. Liu, S. Jiang, and S. Liao. Efficient approximation of cross-validation for kernel methods using Bouligand influence function. In *International Conference on Machine Learning (ICML)*, pages 324–332, 2014.

V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam filtering with naive Bayes – which naive Bayes? In *CEAS*, volume 17, pages 28–69, 2006.

D. Pregibon et al. Logistic regression diagnostics. *Annals of Statistics*, 9(4):705–724, 1981.

K. R. Rad and A. Maleki. A scalable estimate of the extra-sample prediction error via approximate leave-one-out. *arXiv preprint arXiv:1801.10243*, 2018.

A. J. Ratner, C. M. D. Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3567–3575, 2016.

P. Schulam and S. Saria. Can you trust this prediction? Auditing pointwise reliability after learning. In *Artificial Intelligence and Statistics (AISTATS)*, pages 1022–1031, 2019.

J. Steinhardt, P. W. Koh, and P. Liang. Certified defenses for data poisoning attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore. Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records. *BioMed Research International*, 2014, 2014.

H. Wang, B. Ustun, and F. P. Calmon. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. *arXiv preprint arXiv:1901.10501*, 2019.

C. Wei, Y. Peng, R. Leaman, A. P. Davis, C. J. Mattingly, J. Li, T. C. Wiegers, and Z. Lu. Overview of the BioCreative V chemical disease relation (cdr) task. In *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, pages 154–166, 2015.

A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Association for Computational Linguistics (ACL)*, pages 1112–1122, 2018.

J. Zhou, Z. Li, H. Hu, K. Yu, F. Chen, Z. Li, and Y. Wang. Effects of influence on user trust in predictive decision making. In *Conference on Human Factors in Computing Systems (CHI)*, 2019.
A Experimental details for comparing influence vs. actual effects on constructed groups

A.1 Model training

For all experiments in Section 3, we trained a logistic regression model (or softmax for multiclass) using `sklearn.linear_model.LogisticRegression.fit`, fitting the intercept but only applying $L_2$-regularization to the weights. To choose the regularization strength $\lambda$, we conducted 5-fold cross-validation across 10 possible values of $\lambda/n$ logarithmically spaced between $1.0 \times 10^{-4}$ and $1.0 \times 10^{-1}$, inclusive, selecting the regularization that yielded the highest cross-validation accuracy (except on the CDR dataset, where we selected regularization based on cross-validation F1 score to account for class imbalance as per Hancock et al. (2018)'s procedure).

A.2 Group construction

For each dataset, we constructed groups of various sizes relative to the entire dataset by considering 100 sizes linearly spaced between 0.25% and 25% of the dataset. For each of these 100 sizes, we constructed one group with each of the following methods:

1. Shared features: We selected a single feature uniformly at random and sorted the dataset along this selected feature. Next, we selected an training point uniformly at random. We then constructed a group of size $s$ that consisted of the $s$ unique training points that were closest to the chosen point, as measured by their values in the selected feature.

2. Feature clustering: We clustered the dataset with respect to raw features via `scipy.cluster.hierarchy.fclusterdata` with $t$ set to 1, as well as with `sklearn.cluster.KMeans.fit` with `n_clusters` taking on values 4, 8, 16, 32, 64, 128. Since hierarchical clustering determines cluster sizes automatically with a principled heuristic and we try a range of values for `n_clusters` in $k$-means, this recovers clusters with a large range of sizes. The clustering with `n_clusters = 4` also guarantees (via the pigeonhole principle) that there is at least one cluster which contains at least 25% of the dataset. From all the clusters that are at least the size of the desired group, we chose one uniformly at random and chose the group uniformly at random and without replacement from the training points in this cluster.

3. Gradient clustering: We followed the same procedure as “Feature clustering,” except that we clustered the dataset with respect to $\nabla_\theta \ell(x, y; \hat{\theta}(1))$, i.e. each training point was represented by the gradient of the loss on that point.

4. Random within class: We considered all classes with at least as many training points as the desired group. From these classes, we chose one uniformly at random. Then, we chose the group uniformly at random and without replacement from all training points in this class.

5. Random: We picked a group uniformly at random and without replacement from the entire dataset.

The above methods gave us a total of 500 groups (100 groups per method) for each dataset, with the exception of the “random within class” method for MNIST. Since MNIST has 10 classes, each with only 10% of the data, we skipped over groups of size $> 10\%$ just for the “random within class” groups.

In addition, we selected 3 random test points and the 3 test points with highest loss; we intend these to represent the average case and the more extreme case that may be relevant to model developers who want to debug errors that their model outputs. For each of these 6 test points, we selected groups that had large positive influence on its test loss. More specifically, we proceeded in 3 stages:

1. We considered 33 group sizes linearly spaced between 0.25% and 2.5% of the dataset, and for any size $s$ out of these 33, we selected a group uniformly at random and without replacement from training points in the top $1.5 \times 2.5\%$ of the dataset, ordered according to their influence on the test point of interest.

2. This was similar to the first stage, but with 33 sizes spaced between 0.25% and 10% and groups chosen from the top $1.5 \times 10\%$ of the dataset.
3. Finally, we considered 34 sizes spaced between 0.25% and 25%, with groups chosen from the top 1.5 × 25% of the dataset.

Larger groups tend to have lower average influence than smaller groups, since by necessity, we need to construct larger groups out of training points further from the top. This multi-stage approach ensured that we would select small groups with both a high average influence and also with a low average influence, so that we could compare them to larger groups and mitigate confounding the group size with its average influence.

Finally, we repeated this last method of group construction for groups with large negative influence on test point loss.

Using these 6 test points, we generated 1,200 groups (100 subsets per group, with 6 test points, and drawing from the positive and negative tails). In total, we therefore generated 1,700 groups per dataset (except MNIST).

A.3 Comparison of influence and actual effect

To produce Figure 1, we selected groups as described in Appendix A.2. We retrained the model once for each group, excluding the relevant group in order to calculate its actual effect. To compute all groups’ influences, we first calculated the influence of every individual training point using the procedure of Koh and Liang (2017). Then, to compute the influence on test prediction or loss of some group, we simply added the relevant individual influences (in CDR, we weighted these individual influences according to that point’s weight; see Appendix B.1). To compute the influence on self-loss of some group, we summed up the gradients of the loss of each training point to compute $g_1(w)$, we calculated the inverse Hessian vector product $H_{1,1}g_1(w)$ and took its dot product with $g_1(w)$ (again, we modified this with appropriate weighting for individual points in CDR).

B Dataset details

We used the same versions of the Diabetes, Enron, Dogfish, and MNIST datasets as Koh and Liang (2017), since the examination of the accuracy of influence functions for large perturbations is a natural extension of their studies of small perturbations. Additionally, we applied influence to more natural settings in CDR and MultiNLI; here, we discuss their preprocessing pipelines.

B.1 CDR

Hancock et al. (2018) established the BabbleLabble framework for data programming, following the following pipeline: They took labeled examples with natural language explanations, parsed the explanations into programmatic LFs via a semantic parser, and filtered out obviously incorrect LFs. Then, they applied the remaining LFs to unlabeled data to create a sparse label matrix, from which they learned a label aggregator that outputs a noisily labeled training set. Finally, they ran $L_2$-regularized logistic regression on a set of basic linguistic features with the noisy labels.

They demonstrated their method on three datasets: Spouse, CDR, and Protein. The Protein dataset was not publicly available, and the vast majority of Spouse was labeled by a single LF, hence we chose to use CDR. This dataset’s associated task involved identifying whether, according to a given sentence, a given chemical causes a given disease. The sentences and ground truth labels were sourced from the 2015 BioCreative chemical-disease relation dataset (Wei et al., 2015).

In our application, we began with their 28 LFs and the corresponding label matrix. For simplicity, we did not learn a label aggregator; instead, if an example $x$ was given labels $y_{i_1}, y_{i_2}, \ldots, y_{i_k}$ by $k$ LFs $i_1, \ldots, i_k$, then we created $k$ copies of $x$, each with weight $1/k$. The subset of points corresponding to LF $i_1$ then included one instance of $x$ with weight $1/k$. This weighting was taken into account in model training as well as in calculations of influence and actual effect. In addition, we used $L_1$-regularization to reduce the number of features to 330 while still achieving similar F1 score to Hancock et al. (2018); they reported an F1 of 42.3, while we achieved 42.2.

We note that in BabbleLabble, a given LF can never output positive on one example but negative on another. Hence, some LFs are positive (unable to output negative and only able to abstain or output
positive), while the others are negative (unable to output positive and only able to abstain or output negative).

B.2 MultiNLI

Williams et al. (2018) created the MultiNLI dataset for the task of natural language inference: determining if a pair of sentences agree, contradict, or are neutral. To do so, they presented crowdworkers with initial sentences and asked them to generate follow-on sentences that were neutral or in agreement/contradiction. Thus, each of the 380 crowdworkers generated a subset of the dataset. We used these subsets in our application of influence.

The training set consisted of 392,702 examples from five genres. The development set consisted of 10,000 “matched” examples from the same five genres as the training set, as well as 10,000 “mismatched” examples from five new genres. The test set was put on Kaggle as an open competition, hence we do not have its labels and could not use it; instead, the development set was treated as the test set.

The continuous bag-of-words baseline in Williams et al. (2018) first converted the raw text of each sentence in the pair into a vector by treating the sentence as a continuous bag of words and simply averaging the 300D GloVe vector embeddings. This converted a pair of sentences into vectors $a$, $b$. They then concatenated $[a, b, a-b, a \times b]$ into a 1200D vector, where $a \times b$ denotes the element-wise product. Finally, they treated this as input to a neural net with three hidden layers and finetuned the entire model, including word embeddings (more details in Williams et al. (2018)).

For our application, we truncated their baseline and just used the concatenation of $a$ and $b$ as the representation for every example. By running logistic regression on this, we achieved test accuracy of 50.4% (vs. their baseline’s 64.7%). Future work could explore influence in the setting of more complex and higher-performing models.

C Additional experiments

C.1 Representative test points

![Influence on test prediction](image1)

![Influence on test loss](image2)

**Figure C.1**: Influences vs. actual effects of the same coherent groups in Figure 1, but on test points closer to the median (within the 40th to 60th percentile) of the test loss distribution. We consider these to represent average test points. On these, influence on the test prediction remains well-correlated with the actual effect.

C.2 Regularization

In Section 4, our bounds show that influence ought to be closer to actual effect as regularization increases. Here, we support this claim empirically on Diabetes, Enron, Dogfish, and MNIST (small).
Figure C.2: The effect of regularization for a representative test point. Red frame lines indicate the existence of points exceeding those bounds. We did not include the test prediction for MNIST (small) because the margin is not well-defined for a multiclass model.

Figure C.3: As regularization increases, correlation increases between the influences and actual effects on test prediction (Left), test loss (Middle), and self-loss (Right).

To do so, for each dataset, we selected a range of values for $\lambda/n$, and we selected subsets as described in Appendix A.2. We then computed the influence and actual effect of each of these subsets on a representative test point’s prediction, that point’s loss, and on self-loss (Figure C.2).

This experiment required us to retrain the model for every value of $\lambda$ and for every subset. Thus, for computational purposes, we omitted CDR and MultiNLI, and we selected a random 10% subset of MNIST’s training set to use in place of all of MNIST.
In Figure C.3, we observe the trend that correlation generally increases as $\lambda$ does. Specifically, we computed the Spearman $\rho$ between the influence and actual effect for each dataset, each value of $\lambda$, and each evaluation function $f(\cdot)$ of interest (i.e., test prediction, test loss, or self-loss).

### C.3 The effect of loss curvature on the accuracy of influence

One takeaway from the results on test loss in Figure 1-Mid is that the curvature of $f(\theta)$ can significantly increase approximation error; this is expected since the influence $I_f(w)$ linearizes $f(\cdot)$ around $\hat{\theta}(1)$. When possible, choosing a $f(\cdot)$ that has low curvature (e.g., the linear prediction) will result in higher accuracy. We can mitigate this by using influence to approximate the parameters $\hat{\theta}(1-w)$ and then plug that estimate into $f(\cdot)$ (Figure C.4), though this can be more computationally expensive.

Note that Figure C.4 shows that this technique does not help much for measuring self-loss. However, in the context of LOOCV, the computational complexity of the Newton approximation for self-loss (described in Section 4) is similar to that of the influence approximation, so we encourage the use of the Newton approximation for LOOCV (as in [Rad and Maleki (2018)]); Figure C.4 shows that this leads to more accurate approximations for self-loss.

### C.4 Additional analysis of influence functions applied to natural groups of data

In Section 5, we considered the CDR and MultiNLI datasets, which contain the natural subsets of LFs and crowdworkers, respectively. To draw inferences about these subsets, we took the $L_2$-regularized logistic regression model described in Appendix A, calculated the influence of the LF/crowdworker subsets, and retrained the model once for each LF/crowdworker.

**CDR.** As discussed in Appendix B.1, an LF is either positive or negative, where a positive LF can only give positive labels or abstain, and similarly for negative LFs. Because of this stark class separation, we indicate whether an LF is positive or negative, and we consider LF influence on the positive test examples separately from their influence on the negative test examples. To measure an LF’s influence and actual effect on a set of test points, we simply add up its influence and actual effect on the set’s individual test points.

In Figure C.5, we note that influence is a good approximation of an LF’s actual effect, just as with other kinds of subsets as well as other datasets (Figure 1). Furthermore, we observe that positive LFs improve the overall performance of the positively labeled portion of the test set while hurting the negatively labeled portion of the test set, and vice versa for negative LFs. This dichotomous effect further motivates the analysis of influence on the positive test set separately from the negative test set, since the process of adding these two influences to study the influence on the entire test set would obscure the full story.

Next, we define an LF’s *coverage* to be the proportion of the examples that it does not abstain on, which can be measured through the number of examples in its corresponding subset. In Figure C.6 we observe that the magnitude of influence correlates strongly with coverage.

Finally, we define an LF’s *precision* to be the number of examples it labels correctly divided by the number of examples it does not abstain on. Because the dataset had many more negative than positive examples, positive LFs had lower precision than negative LFs. Surprisingly, even when this effect was taken into account and we considered positive LFs separately from negative ones, precision did not correlate with influence (Figure C.7).

We conclude that annotators should aim for heuristic LFs with high coverage, not high precision.

**MultiNLI.** As discussed in Appendix B.2, the training set consisted of five genres, and the test set consisted of a matched portion with the same five genres, as well as a mismatched portion with five new genres. For succinctness, we refer to the influence/actual effect of the set of examples generated by a single crowdworker as that crowdworker’s influence/actual effect.

First, we note in Figure C.8 that influence is a good approximation of a crowdworker’s actual effect for both matched and mismatched test sets, consistent with our findings in Figure 1 for other subset types and datasets.
Figure C.4: The top 3 rows show influence on test loss (with the same test points as in Figure 1), while the bottom 3 show self-loss. Within each set, the first row shows the influence vs. actual effect (as in Figure 1); the second shows the predicted effect obtained by estimating the change in parameters via influence and then evaluating $f(\cdot)$ directly on those parameters; and the third shows the Newton approximation.
Figure C.5: We observe both correlation and underestimation for LFs on the positive and negative test sets. We also see that positive LFs help the positive test set and hurt the negative test set; vice versa for negative LFs.

Figure C.6: The magnitude of LF influence correlates with coverage. This figure is an extension of Figure 4-Left: there, we showed the influence of positive LFs on the positive test set and the influence of negative LFs on the negative test set. Here, we additionally show the influence of positive LFs on the negative test set and vice versa.

Figure C.7: LF influence does not correlate with precision. Similar to Figure C.6, this figure is an extension of Figure 4-Mid: there, we showed the influence of positive LFs on the positive test set and the influence of negative LFs on the negative test set. Here, we additionally show the influence of positive LFs on the negative test set and vice versa.

Unlike in CDR (Figure C.6), we do not find strong correlation between a crowdworker’s influence and the number of examples they contributed; it is possible to contribute many examples but have relatively little influence (Figure C.9).
The most prolific crowdworker contributed 35,000 examples and had large negative influence on the test set. A closer analysis revealed that they had positive influence on the fiction genre but lowered performance on many other genres, despite contributing roughly equally to each genre. This genre-specific trend tended to hold more broadly among the workers: there appear to be two categories of genres (fiction, facetoface, nineeleven vs. travel, government, verbatim, letters, oup) such that each worker tended to have positive influence on all genres in one category and negative influence on all genres in the other (Figure C.10). Moreover, the number of examples a worker contributed to a given genre was not a good indicator for their influence on that genre (Figure C.11).

D Additional analysis on influence vs. actual effect on a test point

D.1 Counterexamples

For Figure 3, we constructed two binary datasets in which the influence of a certain class of subsets on the test prediction of a single test point exhibits pathological behavior.
Figure C.10: Workers tended to have positive influence on fiction, facetoface, and nineeleven and negative influence on travel, government, verbatim, letters, and oup (or vice versa). In this plot, we allowed for full color saturation when the magnitude of the total influence on the test set (matched) exceeded 0.8.

Figure C.11: Influence on a genre does not correlate with number of contributions to that genre.

**Rotation effect.** In Figure 3-Left, our aim was to show that there can be a dataset with subsets such that the cone constraint discussed in Section 4.4 does not hold.

The rotation effect described in Corollary 1 is due to the angular difference between the change in parameters predicted by the influence approximation, \( \Delta \theta_{\text{inf}}(w) \stackrel{\text{def}}{=} H_{\lambda,1}^{-1} \theta_1(w) = H_{\lambda,1}^{-1,2} v_w \), and the change in parameters predicted by the Newton approximation, \( \Delta \theta_{\text{Nt}}(w) = H_{\lambda,1}^{-1} (D(w) + I) v_w \). If \( \Delta \theta_{\text{inf}}(w) \) and \( \Delta \theta_{\text{Nt}}(w) \) are linearly independent, then for any pair of target values \( a, b \in \mathbb{R} \), we can find some \( x_{\text{test}} \) such that \( \mathcal{I}_f(w) = x_{\text{test}}^\top \Delta \theta_{\text{inf}}(w) = a \) and \( \mathcal{I}_{Nt}(w) = x_{\text{test}}^\top \Delta \theta_{\text{Nt}}(w) = b \).

To exploit this, we constructed the MoG dataset as an equal mixture of two standard (identity covariance) Gaussian distributions in \( \mathbb{R}^{60} \), one for each class, and with means \((-1/2, 0, \ldots, 0)\) and \((1/2, 0, \ldots, 0)\), respectively. In particular:

1. We sampled 60 examples from each class for a total of \( n = 120 \) training points, and set the regularization strength \( \lambda = 0.001 \).
2. We then computed \( \Delta \theta_{\text{inf}}(w) \) and \( \Delta \theta_{\text{Nt}}(w) \) for each pair of training points and chose the 120 pairs of training points with the largest angles between \( \Delta \theta_{\text{inf}}(w) \) and \( \Delta \theta_{\text{Nt}}(w) \).
3. Finally, we solved a least-squares optimization problem to find \( x_{\text{test}} \) for which \( \mathcal{I}_f(w) \) and \( \mathcal{I}_{Nt}(w) \) are approximately decorrelated.

Note that we adversarially chose which subsets to study in this counterexample, since our main goal was to show that there existed subsets for which the cone constraint did not hold. For the next counterexample, we instead study all possible subsets in the restricted setting of removing copies of single points.
Scaling effect. In Figure 3-Right, our aim was to construct a dataset such that even if we only removed subsets comprising copies of single distinct points, a low influence need not translate into a low actual effect.

To do so, we constructed the Ortho dataset that contains 2 repeated points of opposite classes on each of the 2 canonical axes of $\mathbb{R}^2$ (for a total of 4 distinct points). By varying their relative distances from the origin, we can control the influence of removing one of these points as well as the rate that the scaling factor $d(w)$ grows as we remove more copies of the same point. Furthermore, because the axes are orthogonal, we can control $d(w)$ independently for each repeated point. We fix the test point $x_{\text{test}} = (1, 1)$. Maximizing $d(w)$ for one axis and minimizing it for the other produces the two distinct lines in Figure 3-Right.

D.2 Scaling effects when removing multiple points

In the general setting of removing subsets of different points, the analogous failure case to a varying scaling factor $d(w)$ (Figure 3-Right) is the varying scaling effect that the error matrix $D(w)$ in Theorem 1 can have on different subsets $w$. The range of this effect is bounded by the spectral norm of $D(w)$. This norm is precisely equal to $d(w)$ in the single-point setting, and it is large when we remove a subset $w$ whose Hessian $H_1(w)$ is almost as large as the full Hessian $H_{\lambda,1}$ in some direction. As with $d(w)$, the spectral norm of $D(w)$ decreases with $\lambda$ (Theorem 1), so as regularization increases, we expect that the influence of a group will track its actual effect more accurately.

E Proofs

E.1 Notation

We first review the notation given in Section 2 and introduce new definitions that will be useful in the sequel. We define the empirical risk as

$$L_s(\theta) \equiv \left[ \sum_{i=1}^{n} s_i \ell(x_i, y_i; \theta) \right] + \frac{\lambda}{2} \| \theta \|_2^2,$$

such that the optimal parameters are $\hat{\theta}(s) \equiv \arg \min_{\theta \in \Theta} L_s(\theta)$.

Given sample weight vectors $r, s \in \mathbb{R}^n$, we define the derivatives

$$g_r(s) \equiv \sum_{i=1}^{n} s_i \nabla_\theta \ell(x_i, y_i; \hat{\theta}(r))$$

$$H_r(s) \equiv \sum_{i=1}^{n} s_i \nabla_\theta^2 \ell(x_i, y_i; \hat{\theta}(r)).$$

If the argument $s$ is omitted, it is assumed to be equal to $r$. For example,

$$H_1 \equiv \sum_{i=1}^{n} \nabla_\theta^2 \ell(x_i, y_i; \hat{\theta}(1)).$$

If $H$ has a $\lambda$ subscript, then we add $\lambda I$. For example,

$$H_{\lambda,1} \equiv H_1 + \lambda I.$$

For a given dataset, we define the following constants:

$$C_\ell = \max_{1 \leq i \leq n} \left\| \nabla_\theta \ell(x_i, y_i; \hat{\theta}(1)) \right\|_2,$$

$$\sigma_{\min} = \text{smallest singular value of } H_1,$$

$$\sigma_{\max} = \text{largest singular value of } H_1.$$
To avoid confusion with the vector $2$-norm, we will use the operator norm $\| \cdot \|_{\text{op}}$ to denote the matrix $2$-norm.

In the sequel, we study the order-3 tensor $\nabla^{3}_\theta f(\hat{\theta}(1))$. We define its product with a vector (which returns a matrix) as a contraction along the last dimension:

$$\left\langle \nabla^{3}_\theta f(\hat{\theta}(1)), v \right\rangle_{ij} \overset{\text{def}}{=} \sum_{k} \frac{\partial^{3}f(\hat{\theta}(1))}{\partial \theta_{i} \partial \theta_{j} \partial \theta_{k}} v_{k}.$$ 

E.2 Assumptions

We make the following assumptions on the derivatives of $\ell(x, y, \cdot)$ and $q(\cdot)$.

**Assumption 1** (Positive-definiteness and Lipschitz continuity of $H$). The loss $\ell(x, y, \theta)$ is convex and twice-differentiable, with positive regularization $\lambda > 0$. Further, there exists $C_H \in \mathbb{R}$ such that

$$\| \nabla^{2}_{\theta} \ell(x, y, \theta_1) - \nabla^{2}_{\theta} \ell(x, y, \theta_2) \|_{\text{op}} \leq C_H \| \theta_1 - \theta_2 \|_2$$

for all $(x, y) \in X \times Y$ and $\theta_1, \theta_2 \in \Theta$. This is a bound on the third derivative of $\ell$, if it exists.

**Assumption 2** (Bounded derivatives of $f$). $f(\theta)$ is thrice-differentiable, with $C_{f, f, 3} \in \mathbb{R}$ such that

$$C_{f} = \sup_{\theta \in \Theta} \| \nabla_{\theta} f(\theta) \|_2, \quad C_{f, 3} = \sup_{v \in \Theta, \|v\|_2 = 1} \left\| \left\langle \nabla^{3}_{\theta} f(\hat{\theta}(1)), v \right\rangle \right\|_{\text{op}}.$$

These assumptions apply to all the results that follow below.

E.3 Bounding the error of the one-step Newton approximation

**Proposition**.[1] Let the Newton error be $\text{Err}_{Nt}(w) \overset{\text{def}}{=} \mathcal{I}^{*}_{f}(w) - \mathcal{I}^{Nt}_{f}(w)$. Then

$$|\text{Err}_{Nt}(w)| \leq \frac{n \|w\|_1^2 C_f C_H C^2_{f}}{\min \|\lambda\|^2 + \lambda^3}.$$

$\text{Err}_{Nt}(w)$ only involves third-order or higher derivatives of the loss, so it is $0$ for quadratic losses.

**Proof.** This proof is adapted to our setting from the standard analysis of the Newton method in convex optimization (Boyd and Vandenberghe [2004]).

First, note that $\text{Err}_{Nt}(w) = \mathcal{I}^{*}_{f}(w) - \mathcal{I}^{Nt}_{f}(w) = f(\hat{\theta}(1 - w)) - f(\hat{\theta}_{Nt}(1 - w))$. We will bound the norm of the difference of the parameters $\| \hat{\theta}(1 - w) - \hat{\theta}_{Nt}(1 - w) \|_2$; the desired bound on $f$ then follows from the assumption that $f$ has gradients bounded by $C_{f}$ and is therefore Lipschitz.

Since $L_{1-w}(\theta)$ is strongly convex (with parameter $\min \|\lambda\|^2 + \lambda$) and minimized by $\hat{\theta}(1 - w)$, we can bound the distance $\| \hat{\theta}(1 - w) - \hat{\theta}_{Nt}(1 - w) \|_2$ in terms of the norm of the gradient at $\hat{\theta}_{Nt}(1 - w)$:

$$\| \hat{\theta}(1 - w) - \hat{\theta}_{Nt}(1 - w) \|_2 \leq \frac{2}{\min \|\lambda\|^2 + \lambda} \| \nabla_{\theta} L_{1-w}(\hat{\theta}_{Nt}(1 - w)) \|_2.$$

Therefore, the problem reduces to bounding $\| \nabla_{\theta} L_{1-w}(\hat{\theta}_{Nt}(1 - w)) \|_2$.
We start by expressing the Newton step $\Delta \theta_{Nt}(w)$ in terms of the first and second derivatives of the empirical risk $L_{1-w}(\theta)$:

$$g_1(w) = \sum_{i=1}^{n} w_i \nabla \theta \ell(x_i, y_i; \hat{\theta}(1))$$

$$= -\sum_{i=1}^{n} (1 - w_i) \nabla \theta \ell(x_i, y_i; \hat{\theta}(1))$$

$$= -\nabla \theta L_{1-w}(\hat{\theta}(1)),$$

$$H_{\lambda,1}(1-w) = \sum_{i=1}^{n} (1 - w_i) \nabla_{\theta}^2 \ell(x_i, y_i; \hat{\theta}(1))$$

$$= \nabla_{\theta}^2 L_{1-w}(\hat{\theta}(1)),$$

$$\Delta \theta_{Nt}(w) = H_{\lambda,1}(1-w)^{-1} g_1(w)$$

$$= -\left[ \nabla_{\theta}^2 L_{1-w}(\hat{\theta}(1)) \right]^{-1} \nabla \theta L_{1-w}(\hat{\theta}(1)),$$

where the second equality for $g_1(w)$ comes from the fact that at the optimum $\hat{\theta}(1)$, the sum of the gradients $\sum_{i=1}^{n} \nabla \ell(x_i, y_i; \hat{\theta}(1))$ is 0.

With these expressions, we bound the norm of the gradient $\nabla \theta L_{1-w}(\hat{\theta}_{Nt}(1-w))$:

$$\left\| \nabla \theta L_{1-w}(\hat{\theta}_{Nt}(1-w)) \right\|_2$$

$$= \left\| \nabla \theta L_{1-w}(\hat{\theta}(1) + \Delta \theta_{Nt}(w)) \right\|_2$$

$$= \left\| \nabla \theta L_{1-w}(\hat{\theta}(1) + \Delta \theta_{Nt}(w)) - \nabla \theta L_{1-w}(\hat{\theta}(1)) - \nabla_{\theta}^2 L_{1-w}(\hat{\theta}(1)) \Delta \theta_{Nt}(w) \right\|_2$$

$$= \left\| \int_{0}^{1} \left( \nabla_{\theta}^2 L_{1-w}(\hat{\theta}(1) + t\Delta \theta_{Nt}(w)) - \nabla_{\theta}^2 L_{1-w}(\hat{\theta}(1)) \right) \Delta \theta_{Nt}(w) \, dt \right\|_2$$

$$\leq \frac{nC_H}{2} \left\| \Delta \theta_{Nt}(w) \right\|_2^2$$

$$= \frac{nC_H}{2} \left\| \left[ \nabla_{\theta}^2 L_{1-w}(\hat{\theta}(1)) \right]^{-1} \nabla \theta L_{1-w}(\hat{\theta}(1)) \right\|_2^2$$

$$\leq \frac{nC_H}{2(\sigma_{\min} + \lambda)^2} \left\| \nabla \theta L_{1-w}(\hat{\theta}(1)) \right\|_2^2$$

$$\leq \frac{n||w||^2 C_H C_f^2}{2(\sigma_{\min} + \lambda)^2}.$$

Putting together the successive bounds gives the result. 

\[ \square \]

### E.4 Characterizing the difference between the Newton approximation and influence

Before proving Theorem 1, we first prove a lemma about the spectrum of the error matrix $D(w)$.

**Lemma 1.** The matrix $D(w) \overset{\text{def}}{=} (I - H_{\lambda,1}^{-\frac{1}{2}} H_1(w) H_{\lambda,1}^{-\frac{1}{2}})^{-1} - I$ has singular values bounded between 0 and $\frac{\sigma_{\max}}{\sigma_{\min} + \lambda}$.

**Proof.** We first show that $H_{\lambda,1}^{-\frac{1}{2}} H_1(w) H_{\lambda,1}^{-\frac{1}{2}}$ has singular values bounded between 0 and $\frac{\sigma_{\max}}{\sigma_{\max} + \lambda}$. The lower bound of 0 comes from the fact that $H_{\lambda,1}^{-\frac{1}{2}} H_1(w) H_{\lambda,1}^{-\frac{1}{2}}$ is symmetric and $H_1(w) \succeq 0$.

To show the upper bound, first note that $H_1(w) \preceq H_1(w) + H_1(1-w) = H_1$ (recalling that $w \in \{0, 1\}^n$), and let $U \Sigma U^\top$ be the singular value decomposition of $H_1$. Since $H_{\lambda,1} = H_1 + \lambda I$,
we have
\[ H_{-\frac{1}{2},1}^{-\frac{1}{2}} H_1(w) H_{-\frac{1}{2},1}^{-\frac{1}{2}} = (H_1 + \lambda I)^{-\frac{1}{2}} H_1(w) (H_1 + \lambda I)^{-\frac{1}{2}} \]
\[ \leq (H_1 + \lambda I)^{-\frac{1}{2}} H_1(w) (H_1 + \lambda I)^{-\frac{1}{2}} \]
\[ = (U(\Sigma + \lambda I)^T)^{-\frac{1}{2}} U\Sigma U^T (U(\Sigma + \lambda I)^T)^{-\frac{1}{2}} \]
\[ = (U(\Sigma + \lambda I)^T)^{-\frac{1}{2}} \Sigma (U(\Sigma + \lambda I)^T)^{-\frac{1}{2}} U^T, \]
so its maximum singular value is upper bounded by \( \frac{\sigma_{\text{max}}}{\sigma_{\text{max}} + \lambda} \).

The bound on the singular values of \( H_{-\frac{1}{2},1}^{-\frac{1}{2}} H_1(w) H_{-\frac{1}{2},1}^{-\frac{1}{2}} \) implies that the singular values of \( I - H_{-\frac{1}{2},1}^{-\frac{1}{2}} H_1(w) H_{-\frac{1}{2},1}^{-\frac{1}{2}} \) lie in \( \left[ \frac{\lambda}{\sigma_{\text{max}} + \lambda}, 1 \right] \). In turn, this implies that the singular values of \( (I - H_{-\frac{1}{2},1}^{-\frac{1}{2}} H_1(w) H_{-\frac{1}{2},1}^{-\frac{1}{2}})^{-1} \) lie in \( \left[ 1, \frac{\sigma_{\text{max}} + \lambda}{\lambda} \right] \). Subtracting 1 from each end (for the identity matrix) gives the desired result.

\[ \square \]

**Theorem 1** The difference between the influence \( I_f(w) \) and the actual effect \( I_f^N(w) \) is
\[
I_f^N(w) - I_f(w) = \nabla_0 f(\hat{\theta}(1))^T H_{-\frac{1}{2},1}^{-\frac{1}{2}} D(w) H_{-\frac{1}{2},1}^{-\frac{1}{2}} g_1(w) + \frac{1}{2} \Delta \theta_{N_1}(w)^T \nabla_0^2 f(\hat{\theta}(1)) \Delta \theta_{N_1}(w) + \text{Err}_f(w) + \text{Err}_{N_1}(w)
\]
where we define the error matrix \( D(w) \) as \( (I - H_{-\frac{1}{2},1}^{-\frac{1}{2}} H_1(w) H_{-\frac{1}{2},1}^{-\frac{1}{2}})^{-1} - I \) with \( H_1(w) \) defined as \( \sum_{i=1}^n w_i \nabla_0^2 f(x_i, y_i; \hat{\theta}(1)) \). Furthermore, \( D(w) \) has eigenvalues between 0 and \( \frac{\sigma_{\text{max}} + \lambda}{\lambda} \), and \( \text{Err}_f(w) \) and \( \text{Err}_{N_1}(w) \) only involve third-order and higher derivatives of \( f \) and \( \ell \) respectively, with
\[
|\text{Err}_f(w)| \leq \frac{\|w\|^2 C_{f,3} C_{f,2}^2}{6(\sigma_{\text{min}} + \lambda)^3} \quad \text{and} \quad |\text{Err}_{N_1}(w)| \leq \frac{n\|w\|^2 C_{f,3} C_{f,2}^2}{(\sigma_{\text{min}} + \lambda)^3}.
\]

**Proof** We can decompose the error for the first-order influence approximation in terms of the error of the Newton approximation:
\[
I_f^N(w) - I_f(w) = I_f^N(w) - I_f(w) = (I_f^N(w) - I_f(w)) + (I_f^N(w) - I_f(w)).
\]

The error of the Newton approximation from the actual influence, \( \text{Err}_{N_1}(w) = I_f^N(w) - I_f^N(w) \), is given by Proposition 1, so we focus on the difference between the influence approximation \( I_f(w) \) and the Newton approximation \( I_f^N(w) \).

From the second-order Taylor expansion of \( f \) about \( \hat{\theta}(1) \), there exists \( 0 \leq \xi \leq 1 \) such that
\[
I_f^N(w) = f(\hat{\theta}(1) - w) - f(\hat{\theta}(1))
\]
\[
= f(\hat{\theta}(1) + \Delta \theta_{N_1}(w)) - f(\hat{\theta}(1))
\]
\[
= \nabla_0 f(\hat{\theta}(1))^T \Delta \theta_{N_1}(w) + \frac{1}{2} \Delta \theta_{N_1}(w)^T \nabla_0^2 f(\hat{\theta}(1)) \Delta \theta_{N_1}(w) + \frac{1}{6} \Delta \theta_{N_1}(w)^T \left\{ \nabla_0^2 f(\hat{\theta}(1) + \xi \Delta \theta_{N_1}(w), \Delta \theta_{N_1}(w)) \right\} \Delta \theta_{N_1}(w).
\]

Define \( \text{Err}_f(w) \) as the error due to third-order and higher derivatives of \( f \). We can bound \( \text{Err}_f(w) \) as follows:
\[
|\text{Err}_f(w)| \leq \frac{C_{f,3}}{6} ||\Delta \theta_{N_1}(w)||_2^2
\]
\[
\leq \frac{\|w\|^2 C_{f,3} C_{f,2}^2}{6(\sigma_{\text{min}} + \lambda)^3}.
\]
Furthermore, we can express the difference between the above first-order Taylor term and the first-order influence approximation $\mathcal{I}_f(w) = q_w'(0) = \nabla_\theta f(\hat{\theta}(1))^\top H_{\lambda,1}^{-1} g_1(w)$ as follows:

\[
\nabla_\theta f(\hat{\theta}(1))^\top \Delta \theta_{N\ell}(w) - \mathcal{I}_f(w) = \nabla_\theta f(\hat{\theta}(1))^\top H_{\lambda,1}^{-1} g_1(w) = \nabla_\theta f(\hat{\theta}(1))^\top \left(H_{\lambda,1}(1-w)^{-1} - H_{\lambda,1}^{-1}\right) g_1(w)
\]

Putting it all together, we have that

\[
\mathcal{I}_{N\ell}^w(w) - \mathcal{I}_f(w) = f(\hat{\theta}_{N\ell}(1-w)) - f(\hat{\theta}(1)) - \nabla_\theta f(\hat{\theta}(1))^\top H_{\lambda,1}^{-1} g_1(w) = \nabla_\theta f(\hat{\theta}(1))^\top \left(H_{\lambda,1}^{-\frac{1}{2}} \left(H_{\lambda,1}(1-w)^{-1} - H_{\lambda,1}^{-\frac{1}{2}}\right) g_1(w) + \frac{1}{2} \Delta \theta_{N\ell}(w)^\top \nabla_\theta^2 f(\hat{\theta}(1)) \Delta \theta_{N\ell}(w) + \text{Err}_f(w).
\]

Adding $\text{Err}_{N\ell}(w)$ from Proposition 1 and applying Lemma 1 to bound the spectrum of $D(w)$ completes the proof. \hfill \Box

### E.5 The influence on self-loss

We first state two linear algebra facts that will be useful in the sequel.

**Lemma 2.** Let $A \succ 0, B \succeq 0 \in \mathbb{R}^{d \times d}$ be a pair of symmetric positive-definite and positive-semidefinite matrices, respectively. Let $\sigma_{A,1}$ be the largest eigenvalue of $A$, $\sigma_{A,d}$ the smallest eigenvalue of $A$, and similarly let $\sigma_{B,1}$ and $\sigma_{B,d}$ be the largest and smallest eigenvalues of $B$, respectively. Then

\[
\frac{\sigma_{B,d}}{\sigma_{A,1}} I \preceq A^{-\frac{1}{2}} B A^{-\frac{1}{2}} \preceq \frac{\sigma_{B,1}}{\sigma_{A,d}} I.
\]

**Proof.** Note that $\frac{1}{\sigma_{A,1}}$ is the smallest eigenvalue of $A^{-1}$, while $\frac{1}{\sigma_{A,d}}$ is its largest. The lemma follows from the fact that the smallest singular value of the product of two matrices is lower bounded by the product of the smallest singular values of each matrix, and similarly the largest singular value of the product is upper bounded by the product of the largest singular values of each matrix. \hfill \Box

The next fact is a consequence of the variational definition of eigenvalues.

**Lemma 3.** Given a symmetric matrix $A \in \mathbb{R}^{d \times d}$ and a vector $v \in \mathbb{R}^d$, we have the following bounds on the quadratic form $v^\top A v$:

\[
\sigma_d \|v\|_2^2 \leq v^\top A v \leq \sigma_1 \|v\|_2^2,
\]

where $\sigma_d$ is the smallest eigenvalue of $A$, and $\sigma_1$ is the largest.

We are now ready to analyze the effect of removing a subset $w$ of $k$ training points on the total loss on those $k$ points.

**Proposition 2.** Suppose that $f(\theta) = \sum_{i=1}^n w_i \ell(x_i, y_i; \theta)$. Then

\[
\mathcal{I}_f(w) + \text{Err}_f(w) + \text{Err}_{N\ell}(w) \leq \mathcal{I}_f^w(w) \leq \left(1 + \frac{3\sigma_{\max}}{2\lambda} + \frac{\sigma_{\max}^2}{2\lambda^2}\right) \mathcal{I}_f(w) + \text{Err}_f(w) + \text{Err}_{N\ell}(w).
\]
Proof. Since \( f(\theta) = \sum_{i=1}^{n} w_i \ell(x_i, y_i; \theta) \), we have that
\[
\nabla_{\theta} f(\hat{\theta}(1)) = \sum_{i=1}^{n} w_i \nabla_{\theta} \ell(x_i, y_i; \hat{\theta}(1)) = g_1(w),
\]
\[
\nabla_{\theta}^2 f(\hat{\theta}(1)) = \sum_{i=1}^{n} w_i \nabla_{\theta}^2 \ell(x_i, y_i; \hat{\theta}(1)) = H_1(w).
\]
Substituting these and \( \Delta \theta \text{N}(w) = H_{\lambda,1}(1-w)^{-1}g_1(w) \) into Theorem 1, we obtain
\[
\mathcal{I}_f^*(w) - \mathcal{I}_f(w) - (\text{Err}_f(w) + \text{Err}_{\text{N}}(w))
\]
\[
= \nabla_{\theta} f(\hat{\theta}(1))^\top H_{\lambda,1}^{-\frac{1}{2}} D(w) H_{\lambda,1}^{-\frac{1}{2}} g_1(w) + \frac{1}{2} \Delta \theta \text{N}(w) + \nabla_{\theta}^2 f(\hat{\theta}(1)) \Delta \theta \text{N}(w)
\]
\[
= g_1(w)^\top H_{\lambda,1}^{-\frac{1}{2}} D(w) H_{\lambda,1}^{-\frac{1}{2}} g_1(w) + \frac{1}{2} g_1(w)^\top H_{\lambda,1}(1-w)^{-1} H_1(w) H_{\lambda,1}(1-w)^{-1} g_1(w)
\]
\[
= g_1(w)^\top H_{\lambda,1}^{-\frac{1}{2}} \left[ D(w) + \frac{1}{2} H_{\lambda,1}^{-\frac{1}{2}} H_{\lambda,1}(1-w)^{-1} H_1(w) H_{\lambda,1}(1-w)^{-1} H_{\lambda,1}^{-\frac{1}{2}} \right] H_{\lambda,1}^{-\frac{1}{2}} g_1(w).
\]
where \( D(w) \overset{\text{def}}{=} (I - H_{\lambda,1}^{-\frac{1}{2}} H_1(w) H_{\lambda,1}^{-\frac{1}{2}})^{-1} - I \) has singular values bounded between 0 and \( \sigma_{\max} \).

From Lemma 2, \( \Lambda(w) \) has singular values bounded between 0 and \( \frac{\sigma_{\max}}{2\lambda^2} \).

Applying Lemma 3 and using \( \mathcal{I}_f(w) = g_1(w)^\top H_{\lambda,1}^{-1} g_1(w) \), we obtain
\[
0 \leq g_1(w)^\top H_{\lambda,1}^{-\frac{1}{2}} \left[ D(w) + \Lambda(w) \right] H_{\lambda,1}^{-\frac{1}{2}} g_1(w)
\]
\[
\leq \left( \frac{\sigma_{\max}}{\lambda} + \frac{\sigma_{\max}(\sigma_{\max} + \lambda)}{2\lambda^2} \right) g_1(w)^\top H_{\lambda,1}^{-1} g_1(w)
\]
\[
= \left( \frac{3\sigma_{\max}}{2\lambda} + \frac{\sigma_{\max}^2}{2\lambda^2} \right) \mathcal{I}_f(w),
\]
which gives us
\[
\mathcal{I}_f(w) + \text{Err}_f(w) + \text{Err}_{\text{N}}(w) \leq \mathcal{I}_f^*(w) \leq \left( 1 + \frac{3\sigma_{\max}}{2\lambda} + \frac{\sigma_{\max}^2}{2\lambda^2} \right) \mathcal{I}_f(w) + \text{Err}_f(w) + \text{Err}_{\text{N}}(w).
\]
Note that \( \text{Err}_{f,2}(w) \overset{\text{def}}{=} \frac{\sigma_{\max}^2}{2\lambda^2} \mathcal{I}_f(w) \) can be bounded as
\[
|\text{Err}_{f,2}(w)| = \left| \frac{\sigma_{\max}^2}{2\lambda^2} \mathcal{I}_f(w) \right|
\]
\[
\leq \frac{\sigma_{\max}^2}{2\lambda^2} \left| \mathcal{I}_f(w) \right|
\]
\[
\leq \frac{\sigma_{\max}^2}{2\lambda^2} \left| g_1(w)^\top H_{\lambda,1}^{-1} g_1(w) \right|
\]
\[
\leq \frac{1}{\lambda^4} C_r \sigma_{\max}^2 \left( \frac{3\sigma_{\max}}{2\lambda} + \frac{\sigma_{\max}^2}{2\lambda^2} \right)
\]
so we can also write
\[
\mathcal{I}_f(w) + O\left( \frac{1}{\lambda^4} \right) \leq \mathcal{I}_f^*(w) \leq \left( 1 + \frac{3\sigma_{\max}}{2\lambda} \right) \mathcal{I}_f(w) + O\left( \frac{1}{\lambda^4} \right).
\]
E.6 The influence on a test point

**Corollary 1** Suppose \( f(\theta) = \theta^T x_{\text{test}} \), and define \( v_{\text{test}} \triangleq H_{\lambda,1}^{-\frac{1}{2}} x_{\text{test}} \) and \( v_w \triangleq H_{\lambda,1}^{-\frac{1}{2}} g_1(w) \). Then \( I_f^* (w) = I_f(w) + v_{\text{test}}^T D(w)v_w + \text{Err}_{Nt}(w) \), where \( D(w) = (I - H_{\lambda,1}^{-\frac{1}{2}} H_1(w) H_{\lambda,1}^{-\frac{1}{2}})^{-1} - I \) is the error matrix from Theorem 1.

**Proof.** Since \( f(\theta) = \theta^T x_{\text{test}} \) is linear, we have that for any \( \theta \in \Theta \),
\[
\nabla \theta f(\theta) = x_{\text{test}},
\]
\[
\nabla^2 \theta f(\theta) = 0,
\]
\[
C_f = 0.
\]

This in turn implies that \( \text{Err}_f(w) = 0 \). Substituting these expressions into Theorem 1 gives us the desired result. \( \square \)

**Proposition 3** Consider a binary classification setting with \( y \in \{-1, +1\} \) and a margin-based model with loss \( \ell(x, y; \theta) = \phi(y \theta^T x) \) for some \( \phi: \mathbb{R} \to \mathbb{R}^+ \). Suppose \( f(\theta) = \theta^T x_{\text{test}} \) and that the subset \( w \) comprises \( \|w\|_1 \) identical training points. Let \((x_k, y_k)\) be a representative training point from the subset. Then the actual effect \( I_f^*(w) \) is related to the influence \( I_f(w) \) according to
\[
I_f^*(w) = I_f(w) + x_{\text{test}}^T H_{\lambda,1}^{-\frac{1}{2}} D(w) H_{\lambda,1}^{-\frac{1}{2}} g_1(w) + \text{Err}_{Nt}(w).
\]

With the additional assumptions on \( w \) and \( \ell(x, y; \theta) \), we have that
\[
\nabla \theta \ell(x, y; \theta) = y \phi'(y \theta^T x)x,
\]
\[
g_1(w) = \sum_{i=1}^n w_i \nabla \theta \ell(x_i, y_i; \hat{\theta}(1)) = \sum_{i=1}^n w_i y_i \phi'(y_i \hat{\theta}(1)^T x_i) x_i = \sum_{i=1}^n w_i y_i \phi'(x_i), \]
\[
\text{def} = \sum_{i=1}^n w_i y_i \phi'(x_i); k = \|w\|_1 \phi_k(x_k),
\]
where in the last equality we use the assumption that we are removing \( \|w\|_1 \) copies of a single distinct point. Similarly,
\[
\nabla^2 \theta \ell(x, y; \theta) = \phi''(y \theta^T x)x x^T,
\]
\[
H_1(w) = \sum_{i=1}^n w_i \nabla^2 \theta \ell(x_i, y_i; \hat{\theta}(1)) = \sum_{i=1}^n w_i \phi''(y_i \hat{\theta}(1)^T x_i) x_i x_i^T = \|w\|_1 \phi_k'' x_k x_k^T.
\]

27
We thus have
\[
D(w) = (I - H_{\lambda,1}^{-\frac{1}{2}} H_1(w) H_{\lambda,1}^{-\frac{1}{2}})^{-1} - I
\]
\[
= \left( I - H_{\lambda,1}^{-\frac{1}{2}} ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}} \right)^{-1} - I
\]
\[
= H_{\lambda,1}^{-\frac{1}{2}} ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}}
\]
\[
= \frac{H_{\lambda,1}^{-\frac{1}{2}} ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}}}{1 - ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}}}
\]
where the third equality comes from the Sherman-Morrison formula. Substituting \(D(w)\) into Corollary \([\ref{corollary}]\) we obtain
\[
\mathcal{I}_f^*(w) = \mathcal{I}_f(w) + x_{test}^\top H_{\lambda,1}^{-\frac{1}{2}} D(w) H_{\lambda,1}^{-\frac{1}{2}} g_1(w) + \text{Err}_{\text{Nt}}(w)
\]
\[
= \mathcal{I}_f(w) + \frac{||w||_1 \phi_k^\ast x_{test}^\top H_{\lambda,1}^{-\frac{1}{2}} x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}} g_1(w)}{1 - ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}}} + \text{Err}_{\text{Nt}}(w)
\]
\[
= \mathcal{I}_f(w) + \frac{x_{test}^\top H_{\lambda,1}^{-\frac{1}{2}} ||w||_1 y_k \phi_k^\ast x_k \cdot ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}} x_k}{1 - ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}} x_k} + \text{Err}_{\text{Nt}}(w)
\]
\[
= \mathcal{I}_f(w) + \frac{\mathcal{I}_f(w) \cdot ||w||_1 z_k^\top H_{\lambda,1}^{-\frac{1}{2}} z_k}{1 - ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}} x_k} + \text{Err}_{\text{Nt}}(w)
\]
\[
= \frac{\mathcal{I}_f(w)}{1 - ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}}} + \text{Err}_{\text{Nt}}(w).
\]
To bound the denominator, we first use the trace trick to rearrange terms
\[
||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}} = \text{tr} \left( ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}} \right)
\]
\[
= \text{tr} \left( H_{\lambda,1}^{-\frac{1}{2}} ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}} \right)
\]
\[
= \text{tr} \left( H_{\lambda,1}^{-\frac{1}{2}} H_1(w) H_{\lambda,1}^{-\frac{1}{2}} \right).
\]
Since \(H_{\lambda,1}^{-\frac{1}{2}} H_1(w) H_{\lambda,1}^{-\frac{1}{2}}\) has rank one under our assumptions, it only has at most one non-zero
eigenvalue. We can therefore apply Lemma \([\ref{lemma}]\) to conclude that
\[
||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}} = \text{tr} \left( H_{\lambda,1}^{-\frac{1}{2}} H_1(w) H_{\lambda,1}^{-\frac{1}{2}} \right) \leq \frac{\sigma_{\text{max}}}{\sigma_{\text{max}} + \lambda},
\]
which in turn implies that \(1 - ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}} x_k \geq \frac{\lambda}{\sigma_{\text{max}} + \lambda},\) so
\[
\frac{1}{1 - ||w||_1 \phi_k^\ast x_k x_k^T H_{\lambda,1}^{-\frac{1}{2}}} \leq \frac{\sigma_{\text{max}} + \lambda}{\lambda} = 1 + \frac{\sigma_{\text{max}}}{\lambda}.
\]