Improving Survey Aggregation with Sparsely Represented Signals

Tianlin Shi*
Stanford University
Stanford, CA 94305
tianlins@stanford.edu

Forest Agostinelli*
University of California - Irvine
Irvine, CA 92697
fagostin@uci.edu

Matthew Staib
Massachusetts Institute of Technology
77 Massachusetts Ave
Cambridge, MA 02139
mstaib@mit.edu

David Wipf
Microsoft Research
Beijing, China
davidwip@microsoft.com

Thomas Moscibroda
Microsoft Research
Beijing, China
moscitho@microsoft.com

ABSTRACT

In this paper, we develop a new aggregation technique to reduce the cost of surveying. Our method aims to jointly estimate a vector of target quantities such as public opinion or voter intent across time and maintain good estimates when using only a fraction of the data. Inspired by the James-Stein estimator, we resolve this challenge by shrinking the estimates to a global mean which is assumed to have a sparse representation in some known basis. This assumption has lead to two different methods for estimating the global mean: orthogonal matching pursuit and deep learning. Both of which significantly reduce the number of samples needed to achieve good estimates of the true means of the data and, in the case of presidential elections, can estimate the outcome of the 2012 United States elections while saving hundreds of thousands of samples and maintaining accuracy.

Keywords

Survey aggregation; Presidential elections; James Stein estimator; Compressive Sensing; Multi-task learning; Deep learning

1. INTRODUCTION

Surveys are a common way of inferring information about an entire population. In social polling for example, people study how different groups of the population respond to some binary questions, such as "do you use Facebook?", "do you think abortion should be legal?" [3] [2], or "who will be the new president?" These polls are frequently used to make predictions (e.g., predicting presidential elections), for market analysis (e.g. to answer questions such as which segments of the population prefers Xbox over PlayStation), or to obtain feedback. Survey -based companies such as Nielsen are represented in more than 100 countries and boast revenues of billions of dollars per year. At the same time, conducting a survey is expensive. For example, for conducting a survey based on a short phone-call interview, survey companies typically charge customers on the order of $20-$30 per call; and even small-scale surveys can easily cost up to multiple tens of thousands of dollars [30]. Nielsen, for example, employs approximately 40,000 people worldwide, reflecting the high-labor cost involved in conducting surveys. In addition, research shows that people are becoming more resistant to answering surveys [19], which may force companies to make good predictions with only a fraction of the data. This drives the search for advanced surveying and sampling strategies that could reduce the number of samples needed in social surveying.

The underlying fundamental problem in aggregating social surveying data is *averaging*. From the sample responses collected, the survey designer applies averaging to estimate the true mean response of each group of interest. At first thought, this problem looks easy: why not just average by "sample means"? That is, we just average all the samples collected within each group. In fact, if there is only one group, then "sample means" is admissible: no other averaging methods could dominate it. However, if there are more groups, then it is known that "sample means" is not necessarily the best due to the so-called Stein's paradox [13]. Stein [27] constructs the James-Stein (JS) estimator that recovers every single mean using information from all groups, and shows that it outperforms "sample means" no matter what the true means are. Efron and Morris [12] cast JS into an empirical Bayes framework and show that JS estimates are the result of a prior *regularization* and *shrinkage* of the sample averages.

In this work, we aim to explore the potential of regularization-based averaging that achieves good estimation while remaining flexible. An inevitable characteristic of the regularizers would be a *sparsity structure*, i.e., a limited number

*Tianlin Shi and Forest Agostinelli contributed equally to this paper.

KDD '16, August 13 - 17, 2016, San Francisco, CA, USA

ISBN 978-1-4503-4232-2/16/08...$15.00

DOI: http://dx.doi.org/10.1145/2939672.2939876

of non-zero parameters. Otherwise, there is little chance that those parameters could be estimated from data, and help the reconstruction of the true means. In fact, sparsity has already been deployed as a killer feature in sensing other types of data such as images, sounds, sensor readings, or network conditions. The well-known *compressive sensing* framework [10] allows sampling such data at a significantly lower rate beyond Shannon-Nyquist limit. This gain owes much to the fact that natural signals can be sparsified – sparsely represented under some known basis such as wavelets for images and speech. It would be great if we could similarly apply compressive sensing techniques to social surveying data. Alas, direct application of compressive sensing to social surveying is difficult, because social data (the true means) are too noisy to be sparsified.

We resolve this challenge by leveraging inspirations from James-Stein estimators. Instead of using sample averages of all groups as in JS, we consider shrinking the sample averages towards some unknown *global vector*, and use compressive sensing to recover this vector from the samples. Therefore, instead of sparsifying the true means, we propose to sparsify this unknown global vector. This idea marries the idea of compressive sensing and averaging estimators in a coherent framework, which we name *compressive averaging*. In this paper, we derive efficient algorithms based on this novel framework for estimating the true means based on survey samples. We also provide theoretical insight into the framework and understanding of when and how it works. Our evaluations based on real-world surveys indicate that compressive averaging can yield substantial improvements over the commonly used surveying methods.

This paper is organized as follows: In section 2, we introduce the problem setting, in section 3 we present compressive averaging, section 4 provides theoretical analysis of compressive averaging, section 5 shows empirical results, section 6 discusses related work, section 7 discusses future work, and section 8 is the conclusion.

2. PRELIMINARIES

A typical economical or sociological survey aims to gather information about characteristics of a population, such as incomes, attitudes, or interests. Formally, we assume that there are T groups in total, and every group is asked the same question. The quantity of interest is the average response per group, which we denote as $\mu[t]$ for group t. Notice that for binary questions, $\mu[t]$ is also the probability of a positive response in group t.

To estimate the quantity $\mu[t]$, the surveying methodology uses two basic steps: *sampling* and *aggregation*. For sampling, researchers send out questionnaires to individual entities in each group, and obtain samples $z_j[t](j = 1, 2, \ldots, n[t])$ for every cell $[t]$. Depending on the response rate and the design of the survey, different number of samples are collected for each group, denoted as $n[t]$. A reasonable model for $z_j[t]$ would be

$$z_j[t] \sim \begin{cases} \text{Ber}(\mu[t]) & \text{binary case} \\ \mathcal{N}(\mu[t], \sigma^2[t]) & \text{continuous case} \end{cases} \quad (1)$$

In both models, the average $y[t] = \sum_j z_j[t]/n[t]$ and the number of samples $n[t]$ are the *sufficient statistics* for estimating $\mu[t]$. For convenience, we treat these two cases in a unified setting:

$$y[t] \sim \mathcal{N}(\mu[t], \frac{\sigma^2[t]}{n[t]}). \quad (2)$$

For the continuous case, this approximation is exact; and for the binary case, the Central Limit Theorem implies that this approximation is good given a sufficient number of samples.

In the aggregation step, the survey designer performs statistical inference to estimate the quantity of interest $\mu[t]$ from samples $z_j[t]$. We would like the estimate to be as accurate as possible. Usually the expenses of data collection dominates the cost of aggregation. Therefore, to reduce the overall cost of surveying, we seek to reduce the total number of samples needed to incur a bearable amount of error.

Finally, note that for convenience we interchangeably use μ as a function of t, or as a vector. The same thing goes for y, n, and σ^2.

2.1 Sample Averaging

The straightforward estimator would just use the "sample means" $y[t]$ of each group, namely,

$$\hat{\mu}_{\text{MLE}}[t] = y[t] = \frac{1}{n[t]} \sum_j z_j[t]. \quad (3)$$

Sample averaging is also the maximum likelihood estimator, and is optimal when $T = 1$.

2.2 James-Stein Estimators

The James-Stein estimator (JS) [27] shrinks the sample averages y towards the global mean $f = T^{-1} \sum_{t=1}^{T} y[t]$, the unweighted average of all $y[t]$. Precisely,

$$\hat{\mu}_{\text{JS}}[t] = f + (1 - \gamma)_+ \cdot (y[t] - f), \quad (4)$$

where $(\cdot)_+ = \max(0, \cdot)$, and the shrinkage factor γ is

$$\gamma = (T - 3)\Big(\sum_t \frac{n[t]}{\sigma^2[t]} (y[t] - f)^2 \Big)^{-1}. \quad (5)$$

Where σ^2 is the true variance of the data. The smaller the distance between y and f, the more aggressive the shrinkage is. So the optimal amount of shrinkage is adapted to the data. By doing so, the James-Stein estimator makes a better tradeoff between variance and bias, and achieves a better estimation performance.

3. COMPRESSIVE AVERAGING

3.1 Intuition from Example

To see how sparsity might help improve the estimates, we first look at an example as shown in Figure 1, where James-Stein estimators do not work well. Suppose we have the true means

t	1	2	...	8	9	10
$\mu[t]$	1.0	1.0	...	1.0	99.0	101.0

and 10 samples are collected from each category with noise variance $\sigma^2[t] = 50$ for all t. It is well known that JS performs worse when such outliers exist [11] [13]. If we shrink the sample means towards the global mean, $f = 26.95$, the shrinkage parameter becomes very small, $\gamma = 0.002$. That is, JS almost becomes sample means.

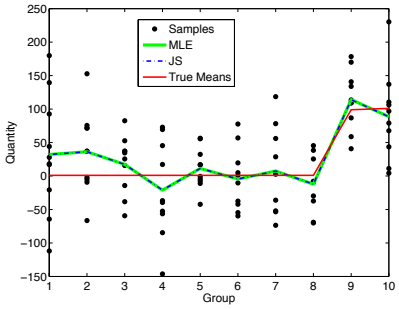


Figure 1: Intuitive example where James-Stein estimator (JS) does not work well. As the sample averages y (MLE) deviate a lot from global mean f, the shrinkage effect that JS exploits becomes negligible. So JS reduces to MLE.

A natural idea to improve JS would be to prevent the cases $t \geq 9$ and $t < 9$ from sharing the same global mean. For example, we could construct *a global mean* $f[t]$,

$$f[t] = \begin{cases} 1 & t \leq 8 \\ 100 & t = 9, 10 \end{cases}, \tag{6}$$

then we could shrink the estimates towards $f[t]$ so that the overall risk is much smaller. The problem with this idea is that when carrying out a survey, such global mean $f[t]$ is not known a priori. Therefore, it must be estimated from the data.

3.2 Modeling the Global Mean

At the conceptual level, all we need is a function $f[t]$ that fits the data well and comes from a restricted model family. For the outlier example in Figure 1, we might assume that $f[t]$ is a sparse combination of various components: a constant global mean $f_0[t] := 1$, and singleton functions $f_i[t] := \mathbb{I}[t = i]$. The sparse solution $f = f_0 + f_9 + f_{10}$ is exactly the function we wish for in Equation (6) and is quite stable due to the occurrence of f_0.

In general, we consider a model family $\mathcal{F}$ and a function $f[t] \in \mathcal{F}$. Then we explain data through a generative model:

$$\mu[t] \sim \mathcal{N}(f[t], A) \tag{7}$$

$$y[t] \sim \mathcal{N}(\mu[t], \sigma^2[t]/n[t]) \tag{8}$$

For some constant A and with known noise level $\sigma^2[t]$. The marginal distribution of y given f is

$$y[t] \sim \mathcal{N}(f[t], A + \frac{\sigma_t^2}{n[t]}). \tag{9}$$

The function f encodes our flexible prior about the latent quantities μ, and A controls how strong that prior ought to be. We jointly estimate both of them from data by minimizing the negative log likelihood of equation 9 which leads to the following optimization problem:

$$\min_{f \in \mathcal{F}, A \geq 0} \quad \sum_{t=1}^{T} \frac{(f[t] - y[t])^2}{A + \sigma^2[t]/n[t]} + \log(A + \frac{\sigma^2[t]}{n[t]}) \tag{10}$$

In the case of binary data, we find that modeling the $u[t]$ and $y[t]$ as being generated from a Beta distribution is intractable. Since the maximum variance for a Bernoulli random variable is 0.25, we put a prior on A to add a penalty for improbable A and only search in the range of possible values for A.

Once f and A are computed from the data, the means $\hat{\mu}$ can be estimated using the posterior mode, which is

$$\hat{\mu}[t] = f[t] + (1 - \frac{\sigma^2[t]/n[t]}{A + \sigma^2[t]/n[t]})_+ \cdot (y[t] - f[t]). \tag{11}$$

The core challenge in using Eq.(11) is to devise the family $\mathcal{F}$. If we allow $f[t]$ to be arbitrary the optimal solution would be to set $f = y$ which reduces to sample averaging.

Real social data can be noisy and therefore there does not generally exist a sparse representation for the true means μ. The idea of sparsity-based aggregation is that we instead exploit the *sparse representation of regularizers* f, and then use f to improve the estimates $\hat{\mu}$ of the true means.

To formalize this intuition in the example, we first construct a basis $\Phi = \{\phi_1, \phi_2, \ldots, \phi_k\}$ with $\phi_i \in \mathbb{R}^T$ such that

$$f[t] = \sum_i \alpha_i \phi_i[t], \tag{12}$$

where $\alpha \in \mathbb{R}^K$ is a sparse coefficient vector with $||\alpha||_0 \leq k \ll K$.

3.3 Estimating the Global Mean

Now that we know how to model f, we need a method to determine what f should be. In the next two sections we present two different ways to do this: orthogonal matching pursuit and deep learning.

3.3.1 Orthogonal Matching Pursuit

Based on our framework, the first step is to recover α (and hence f) as well as A empirically from the data. We solve the optimization problem Eq.(10) by iterating between A and α.

- **Fix α, solve for A.** Since A is only a single variable and we can therefore just use linear search algorithm to find the optimum.
- **Fix A, solve for α.** The optimization problem becomes

$$\begin{aligned} \min_{f \in \mathcal{F}} \quad & \sum_{t=1}^{T} \frac{(f[t] - y[t])^2}{A + \sigma^2[t]/n[t]} \\ \text{s.t.:} \quad & ||\alpha||_0 \leq k \end{aligned} \tag{13}$$

This problem resembles the sparse recovery formulations in compressive sensing [10]. Therefore, we solve it via similar techniques. We present a greedy algorithm (Algorithm 1) based on Orthogonal Matching Pursuit (OMP) [32] with modifications to handle the heterogenous noise levels.

Due to the connection to compressive sensing, we call this approach "compressive averaging" (or "compressive surveying") for aggregating survey data.

3.3.2 Deep Learning

Deep learning [20] has been shown to excel on a wide variety of tasks such as computer vision [28], speech recognition [17], and high energy physics [7]. We propose using deep learning to learn f. The input to the network is the y, σ^2, and a scaled n. The target output is the global mean f.

Using the assumption that the global mean is sparsely represented in some given basis, Φ, we can generate examples

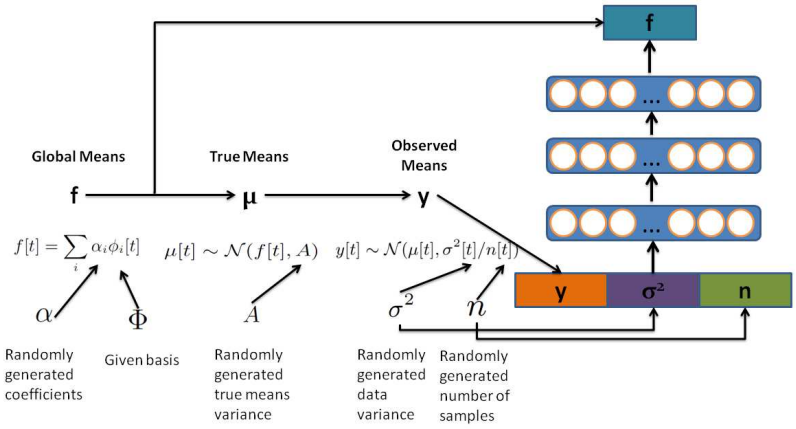


Figure 2: The process of generating training data for the deep neural network.

Algorithm 1 Modified Orthogonal Matching Pursuit for solving α.

Input. Sample average y, basis $\Phi = (\phi_1, \ldots, \phi_K)$.
Let $\Lambda = \text{diag}(n[1]/\sigma^2[1], ..., n[T]/\sigma^2[T])$.
Compute $y' = \Lambda^{\frac{1}{2}} y$, $\Phi' = \Lambda^{\frac{1}{2}} \Phi$.
Let residual $r_0 = y$, support set $S_0 = \emptyset$.
for $i = 1 \to k$ **do**
  Find $j = \arg\max_j r_{i-1}^\top \phi'_j / ||\phi'_j||_2$
  $S_i = S_{i-1} \cup \{j\}$.
  Concatenate $\Phi'_{S_i} = (\Phi'_{S_{i-1}}, \phi'_j)$.
  Compute $\alpha_{S_i} = (\Phi'^\top_{S_i} \Phi'_{S_i})^{-1} \Phi'^\top_{S_i} y$.
  Update residual $r_i = y - \Phi'_{S_i} \alpha_{S_i}$.
end for
Output. Support set S_k and coefficients α_{S_k}.

of a global mean by first randomly generating sparse coefficients α such that $||\alpha||_0 \leq k \ll K$ and then, using equation 12, produce a global mean f. Given the global mean f and variance A (where A is randomly selected from a given range of numbers), we can generate true means μ using equation 7. Given the true means $\mu[t]$, randomly generated $\sigma^2[t]$, and randomly generated $n[t]$, we can then get $y[t]$ using equation 8. For the binary case, $\sigma^2[t]$ is determined by $\mu[t]$. Figure 2 gives a visualization of the process of generating data to train the neural network. The benefit to this approach is we have, for all intents and purposes, infinite training data since we can always generate new examples. We take advantage of this by generating new data every 10 epochs.

The deep neural network (DNN) we used has 3 layers with 1000 hidden units in each layer. We use the adaptive piecewise linear (APL) activation units [5]. For every hidden layer we initialize the weight matrices with the "Xavier" filler [16]. The loss function is half the mean squared error between the output of the DNN and the global means:

$$L(\theta) = \frac{1}{2M} \sum_{m=0}^{M} \frac{1}{T} \sum_{t=0}^{T} (f_m[t] - \hat{f}_m[t])^2 \qquad (14)$$

Where $\hat{f}$ is the output of the DNN and M is the batch size. We use $M = 100$ for our experiments. We train the DNN using backpropagation and momentum [23] for 60,000 iterations. There learning rate starts at 0.1 and decays according to $\frac{0.1}{1.0001^i}$ where i is the iteration. The momentum starts at 0.5 and goes to 0.9 over 5000 iterations. No weight regularization is used.

After training the DNN, we optimize equation 10 by first obtaining f by doing a feedforward pass through the DNN. Then we can obtain A by fixing f to be the output of the DNN and optimizing equation 10.

As a note, we tried using a DNN to directly predict the true means μ, however, this proved too hard for the DNN to learn and performance was not any better than sample averaging. It seems that it is much better to first train a DNN to predict f and then shrink y to this prediction.

3.4 Choosing the Basis

A wide range of social surveys are conducted periodically, such as the General Social Survey [1]. For example, to study how presidential approval rates evolve, investigation institutes send out surveys monthly or yearly. So each category t corresponds to a month/year. Borrowing techniques from compressive sensing literature, we treat $y[t]$ as a discrete signal in time t, and use wavelets as the basis.

The basis we use in all the experiments is the Daubechies least-asymmetric wavelet packet ("wpsym" in MATLAB). The wavelet can be constructed at multiple scales, and we use the first four coarse scales (equivalent to 8 lowest frequency components). The sparsity level k is set to be 3.

The reason we select this basis is because the components of the basis can easily be used to make smooth curves. A basis such as the Haar wavelets would be more difficult since each component is discontinuous. $k = 3$ was chosen because it will lead to sparse solutions. We use the 8 lowest frequency components instead of all the frequency components of the basis because the higher frequency components will fit high variations in the data, which, in the case of noisy data, will lead to poor solutions.

However, in section 5.3 our experiments show that other smooth basis functions such as the Discrete cosine transform-II basis and the polynomial basis both give good results, while the discontinuous Haar wavelets sometimes perform poorly. Our experiments show that good results can be obtained with varying levels of k. In addition, our experiments

show that using the low frequency components in the basis is important for good performance.

4. THEORETICAL ANALYSIS

In this section, we provide theoretical insights into the performance of the orthogonal matching pursuit (compressive averaging) algorithm for estimating the global mean. For simplicity, we assume all noise levels are the same and equal to 1, i.e. $\sigma^2[t]/n[t] = 1$.

LEMMA 4.1. *Suppose the support of α is S, and let Φ_S denote the sub-dictionary with columns restricted to index S. If Φ_S is full-rank in column, then*

$$f = \Phi_S(\Phi_S^\top \Phi_S)^{-1}\Phi_S^\top y \tag{15}$$

PROOF. Notice when we assume $\sigma^2[t]/n[t] = 1$, the optimization problem Eq.(13) for solving α does not involve A at all. Given the support of α, the problem has a quadratic form, and the optimal solution is just Eq.(15). □

Therefore, given S, the sensitivity matrix

$$G_{ij} = \frac{\partial f[i]}{\partial y[j]} = \Phi_S(\Phi_S^\top \Phi_S)^{-1}\Phi_S^\top \tag{16}$$

is also the projection operator that $f = Gy$.

LEMMA 4.2. *Every optimal solution (A, f) of Eq.(10) satisfies the following equation,*

$$\frac{1}{A+1} = \frac{T}{||f-y||_2^2}. \tag{17}$$

PROOF. Given any solution of f, the optimal solution for A must satisfy Eq.(17). □

THEOREM 4.3. *Define λ as*

$$\lambda = \frac{||G(M-y)||_2^2}{||M-y||_2^2},$$

and

$$k = ||\alpha||_0$$

Notice λ, k are also random variables dependent on y. We have

$$\mathbb{E}_\mu||\hat{\mu}-\mu||^2 = T - \mathbb{E}_\mu[\frac{T^2-2T}{||f-y||_2^2} + \frac{2T(k-\lambda)}{||f-y||_2^2}] \tag{18}$$

PROOF. The total squared error can be decomposed as

$$\mathbb{E}_\mu||\hat{\mu}-\mu||^2 = \mathbb{E}_\mu[||y-\hat{\mu}||^2] - T + 2\cdot\mathbb{E}_\mu[\frac{\partial\hat{\mu}[i]}{\partial y[i]}]$$

Plug in the expression of $\hat{\mu}$ from Eq.(11) , we would have

$$\mathbb{E}_\mu||\hat{\mu}-\mu||^2 = T - \mathbb{E}_\mu[\frac{T^2-2T}{||f-y||_2^2} + 2T\frac{\mathrm{Tr}(G)||f-y||_2^2 - (f-y)^\top G^\top (f-y)}{||f-y||_2^4}]$$

For $G = \Phi_S(\Phi_S^\top\Phi_S)^{-1}\Phi_S^\top$, we have $G^\top = G$, $G\cdot G = G$ and $\mathrm{Tr}(G) = k$. Then we can prove the theorem. □

This theorem sheds light into when compressive averaging is useful. First of all, the following corollary shows it is safe to use compressive averaging whenever sampling averaging is applicable.

COROLLARY 4.4. *Compressive averaging dominates sample averaging for $T \geq 2$, i.e. for all μ,*

$$R_\mu(\hat{\mu}_{CA}) < R_\mu(\hat{\mu}_{MLE}). \tag{19}$$

Next, compared to James-Stein estimator, the following corollary shows that compressive averaging could introduce an overhead compared to JS. And the overhead is small when the representation of $f[t]$ is sparse.

COROLLARY 4.5. *The relationship of compressive averaging and James-Stein estimator can be established as*

$$R_\mu(\hat{\mu}_{CA}) - R_\mu(\hat{\mu}_{JS}) = \mathbb{E}_\mu[\frac{2n(\lambda-k)}{||f-y||^2}]. \tag{20}$$

Notice that $\lambda \in [\lambda_{\max}, \lambda_{\min}]$ where $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalue of G. This implies $k \geq \lambda$. Therefore, the right-hand side is always non-positive.

So when do we expect this approach to work? This approach is useful because it address a fundamental problem in James-Stein estimators: global means could be an inappropriate prior and lead to small shrinkage. This could have significant advantage when sparsity structure does exist in the problem. But we also must be aware that compressive averaging makes a tradeoff by reducing the deviation $||f-y||_2$ at the cost of an additional possible penalty term that is related to the inherent sparsity of the regularizing global function.

5. EXPERIMENTS

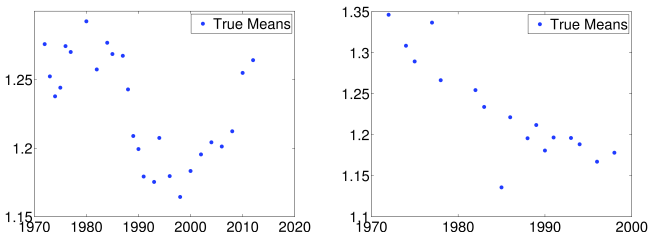
As baselines, we include sample averaging (`Avg`) and James-Stein estimator (`JS`). We also include multi-task averaging [15] (`MTAvg`), which uses a regularization that enforces all estimates to be close to one another. We use constant multi-task averaging since experiments from [15] show that this often gives the best performance. When referring to compressive averaging in our plots, using orthogonal matching pursuit to predict f is labeled as `OMP`, whereas the using a deep neural network (DNN) to predict f is labeled as `DNN`.

Sections 5.1 and 5.2 show the results of estimating the means of the GSS and Xbox datasets (the datasets are explained in their respective sections). The evaluation metrics used in section 5.1 and 5.2.2 is the mean absolute difference between each true mean (the mean when 100% of the data is used) and each estimated mean. The estimated and true means are first converted to percentages and then the mean absolute difference between the two percentages is obtained. For example, for the presidential election, a mean of 1.8 means that 80% of the people voted for Obama. The evaluation metrics in section 5.2.3 are slightly different because there is only one true mean: the exit poll data on election day. So, the mean absolute difference is taken between the estimated mean at the last day of the survey and the exit poll data.

Section 5.3 investigates how sensitive compressive averaging is to changes in parameters and basis functions. Section 5.4 shows how each method performs when the data is particularly noisy.

5.1 GSS Data

We take columns from the yearly General Social Survey (GSS) [1] data: a question regarding gun law (`GUNLAW`) and a question regarding women working (`FEWORK`). `GUNLAW` asks



(a) GUNLAW (b) FEWORK

Figure 3: Scatter plot of the true means on two datasets extracted from the General Social Survey. As we see, the true means exhibit patterns on the macroscopic scale, but contains inherent noise in individual groups.

people's opinions towards whether a police permit should be required before a person can buy a gun. We put 1 if person answers "FAVOR" and 2 if a person answers "OPPOSE." Other answers such as "Don't Know" are ignored. The data for this survey is available from 1972 to 2012, with a total of 36,921 samples. We would like to estimate the average approval rate of carrying a gun without a permit per year. Similarly, `FEWORK` is a question about people's opinion regarding married women earning money in business or industry given their husband is capable of supporting her. Label "1" means APPROVE and label "2" means DISAPPROVE. It contains a total of 24,401 samples. Figure 3 shows the scatter plot of true means of both datasets. The plots show that the true means of both datasets exhibit macroscopic patterns, but for each category, there exists inherent noise. We vary the sampling percentage from 1% to 30%, run each estimator for 30 runs, and compute the mean absolute difference between the estimated means and the true means (the means when 100% of the data is used). The mean absolute difference is shown in terms of percentage to provide the readers with an intuitive interpretation of the results. For example, if the estimated mean is 1.6 and the true mean is 1.7, this means that it was estimated that 60% percent disapprove but the truth is that 70% disapprove, leading to a mean absolute difference of 10%.

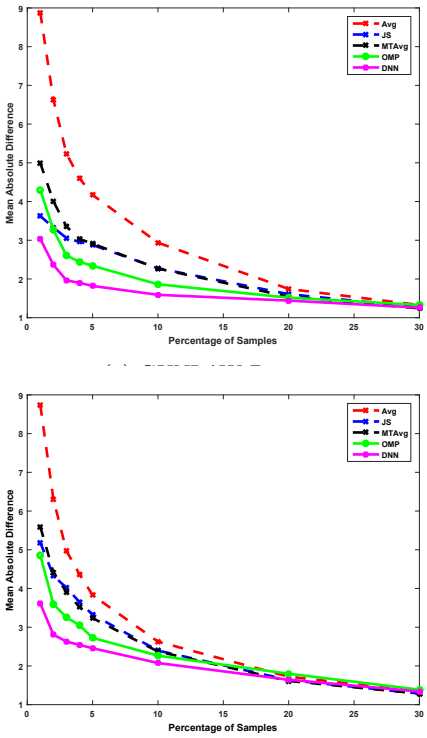
The results are shown in figure 4. One can see that, when taking small percentages of the total data, the `DNN` method outperforms all the other methods and the `OMP` method almost always outperforms all the other methods. For example, when 3% of the data is given the `DNN` method reduces error rates up to 30% over `JS` and 45%-61% over `Avg`.

5.2 Xbox Data for Presidential Election

5.2.1 *Preliminaries of Xbox Dataset*

We would like to apply compressive averaging to the data in [33]. The dataset in this paper contains non-representative polls taken from Xbox users about who they would vote for in the 2012 presidential elections. The surveys were taken in the 45 days leading up to the presidential elections and focuses on the two party (only Obama vs Romney) outcome. We collect a 1 for "Mitt Romney" and 2 for "Barack Obama."

Since the Xbox data was not representative of the entire population, post-stratification [22] was used to correct for the differences between the population that was sampled from the Xbox and the true population. To do this, [33] first splits the samples into cells based on the demographic information of each sample.



(b) FEWORK dataset.

Figure 4: Performance of different averaging methods on estimating the true mean of the dataset.

Given the cell values we can apply post-stratification using equation 21.

$$\hat{y}^{PS} = \frac{\sum_{j=1}^{J} N_j \hat{y}_j}{\sum_{j=1}^{J} N_j} \tag{21}$$

Where y_j is the estimate for cell j, J is the number of cells, and N_j is the true size of the population in the jth cell. In addition, the post-stratification estimate for a subpopulation (i.e. how all college graduates will vote), can be computed by equation 22.

$$\hat{y_s}^{PS} = \frac{\sum_{j \in J_s} N_j \hat{y}_j}{\sum_{j \in J_s} N_j} \tag{22}$$

Where J_s is the indexes of all the cells that contain the subpopulation s.

5.2.2 *Estimating the Means of the Raw Samples*

Since the dataset is not representative, the means of the sampled data will not, by themselves, give useful information. However, we can still verify compressive averaging does a good job at adjusting means by adjusting the means of the raw samples as shown in figure 5.

5.2.3 *Comparing Estimated Means with Exit Poll Data*

To determine how well the algorithm works, the 2012 exit poll data can be used to estimate the true value for how subpopulations voted in the 2012 elections. Each subgroup in the demographics of sex, race, age, education, party ID, and ideology was evaluated for accuracy. An example of how the subgroups of demographics vary across time is shown in

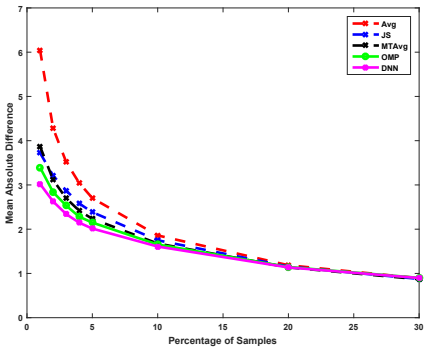


Figure 5: Performance of predicting the means of the raw samples of the Xbox dataset.

figure 6. In addition, analysis was performed on the largest 30 two-dimensional demographic subgroups (i.e conservative males, females between 45 and 64 years old, etc.). For each subgroup or two-dimensional subgroup that is to be analyzed, we post-stratify to that subgroup for each timepoint (i.e. get post-stratified estimate for college graduates at each timepoint) and use the post-stratified estimates as y, obtain the estimated means $\hat{\mu}$, and then get the mean absolute difference between the estimated mean at the last timepoint and the 2012 exit poll data. After we have done this for all the subgroups we obtain the mean and median absolute differences.

Figure 7 shows the results. We post-stratify on the age, sex, race, education, party, and ideology demographics. There were 246,683 respondents surveyed in total. Using only 10% of the data saves 221,967 samples.

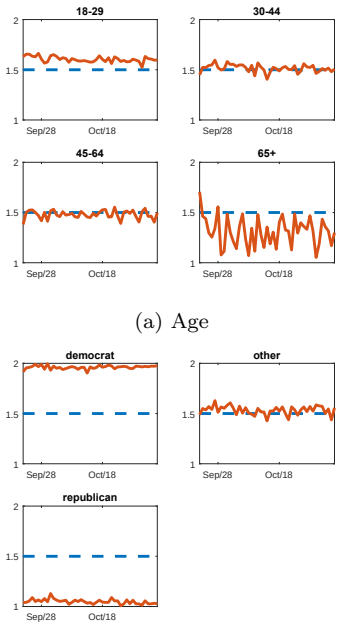
The results show that compressive averaging has around the same accuracy when using 10% of the data as it does when using 100% of the data. In addition, when compressive averaging uses 10% of the data it performs comparable to, and often even better than, when the `Avg` method uses 100% of the data (This is shown in figure Figure 8). The mean or median absolute difference of the `OMP` method at 10% is, *at most*, greater than the `Avg` method at 100% by 0.1.

One can notice that in figure 7b that for all the methods but `Avg`, there are percentages smaller than 100% at which they perform better. One reason this might be is that perhaps the value n is not precise because of post-stratification. It could be the value of n should be smaller, meaning we should trust y less, because of noise induced by post-stratification. Of course, n is smaller for smaller percentages, which may mean using 10% of the data gives a better estimate of n. However, we are not absolutely certain this is the correct explanation.

In addition, figure 7 is the only case in which the `OMP` method performs better than the `DNN` method. This could be due to the different evaluation methods. In the previous sections, we were evaluating how close the estimated means were to the true means. For this task, we are evaluating how the estimated mean at the very last day compares to the 2012 exit poll data.

5.3 Sensitivity to Basis and Sparsity Parameters

We test how sensitive compressive averaging is to the sparsity parameter k and the choice of the basis. The results for how the performance varies for choices of k between 1 and



(a) Age

(b) Party

Figure 6: The average of who the categories in the age and party demographics said they would vote for across time. The dashed line represents 1.5 (or 50% voting for Obama).

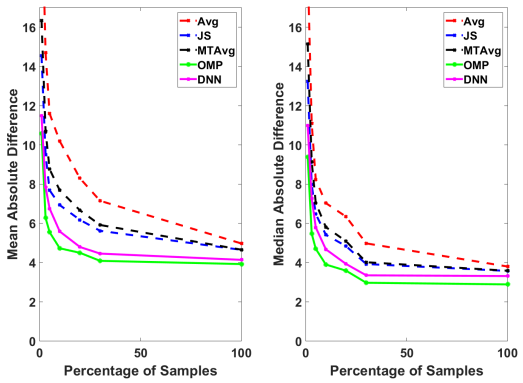
6 is shown in figure 9. The results show that any choice of k greater than or equal to 3 does not lead to any significant decrease in performance.

Figure 10 shows how changing the amount of L lowest frequency components affects performance. Results show that it is necessary to use a small L. Higher L leads to worse performance. This is due to the higher frequency components fitting high variations in the data; which is especially problematic for noisy data. Using the 4 lowest frequency components actually leads to better performance than using the 8 lowest frequency like we did in our experiments.

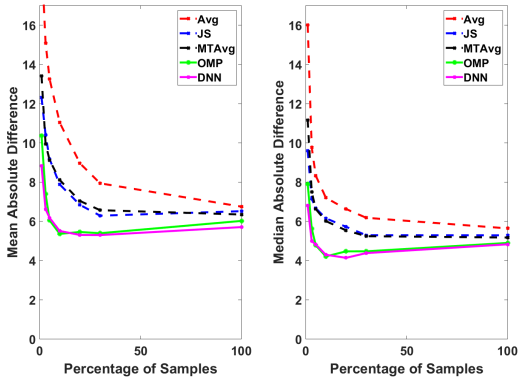
Figure 11 shows how the performance varies when using a different basis. One can see that using the Harr wavelets results in a significant decrease in performance for the `GUNLAW` dataset. This was expected because Haar wavelets are discontinuous. The discrete cosine transform-II performs slightly worse than the basis used in our experiments (wmpsym) but performs better than sample averaging, multi-task averaging, and James Stein in most cases. The polynomial basis actually outperforms wmpsym.

5.4 Resilience to Noise

We would like to investigate how the estimated means across time actually look under high noise, we take small fraction of the samples (as low as only one sample per timepoint) for the `GUNLAW` dataset and visualize the estimated means along with the sample means (y) and true means (μ). The plots are shown in figure 12. Of course, most methods perform very poorly when there is only one sample per timepoint, however, the `DNN` method is the closest to the true means in this case. As more samples are added the estimated means become closer to the true means, but even with 1146 total samples, the James Stein and multi-



(a) All demographic subgroups.



(b) The largest two-dimensional demographic subgroups.

Figure 7: The mean and median absolute differences of all the demographic subgroups as a function of sample ratio. No adjustments or imputations were made to the cell estimates. We run 30 different trials at each percentage.

task averaging methods deviate significantly more from the true mean than the `OMP` and `DNN` methods.

6. RELATED WORK

An explanation of Stein's phenomenon is that the estimator shrinks unbiased maximum likelihood $y[t]$. Efron and Morris formalize this intuition through an Empirical Bayes argument [12]. The key assumption is a hierarchical Bayesian model, where the means to be estimated are drawn from a normal distribution centered around zero. This assumption implies that the maximum posterior estimate of the true means are shrinking the sample averages.

Several extensions of James-Stein estimator have been introduced. Bock [9] considers dependencies between the true means and uses a covariance matrix to model the sample averages. Senda and Taniguchi [24] develop a type of James-Stein estimator for time series regression models. It has been shown in [21] that the James-Stein estimator itself is not admissible, and is dominated by the positive part of James-Stein estimator.

The idea of using information from all groups to improve the estimate of the quantity of a single group falls into the big realm of Multi-Task Learning (MTL) in the machine learning and data mining community. Early work in

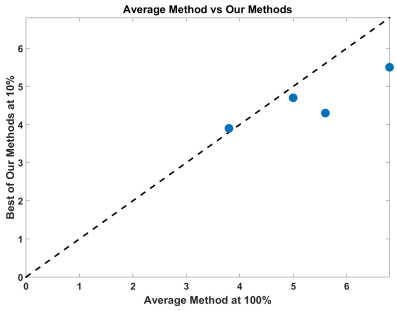


Figure 8: This plot shows the results from figures 7 of the average method with 100% of the samples vs the performance OMP with only 10% of the samples. Points to the right of the dashed line means that our method performs better than the `Avg` method while using 200,000 fewer samples.

MTL includes Thrun's "learning to learn" [31], and Baxter's "learning internal representation" [8]. One approach to modern multi-task learning is to build a hierarchical Bayesian model that infers characteristics shared by all groups [29]. Another line of work relies on a regularizer that penalizes the estimates for different groups. The types of regularization include distance to means [14], trace norm [4], pairwise distance [18] constraints, etc. These multi-task learning algorithms are usually tailored for regression [26], classification [34], feature learning [6], etc. For the survey aggregation problem, recently Sergey et al. [15] proposed a Multi-Task Averaging (MTA) approach that adds a penalty term invented in MTL literature to improve the estimates.

7. FUTURE WORK

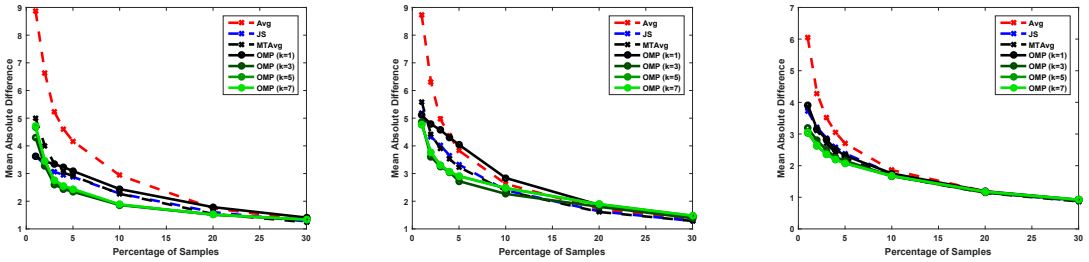
Results from [33] show that one can use an adjustment model to impute missing cell data to get even better accuracy when doing post-stratification. This introduces a new problem of adjusting $n[t]$ (the number of samples at each timepoint t) to account for the artificially added information which is, to the best of our knowledge, an open research problem. Preliminary results incorporating this method look promising.

We have shown how assuming the global mean of a temporal signal is sparsely represented in some basis can improve the estimation of the means. However, spatial data can be represented as a graph, which [25] showed can also be sparsely represented in some basis. Using this knowledge, it is possible that spatial data can also benefit from compressive averaging.

We would also like to extend this method to multi-way survey data, which can also be applied to the Xbox data. By dividing the population into demographic cells we could utilize low-rank matrix factorization techniques to further reduce the sampling rate required for these types of survey data.

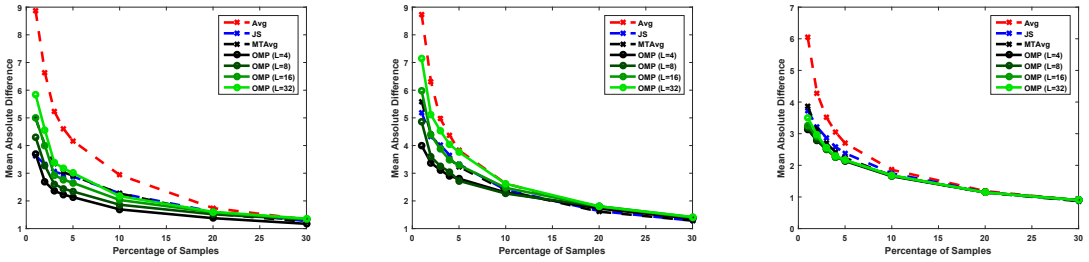
8. CONCLUSION

In this paper we exploit the temporal relationships between data by shrinking to an unknown global mean that is assumed to be sparsely represented in a given basis. We present two ways of estimating this global mean: orthogonal matching pursuit and deep learning. In our experiments, we were able to increase accuracy over sample averaging, James



(a) Gunlaw (b) Fework (c) Raw Xbox

Figure 9: The performance of the OMP method with different levels of sparsity. (k=3) is what we use in our experiments.



(a) Gunlaw (b) Fework (c) Raw Xbox

Figure 10: The performance of OMP when using the L lowest frequency components in the basis. ($L = 8$) is what we use in our experiments.

Stein estimators, and multi-task averaging. In addition, we were able to estimate the results of the 2012 presidential election using 10% of the samples (saving 221,967 samples) and still getting accuracy similar to or better than sample averaging with 100% of the samples.

9. ACKNOWLEDGMENTS

We would like to thank David Rothschild for providing the Xbox data for the 2012 presidential elections. Forest Agostinelli was supported by the NSF Graduate Research Fellowship Program (GRFP). Matthew Staib was supported by the Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

10. REFERENCES

[1] General Social Survey (GSS). *In P. Lavrakas (Ed.), Encyclopedia of survey research methods. (pp. 301-303)*, 2008.

[2] Coming and Going on Facebook. *Pew Research Center's Internet & American Life Project*, 2013.

[3] Widening Regional Divide over Abortion Laws. *Pew Research Center for the People & the Press*, 2013.

[4] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *The Journal of Machine Learning Research*, 10:803–826, 2009.

[5] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830*, 2014.

[6] A. Argyriou and T. Evgeniou. Multi-task feature learning. In *Advances in neural information processing systems*, 2007.

[7] P. Baldi, P. Sadowski, and D. Whiteson. Enhanced higgs boson to $\tau+ \tau-$ search with deep learning. *Physical review letters*, 114(11):111801, 2015.

[8] J. Baxter. Learning internal representations. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 311–320. ACM, 1995.

[9] M. Bock. Minimax estimators of the mean of a multivariate normal distribution. *The Annals of Statistics*, pages 209–218, 1975.

[10] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008.

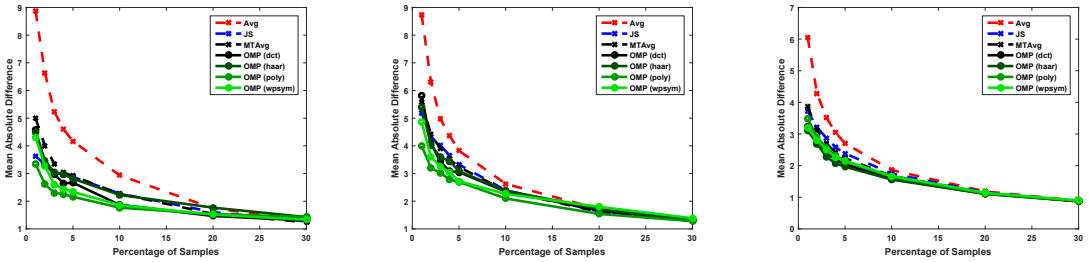
[11] B. Efron. *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*, volume 1. Cambridge University Press, 2010.

[12] B. Efron and C. Morris. Limiting the risk of Bayes and empirical Bayes estimators -âĂŤ Part II: The empirical Bayes case. *Journal of the American Statistical Association*, 67(337):130–139, 1972.

[13] B. Efron and C. Morris. Stein's paradox in statistics. *Scientific American*, 236:119–127, 1977.

[14] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004.

[15] S. Feldman, M. Gupta, and B. Frigyik. Multi-task averaging. In *Advances in Neural Information Processing Systems*, 2012.



(a) Gunlaw (b) Fework (c) Raw Xbox

Figure 11: The performance of the OMP method when using a different basis. We compare the Haar basis (haar), discrete cosine transform-II basis (dct), the polynomial basis (poly), and the Daubechies least-asymmetric wavelet packet (wpsym). The wpsym basis is the basis used in our experiments.

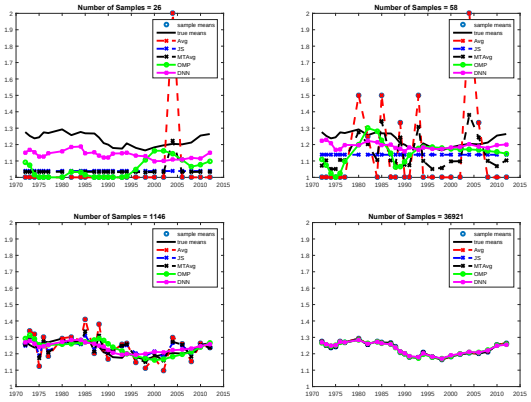


Figure 12: Visualization of the estimated means from `GUNLAW` survey taken between 1972 and 2012. The plots go from the extreme case of only one sample per timepoint (26 total samples because the survey was not conducted every year) to having all of the samples for each timepoint (36,921 samples).

[16] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.

[17] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al. Deepspeech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[18] J. Honorio and D. Samaras. Multi-task learning of gaussian graphical models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 447–454, 2010.

[19] A. Kohut, C. Doherty, S. Keeter, et al. Polls face growing resistance, but still representative. *Pew Center for Research on People and the Press, Washington, DC*, 2004.

[20] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[21] E. L. Lehmann and G. Casella. *Theory of point estimation*, volume 31. Springer Science & Business Media, 1998.

[22] R. J. Little. Post-stratification: a modeler's perspective. *Journal of the American Statistical Association*, 88(423):1001–1012, 1993.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[24] M. Senda and M. Taniguchi. James-stein estimators for time series regression models. *Journal of multivariate analysis*, 97(9):1984–1996, 2006.

[25] T. Shi, D. Tang, L. Xu, and T. Moscibroda. Correlated compressive sensing for networked data. In *Uncertainty in AI*, 2014.

[26] M. Solnon, S. Arlot, and F. Bach. Multi-task regression using minimal penalties. *The Journal of Machine Learning Research*, 13(1):2773–2812, 2012.

[27] C. Stein et al. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley symposium on mathematical statistics and probability*, volume 1, pages 197–206, 1956.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[29] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011.

[30] The Wallace Foundation. Workbook F: Telephone Surveys. Accessed: 2014-09-01.

[31] S. Thrun. Learning to learn: Introduction. In *In Learning To Learn*. Citeseer, 1996.

[32] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[33] W. Wang, D. Rothschild, S. Goel, and A. Gelman. Forecasting elections with non-representative polls. *International Journal of Forecasting*, 2014.

[34] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *The Journal of Machine Learning Research*, 8:35–63, 2007.