Automatic Algorithm Configuration for Compressive Sensing

Sharan Vaswani

Computer Science Department, University of British Columbia, 2366 Main Mall, Vancouver, BC, Canada V6T 1Z4
{sharanv}@cs.ubc.ca

Abstract. Compressive sensing involves efficiently recovering signals from fewer samples than that dictated by the Nyquist-Shannon sampling theorem by exploiting some apriori information about the signal. Most compressive sensing algorithms proposed in the literature have a number of tunable parameters. These are tuned manually by a trial and error scheme. In this paper, we study the importance of systematic parameter tuning for such algorithms. To demonstrate our approach, we perform automatic algorithm configuration for a sample example, namely for image reconstruction using a Total variation based algorithm (TVAL3) in the context of the Single Pixel Camera. We use two algorithm configurators - an iterative local search algorithm (ParamILS) and a sequential model based algorithm (SMAC) for systematically tuning the parameters. The study reveals that simple parameter tuning can lead to gains in the speed and quality of reconstruction. Although this study is restricted to the specific application and algorithm chosen, the ideas developed can be used for any compressive sensing algorithm having a number of tunable parameters.

Keywords: compressive sensing, single pixel camera, image reconstruction, automatic algorithm configuration

1 Introduction

There has been much work on compressive sensing algorithms since the seminal paper on sparse signal recovery by Candes et.al [4]. The central results state that a vector $x \in R^N$ sparse in some domain can be recovered from a small number of linear measurements $y = Ax \in R^K$, $K \ll N$ (or y = Ax + e when there e is measurement noise). Compressive sensing has found applications in a large number of domains such as image deblurring [7], image denoising [21], image reconstruction for MRI modalities [14] [13] , development of the single pixel camera [6] [20] with applications to hyper-spectral and infrared imaging [1].

There are several compressive sensing applications that demand a real time solution. One such application is the Single Pixel Camera (SPC) developed at Rice university. The single pixel in SPC refers to the single photo-diode used in the camera. Instead of using multiple sensors to measure the incident light from the image to be taken, SPC measures a linear superposition of light intensities from different pixels using a single photo-diode. By measuring multiple such linear superpositions and exploiting the spatial smoothness in the image it reconstructs the original image. SPC has been used for medical diagnostics, surveillance, fault detection applications in the infrared domain. [1]. SPC is highly effective in domains such as infrared where the cost of each sensor is very high. Details about the SPC are presented in the next section.

One of the challenges in compressive sensing is the high computational complexity of the reconstruction algorithms .Every compressive sensing algorithm is iterative in nature and typically has a computational complexity of $\mathcal{O}(n^4)$ where n is the size of the signal. Because of this, it becomes difficult to accurately reconstruct the signal for applications which demand a real time solution like those described above. Hence, the development of efficient algorithms achieving a reasonably accurate reconstruction within time constraints is an active area of research in compressive sensing. [19] [3] [23]. One such algorithm is TV minimization by Augmented Lagrangian and ALternating direction Algorithm (TVAL3) [12]. [12] proposes an algorithm for efficient image reconstruction by exploiting the spatial smoothness in the image. The technical details of this algorithm are explained in the next section.

Another major challenge in designing practical compressive sensing algorithms is their dependence on tunable parameters. Most reconstruction algorithms proposed have a number of tunable parameters. The speed and quality of reconstruction is usually very sensitive to the chosen parameter values and hence assigning appropriate values to these parameters becomes important. Algorithm designers tune the parameters manually using a trial and error scheme. [15] gives a compelling justification for the use of automated methods for tuning parameters for compressive sensing algorithms. This brings us to the realm of automatic algorithm configuration. There has been some previous work in the literature on automatic parameter tuning for compressive sensing applications. [15] uses the concept of phase transitions for parameter tuning for iterative algorithms like [16] and [5] which employ thresholding and relaxation techniques to solve the under-determined system of equations. Parameter tuning for the specific case of image denoising was explored in [18]. In this paper, the parameter space is searched exhaustively. Domain specific knowledge is used to prune parts of the search space. [24] uses the EM algorithm for automatic parameter tuning for an algorithm solving the multiple measurement vector problem. To the best of our knowledge this study is the first which uses general automatic configuration methods for compressive sensing applications without using any domain specific knowledge.

The remaining paper is structured as follows: Section 2 gives the necessary background and precisely formulates the problem solved in this study. Section 3 gives some important details about the working of two automatic algorithm configurators - ParamILS and SMAC. Section 4 describes our experimental setup and results. Section 5 concludes our paper and we outline some directions for future work in Section 6.

2 Background

2.1 Compressive Sensing

As explained in the previous section, compressive sensing involves reconstructing a higher dimensional signal (x) from a small number of linear measurements y. Since the number of measurements is much less than the number of samples i.e. the matrix A has lesser number of rows than columns, there are be multiple solutions capable of explaining the observations. Such an under-determined system admits multiple solutions. In order to resolve the ambiguity among the solutions and to obtain a meaningful result some apriori information about the signal is required. The information needed is problem specific. Some typical apriori information used for reconstruction of natural images is sparsity in wavelet domain, spatial smoothness of the image etc. In an nutshell, the recovery of the original signal from incomplete measurements involves solving an inverse problem using as a priori information the sparsity of the signal in some domain. This notion can be formulated formally as:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \eta, \tag{1}$$

where $\mathbf{A}$ is the measurement matrix (also known as sampling matrix) of size $M \times N$ (in the context of compressive sensing $M < N$), $\mathbf{y}$ is the measured data of size M, $\mathbf{x}$ is the vector of unknowns and is of size N, and η is the noise in the measurement. The apriori information is also required to be mathematically modeled and augmented with the linear system in a meaningful way to formulate the problem completely. For example, apriori information that the vector is $\mathbf{x}$ is sparse in a certain basis is modeled as:

$$\arg\min_{x} \|\mathcal{B}^{*}\mathbf{x}\|_0, \tag{2}$$

where $\mathcal{B}^{*}$x represents the projection of the vector x onto B, $\|.\|_0$ denotes the ℓ_0 'norm' i.e. the counting 'norm' of the vector. Using ℓ_0 'norm' leads to an intractable combinatorial optimization problem. In order to overcome the problem with ℓ_0 'norm' the convex relaxation of ℓ_0 'norm' is used, e.g. the ℓ_1 norm.

An example of apriori information suitable for natural images is spatial smoothness. Typically spatial smoothness is enforced by minimizing the total variation (TV) of the image. The total variation (TV) of an image is computed as the difference between neighboring pixel values. Putting all of this together, a sample formulation of a compressive sensing reconstruction problem using spatial smoothness as apriori information (i.e. penalizing the total variation) is as follows:

$$\begin{aligned} &\arg\min_{x} \|TV(\mathbf{x})\|, \\ &s.t. \|Ax - y\| \leq \epsilon \end{aligned} \tag{3}$$

This is exactly the problem solved by the TVAL3 algorithm described subsequently.

2.2 Single Pixel Camera

As described in the previous section, the Single Pixel Camera or SPC directly acquires random measurements of a scene without collecting all the pixels. SPC makes compressive measurements in the sense that it samples the scene fewer times than the number of pixels/voxels and is able to accurately reconstruct the scene using some apriori information about the captured image. The camera employs a digital micromirror device (DMD) to optically calculate linear projections of the scene onto pseudorandom binary patterns. Each mirror in the DMD can be prograatically configured reflect the light coming from the scene either towards or away from the photodiode. Hence the light intensity captured by the photodiode is proportional to the linear superposition of the pixel intensities in the original scene for which the mirrors reflected the light in the direction of the photo-diode. Multiple measurements of the scene are taken by generating different random binary patterns on the DMD.

Figure 1 shows the architecture of the Single Pixel Camera. Each entry in the measurement vector b corresponds to a light intensity measurement at the photodiode. Each random binary pattern input to the DMD is one row of the measurement matrix A. The number of rows in A is equal to the number of measurements taken. If x represents the original scene in vectorized form and if we use spatial smoothness of the scene as the apriori information, we need to solve exactly 3 to reconstruct the original image.

2.3 TVAL3

TVAL3 stands for TV minimization by Augmented Lagrangian and ALternating direction Algorithm. This algorithm is the state of the art in terms of both speed and quality of reconstruction for image reconstruction using spatial smoothness as apriori information as formulated in 3.

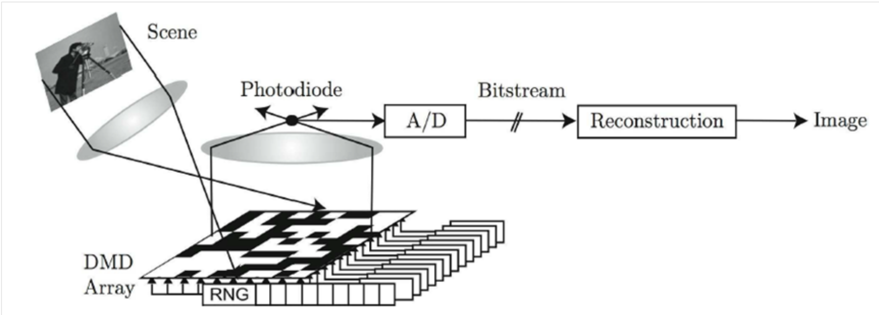


Fig. 1. Architecture of Single Pixel Camera

More precisely, it solves the constrained optimization problem given in 4.

$$\min_x \sum_i \|D_i x\| \quad s.t. Ax = y \tag{4}$$

where $x \in R^N$ is the original image, D_i is the difference gradient of x at pixel i, $A \in R^{M \times N}$ is the measurement matrix, y $\in R^M$ is the observation obtained via linear measurements, M is the number of measurement, N is the number of the pixels in the image and $\|\|$ can be the first or second norm. The ratio $\frac{M}{N}$ is called the under-sampling ratio f. Lesser the under-sampling ratio, lesser the number of measurements and the harder it is to reconstruct the original image.

TVAL3 formulates the problem as an augmented Lagrangian which is a Lagrangian formulation of the constrained optimization problem augmented with a quadratic penalty. The problem to be solved then translates to equation 5.

$$\min L_A(w_i, x) = \sum_i (\|w_i\| - \nu_i^T(D_i x - w_i) + \frac{\beta_i}{2}\|D_i x - w_i\|_2^2) - \lambda^T(Ax - y) + \frac{\mu}{2}\|Ax - y\|_2^2) \tag{5}$$

where w is a slack variable, ν and λ are Lagrangian multipliers whereas μ and β are quadratic penalty parameters. It is possible to decompose the above problem into the x-subproblem and w-subproblem. TVAL3 uses an alternating direction approach to solve this problem i.e. the x and w sub-problems are alternatively solved in each iteration. The w-subproblem is formulated as equation 6.

$$\min_i \sum_i (\|w_i\| - \nu_i^T(D_i x - w_i) + \frac{\beta_i}{2}\|D_i x - w_i\|_2^2) \tag{6}$$

The above problem is separable with respect to w and hence can be solved by using the shrinkage formulae presented in [12]. The x-subproblem is formulated in equation 7.

$$\min_x \sum_i -\nu_i^T(D_i x - w_i) + \frac{\beta_i}{2}\|D_i x - w_i\|_2^2 - \lambda^T(Ax - y) + \frac{\mu}{2}\|Ax - y\|_2^2) \tag{7}$$

The x-subproblem is represents an inverse problem and takes up majority of the computation time for the algorithm. In the case of TVAL3, it is solved by the one step steepest descent method with the step length chosen by the Barzilai and Borwein [2] method. The x-subproblem can be solved by alternative approaches such as conjugate gradient, BFGS.

After each iteration of minimizing w and x, the quadratic penalty parameters are increased by a certain fixed amount. This corresponds to the intuition that as the solution converges to the true solution, constraint violations should be penalized more. The Lagrangian multipliers λ and ν are updated according to the formula suggested by Hestenes [8] and Powell [17] given in equation 9

$$\nu_i = \nu_i - \beta_i(D_i - w_i) \tag{8}$$

$$\lambda = \lambda - \mu(Ax - y) \tag{9}$$

As we iteratively minimize the x and v subproblem, the error decreases i.e. x tends to the original image. We stop the alternate iterative minimization after either a certain maximum number of iterations ($outer - iter$) has been reached or when the change in the reconstructed image $x$is less than a certain threshold($outer - tol$).

2.4 Problem Definition

We can immediately see the number of parameters involved in the optimization process. For the TVAL3 case, the tunable parameters include the ones given in table 1.

Parameter	Symbol
Maximum number of outer iterations	$outer - iter$
Threshold tolerance for outer iterations	$outer - tol$
Initial values of quadratic penalty	μ^0, β^0
Rate of increase in quadratic penalty	$rate$
Maximum number of inner iterations (used for steepest descent)	$inner - iter$
Threshold tolerance for inner iterations	$inner - tol$
Maximum number of iterations (outer + inner)	$max - iter$

Table 1. Tunable parameters for TVAL3

The author gives typical ranges for these parameters and also identifies the best value (to obtain the maximum reconstruction quality) for each of the parameters. These values have been obtained by manually tuning the parameters. The values of the parameters depend on the amount of noise level in the images and also on the domain under consideration. The sensitivity of reconstruction to tunable parameters in TVAL3 and the need for tuning the parameters for each domain or noise level provides motivates the use of automatic algorithm configuration methods. In this paper, we address the problem of systematic parameter tuning for the TVAL3 algorithm used for image reconstruction in the context of single pixel camera described above.

3 Automatic Algorithm Configuration

Figure 2 shows the general procedure for automatic algorithm configuration. The configurator is initialized using some default parameter values. The configurator evaluates candidate parameter configurations by running the target algorithm with the candidate parameter configuration on a set of benchmark instances from the training set. After obtaining the solution cost (maybe the quality of the solution, or the runtime) the configurator picks another candidate configuration to compare against. How the next candidate configuration is picked and how two configurations are compared is dependent on the configurator being used. At each point of time, the configurator maintains the incumbent configuration which is the best parameter configuration found so far.

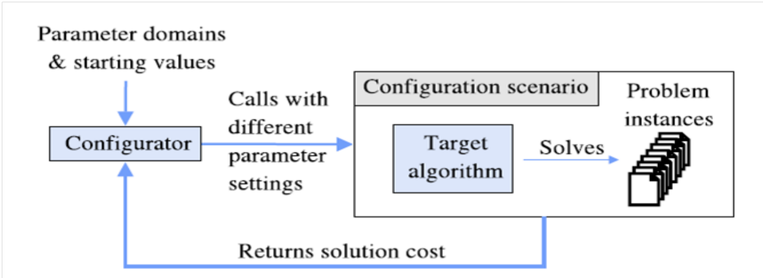


Fig. 2. General procedure of automatic algorithm configuration. (Image taken from [11])

Parameters for the configurator include the tuner-timeout (the time for which the configurator searches the parameter space) and the captime. We briefly explain the notion of capping - often the search for a performance-optimizing parameter setting spends a lot of time in evaluating a parameter configuration that is much worse than other, previously-seen configurations. For example, say want to find the configuration which minimizes the time required to solve a particular instance. If one of the candidate configurations is taking a large time (more than a certain threshold (called the captime) or far greater than the older configurations) we can prematurely terminate the evaluation of this particular configuration and conclude that it is worse than the previous configurations. This helps in terminating the evaluation of bad configurations which speeds up the search. As mentioned before, we consider two automatic algorithm configurators namely ParamILS and SMAC. We briefly describe each of them in the next subsections.

3.1 ParamILS

ParamILS [11] is a model free automatic algorithm configuration method based on Iterated Local Search through the discrete parameter configuration space. It uses a combination of default and random settings for initialization, employs iterative first improvement as a subsidiary local search procedure, uses a fixed number of random moves for perturbation, and always accepts better or equally-good parameter configurations, but re-initializes the search at random with a certain non-zero probability. It is based on a one-exchange neighborhood, that is, we always consider changing only one parameter at a time. Depending on how they choose the training instances to compare two candidate configurations, there exist 2 variants of ParamILS - BasicILS and FocusedILS. We refer the reader to [11] for details on the working of ParamILS and the capping mechanism used by it. Since ParamILS searches through the discrete parameter space and cannot directly handle continuous parameters, these are handled by appropriate discretization.

3.2 SMAC

SMAC [10] stands for Sequential Model Based Optimization for Algorithm Configuration. It is currently the state of the art in automatic algorithm configuration. It is an example of a model based automatic algorithm configuration method. Such methods construct a regression model (often called a response surface model) that predicts performance and then use this model for optimization. Sequential model-based optimization (SMBO) iterates between fitting a model and gathering additional data based on this model. SMAC constructs a random forest model and uses it to choose promising candidate configurations. It uses the models predictive distribution to compute its expected positive improvement over the the incumbent. This approach automatically trades off exploitation and exploration. SMAC also offers the flexibility of using instance specific features to more efficiently select training instances to compare two candidate configurations. These features can be integrated easily into the random forest model. The reader is advised to go through [10] for internal details about SMAC. SMAC is able to directly handle continuous as well as categorical parameters.

4 Experiments

In this section, we describe experiments for automatic algorithm configuration for the TVAL3 algorithm and present some preliminary results on the effectiveness of our approach.

4.1 Experimental Setup

We need a set of instances (the training set) to find the optimal parameter configuration and another disjoint set of instances (the test set) to evaluate the effectiveness of the chosen configuration. For our experiments, we choose 18 standard images (images like lena, cameraman) as benchmark images. We divide this set of images into the training and test sets of size 12 and 6 respectively. We use images with resolution 32×32 in our experiments mainly because of the faster reconstruction time of small image sizes. Given enough computational resources the same set of experiments can be performed for higher image sizes as well. For each image, we choose different undersampling ratios f from 0.1 to 0.9 in steps of 0.1. We also choose different seed values for generating the random measurement matrix A. Hence both the training and test sets consists of 32×32 images with various seeds and undersampling ratios. The number of instances chosen for training is 200 whereas the test set consists of 100 such instances.

The metric used to assess the quality of a reconstruction is the relative error between the original image and the reconstructed image. For each of the two configurators - ParamILS and SMAC, we perform 5 independent configuration runs. The configuration which gives the least error on the training set is considered as 'best' and is used for reconstruction on the test set. All plots presented in this paper represent results on the test set. The tuner timeout for ParamILS is set to 30 minutes and for SMAC it is set to 30 minutes for the discrete case and 1 hour for the continuous case. Since our study considers only 32×32 images which take very small amount of time (typically less than a second) for reconstruction, we don't use capping and thus use large captimes of 5 seconds in all our experiments. Since the computational complexity for the iterative TVAL3 procedure is high, larger image sizes take a significantly longer time for reconstruction (for example it takes around 30 seconds for reconstructing images of size 128×128.) Thus capping will play an important role if we use higher image sizes in the algorithm configuration.

Amongst the parameters for TVAL3 listed in table 1, it has been found by the author of TVAL3 and also verified independently by us that the values of mu^0, $beta^0$, $max - iter$ and $outer - tol$ affect the reconstruction the most. For this study, we verify our approach by automatically tuning these parameters. The other parameters are set to their default values. The ranges for each of these parameters is estimated empirically. The following table gives the maximum and minimum values considered for each of the 4 parameters mentioned above. It also shows the configuration space for the discrete and continuous optimization.

Parameter	Range	Discrete Configuration Space	Continuous Configuration Space
$max - iter$	1 - 400	1,10,50,100,200,300,400	continuous in range [1,400]
μ	2^4 - 2^{13}	$2^4, 2^5....2^{12}, 2^{13}$	continuous in range $[2^4, 2^{13}]$
β	2^4 - 2^{13}	$2^4, 2^5....2^{12}, 2^{13}$	continuous in range $[2^4, 2^{13}]$
$outer - tol$	$10^{-1} - 10^{-12}$	$10^{-1}, 10^{-2}....10^{-11}, 10^{-12}$	categorical in { $10^{-1}, 10^{-2}....10^{-11}, 10^{-12}$ }

Table 2. Ranges for parameters considered in experiments

All our experiments are run on a 32 bit Intel Core2-Duo Dell laptop having a clock frequency of 2 GHz and 3GB main memory.

4.2 Results

In the subsequent experiments, we use automatic algorithm configuration to optimize the quality (minimize the error metric) of reconstruction. The first experiment was performed using ParamILS as the algorithm configurator. This experiment shows that the quality of reconstruction can be arbitrarily bad if we use random parameter configurations and starting from such bad configurations, ParamILS is able to find a configuration which gives a better reconstruction quality. This result is shown in figure 3. In this figure, the X axis gives the error using a random initial configuration whereas the Y axis gives the error obtained using the 'best' (as explained in the previous subsection) configuration from ParamILS. Each point in the plot represents a different instance in the test set. The y = x line divides the quadrant into 2 parts - the bottom region where the error due to the tuned parameters is less than that obtained using the random configuration and vice versa for the other region. Points lying on the y = x indicate similar performance between the tuned and random parameters. The same convention is followed for all subsequent results.

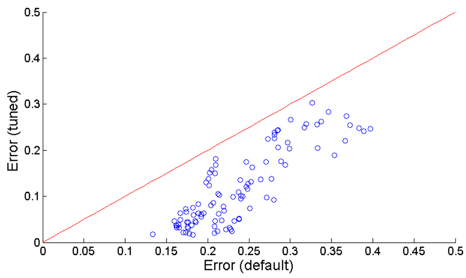
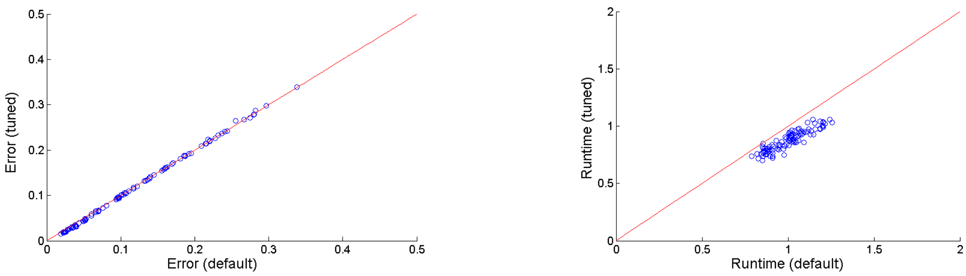


Fig. 3. Experiment 1: Scatter plot of the error obtained using a random configuration vs error obtained using the tuned configuration obtained by ParamILS.

Experiment 2 compares the reconstruction quality between the default configuration of parameters (as suggested by the author of TVAL3) and the parameters found by ParamILS. The 'best' configuration obtained by ParamILS is $max\_iter = 150$, $\beta = 2^4$, $\mu = 2^9$, $outer\_tol = 10^{-12}$ which is quite close to the default configuration $max\_iter = 300$, $\beta = 2^5$, $\mu = 2^8$, $outer\_tol = 10^{-8}$. Figure 4(a) shows similar quality for both the configurations. Hence, starting from a random initial configuration, ParamILS is able to find a parameter configuration which is as good as the default configuration suggested by the author. The reconstruction quality obtained using the tuned parameters is better than that obtained using the default parameters in 82% of the case with an average improvement of 2%. After an extensive evaluation consisting of more configuration runs and different initial configurations, we obtain similar (marginally better) quality measures. For the same experiment, we also plotted the reconstruction time for the test instances in a similar way. Figure 4(b) shows the results obtained. The axes in this case denote the reconstruction times for each of the configurations. Using tuned parameters gives a faster reconstruction in all cases and an average improvement of 13% in runtime. Hence we can obtain a similar quality reconstruction in a shorter time.

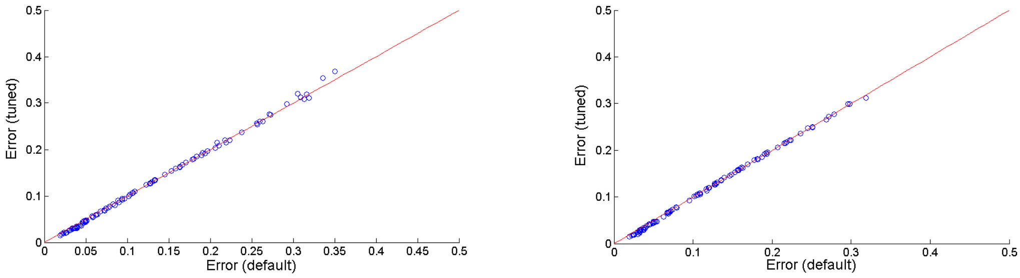


(a) (b)

Fig. 4. Experiment 2: (a) Scatter plot for the error obtained using the default configuration (suggested by the authors) vs error obtained using the tuned configuration obtained by ParamILS. (b) Scatter plot for the reconstruction time using the default configuration vs the reconstruction time using the tuned configuration obtained by ParamILS.

Experiment 3 uses SMAC as the algorithm configurator but optimizes the parameters in the discrete space (as in ParamILS). No instance based features are used in this experiment. Figure 5(a) shows a plot similar to the one obtained in figure 4(a) i.e. it shows the comparison between the tuned parameters obtained from SMAC and the default parameters. Again, the parameters obtained are as good as the default parameters. This shows that for our problem SMAC does as well as ParamILS. SMAC is able to search for optimal configurations in the continuous space of parameters, can easily handle categorical parameters and allows the flexibility to define instance specific features. Since SMAC offers a greater degree of flexibility, we use SMAC as the algorithm configurator in subsequent experiments. Experiment 4 optimizes over parameters in the continuous space. The range for the parameters are listed in table 2 and the configurator parameters are described in the previous subsection. In addition, we also use some instance specific features to better choose training instances to evaluate candidate configurations. We use 3 features namely the undersampling ratio of the given instance f, the dynamic range of the image (maximum intensity - minimum intensity) and the size of the image (32×32 for all images in current experiments). Figure 5(b) shows the result obtained by using SMAC to configure the parameters. The tuned configuration performs better in 80% of the cases resulting in an average improvement of 3%. Thus the quality of reconstruction is similar (marginally better) than the default configuration.

This lets us to believe that the author has set parameters in a conservative fashion (so as to obtain maximum reconstruction quality without much constraint on the time). Experiment 2 showed that we can obtain faster reconstructions with the same quality. This becomes important in online applications like surveillance where reconstruction speed is of essence. We explore this line of inquiry in subsequent experiments.



(a) (b)

Fig. 5. (a)Experiment 3: Scatter plot for the error obtained using the default configuration vs error obtained using the tuned configuration obtained by SMAC in the discrete parameter space (b) Experiment 4: Scatter plot for the error obtained using the default configuration vs error obtained using the tuned configuration obtained by SMAC with 3 features in the continuous parameter space.

Contrary to the previous experiments, in the next two experiments the objective function is to minimize the reconstruction time to obtain a quality better than a specified lower bound on quality. In SAT terminology a problem instance in our case is considered to be 'solved' if the quality of the reconstruction is higher than the lower bound on the quality and like in SAT, we need to find the parameter configuration which reduces the time taken to solve the problem. We define the lower bound on the quality as a fraction (P) of the optimal quality (optimal here is defined as that obtained from using the default configuration of parameters) for each problem instance in the training set. Specifically, we find the optimal quality for each image and undersampling ratio. Since the lower bound on the quality is a fraction of the optimal quality, we sacrifice on the quality to gain a speedup. The factor P essentially trades off quality vs time. For our initial experiments, we use $P = 0.98$. This was motivated by the intuition that in our applications of interest like surveillance, taking a small hit in quality is acceptable. Future work involves exploring different values for P. A case of particular interest is when $P = 1$. This corresponds to the question what is the maximum speedup we can obtain without loss in quality. However since the optimal quality is only on the training set, it is not guaranteed that we will get solutions with optimal quality on the test set.

Experiment 5 was performed for minimizing the runtime, with $P = 0.98$ and using SMAC with continuous parameters and the features described in the previous experiment. Figure 6(a) compares the reconstruction time for both the default configuration and the tuned configuration obtained from SMAC for 32×32 images. We see that all points lie well below the y = x line indicating good speedups. The average speedup for the 'best' configuration is 50% whereas the loss in quality on the test set is only 10%. Another experiment (experiment 6) we conduct is the transfer of the parameters learnt for the 32×32 image size to higher dimensions. In particular, we use the parameters learnt in the previous experiment to reconstruct images of size 64×64. Figure 6(b) shows the comparison for reconstruction times obtained using default and the tuned parameters. For the 64×64 case too, we obtain speedups of 45% with a 9% loss in quality. Hence we see that the transfer to higher dimensions is very effective. For a typical 64×64 image, the reconstruction time goes down from 6 seconds to 3 seconds. These savings in reconstruction time will become important as we scale to even higher dimensions.

Finally we show some reconstructed images(using both the default and tuned parameters) for experiment 6 in Figure 7. As we can see there is not much loss in visual quality, and applications

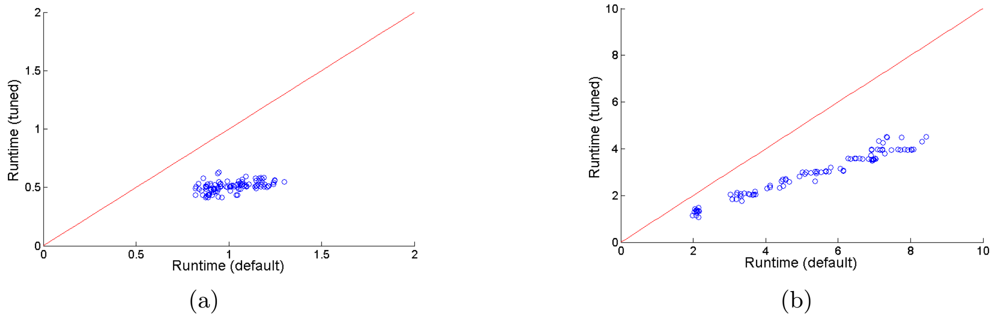


Fig. 6. (a) Experiment 5: Scatter plot for the reconstruction time for 32×32 images using the default configuration vs the tuned configuration obtained by SMAC with 3 features in continuous space (b) Experiment 6: Scatter plot for the reconstruction time for 64×64 images using the default configuration vs the configuration obtained by SMAC for 32×32 images in the previous experiment.

which do not run sophisticated image processing techniques on the reconstructed images, such a small loss in quality is acceptable.

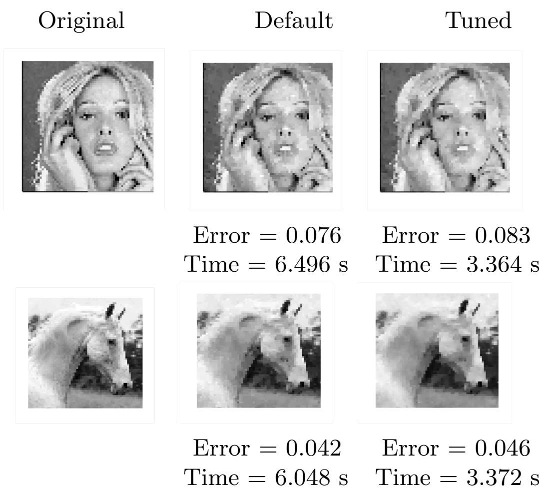


Fig. 7. Original and reconstructed 64×64 images with the $f = 0.6$. The leftmost image in each row represents the original image whereas the next two images represent the reconstructed images, the middle one using the default parameters whereas the rightmost image is obtained by using the tuned parameters for 32×32 images in experiment 5.

5 Conclusion

In this paper, we demonstrate the effectiveness of using automatic algorithm configuration techniques for a particular compressive sensing problem. In particular, we demonstrate our approach by systematically tuning 4 parameters of the TVAL3 algorithm used for image reconstruction. We observe that we can obtain better (or at least as good as) solutions (in terms of both reconstruction quality and speed) by systematic parameter tuning as compared to random configurations or those obtained by the manual hit and trial method. Also the use of automatic algorithm configuration methods is a more principled approach and increases our confidence in the results obtained especially for algorithms like in compressive sensing where the solution is very sensitive to the parameter settings. Although we demonstrate our approach for the specific case of the TVAL3 algorithm for image reconstruction in the context of Single Pixel Camera, the ideas developed in this paper can be used for algorithm configuration for other compressive sensing algorithms. For applications which desire an online real time reconstruction like those in surveillance, it is acceptable to take a hit in the quality if we can get large savings in the time taken for the reconstruction. In such cases it is best that we optimize over runtime to find the optimal parameter settings. In this paper, we can adjust the parameter P to trade-off speed and accuracy of the reconstruction. For applications like medical diagnostics in which sophisticated image processing algorithms are run on the reconstructed image, it is better to minimize the reconstruction error to find the parameter configuration which is able to obtain the best quality.

6 Future Work

We now give some directions for future work. The first direction of future work involves scaling up all the experiments conducted in this study. For example, using greater number of configuration runs with higher tuner timeouts, using larger training and testing sets. Such experiments will increase our confidence about the conclusions drawn from this paper. A direction of future work is to try and obtain higher qualities and/or faster reconstructions using richer instance specific features for SMAC. Another direction of future work involves parameter configuration and testing for larger image sizes upto 512×512. We also plan to use more principled error metrics like SNR or SSIM [22] and to expose other parameters in given in table 1. Other than these parameters, there are a number of design decisions which need to be made - the inner solver to be used for solving the x subproblem (TVAL3 uses one steepest descent), whether we can use a structured measurement matrix like Hadamard, circulant matrices [23] and obtain the same quality in a shorter time ([12] investigates the use of Hadamard matrices to speed up the reconstruction). Instead of making design decisions in the initial stage of the algorithm development process, we defer these decisions and let the automatic configurator make the decision which gives the best performance. This is inspired from the idea of Programming by Optimization, a more general software development procedure proposed by Hoos in [9].

References

1. Commercial applications of single pixel camera. `http://inviewcorp.com/`.
2. Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.
3. Alexandre Borghi, Jérôme Darbon, Sylvain Peyronnet, Tony F Chan, and Stanley Osher. A simple compressive sensing algorithm for parallel many-core architectures. *Journal of Signal Processing Systems*, pages 1–20, 2012.
4. Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.
5. Wei Dai and Olgica Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *Information Theory, IEEE Transactions on*, 55(5):2230–2249, 2009.
6. Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):83–91, 2008.
7. Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.
8. Magnus R Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.
9. Holger H Hoos. Programming by optimization. *Communications of the ACM*, 55(2):70–80, 2012.
10. Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.
11. Frank Hutter, Holger H Hoos, Kevin Leyton-Brown, and Thomas Stützle. Paramils: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306, 2009.
12. Chengbo Li. *An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing*. PhD thesis, Citeseer, 2009.
13. Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic resonance in medicine*, 58(6):1182–1195, 2007.
14. Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing mri. *Signal Processing Magazine, IEEE*, 25(2):72–82, 2008.
15. Arian Maleki and David L Donoho. Optimally tuned iterative reconstruction algorithms for compressed sensing. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):330–341, 2010.
16. Deanna Needell and Joel A Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.
17. Michael JD Powell. *" A method for non-linear constraints in minimization problems"*. UKAEA, 1967.
18. Fabrizio Russo. Technique for image denoising based on adaptive piecewise linear filters and automatic parameter tuning. *Instrumentation and Measurement, IEEE Transactions on*, 55(4):1362–1367, 2006.
19. David S Smith, John C Gore, Thomas E Yankeelov, and E Brian Welch. Real-time compressive sensing mri reconstruction using gpu computing and split bregman methods. *International Journal of Biomedical Imaging*, 2012, 2012.
20. Dharmpal Takhar, Jason N Laska, Michael B Wakin, Marco F Duarte, Dror Baron, Shriram Sarvotham, Kevin F Kelly, and Richard G Baraniuk. A new compressive imaging camera architecture using optical-domain compression. In *Electronic Imaging 2006*, pages 606509–606509. International Society for Optics and Photonics, 2006.
21. Curtis R Vogel and Mary E Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.
22. Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

23. Wotao Yin, Simon Morgan, Junfeng Yang, and Yin Zhang. Practical compressive sensing with toeplitz and circulant matrices. In *Visual Communications and Image Processing 2010*, pages 77440K–77440K. International Society for Optics and Photonics, 2010.
24. Justin Ziniel and Philip Schniter. Efficient message passing-based inference in the multiple measurement vector problem. In *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pages 1447–1451. IEEE, 2011.