

Maximum Consensus by Weighted Influences of Monotone Boolean Functions

Erchuan Zhang* David Suter*,‡ Ruwan Tennakoon** Tat-Jun Chin†
Alireza Bab-Hadiashar** Giang Truong* Syed Zulqarnain Gilani*,‡

*Centre for AI & ML, Edith Cowan University, Perth Australia.

‡ Nutrition and Health Innovation Research Institute, Edith Cowan University, Perth Australia.

**RMIT University, Melbourne Australia. †University of Adelaide, Adelaide Australia.

{erchuan.zhang, d.suter, h.truong, s.gilani}@ecu.edu.au, {ruwan.tennakoon, abh}@rmit.edu.au
tat-jun.chin@adelaide.edu.au

Abstract

Maximisation of Consensus (MaxCon) is one of the most widely used robust criteria in computer vision. Tennakoon et al. (CVPR2021), made a connection between MaxCon and estimation of influences of a Monotone Boolean function. In such, there are two distributions involved: the distribution defining the influence measure; and the distribution used for sampling to estimate the influence measure. This paper studies the concept of weighted influences for solving MaxCon. In particular, we study the Bernoulli measures. Theoretically, we prove the weighted influences, under this measure, of points belonging to larger structures are smaller than those of points belonging to smaller structures in general. We also consider another “natural” family of weighting strategies: sampling with uniform measure concentrated on a particular (Hamming) level of the cube. One can choose to have matching distributions: the same for defining the measure as for implementing the sampling. This has the advantage that the sampler is an unbiased estimator of the measure. Based on weighted sampling, we modify the algorithm of Tennakoon et al., and test on both synthetic and real datasets. We show some modest gains of Bernoulli sampling, and we illuminate some of the interactions between structure in data and weighted measures and weighted sampling.

1. Introduction

Robust model fitting is a fundamental problem in processing data in the presence of outliers, which is a long-standing and challenging research topic. *Maximisation of Consensus* (MaxCon) is one of the most popular criteria for robust fitting, which aims at finding a model that has the largest consensus on the given measurements. Formally, given a data set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ of size n and a prescribed

threshold $\varepsilon > 0$, the maximum consensus criterion for prescribed type of mathematical model is formulated as,

$$\begin{aligned} \max_{\boldsymbol{\theta} \in \Omega, \mathcal{I} \subseteq \mathcal{X}} & |\mathcal{I}|, \\ \text{s.t. } & r(\mathbf{x}_i, \boldsymbol{\theta}) \leq \varepsilon, \quad \forall \mathbf{x}_i \in \mathcal{I}, \end{aligned} \quad (1)$$

where Ω is the set of allowed parameters for prescribed mathematical model, $\boldsymbol{\theta}$ is the model parameter, $|\mathcal{I}|$ is the size of the subset $\mathcal{I} \subseteq \mathcal{X}$, $r(\mathbf{x}_i, \boldsymbol{\theta})$ is the fitting residual of datum \mathbf{x}_i with respect to the model $\boldsymbol{\theta}$.

Random sample consensus (RANSAC), “the original MaxCon algorithm”, is probably one of the most influential and widely used method since it was proposed in the 1980s [10]. The key idea is to adopt a hypothesize and verification procedure, which randomly samples subsets to fit the model and computes the consensus with respect to the obtained model. Such randomized methods can be very fast but there is no guarantee for the optimality of solutions. Variants of RANSAC, such as LO-RANSAC [6] and PROSAC [5], may improve the original RANSAC, but, they still inherit the drawback of randomized methods.

Related approaches are based on the M -estimator paradigm, or other ways to modify the residual measure. Examples include: l_1 method [20], l_∞ method [22, 23], iteratively reweighted least squares [1], reweighted l_1 method [21], exact penalty method [15, 16].

It is known that the MaxCon problem is combinatorially hard [3]. Optimal methods, (branch-and-bound (BnB) search [17] and A^* tree search [2, 4]) exist but their (worst case) runtime increases exponentially with the size of the problem, which makes them ineffective for large scale and high dimensional problems.

This work directly follows [24] - here examining alternative definitions of influences and associated estimation methods. That work, in turn, uses the same basic “tree” strategy of Chin et al. [4], which used A^* search on a tree structure, founded on the notion of “basis” in LP-type the-

ory. The A^* search is able to find global optimal solutions. However, this method is slow in general, despite speedups introduced by Cai *et al.* [2]. In [25], an unsupervised learning approach was proposed to determine which point in a basis to remove in solving robust model fitting problems. This approach adopted the framework of reinforcement learning, where removing points are guided by maximising *rewards*. Like many learning based approaches, it may take a long time to train and it is hard to analyse the method and its ability to generalise. Like [25] and [24], modifications to the basic tree search that lose the priority queue guarantees of A^* , sacrifice optimality guarantees for speed. This paper is also spiritually in that context.

1.1. Influence and consensus maximisation

Recently, Tennakoon *et al.* [24] characterized the maximum consensus problem by *Monotone Boolean Function Theory* and investigated the characterisation of the MaxCon solutions by *influences* of a monotone Boolean function. In detail, any subset $\mathcal{I} \subseteq \mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ can be represented by a bit vector \mathbf{b} of length n , where its i -th component $b_i = 0$ denotes the exclusion of the datum \mathbf{x}_i and $b_i = 1$ denotes the inclusion of the datum \mathbf{x}_i . Then, any subset \mathcal{I} is nothing more than a vertex of an n -dimensional Boolean cube. A Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$:

$$f(\mathbf{b}) := \begin{cases} 1, & \text{if } r(\boldsymbol{\theta}, \mathbf{x}_i) > \varepsilon, \forall i \in \{j : b_j = 1\}. \\ 0, & \text{if } r(\boldsymbol{\theta}, \mathbf{x}_i) \leq \varepsilon \end{cases} \quad (2)$$

defines feasibility (the subset $\{\mathbf{x}_i\}$ denoted by \mathbf{b} is called *feasible* if $f(\mathbf{b}) = 0$ and *infeasible* otherwise). Solving MaxCon is a search for the feasible subset of maximal size.

A Boolean function is called *monotone* if $f(\mathbf{a}) \leq f(\mathbf{b})$ holds for any $\mathbf{a} \prec \mathbf{b}$, where the ordering relation \prec means i -th component $a_i \leq b_i$ for any $1 \leq i \leq n$. We can easily see that the Boolean function (2) is monotone. The reason is that adding data points to infeasible subsets still keeps them *infeasible* while deleting data points from feasible subsets keeps them *feasible*. An *upper zero* \mathbf{b} of a monotone Boolean function f is a vertex such that $f(\mathbf{b}) = 0$ and $f(\mathbf{a}) = 1$ for any \mathbf{a} satisfying $\mathbf{b} \prec \mathbf{a}$. A *maximum upper zero* \mathbf{b} of f is an upper zero with the maximal size, *i.e.*, $f(\mathbf{b}) = 0$ and $f(\mathbf{a}) = 1$ for any \mathbf{a} satisfying $\|\mathbf{a}\|_1 > \|\mathbf{b}\|_1$, where $\|\cdot\|_1$ is the l_1 norm. The upper zeros essentially characterise all possible candidate consensus solutions systematically and finding maximum consensus is equivalent to finding the maximum upper zero of associated Boolean function. See Figure 1 for a toy example of 2D line fitting, which illustrates the connection between monotone Boolean functions and the MaxCon problem.

The key concept in [24] is *influence*. When endowing the Boolean cube with a uniform measure, the *influence* of

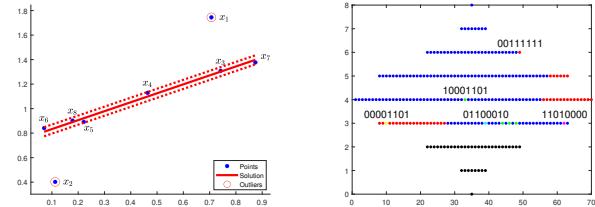


Figure 1. Left: 2D line fitting problem with 8 points. Right: Boolean cube associated with the fitting problem. The 8D Boolean cube is flattened on the plane, where every blue dot represents an infeasible subset of the line fitting problem and other colored dots represent feasible subsets. These dots are arranged by the ordering relation \prec from top to bottom. In this toy example, there are four upper zeros, *i.e.*, $\mathbf{b}^6 = 00111111$ (maximum upper zero, superscript represents how many data points are chosen), $\mathbf{b}^4 = 10001101$, $\mathbf{b}^3 = 01100010$, $\mathbf{b}^{3'} = 11010000$, which are highlighted in the cube and the sub-cubes determined by these vertices are distinguished by different colors. The vertex $\mathbf{b}^{3''} = 00001101$ is a pseudo upper zero (see definition in Section 3). Note that all vertices below level 3, *i.e.*, less than 3 points are fitted, must be feasible.

b_i of a Boolean function f is defined [19] as,

$$\text{Inf}_i[f] := \mathbb{P}_{\mathbf{b} \sim \{0, 1\}^n} [f(\mathbf{b}) \neq f(\mathbf{b}^{\oplus i})], \quad (3)$$

where $\mathbf{b}^{\oplus i}$ denotes flipping the i -th bit in \mathbf{b} , $\mathbf{b} \sim \{0, 1\}^n$ means \mathbf{b} is uniformly distributed on the Boolean cube.

To estimate the influences in equation (3), the most obvious and natural way would be to uniformly sample, and use the empirical counts as (unbiased) estimates. The whole Boolean Cube is, unfortunately, so vast, that spreading the samples uniformly is not an efficient sampling strategy. This undoubtedly led to the “uniform at level p' ” (for p' slightly larger than the combinatorial dimension p) sampling strategy of [24], which will produce *biased estimates*.

1.2. Motivation of this paper

Given the above, one can thus view the method of [24] as employing *biased* estimates of *uniform measure* influence (equation (3)). Alternatively, one could take the sampling distribution used in [24] as implicitly defining a *new (weighted) influence measure* (and the sampling strategy is now an unbiased estimator of this new influence measure). Both views are really two sides of the same coin. However, the primary issue is whether the biased estimates of uniform measure influence (first viewpoint: biased estimator - second viewpoint: new influence measure) actually preserve the ordering of the sizes of influences.

Put simply, in [24] there is a disconnect between the measure used in the definition (and also in the proof of influence being greater for outliers) and the measure used in the empirical estimation: but one can reconcile this by using a definition of influences that entails the same biased measure

as would be used in sampling (to estimate the influences). We are thus led to a more systematic and coherent study of a topic called “weighted” (or biased) influences.

In section 2.1, we equip the Boolean cube $\{0, 1\}^n$ with the Bernoulli(q) measure μ_q , namely, the probability of $b_i = 1$ is q and the probability of $b_i = 0$ is $1 - q$, which includes the uniform measure as a special case ($q = \frac{1}{2}$). For this, an unbiased estimator is:

$$\text{Inf}_i^{(q)}[f] = \mathbb{E}_{\mathbf{x} \sim \mu_q(\mathbf{x})} [\mathbb{I}(f(\mathbf{x}) \neq f(\mathbf{x}^{\oplus i}))], \quad (4)$$

where $\mathbb{I}(\cdot)$ is the indicator function, $\mathbb{E}[\cdot]$ is the sample mean of random variables. Empirical evidence (see section 4) shows that a Bernoulli weighted influence based method can achieve similar optimal solutions with less runtime or even better results with similar runtime. But perhaps more importantly, we analytically prove that this weighted measure preserves the necessary ordering of influences - see section 3. In fact, since the expressions we derive for Bernoulli weighted influence subsume uniform influence as a special case, and since we cover more general data characteristics (section 3.2) than that in the corresponding sections of [24], this paper provides a more comprehensive theoretical analysis of the uniform measure as well.

Another family of measures, we call Hamming(k), puts equal mass on subsets of a fixed size (fixed Hamming norm) k , and zero elsewhere. A particular case of which is applied in [24]. For these measures, it turns out that we do not need to conduct an analysis from scratch. The expressions we have for Bernoulli(q) weighted influence naturally stratify by level (as do of course the corresponding equations in [24] which are the special case Bernoulli(0.5)). This allows us to obtain the number of feasibility/infeasibility flips per level by inspection (and the corresponding weighted influence is then just the appropriate normalisation produced by dividing these counts by the number of vertices/subsets in the given level). Nonetheless, this was not pointed out in [24], nor were any resulting observations made. Moreover, since we analyse a more general set of data assumptions we also have a more comprehensive picture than can be deduced simply by inspection of equations in [24].

Our main contributions can be summarised as:

- We characterize the influence of outliers/inliers in MaxCon by the concept of weighted influence. We prove that the weighted influences of points belonging to the larger structure in the data are generally smaller for both ideal and non-ideal cases (see section 3 for definitions). Note that our analysis is much more general, than that in [24].
- We empirically test modifications made to the basic algorithm of [24], to accommodate weighted sampling by Bernoulli measure. On several robust fitting tasks,

our weighted influence based variant is an effective alternative to the version in [24].

We also provide a conjecture, motivated by the structure of the equations we derive, that possibly links and explains the inherent structural complexity of MaxCon, by some recent constructs of what characterises hard problems - see Conjecture 3.1 section 3.2.

The aim of the paper is to investigate MaxCon under new definitions of influence, and corresponding estimation strategies. The spirit is the study of those issues - not of producing a “leaderboard winning” algorithm.

1.3. Notation

We adopt the following notation: (1) Bold \mathbf{b} denotes a vertex (a vector of n bits) in the Boolean cube $\{0, 1\}^n$ and its i -th component is denoted by b_i . (2) $[n] = \{1, 2, \dots, n\}$. (3) $L_k := \{\mathbf{b} \in \{0, 1\}^n \mid \|\mathbf{b}\|_1 = k\}$ denotes level k of the Boolean cube $\{0, 1\}^n$ for $0 \leq k \leq n$. (4) $L_{\leq k} := \{\mathbf{b} \in \{0, 1\}^n \mid \|\mathbf{b}\|_1 \leq k\}$ represents levels below level $k + 1$ in the cube $\{0, 1\}^n$. Similar definitions for $L_{<k}$, $L_{\geq k}$, $L_{>k}$. (5) For $\mathbf{b}^* \in L_k$, $B_{\mathbf{b}^*} := \{\mathbf{b} \in \{0, 1\}^n \mid \Delta(\mathbf{b}, \mathbf{b}^*) = l, \mathbf{b} \in L_{k-l}, 1 \leq l \leq k - p - 1\}$ denotes the sub-cube determined by \mathbf{b}^* , where $\Delta(\mathbf{b}, \mathbf{b}^*) = |\{i : b_i \neq b_i^*\}|$ is the Hamming distance between \mathbf{b} and \mathbf{b}^* . (6) $S_{\mathbf{b}^{k_i}}^j := \{l \in [n] \mid b_l^{k_i} = j\}$, where $j = 0, 1$. $S_{\mathbf{b}^{k_i}}^1$ and $S_{\mathbf{b}^{k_i}}^0$ represent the set of indices of data points who are inliers and outliers with respect to $\mathbf{b}^{k_i} \in L_{k_i}$, respectively. (7) C_k^n denotes the number of k -combinations of a set with n elements, which is $n!/k!/(n-k)!$, where $1 \leq k \leq n$.

2. Weighted Influences

In this section, we will introduce the definitions of (Bernoulli) weighted influences of monotone Boolean functions (section 2.1) and (Uniform Hamming) weighted influences (section 2.2).

2.1. Bernoulli(q) weighted influences

Let $\Omega_n = \{0, 1\}^n$ be the discrete n -dimensional cube endowed with the Bernoulli measure μ_q , where $\mu(\{\mathbf{b} : b_i = 1\}) = q \in (0, 1)$, namely, $\mu_q(\mathbf{b}) = q^{|\mathbf{b}|_1}(1 - q)^{n - |\mathbf{b}|_1}$. Then, we can define a metric on Ω_n [19],

$$\langle f_1, f_2 \rangle := \sum_{\mathbf{b} \sim \Omega_n} f_1(\mathbf{b}) f_2(\mathbf{b}) \mu_q(\mathbf{b})$$

for any f_1, f_2 defined on Ω_n . The parity functions

$$\chi_S^q(\mathbf{b}) := q^{\sum_{i \in S} b_i} \cdot q_+^{|S| - \sum_{i \in S} b_i}$$

form a basis of the space $(\Omega_n, \langle \cdot, \cdot \rangle)$, where S is a subset of $[n]$, $\mathbf{b} \in \Omega_n$, $q_- = -\sqrt{\frac{1-q}{q}}$, $q_+ = \sqrt{\frac{q}{1-q}}$. $\chi_\emptyset^q(\mathbf{b}) \equiv 1$ by convention.

The *weighted first-order Fourier coefficient* of a Boolean function f on i -th variable can be given by

$$\hat{f}^q(\{i\}) = \langle f, \chi_{\{i\}}^q \rangle = \sum_{\mathbf{b} \sim \Omega_n} f(\mathbf{b}) q_-^{b_i} q_+^{1-b_i} \mu_q(\mathbf{b}). \quad (5)$$

With respect to the Bernoulli measure μ_q , the *weighted influence* of the i -th variable on a Boolean function f defined on Ω_n is defined as (see Page 9 in [14])

$$\text{Inf}_i^q[f] := \mu_q(\{\mathbf{b} : f(\mathbf{b}) \neq f(\mathbf{b}^{\oplus i})\}), \quad (6)$$

where $\hat{f}^q(\{i\})$ and $\text{Inf}_i^q[f]$ are related by¹:

Theorem 2.1. If $f : \{0,1\}^n \rightarrow \{0,1\}$ is a monotone Boolean function, then $\text{Inf}_i^q[f] = -\frac{1}{\sqrt{q(1-q)}} \hat{f}^q(\{i\})$.

In practice, it is not very realistic to calculate the exact value of (5) or (6) since the summation contains 2^n possible \mathbf{b} to evaluate. Since we will take advantage of the order information of all weighted influences rather than their exact values to design algorithms for MaxCon, it is sufficient to get approximated weighted influences of good quality.

The way we estimate weighted influences is illustrated as follows: Given a sample size $h > p$ and a weight $q \in (0, 1)$, we sample half of a set of vertices $\{\mathbf{b}_j\}_{j=1}^{\lfloor \frac{h}{2} \rfloor}$ on the Boolean cube Ω_n according to the Bernoulli measure μ_q and then flip i -th bit in all \mathbf{b}_j to generate another half samples $\{\mathbf{b}_j\}_{j=\lfloor \frac{h}{2} \rfloor + 1}^h$. The monotone Boolean function f is evaluated on these vertices. Therefore, by (5), (6) and Theorem 2.1, we estimate the weight influence $\widetilde{\text{Inf}}_i^q[f]$ as

$$\widetilde{\text{Inf}}_i^q[f] = -\frac{1}{h\sqrt{q(1-q)}} \sum_{j=1}^h f(\mathbf{b}_j) q_-^{b_{j,i}} q_+^{1-b_{j,i}} \mu_q(\mathbf{b}_j), \quad (7)$$

where $b_{j,i}$ is the i -th component of \mathbf{b}_j . Note: evaluation can be simplified by the monotonicity of f : if $f(\mathbf{b}_j) = 0$ and $b_{j,i} = 1$, then $f(\mathbf{b}_j^{\oplus i}) = 0$; if $f(\mathbf{b}_j) = 1$ and $b_{j,i} = 0$, then $f(\mathbf{b}_j^{\oplus i}) = 1$. Figure 2 shows the estimated and exact weighted influences of data points illustrated in Figure 1, where influences are normalized by maximal influence.

One might ask - can we recommend the best q to use? Unfortunately, though we can illustrate that there is an optimal q for a toy problem (see for example Fig. 3), the best q is probably problem dependent. Moreover, sampling the measure is another complication.

2.2. Hamming(k) weighted influences

Defining a Fourier transform (and in general analysis) restricted to the Boolean slice (level of equal Hamming norm) is actually a more complex topic, only recently researched.

¹Refer to the supplementary material for the proof. A similar result can be found in [14] (Page 20) but without proof.

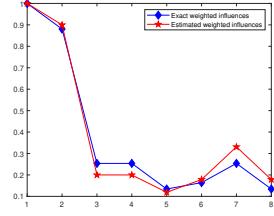


Figure 2. Comparison of normalized exact and estimated weighted influences of all data points used in Figure 1, where q is set to 0.375 and we set the number of samples $h = 100$.

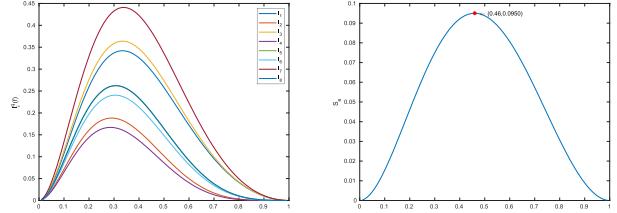


Figure 3. Example (3 outliers) of various q for separating MaxCon inliers and outliers. Left: Influences of all data points - top 3 outliers. Right: The separation function (distance between smallest outlier and largest inlier influence): the best choice for q is 0.46 in terms of separation, and for this data set. Note: sampling to estimate this measure is a different issue altogether.

See for example, [8, 9, 27] - arguments that might appeal to Fourier theory for the uniform cube, often side-step the more complicated Fourier picture on the slice. Likewise, we will make no reference to Fourier coefficients.

In essence, one can go “direct” from the definition of expectations of the counts of transitions (from feasibility to infeasibility) by flipping the i -th bit (including or excluding the i -th data point) to the appropriate definition of influence for the slice-concentrated measures (and thereby for sampling/estimation strategies). Moreover, since these transitions are counted by level (slice) in the appropriate formulae for the Bernoulli(q) influences (given in section 3 or in [24] for the Bernoulli(0.5) measure): one does not have to repeat complicated derivations/analysis but can merely “pick off” the relevant counts for the level under consideration, and then normalize by the total count measure on a slice C_k^N for slice k . Moreover, since one is only interested in relative sizes of influences, normalization can be omitted.

3. Analysis of weighted influence

In this section, we theoretically show that points belonging to larger structures have smaller weighted influences: intuitively, the weighted influence of an “inlier” data point is smaller than that of an “outlier” data point. Proofs of all theoretical results in the following two subsections can be found in the supplementary material. Informed by our analysis, one can search for the MaxCon solution, using weighted sampling for influence: with some assurance the

estimates have the right relative order to identify outliers.

3.1. Bernoulli(q) - Ideal case

Real datasets are “non-ideal”, *i.e.*, there are multiple structures in the dataset and these often share many data points; yet the ideal case is a useful starting point.

Definition 3.1. Suppose $\{\mathbf{b}^{k_i}\}_{i=1}^{n_0}$ are upper zeros of a monotone Boolean function, where $\mathbf{b}^{k_i} \in L_{k_i}$, $p+1 \leq k_1 \leq k_2 \leq \dots \leq k_{n_0} \leq n$. Then

f is called ideal if $\forall i, j \in [n_0]$, $|S_{\mathbf{b}^{k_i}}^1 \cap S_{\mathbf{b}^{k_j}}^1| \leq p$, namely, the structures have very small overlap.

f is called non-ideal if $\exists i, j \in [n_0]$, $|S_{\mathbf{b}^{k_i}}^1 \cap S_{\mathbf{b}^{k_j}}^1| > p$. Namely, at least two structures have significant overlap. We call $\mathbf{b}^\alpha = (b_1^\alpha, \dots, b_n^\alpha)$ defined by

$$b_l^\alpha := \begin{cases} 1, & l \in S_{\mathbf{b}^{k_i}}^1 \cap S_{\mathbf{b}^{k_j}}^1, \forall l \in [n], \\ 0, & \text{otherwise,} \end{cases}$$

a pseudo upper zero with respect to \mathbf{b}^{k_i} and \mathbf{b}^{k_j} , where $\alpha := |S_{\mathbf{b}^{k_i}}^1 \cap S_{\mathbf{b}^{k_j}}^1|$ is the level that the pseudo upper zero belongs to. Let $\{\mathbf{b}^{\alpha_i}\}_{i=1}^{m_0}$ ($p+1 \leq \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_{m_0} < k_{n_0}$) be the set of all pseudo upper zeros if f is non-ideal.

Theorem 3.1. (Ideal Single Structure) If $n_0 = 1$, namely, f is ideal with respect to a single maximum upper zero \mathbf{b}^{k_1} , then, for $i \in S_{\mathbf{b}^{k_1}}^1$ (inliers),

$$\text{Inf}_i^q[f] = (C_p^{n-1} - C_p^{k_1-1})q^p(1-q)^{n-p-1}, \quad (8)$$

and for $i \in S_{\mathbf{b}^{k_1}}^0$ (outliers),

$$\text{Inf}_i^q[f] = C_p^{n-1}q^p(1-q)^{n-p-1} + \sum_{l=p+1}^{k_1} C_l^{k_1}q^l(1-q)^{n-l-1}, \quad (9)$$

which implies $\text{Inf}_{i_1}^q[f] - \text{Inf}_{i_2}^q[f] = C_p^{k_1-1}q^p(1-q)^{n-p-1} + \sum_{l=p+1}^{k_1} C_l^{k_1}q^l(1-q)^{n-l-1} > 0$, where $i_1 \in S_{\mathbf{b}^{k_1}}^0$, $i_2 \in S_{\mathbf{b}^{k_1}}^1$.

This theorem indicates that if there is only one ideal structure, namely, points are either inliers or outliers with respect to that structure, the weighted influences of outliers (all outliers share the same influences) are strictly larger than those of inliers (all inliers share the same influences).

Theorem 3.2. (Ideal Multiple Structure) If $n_0 > 1$, *i.e.*, f is ideal with several upper zeros, then, $\forall i \in \bigcap_{l=1}^{n_0} S_{\mathbf{b}^{k_l}}^i$ (if non-empty), $\text{Inf}_i^q[f]$ is

$$\begin{aligned} & (C_p^{n-1} - \sum_{i_s=1}^{k_s} C_p^{k_s-1})q^p(1-q)^{n-p-1} \\ & + \sum_{i_s=0}^{k_s} \sum_{l=p+1}^{k_s} C_l^{k_s}q^l(1-q)^{n-l-1}. \end{aligned} \quad (10)$$

Put simply, we can see (from the summations over i_s) that the influence of a data point decreases when belonging to a structure (and the decrease is more the larger the structure is) and increases for every structure the point does not belong to (again, by more if that structure is large). These increases and decreases are weighted differently (terms involving q), complicating the relationship.

Note: with multiple structures, one has to be careful to qualify “inlier” and “outlier”. These only have meaning with respect to a nominated single structure. For that structure, all points “inlier” to other structures are actually outlier to the nominated one.

From (10), the influence $\text{Inf}_i^q[f]$ of data point i is larger if it is an outlier with respect to more upper zeros (and data outlier to all structures will have the largest influence of all).

Here, we introduce a new Boolean cube $\{0, 1\}^{n_0}$, for any $i \in \bigcap_{l=1}^{n_0} S_{\mathbf{b}^{k_l}}^i$, it corresponds to a vertex $(i_1, i_2, \dots, i_{n_0}) \in \{0, 1\}^{n_0}$. Then, $\text{Inf}_i^q[f]$ is a real-valued Boolean function on $\{0, 1\}^{n_0}$. To shorten notation, we denote $\text{Inf}_i^q[f]$ for $i \in \bigcap_{l=1}^{n_0} S_{\mathbf{b}^{k_l}}^i$ by $\tilde{f}^q(S^i)$.

Corollary 3.2.1. The influences (10) have the following order relationship

$$\forall \mathbf{i}, \mathbf{j} \in \{0, 1\}^{n_0}, \mathbf{i} \succ \mathbf{j} \implies \tilde{f}^q(S^i) < \tilde{f}^q(S^j). \quad (11)$$

Note that $\tilde{f}^q(S^\bullet)$ exists only if S^\bullet is non-empty.

3.2. Bernoulli(q) - Non-ideal case

Given a pseudo upper zero \mathbf{b}^{α_i} , we introduce the notation $S_{\mathbf{b}^{\alpha_i}}^j = \{l \in [n] | b_l^{\alpha_i} = 1-j\}$, where $j = 0, 1$. For any $i \in (\bigcap_{l=1}^{n_0} S_{\mathbf{b}^{k_l}}^i) \cap (\bigcap_{l=1}^{m_0} S_{\mathbf{b}^{\alpha_{n_0+l}}}^i)$ (if non-empty), denote the influence $\text{Inf}_i^q[f]$ of i -th variable on f by $\tilde{f}^q(S^i)$, where $\mathbf{i} = (i_1, i_2, \dots, i_{n_0+m_0})$.

Observe that if $\mathbf{b}^{\alpha_{i_0}}$ is a pseudo upper zero with respect to upper zeros $\mathbf{b}^{k_{i_1}}$ and $\mathbf{b}^{k_{i_2}}$, a data point i is an inlier with respect to $\mathbf{b}^{k_{i_1}}$ and $\mathbf{b}^{k_{i_2}}$, then it must be an inlier with respect to $\mathbf{b}^{\alpha_{i_0}}$; if i is an outlier with respect to any upper zero, then it must be an outlier with respect to $\mathbf{b}^{\alpha_{i_0}}$.

Theorem 3.3. If f is non-ideal, then the weighted influence $\tilde{f}^q(S^i)$ is given by

$$\begin{aligned} & (C_p^{n-1} - \sum_{\substack{i_s=1 \\ 1 \leq s \leq n_0}}^{k_s} C_p^{k_s-1} + \sum_{\substack{i_{n_0+s}=0 \\ 1 \leq s \leq m_0}}^{k_s} C_p^{\alpha_s-1})q^p(1-q)^{n-p-1} \\ & + \left(\sum_{i_s=0}^{k_s} \sum_{l=p+1}^{k_s} C_l^{k_s} - \sum_{i_{n_0+s}=1}^{\alpha_s} \sum_{l=p+1}^{k_s} C_l^{\alpha_s} \right) q^l(1-q)^{n-l-1}. \end{aligned}$$

where $\tilde{f}^q(S^\bullet)$ doesn't exist if $S^\bullet = \emptyset$.

The core strategy to prove this theorem is by induction on the number of pseudo upper zeros and taking advantage of Theorem 3.2. Similar to the ideal case, by Theorem 3.3,

we can see that the weighted influences of points belonging to larger structures are smaller. What becomes apparent is that with many structures, the expressions become hugely complicated by the combinatorics of possible intersections/overlap between them - complicating the measure of inlier/outlier dichotomy (with respect to any given structure). This leads us to conjecture:

Conjecture 3.1. (Complexity of MaxCon and Overlap Gap Property (OGP)) *In [11] (and the works referred therein), the computational complexity of many algorithms has been linked to what is called the OGP. Essentially it is argued that if the solution space is highly clustered (and here locality is measured by overlap of the solutions) then the problem will be hard for whole classes of algorithms. In short, the existence of potential solutions that are either very similar/have large overlap, or are widely separated (few intermediate separated) - is a “signature” of a hard problem to solve. We conjecture that this is true of the MaxCon landscape of possible data configurations, and the above reflects (through the lens of influence) how this complicated data instance scenario manifests in a property related to the sought solution - influence.*

This conjecture is particularly intriguing because of the link between maximum independent set (studied in [11]) and MaxCon - it can be shown that the MaxCon solution is the maximum independent set of the (hyper)graph formed by the infeasible minimal sized subsets (and also to the minimum vertex cover of the complement hypergraph).

3.3. Uniform Hamming measure Influence

Consider the ideal single structure case define in Theorem 3.1. Firstly, it is easy to see the for the ideal single structure, any algorithm that starts with a feasible set of size large than the combinatorial dimension, and then greedily adds points if the larger set remains feasible, will obtain the MaxCon solution. So it is an easy problem with obvious solution strategies. But if we did decide to use influences we can note that from equation (8) the inlier influences came from counting feasible/infeasible transitions between levels p and $p + 1$ only. Thus for sampling uniform Hamming level above level $p + 1$, there are no feasibility/infeasibility transitions caused by inliers. In other words, the Uniform Hamming influence measure, at that level or above, will be exactly zero for inliers. Since the influences of outliers, for the same measure will be non-zero, this seems to promise a remarkably efficient sampling strategy - one could eliminate outliers at the first sample that revealed a count for the associated influence - without the need to continue with the full estimation process. Of course, practically, this is too good to hold for real data - it is very brittle to our strict assumptions here. Nonetheless, it does hint at the usefulness of a less brittle strategy of early termination of the counting

process once a count reaches some degree of statistically significantly higher than the rest, rather than a set number of samples always being used: a future research topic.

Now consider the ideal multiple structure setting. From equation 10, the influence accrued by being inlier to some structure (first term) is only accrued between level p and $p + 1$. The subtraction is due to feasibility transitions “that didn’t occur” because the subset with added inlier remains within the same structure). So once again, sampling above level p will not “see” those counts. But since inliers to one structure are outliers to another (we assumed no significant overlap): hence the influence of inliers to any structure will not be zero - different to the single structure case, as all structures are outliers to to (all) other structures and thus accrue influence from the second term in the equation. It is also easy to see that so long as the level is “not too high” (above the largest structure) the Hamming sampled influences will be an appropriate guide (influence of inliers of a larger structure will be smaller). (In that second term the largest structure is excluded from the sum over $i_s = 0$, when calculating influence for that structure.)

Further discussion is in the supplementary material.

4. Experiments with Bernoulli measure

In this section, we empirically demonstrate that Bernoulli weighted influence behaves generally as our analysis predicts (essentially retaining the correct ordering of influences for inlier/outlier separation) across several robust model fitting tasks on both synthetic and real datasets. The main algorithm we use is presented in the supplementary material (Algorithm 1 - which follows [24]). To reduce the risk of losing genuine inliers, we also implement the local expansion (see Algorithm 2 in the supplementary material) at the last step in Algorithm 1, which likewise follows the strategy of [24]. Note: the algorithm of [24] is using uniform sampling at fixed Hamming level (so is implicitly providing a Hamming measure based result), though there it was interpreted as a biased way to estimate uniform measure influence, rather than, as here, a new measure of influence. [24] empirically optimised their sampling level - hence we do not experiment with Hamming sampling here.

For comparison we use: RANSAC [10], Lo-RANSAC [6], A*-NAPA-DIBP [2] (We only use it to generate ground truth solutions for synthetic data with low rate of outliers.), MBF [24], L1 [20]. Our method using Weighted Influence is short for WI. Experiments employed Matlab R2020b on a computer with Intel(R) i7-8700K CPU and 32GB RAM.

4.1. Robust linear regression

Analysis of parameters in weighted influences estimation: Evaluation of the weighted influence (7) requires two parameters: sample probability q and sample size h . Here, we study the effect of q and h using synthetic data

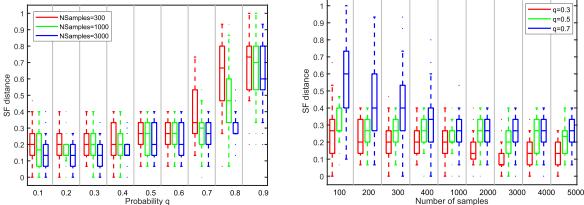


Figure 4. Comparison results for 2-dimensional robust linear regression: Variation of the SF distance with respect to different (Left) sample probability q and (Right) number of samples h . All experiments were run 50 times.

on the 2-dimensional line fitting problem. We generate $n = 15$ points² around a straight line and randomly select 30% points as outliers, which are perturbed with uniformly distributed noise in range $[-4, -0.1] \cap (0.1, 4]$. The remaining points are perturbed with uniformly distributed noise in $[-0.1, 0.1]$. We set the inliers threshold ε as 0.1 in all experiments throughout this subsection.

Let \mathbf{r}_{es} and \mathbf{r}_{ex} be the top k (in decreasing order) of estimated weighted influences (calculated by (7)) and exact weighted influences (calculated by (6)) of all points, respectively. The value k is determined by the number of outliers (found by A^* -NAPA-DIBP). We measure the difference between \mathbf{r}_{es} and \mathbf{r}_{ex} by the normalized Spearman Footrule (SF) distance [7]. Denote a set of elements contained in³ \mathbf{r}_{\bullet} by $\mathcal{R}(\mathbf{r}_{\bullet})$ and the position of the element $z \in \mathcal{R}(\mathbf{r}_{\bullet})$ in \mathbf{r}_{\bullet} by $z^{\mathbf{r}_{\bullet}}$. The normalized SF distance between \mathbf{r}_{es} and \mathbf{r}_{ex} is given by

$$F(\mathbf{r}_{es}, \mathbf{r}_{ex}) = \frac{1}{k(k+1)} \sum_{z \in \mathcal{R}(\mathbf{r}_{es}) \cap \mathcal{R}(\mathbf{r}_{ex})} |z^{\mathbf{r}_{es}} - z^{\mathbf{r}_{ex}}|,$$

where we use $k+1$ for $z^{\mathbf{r}_{\bullet}}$ if $z \notin \mathcal{R}(\mathbf{r}_{\bullet})$. The smaller the SF distance is, the more likely that \mathbf{r}_{es} and \mathbf{r}_{ex} share the same top k order of weighted influences.

For fixed number of samples ($h = 300, 1000, 3000$), we vary the probability q from 0.1 to 0.9; and for fixed probability ($q = 0.3, 0.5, 0.7$), we vary the number of samples from 100 to 5000. The results are shown in Figure 4, from which we can see that the SF distance is relatively small when $q \in (0.2, 0.4)$; and for more samples, the SF distance is smaller. In the following experiments, we will set the probability q between $(p+1)/n$ and 0.4 and sample size h between 100 to 500 to maintain a trade-off between estimation accuracy and run time. Note that the SF distance is not able to distinguish points in \mathbf{r}_{es} or \mathbf{r}_{ex} who share the same weighted influences. These experiments also show the necessity of iteratively re-estimate weighted influences since evaluating all weighted influences once does not match the order of exact weighted influences perfectly.

²We choose the number of data points very low since the exact weighted influences require exponential time to compute as n increases.

³• means es or ex .

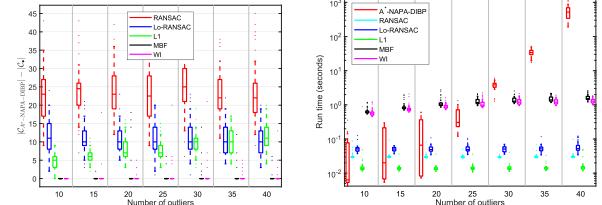


Figure 5. Comparison results for 8-dimensional robust linear regression: Compare (Left) consensus error and (Right) run time with different number of outliers. The ground truth is found by A^* -NAPA-DIBP. All experiments were repeated 50 times.

Relative Performance: In this experiment, we consider 8-dimensional linear regression problem, where the synthetic data are generated in the same way as we did in last experiments. The number of data points n is chosen as 200 and we vary the number of outliers from 10 to 40, which is limited by the computation time of A^* -NAPA-DIBP for generating ground truth. From Figure 5, we can find both Bernoulli weighted influences and the original MBF can achieve optimal solutions and the Bernoulli weighted method is faster than MBF in general. The effect of local expansion is compared in the supplementary.

4.2. Linearised fundamental matrix estimation

We further test the Bernoulli weighted influence based approach on five image pairs from sequence “00” of the KITTI Odometry dataset [12] in this subsection. For two given image pairs, SIFT keypoints [18] are detected and matched by the VLFeat toolbox [26]. Suppose $\mathbf{p}_1, \mathbf{p}_2$ are two features that are matched, where $\mathbf{p}_i = (x_i, y_i, 1)^T$ is the coordinate in view i , $i = 1, 2$, then each correspondence provides a linear constraint on the fundamental matrix $\mathbf{F} \in \mathbb{R}^{3 \times 3}$ as $\mathbf{p}_1^T \mathbf{F} \mathbf{p}_2 = 0$. In this experiment, we consider the linearised version of fundamental matrix estimation [13], where the inliers threshold is set to 0.02 for all image pairs. The iteration numbers of RANSAC and Lo-RANSAC are set to match the runtime of WI.

We report in Table 1 the average (running all algorithms 20 times) consensus and runtime. Observe that: (1) Although L1 is generally very fast, the consensus size found is smaller than that of the Bernoulli based approach WI; (2) RANSAC and Lo-RANSAC are allowed to run the same time as WI, yet WI is still better in terms of returned consensus size; (3) WI achieves almost the same consensus size as MBF but with less time cost (on image pairs 198-201, 417-420, 579-582, WI finds a slightly higher average consensus size than MBF). Further detail is in the supplementary.

4.3. Homography estimation

This experiment considers homography estimation together with linear residual model. For a set of correspondences from two views represented by $\{(\mathbf{x}_i, \mathbf{y}_i)\}$ with

Table 1. Results for linearised fundamental matrix estimation. Runtime of RANSAC and Lo-RANSAC is set to that of WI.

		104-108	198-201	417-420	579-582	738-742
L1	Consensus	252	264	317	497	429
	Time (s)	0.85	0.07	0.04	0.06	0.09
RANSAC	Consensus	266.55 (262,270)	284.00 (283,285)	353.30 (351,356)	497.05 (492,503)	432.60 (428,440)
Lo-RANSAC	Consensus	268.55 (266,273)	284.90 (284,287)	355.25 (354,356)	501.45 (496,506)	439.35 (432,443)
MBF	Consensus	271.70 (268,274)	287.70 (284,289)	358.95 (357,360)	507.15 (503,510)	444.20 (442,446)
	Time (s)	3.17 (2.65,4.41)	1.88 (1.65,2.47)	2.36 (1.96,2.66)	3.61 (3.17,4.36)	3.19 (2.68,3.56)
	WI	271.55 (269,274)	288.35 (286,289)	359.05 (356,360)	507.65 (502,510)	444.15 (442,446)
WI	Time (s)	2.64 (2.26,3.31)	1.75 (1.43,2.03)	2.10 (1.82,2.51)	3.42 (2.93,4.15)	3.09 (2.80,3.57)

$\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^2$, let $\tilde{\mathbf{x}}_i = (\mathbf{x}_i, 1)^T$, $\tilde{\mathbf{y}}_i = (\mathbf{y}_i, 1)^T$ be the homogeneous representation, homography estimation is to find a matrix $\mathbf{H} \in \mathbb{R}^{3 \times 3}$ such that $\tilde{\mathbf{y}}_i = \mathbf{H}\tilde{\mathbf{x}}_i$ for inliers. Since \mathbf{H} is 8 dimensional uniquely defined up to scale, it is possible to express homography estimation with linear residual as a linear regression problem (see Chapter 4 in [13] for details). We use 4 image pairs from the Zurich Buildings datasets (Building 5, 22, 28, 37). Similar to linearised fundamental matrix estimation, VLFeat toolbox is used for extracting SIFT features and correspondence matching. Note that each correspondence match produces two residual functions, which means the number of data points, inliers/outliers are doubled in optimisation.

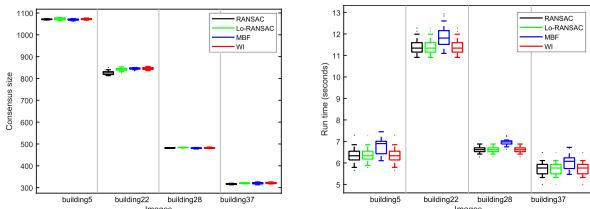


Figure 6. Comparison results for linearised homography estimation on 4 Zurich Building image pairs: (Left) Compare consensus size and (Right) run time. All experiment were repeated 20 times.

In this experiment, we allow RANSAC and Lo-RANSAC to run for the same time as WI - simply to show what they are capable of with such a budget. The inliers threshold is set to 0.1. From Figure 6, we find that WI can achieve (roughly) the best performance among the four sampling based methods. The run time of WI is smaller than that of MBF on average. Further comparison between WI and MBF can be found in the supplementary material.

5. Conclusion

We studied weighted influence in relation to the Max-Con problem. Our analysis is more general than [24] in terms of both the inclusion of weighted measures and in the study of the “non-ideal case”. Incorporating weighted influences into the algorithm of [24], is straightforward. As demonstrated by experiments on both synthetic and real data, doing so is an effective alternative of [24] (roughly, saving some run time while achieving similar consensus or achieve better consensus with similar time budget). However, the main message is that, rather than viewing biased sampling as some way to (more efficiently) estimate uniform influence, and “trusting” that the bias (in the estimates) thereby introduced has no detrimental effect on the ordering of influences (specifically of outliers having larger influence than inliers), a proper study of biased measures allows us to characterise the changes in these influence measures and check for retention of the correct ordering. We address this for two “natural” families of biased measures: Bernoulli and Uniform at a fixed Hamming level.

Of course our work has two obvious limitations: our analysis (of ideal cases) necessarily involves assumptions whose relevance to real world data could be challenged, and our experiments are not exhaustive.

Acknowledgements: This work was partly supported by the Australian Research Council grants DP200103448 (D. Suter & E. Zhang) and DP200101675 (T.-J. Chin).

References

- [1] Khurram Aftab and Richard Hartley. Convergence of iteratively re-weighted least squares to robust m-estimators. In

2015 IEEE Winter Conference on Applications of Computer Vision, pages 480–487, 2015. 1

[2] Zhipeng Cai, Tat-Jun Chin, and Vladlen Koltun. Consensus maximization tree search revisited. In *ICCV*, pages 1637–1645, 2019. 1, 2, 6

[3] Tat-Jun Chin, Zhipeng Cai, and Frank Neumann. Robust fitting in computer vision: Easy or hard? In *ECCV*, pages 701–716, 2018. 1

[4] Tat-Jun Chin, Pulak Purkait, Anders Eriksson, and David Suter. Efficient globally optimal consensus maximisation with tree search. In *CVPR*, pages 2413–2421, 2015. 1

[5] Ondrej Chum and Jiri Matas. Matching with prosac—progressive sample consensus. In *CVPR*, pages 220–226, 2005. 1

[6] Ondrej Chum, Jiri Matas, and Josef Kittler. Locally optimized ransac. In *Joint Pattern Recognition Symposium*, pages 236–243, 2003. 1, 6

[7] Ronald Fagin, Ravi Kumar, and Dakshinamurthy Sivakumar. Comparing top k lists. *SIAM Journal on discrete mathematics*, 17(1):134–160, 2003. 7

[8] Yuval Filmus. Friedgut–Kalai–Naor theorem for slices of the Boolean cube. *Chicago Journal of Theoretical Computer Science*, pages 14:1–14:17, 2016. 4

[9] Yuval Filmus. Orthogonal basis for functions over a slice of the Boolean hypercube. *Electronic Journal of Combinatorics*, 23(1):P1.23, 2016. 4

[10] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 6

[11] David Gamarnik. The overlap gap property: A topological barrier to optimizing over random structures. *Proceedings of the National Academy of Sciences*, 118(41), 2021. 6

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012. 7

[13] Richard Hartley and Zisserman Andrew. *Multiple view geometry in computer vision*. Cambridge university press, 2000. 7, 8

[14] G Kalai and S Safra. Threshold phenomena and influence, with some perspectives from mathematics, computer science and economics. Technical report, Discussion Paper Series dp398, Center for Rationality and Interactive Decision Theory, Hebrew University, Jerusalem, 2005. 4

[15] Huu Le, Tat-Jun Chin, and David Suter. An exact penalty method for locally convergent maximum consensus. In *CVPR*, pages 1888–1896, 2017. 1

[16] Huu Minh Le, Tat-Jun Chin, Anders Eriksson, Thanh-Toan Do, and David Suter. Deterministic approximate methods for maximum consensus robust fitting. *IEEE TPAMI*, 43(3):842–857, 2019. 1

[17] Hongdong Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *ICCV*, pages 1074–1080, 2009. 1

[18] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157, 1999. 7

[19] Ryan O’Donnell. *Analysis of boolean functions*. Cambridge university press, 2014. 2, 3

[20] Carl Olsson, Anders Eriksson, and Richard Hartley. Outlier removal using duality. In *CVPR*, pages 1450–1457, 2010. 1, 6

[21] Pulak Purkait, Christopher Zach, and Anders Eriksson. Maximum consensus parameter estimation by reweighted l_1 methods. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 312–327, 2017. 1

[22] Yongduek Seo, Hyunjung Lee, and Sang Wook Lee. Outlier removal by convex optimization for l_∞ approaches. In *Pacific-Rim Symposium on Image and Video Technology*, pages 203–214, 2009. 1

[23] Kristy Sim and Richard Hartley. Removing outliers using the l_∞ norm. In *CVPR*, pages 485–494, 2006. 1

[24] Ruwan Tennakoon, David Suter, Erchuan Zhang, Tat-Jun Chin, and Alireza Bab-Hadiashar. Consensus maximisation using influences of monotone boolean functions. In *CVPR*, pages 2866–2875, 2021. 1, 2, 3, 4, 6, 8

[25] Giang Truong, Huu Le, David Suter, Erchuan Zhang, and Syed Zulqarnain Gilani. Unsupervised learning for robust fitting: A reinforcement learning approach. In *CVPR*, pages 10348–10357, 2021. 2

[26] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1469–1472, 2010. 7

[27] Karl Wimmer. Low influence functions over slices of the boolean hypercube depend on few coordinates. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 120–131, 2014. 4