ConvexVST: A Convex Optimization Approach to Variance-stabilizing Transformation

Mengfan Wang [1] **Boyu Lyu** [1] **Guoqiang Yu** [1]

Abstract

The variance-stabilizing transformation (VST) problem is to transform heteroscedastic data to homoscedastic data so that they are more tractable for subsequent analysis. However, most of the existing approaches focus on finding an analytical solution for a certain parametric distribution, which severely limits the applications, because simple distributions cannot faithfully describe the real data while more complicated distributions cannot be analytically solved. In this paper, we converted the VST problem into a convex optimization problem, which can always be efficiently solved, identified the specific structure of the convex problem, which further improved the efficiency of the proposed algorithm, and showed that any finite discrete distributions and the discretized version of any continuous distributions from real data can be variance-stabilized in an easy and nonparametric way. We demonstrated the new approach on bioimaging data and achieved superior performance compared to peer algorithms in terms of not only the variance homoscedasticity but also the impact on subsequent analysis such as denoising. Source codes are available at https://github.com/yu-lab-vt/ConvexVST.

1. Introduction

The variance-stabilizing transformation (VST) problem is to find a transformation function, which can transform a family of heteroscedastic random variables (r.v.) to homoscedastic r.v. In mathematical terms, it is to find a function f for a family of r.v. X_θ indexed by the parameter θ so that $Var[f(X_\theta)]$ or $MSE[f(X_\theta)]$ are functionally independent with θ.

As a fundamental preprocessing step for data analysis, the VST problem has been the subject of extensive theoretical and experimental studies because of its broad applications. In numerous papers, VST, making the resulting distribution more homoscedastic, not only eases but also improves the subsequent analysis from the basic logistic regression (Yan & Su, 2009) to advanced neural networks (Zhang et al., 2019a). Without VST, people have to either model the different variances for different observations, which is often difficult and sometimes impossible, or assume wrongly a single variance for all observations. Indeed, VST has been applied to a wide variety of different fields, such as image denoising (Azzari & Foi, 2016), machine learning (Zhang et al., 2019a), microarray data calibration (Huber et al., 2002), traffic control (Guo et al., 2012), managerial accounting (Ittner & Larcker, 2001), and clinical trials (Enright et al., 2003).

The most original and classical variance stabilizer (Tippett, 1935)

$$f(z) = \int^z \frac{1}{\sqrt{Var[X_\theta]}} dE[X_\theta] \tag{1}$$

was proposed to transform an arbitrary family of basic probability distributions and provide good asymptotic properties. After that, most of the contributions focused on finding a closed-form expression for a particular distribution, for example, the Anscombe transformation $f : z \to 2\sqrt{z + 3/8}$, which can stabilize the variance of Poisson distribution approaching 1 when z increasing (Anscombe, 1948).

These theoretical researches did not stabilize variance very well in real applications. We reasoned that the less satisfactory performance is due to the following two facts. Firstly, simple parametric transformation functions cannot match well with the real data, while more complicated models are hard to be analytically solved. Secondly, almost all works aimed at achieving the stabilization in the asymptotic sense, which can only guarantee good performance when the variance is small compared to its mean. Thus, some researchers turned to develop data-driven algorithms and tried to find numerical solutions by optimization approach. We mention the AVAS (Tibshirani, 1988) as the most classical one, which is further improved in recent years by (Foi, 2009). Both of them refine the VST functions by calculating an integral formula like Equation 1 iteratively, hoping for better

[1]Bradley Department of Electrical and Computer Engineering, Virginia Tech, VA, USA. Correspondence to: Guoqiang Yu <yug@vt.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

and better results. However, their models led to challenging nonconvex optimization problems, whose solutions are difficult to find and convergence cannot be guaranteed.

This paper proposes an optimization approach to solve the VST problem as well, but it is the first time that the VST problem is converted into a convex optimization problem, which can guarantee the convergence and be solved efficiently. What's more, we took advantage of the special structure of the resultant optimization problem and showed that the problem can be solved much efficiently in the time complexity of $O(nm)$ instead of the off-the-shelf convex solvers of $O(n^3m^3)$, where n is the number of distributions and m is the number of elements in the sample space. Extensive experiments on both synthetic and real datasets under various conditions confirmed the superiority of our proposed approach compared to peer methods.

2. Methods

2.1. Problem Formulation

The VST problem is to find a nonlinear monotonic transformation function f, which can transform the r.v. X_θ to a new one $Y_\theta = f(X_\theta)$ such that Y_θ has the constant variance $Var[Y_\theta] = c$ or the constant MSE $MSE[Y_\theta] = c$. To rigorously formulate the problem, we have the following definitions.

Definition 1 – heteroscedastic random variable. The random variable in this paper is from a family of heteroscedastic r.v. composed of a series of true signals $\theta \in \Theta$ and noise distributions N_θ:

$$X_\theta = \theta + N_\theta, \tag{2}$$

where N_θ is dependent with θ and so does $Var[X_\theta]$ or $MSE[X_\theta]$. Θ is the parameter to index the distribution. For example, for Poisson distribution, we have $E[X_\theta] = Var[X_\theta] = \theta$, where $\Theta = [0, \infty) \subset \mathbb{Z}$.

Definition 2 – finite discrete distributions. A family of distributions whose choices of the parameter Θ are discrete and finite, and for a certain parameter θ, the sample space $\mathcal{S}_x = \{x_0, x_1, \ldots, x_{m-1}\} \subset \mathbb{R}$ are also discrete and finite with a probability $Pr[X = x_k|\Theta = \theta] = p_{\theta k}$.

Our algorithm is developed for finite discrete distributions because of the property of data from digital systems. However, this is not a limitation for practical applications. For theoretical continuous or infinite discrete distributions, such as Poisson distribution or multiplicative normal distribution (Foi, 2009), an approximate solution can also be achieved by discretization or manually clipping.

Definition 3 - transform function. $f : \mathcal{S}_x \to \mathcal{S}_y$ is a valid transform function if: (1) it is a bijection; (2) for all $x_1 < x_2$, we have $y_1 < y_2$ where $f(x_1) = y_1$ and $f(x_2) = y_2$.

Definition 4 – variance. The expected squared deviation from the mean to a r.v.

$$Var[X_\theta] = E[(X_\theta - E[X_\theta])^2] \tag{3}$$

Definition 5 – mean squared error. The expected squared deviation from the true signal to a r.v.

$$MSE[X_\theta] = E[(X_\theta - \theta)^2] \tag{4}$$

The VST problem was proposed to stabilize the variance theoretically, but people may concern more with how much a r.v. differs from its signal in real applications. For symmetric distributions, $E[X_\theta] = \theta$ so that the two metrics basically equal. When distributions are clipped, such as the clipped Poissonian-Gaussian model in Section 3, the two metrics could be quite different. Our optimization algorithm can be applied to both of the two metrics. However, for more reasonable results and clearer interpretation, we only stabilize the MSE in the remaining part and the algorithm stabilizing the variance is explained in the supplementary.

2.2. Algorithm

We introduce the cost function proposed by (Foi, 2009) to measure the performance of a transformation f:

$$C_f = \sum_\theta |\sigma^2_{\theta,f} - c| \tag{5}$$

where $\sigma^2_{\theta,f}$ is the MSE of $f(X_\theta)$:

$$\sigma^2_{\theta,f} = E\{[f(X_\theta) - f(\theta)]^2\} \tag{6}$$

It is hard to minimize the highly non-convex cost function directly. As a substitution, we turn to minimize:

$$C'_f = \max_\theta \sigma^2_{\theta,f}, \tag{7}$$

Lemma 1. *C'_f is always convex.*

This substitution is the most critical idea that enables our new method. Before we provide the motivation behind, we need to introduce several new notations. The samples in $\mathcal{S}_x$ can be sorted in ascending order so that $x_0 < x_1 < \cdots < x_{m-1}$. After transformation, the "distances" between adjacent samples are changed. Let us use new variables $\Delta\mathbf{x}$ to represent their distances: $\Delta x_k = f(x_k) - f(x_{k-1})$. Clearly, with the starting point x_0 fixed there is a one-to-one correspondence between $\Delta\mathbf{x}$ and $f(x)$. Thus, we can directly optimize $\Delta\mathbf{x}$ instead of the transform function $f(x)$. On the one hand, when minimize C'_f or the maximum $\sigma^2_{\theta,f}$, we have to shrink some related variables in $\Delta\mathbf{x}$ because they are positively correlated to $\sigma^2_{\theta,f}$. On the other hand,

because the sum of all variables ($\Delta\mathbf{x}^T\mathbf{1}$) is fixed, the remaining $\Delta\mathbf{x}$ must be enlarged to satisfy the constraint, and the variances of other distributions will increase. As a result, large variances decrease and small variances increase even though we only tried to explicitly decrease the maximum one. That is, even though we want to minimize C'_f only, C_f also decreased because of small variances increasing. In Section 3, we experimentally showed that minimizing C'_f can decrease C_f effectively.

Lemma 2. *C'_f can be represented in the matrix form $C'_f = \max_\theta \Delta\mathbf{x}^T\mathbf{Q}_\theta\Delta\mathbf{x}$, where $\Delta\mathbf{x} = [\Delta x_1 \quad \Delta x_2 \dots \Delta x_{m-1}]^T$. $\mathbf{Q}_\theta$ is a positive definite matrix with a special Cholesky decomposition, whose inverse can be derived directly.*

There are still some restrictions to guarantee good results. Firstly, $\Delta\mathbf{x}$ should be positive to keep f monotonically increasing. What's more, the sum of $\Delta\mathbf{x}$, or the total length of $\mathcal{S}_x$ should not be changed after stabilization. Otherwise we can set $\Delta\mathbf{x}$ close to $\mathbf{0}$ to minimize C'_f. In conclusion, we formulate an optimization problem:

$$\min_{\Delta\mathbf{x}} \max_{\theta} \quad \Delta\mathbf{x}^T\mathbf{Q}_\theta\Delta\mathbf{x} \tag{8}$$

$$s.j. \quad \Delta\mathbf{x} > \mathbf{0} \tag{9}$$

$$\Delta\mathbf{x}^T\mathbf{1} = x_{m-1} - x_0 \tag{10}$$

Lemma 3. *This convex optimization problem can be converted into the second-order cone programming (SOCP) problem and solved in $\mathcal{O}(nm)$ time complexity operations, where n is the size of Θ and m is the size of $\mathcal{S}_x$.*

Mature optimizers like MOSEK (Andersen et al., 2003) can solve an arbitrary SOCP problem in $\mathcal{O}(n^3m^3)$ time complexity. We decrease the time complexity to $\mathcal{O}(nm)$ and could get the global optimum $\Delta\mathbf{x}^*$ efficiently because of the special structure of our problem mentioned in Lemma 2.

Solving the optimization problem is the first part of the algorithm. After that, we have a preliminary solution $f_0 : x_k \to \sum_{j=1}^{k}\Delta x_j^* + x_0$. And the corresponding MSEs are $\sigma^2_{\theta,f_0} = \Delta\mathbf{x}^{*T}\mathbf{Q}_\theta\Delta\mathbf{x}^*$. For most of the cases, this part can provide good-enough results. However, the objective function only constrains the maximum MSE, which may ignore some extremely small MSEs. To compensate for this drawback, we propose the second part of the algorithm.

In the first part, we constrain all MSEs $\sigma^2_{\theta,f} \le \max_\theta \sigma^2_{\theta,f_0} = c$. In the second part, we hope all MSEs can be as close to c as possible:

$$\min_f \max_\theta |\sigma^2_{\theta,f} - c| \tag{11}$$

This objective function is still highly non-convex, but the local optimum can be achieved by the majorize-minimization (MM) algorithm. We can rewrite the function as:

$$\min_f \max_\theta \{\sigma^2_{\theta,f} - c, c - \sigma^2_{\theta,f}\} \tag{12}$$

For any θ, $\sigma^2_{\theta,f} - c$ is always convex but $c - \sigma^2_{\theta,f}$ is concave. Assume we have the initialization $\Delta\mathbf{x}_0$, the first-order Taylor expansion of $c - \sigma^2_{\theta,f}$ at $\Delta\mathbf{x}_0$ will be:

$$c - \sigma^2_{\theta,f} \le -2\Delta\mathbf{x}_0^T\mathbf{Q}_\theta\Delta\mathbf{x} + \Delta\mathbf{x}_0^T\mathbf{Q}_\theta\Delta\mathbf{x}_0 + c \tag{13}$$

Under the same constraints, the final optimization problem of the second part is:

$$\begin{aligned}\min_{\Delta\mathbf{x}} \max_{\theta} \quad & \{\Delta\mathbf{x}^T\mathbf{Q}_\theta\Delta\mathbf{x} - c, \\ & -2\Delta\mathbf{x}_0^T\mathbf{Q}_\theta\Delta\mathbf{x} + \Delta\mathbf{x}_0^T\mathbf{Q}_\theta\Delta\mathbf{x}_0 + c\}\end{aligned} \tag{14}$$

$$s.j. \quad \Delta\mathbf{x} > \mathbf{0} \tag{15}$$

$$\Delta\mathbf{x}^T\mathbf{1} = x_{m-1} - x_0 \tag{16}$$

Lemma 4. *The solution of Problem 14 will always converge to a local minimum of Function 11 when we solve it iteratively. Similar to the first part, it can also be converted into a SOCP problem with the same time complexity.*

The second part is optional which has the potential to overfit the distribution when N_θ is inaccurately estimated from data.

3. Applications and Experiments

In this section, we introduce the noise model of imaging data (Section 3.1) and two peer methods (Section 3.2), and then compare the three methods on various types of data. Firstly we variance-stabilized the theoretical clipped Poissonian-Gaussian model directly with the assumption that the distributions are known (Section 3.3), which theoretically favors peer methods. We also simulated synthetic dynamic bioimaging data with different noise levels and tested the performance (Section 3.4). Last but not least, we stabilized the variances of noise on a public bioimage dataset with static images (Section 3.5), and investigated the influence of stabilization on denoising performance based on the BM3D algorithm (Section 3.6). For all experiments, we compared our method to two peer methods.

3.1. Clipped Poissonian-Gaussian Model

Clipped Poissonian-Gaussian distribution is a mathematical model to describe the noise distribution of raw imaging data (Foi et al., 2008). For digital imaging sensors, the basic noise model is the Poissonian-Gaussian model. It is a signal-dependent model consisting of two independent parts, a Poissonian part P_θ modeling the photo-sensing and a Gaussian part G modeling the remaining signal-independent noise in the camera system:

Table 1. Mean Absolute Error and Maximum Error on Theoretical Model

Algorithm	Mean Absolute Error				Maximum Error			
	ConvexVST	GAT	Foi	ConvexVST (open interval)	ConvexVST	GAT	Foi	ConvexVST (open interval)
$\chi = 5$	**1.504**	44.81	7.709	*0.379*	190.9	377.4	**184.2**	*0.876*
$\chi = 10$	**2.695**	115.5	18.98	*0.773*	355.4	728.0	**313.3**	*1.633*
$\chi = 15$	**3.376**	200.2	35.05	*0.756*	517.3	1211	**463.4**	*1.978*

$$N_\theta = P_\theta + G, \tag{17}$$

where $P_\theta + \theta \sim \mathcal{P}(\chi\theta)$ and $G \sim \mathcal{N}(0, \sigma_G)$. $\chi > 0$ and $\sigma_G \geq 0$ are real scalar parameters, and $\mathcal{P}$ and $\mathcal{N}$ are the Poisson and Gaussian distributions. Without the influence of clipping, the variance and MSE of Poissonian-Gaussian model is $Var[N_\theta] = MSE[N_\theta] = \chi\theta + \sigma_G^2$.

However, imaging data can only be represented in a range. For example, the values of the 8-bit image are between 0 to 255. If an observation is out of range, it will be over-exposed to 255 or under-exposed to 0. We assume the data range is $[a, b]$, then the clipped Poissonian-Gaussian noise is:

$$\tilde{N}_\theta = \begin{cases} b - \theta & N_\theta > b - \theta \\ N_\theta & a - \theta \leq N_\theta \leq b - \theta \\ a - \theta & N_\theta < a - \theta \end{cases} \tag{18}$$

And the clipped Poissonian-Gaussian model is:

$$X_\theta = \theta + \tilde{N}_\theta \tag{19}$$

X_θ has a finite discrete distribution.

3.2. Peer Methods

We introduce the two peer methods before the experiments. The first one is the generalized Anscombe transformation (GAT) (Starck et al., 1998). Compared to the Anscombe transformation, it takes the Gaussian part into account:

$$f_{GAT} : z \to \frac{2}{\chi}\sqrt{\chi z + \frac{3}{8}\chi^2 + \sigma_G^2} \tag{20}$$

f_{GAT} can stabilize the variance of Poissonian-Gaussian distribution to approximately 1 when z approaches infinite, which is a stabilizer with the most extensive applications up to now. However, it cannot deal with the clipping problem.

Few state-of-art methods are focusing on arbitrary distributions variance stabilization or specific for the clipped model. We choose the most appropriate one, a recursive optimization approach (Foi, 2009). The basic idea is to select a constant variance c and refine the VST function iteratively so that the variance curve will more and more close to c.

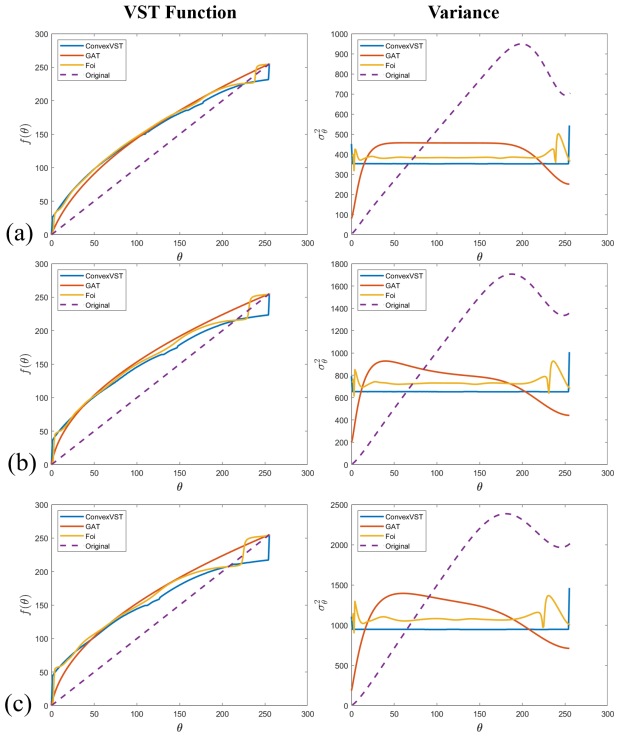

Figure 1. The experiment result on the theoretical models. (a) $\chi = 5$. (b) $\chi = 10$. (c) $\chi = 15$. The first row shows VST functions of all methods, where the dashed lines are the function $f : z \to z$. The second row shows the stabilized variance curves, where the dashed lines are the original curves.

The author did not provide the source code and we reimplemented it. All parameters are suggested by the author except c. In the original paper, the author chose $c = 1$ when $\theta \in [0, 1]$. We found c had the most significant influence and not all distributions could be variance-stabilized when $c = 1$. For the best performance, we set c equals to a theoretically optimal stabilized-variance.

In the following parts, figures, and tables, we use "ConvexVST", "GAT" and "Foi" to represent our method, GAT, and the recursive optimization approach.

Table 2. Mean Absolute Error and Maximum Error on Synthetic Dynamic Data

	Mean Absolute Error				Maximum Error			
Algorithm	ConvexVST	GAT	Foi	ConvexVST (open interval)	ConvexVST	GAT	Foi	ConvexVST (open interval)
$\chi = 5$	**10.72**	40.42	73.65	*8.715*	**216.0**	366.3	452.6	*46.51*
$\chi = 10$	**11.01**	111.2	131.7	*7.437*	**409.5**	731.1	724.3	*79.84*
$\chi = 15$	**20.84**	195.8	216.8	*15.46*	**601.2**	1224	1153	*127.2*

3.3. Stabilization Results on Clipped Poissonian-Gaussian Model

In the experiment, we assume $\theta \in [0, 255] \subset \mathbb{Z}$, and all observations are clipped between $[0, 255]$. The variance of the Gaussian part is $\sigma_G^2 = 20$, and the parameter χ is 5, 10, and 15 to simulate real noise distributions.

We used both the first part and the second part of ConvexVST to achieve the best result, while even only the first part is also much better than peer methods. The second part is iterated 5 times to converge. In the experiment, we realized that it was almost impossible to stabilize variances for all θ. To improve the overall performance, we release the constraint on the extreme value in Θ. Thus, we minimize $\max_{\theta \in [1,254]} \sigma_{\theta,f}^2$ without the consideration of $\theta = 0$ or 255. For Foi's method, we found it cannot converge probably because of its highly non-convexity and iterated it 150 times. With more iterations, the VST functions have a stair-like shape, where the steps are likely due to overfitting and the performance is decreased.

Figure 1 shows the VST functions and stabilized variances. Even the three methods take different strategies, the corresponding VST functions have similarities to some extent. Foi's method can stabilize the variances but there are still some fluctuations. The performance of GAT depends on the clipping effect. When χ is bigger, the noise is stronger, and the clipping effect is larger, GAT performs worse. ConvexVST performs better than the other two with almost flat variance curves except for the ends.

Table 1 shows the quantitative results. The end variances ($\theta = 0$ or 255) have a huge impact on ConvexVST, whose performance tends to be underestimated for applications whose signals of interest does not contain them. As a reference for applications with a local region of interest, the quantitative evaluation of ConvexVST without the consideration of end variances is also shown in the table and labeled as "ConvexVST(open interval)", the same as the three methods. The best results of the three methods are in bold in all tables. ConvexVST(open interval) is in italics and not compared. Indeed, it always has much better performance than peer methods without including the end variances.

As Equation 5, we measure the mean absolute error (MAE)

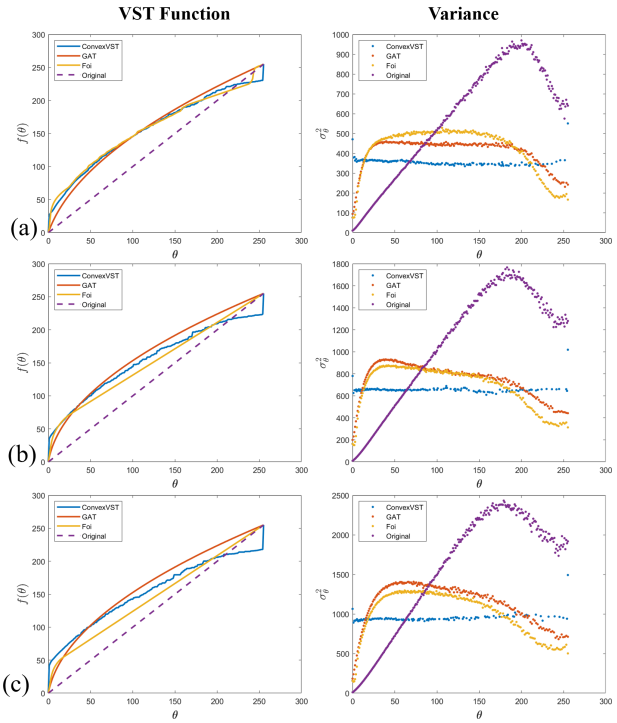

Figure 2. The experiment result on the synthetic dynamic data. (a) χ = 5. (b) χ = 10. (c) χ = 15. The first row shows the VST functions of all methods, and the second row shows the stabilized variances.

$MAE_f = \frac{1}{n}\sum_{\theta=0}^{n-1} |\sigma_{\theta,f}^2 - c|$ to a constant c firstly. We choose $c = Med(\sigma_{\theta,f}^2)$ because the median is the optimal solution to minimize MAE (Poor, 2013). ConvexVST has the smallest MAE on all models.

To compare the performance of these methods in the extreme case rather than the average performance, we also proposed the maximum error (ME) $ME_f = \max_\theta(\sigma_{\theta,f}^2) - \min_\theta(\sigma_{\theta,f}^2)$ as a metric. Foi's method is slightly better because our method releases the constraints at the ends to improve the average performance.

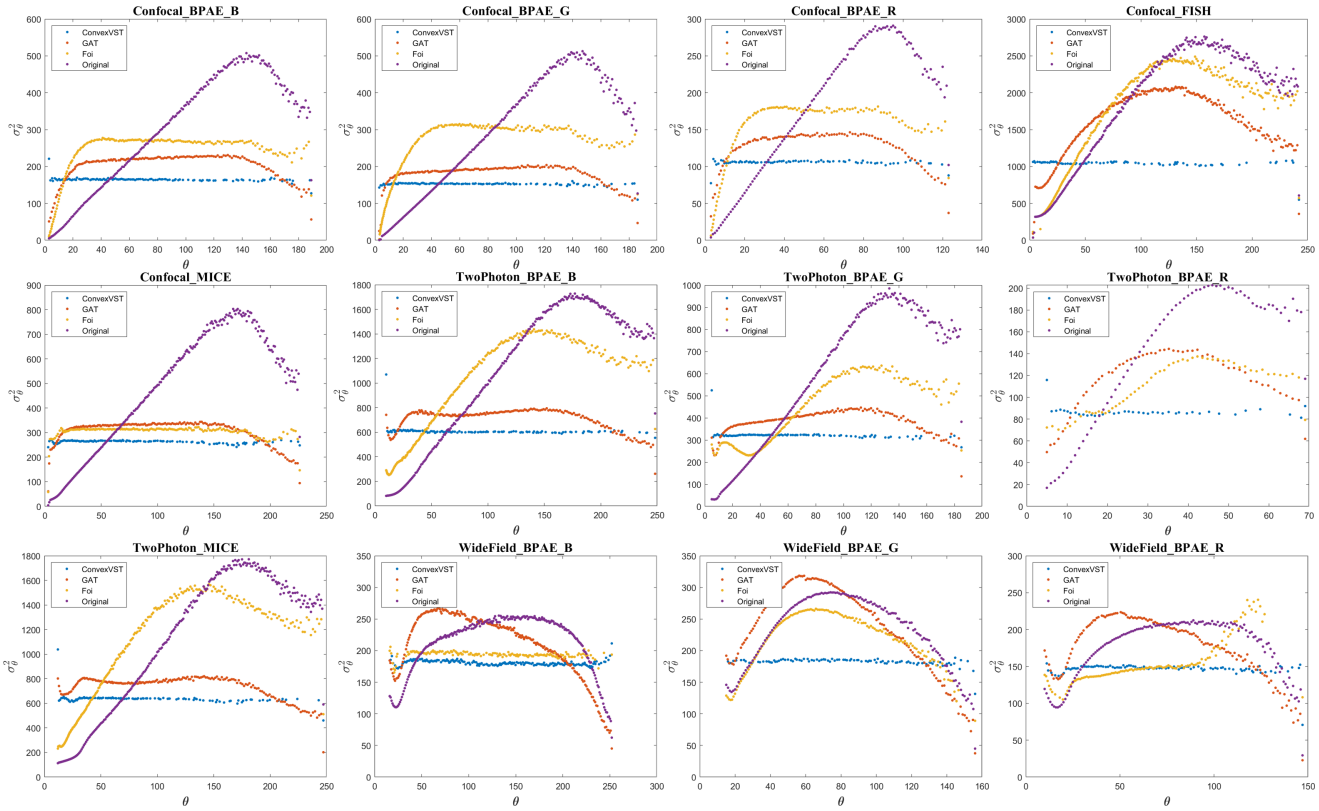

Figure 3. The experiment on the real static imaging data. Every subfigure plots the original and stabilized variances of a subset.

3.4. Stabilization Results on Synthetic Dynamic Imaging Data

In Experiment 3.4 and 3.5, we will show the performance of these methods on both synthetic and real bioimaging data. For data-driven approaches, the noise model estimation is necessary to acquire enough information. Static images without any activity, such as the part of a video before the experimental manipulation, are excellent materials to estimate noise models for variance stabilization. However, we should not expect data containing static images all the time. Therefore, we consider the worst case that no static image can be utilized and all images are dynamic firstly, where every signal only generates one observation. It is impossible to evaluate the performance on real dynamic imaging data because of the lack of ground truth. Instead, we did simulation based on our in-house 4D two-photon fluorescence imaging data measuring the motility of microglia in the mouse brain. The data resolution is $512 \times 512 \times 61$ and there are 56 frames. The data were denoised by 3×3 median filters to generate the ground truth, and then added three levels of noise the same as Experiment 3.3. The training set and the test set are the same, but we learned the model from the data directly and tested the performance with the ground truth to avoid overfitting.

Figure 2 shows the VST functions and stabilized variances. GAT and Foi's method can stabilize variances to some degree with better performances under smaller noise levels. ConvexVST is still the best method, which performs as good as on the static data. As before, Table 2 shows the MAE and ME on the synthetic dynamic data. In this experiment, ConvexVST has the best performance under any noise level and any metric. It shows our method is more stable than peer methods when prior knowledge is unknown.

3.5. Stabilization Results on Public Bioimaging Dataset

In this experiment, we utilized the Fluorescence Microscopy Denoising (FMD) dataset (Zhang et al., 2019b) dedicated to Poissonian-Gaussian denoising to evaluate the methods. The dataset consists of 12,000 real fluorescence microscopy images obtained with commercial confocal, two-photon, and wide-field microscopes and representative biological samples such as cells, zebrafish, and mouse brain tissues. There are 12 subsets containing a certain kind of samples taken by one microscope. Every subset contains 20 fields of view (FOV), and every FOV was taken 50 times. The resolution of all images is 512×512. All images are 8-bit

Table 3. Mean Absolute Error and Maximum Error on Real Static Data (the last column is open interval)

Algorithm	Mean Absolute Error				Maximum Error			
	ConvexVST	GAT	Foi	ConvexVST	ConvexVST	GAT	Foi	ConvexVST
Confocal_BPAE_B	**2.265**	23.02	36.28	*1.516*	**93.66**	180.0	267.0	*16.14*
Confocal_BPAE_G	**1.862**	16.04	60.86	*1.421*	**50.36**	178.8	314.2	*16.13*
Confocal_BPAE_R	**1.727**	16.01	28.94	*1.158*	**32.78**	114.0	175.1	*8.412*
Confocal_FISH	**19.34**	332.0	523.5	*13.27*	**530.6**	1991	2377	*71.21*
Confocal_MICE	**4.022**	27.10	10.05	*3.759*	**30.45**	283.0	264.4	*29.51*
TwoPhoton_BPAE_B	**9.277**	53.80	348.4	*5.068*	**515.2**	532.8	1186	*31.69*
TwoPhoton_BPAE_G	**5.546**	37.04	136.8	*2.952*	**257.5**	310.8	405.7	*23.08*
TwoPhoton_BPAE_R	**2.070**	20.33	19.23	*1.251*	**34.59**	94.61	67.03	*7.771*
TwoPhoton_MICE	**12.87**	54.20	382.3	*7.986*	**577.6**	618.7	1336	*54.90*
WideField_BPAE_B	3.684	38.67	**3.123**	*2.804*	146.8	222.2	**40.54**	*20.19*
WideField_BPAE_G	**4.044**	56.13	35.92	*2.277*	181.2	281.0	**177.2**	*20.91*
WideField_BPAE_R	**4.066**	31.75	20.45	*2.211*	189.9	200.8	**135.2**	*16.98*

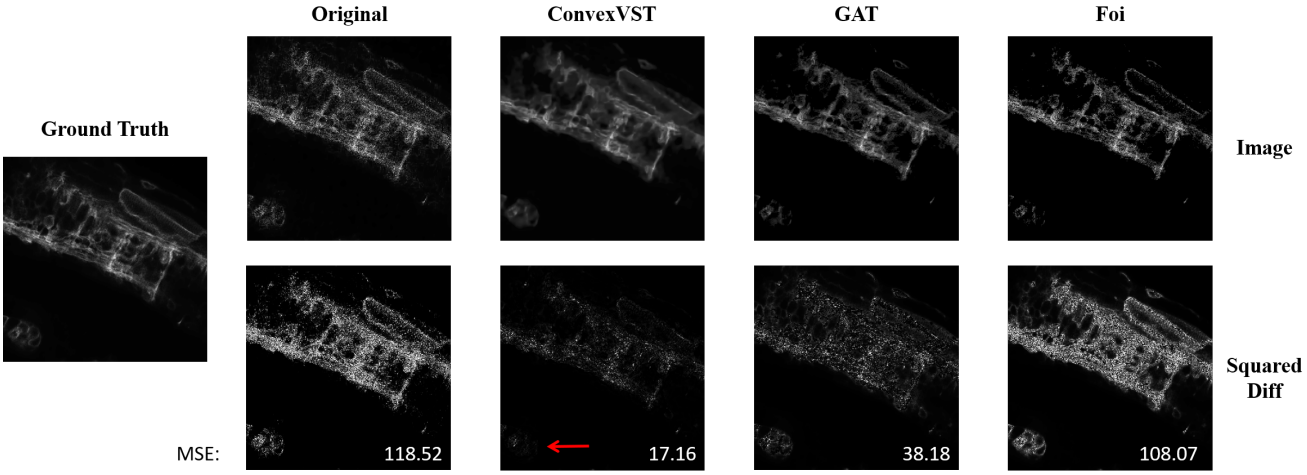

Figure 4. A visual comparison of the denoising task. The first row shows the original and denoised images, while the second row shows the squared differences.

between 0 to 255.

We tested all 12 subsets separately. For each subset, we allocated half of the images (25 images of all FOVs) as the training set and half of them as the test set. The training set was used to estimate the distribution and generate VST functions, and we evaluated the performance on the test set.

Figure 3 shows the stabilized variances of each subset. For confocal and two-photon data, the noise follows the clipped Poissonian-Gaussian distribution, whose original variance curves are similar to Figure 1. However, for wide-field data, the noise distribution is quite different on account of unclear reasons. ConvexVST performs well on all the datasets. On most of the confocal and two-photon data, GAT can stabilize variances but still performs worse than ConvexVST. The performance decreases greatly when data don't follow the assumptions, such as the wide-field data or "Confocal_FISH", where the latter has strong noise and is clipped severely. Foi's method has extremely unstable performance on the dataset. On one hand, on some subsets like "Confocal_MICE" and "WideField_BPAE_B", it is comparable to ConvexVST and even better; on the other hand, on some others like "Confocal_FISH" and "TwoPhoton_MICE", it cannot stabilize the variances at all.

We also analyzed the results quantitatively. Table 3 shows the MAE and ME on each subset. ConvexVST beats the other two on 11 of 12 subsets with much better MAEs, and has the best performance on all confocal and two-photon subsets under the evaluation on ME even without excluding the end variances.

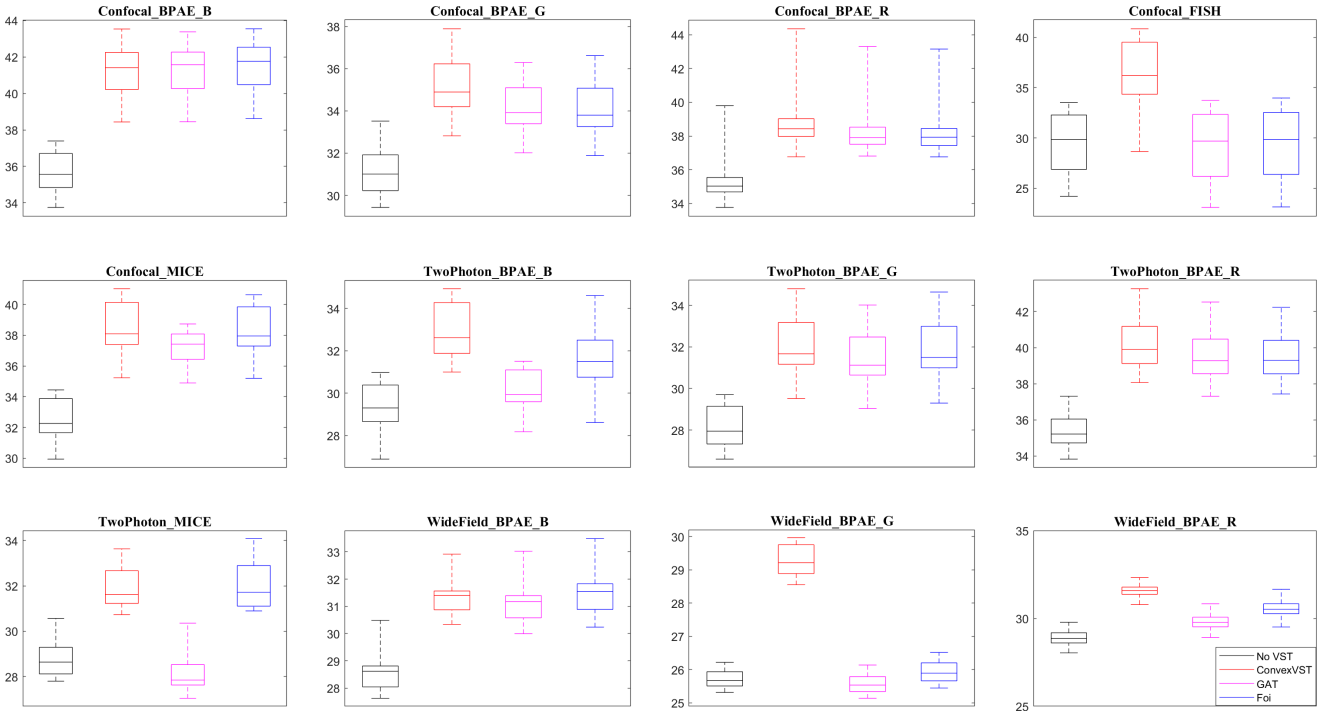

Figure 5. PSNR of all groups of each subset. Every subfigure plots the PSNR of images stabilized by three VST methods with the comparison to the directly denoised image.

3.6. Denoising performance on Public Bioimaging Dataset

In the last experiment, we investigate whether our approach contributes to practical problems. The most popular application of VST is denoising, where VST can make the image noise distribution more homoscedastic, change the clipped Poissonian-Gaussian distributions to Gaussian-like distributions so that the denoising algorithm can simply assume the noise follows the Gaussian distribution.

As the most effective one with the assumption of additive i.i.d. Gaussian noise, we utilize BM3D (Dabov et al., 2007) as the denoising algorithm in this part. BM3D combines the idea from non-local means (Buades et al., 2005) and wavelet-based denoising methods, groups similar blocks in a 3D-array, collaborative filters these blocks in the transform domain, and then gives a final estimation by aggregating all local estimates.

The division of the training and test sets are the same as Experiment 3.5, and so are other configurations and parameters. The dataset is variance-stabilized by the three approaches, denoised by BM3D, and then transformed inversely to recover original signals. One more group without stabilization and directly denoised is set as a control group.

The image quality after denoising is measured by PSNR (peak signal-to-noise ratio). PSNR is arguably the most popular measure on the image quality in the field of denoising. Figure 5 shows the box plot of PSNR of groups with or without stabilization. Every subfigure represents the PSNRs of the foreground region of all FOVs in one subset. All VST methods contribute to improving the denoising performance with higher PSNRs in most of the cases, but a transformation can also have a negative impact occasionally if not good enough. The stabilized variance curves are highly correlated to the denoising performance, but not always.

Our method has the best denoising performance in 9 of all 12 subsets, and is comparable with the better of the peer methods in the remaining three subsets. Table 4 shows the overall PSNR comparison on the whole dataset. Compared to the better one of the peer methods, our approach helps to improve PSNR by almost 2dB. It is worth noting that an improvement of 2 dB is a significant number in the field of denoising.

Table 4. Overall PSNR after denoising

No VST	ConvexVST	GAT	Foi
29.5768	33.3161	30.5624	31.3839

A visual comparison of the denoising task is also presented

in Figure 4. The first row shows the original and denoised images from the Confocal_FISH subset, while the second row shows the squared differences between the first row and the ground truth. The differences were rescaled for visualization purpose. Corresponding MSEs are shown at the bottom. These results show that our method outperforms peer methods not only quantitatively but also qualitatively. Take the bottom corner indicated by the red arrow as an example, our method results in a much smaller difference in this region than other methods.

4. Conclusion

This paper proposed the first convex optimization-based VST algorithm, which aims at unifying the variance or MSE of data and making them more consistent to improve the subsequent analysis. It is more stable and flexible compared to previous works, which guarantees to converge and can stabilize the variance or MSE of any non-analytical distributions from real data.

The key idea is to minimize the maximum of variances of all θ. When the maximum variance is decreased, other variances will tend to increase because the resultant range of the transformed value is constrained to be the same as before the transformation. In this way, all variances will move close to each other and exhibit a stabilized curve.

We tested the algorithm on theoretical distributions and real bioimaging data, and compared them to two peer methods in four different experiments. The results are analyzed qualitatively and quantitatively. Our algorithm achieved much superior average performance than peer methods in all experiments and achieved better extreme performance in most of the real and synthetic data experiments. The denoising experiment exhibited great potential of our approach in practical applications.

As a general data transformation approach, the algorithm can be applied to a broad spectrum of problems and be a preprocessing step before training any models on heteroscedastic data, which has the potential to be as important as the Z-score normalization and Box-Cox transformation (Sakia, 1992). However, it is also limited to reliable estimation of the distribution and requires large scale of data. Further more, discretization is necessary for continuous distributions, which is worth investigating for joint optimal discretization and variance stabilization in future studies.

Acknowledgements

We thank the anonymous reviewers for their helpful suggestions. This work was supported by grants NIH R01MH110504 and NSF 1750931.

References

Andersen, E. D., Roos, C., and Terlaky, T. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(2):249–277, 2003.

Anscombe, F. J. The transformation of poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254, 1948.

Azzari, L. and Foi, A. Variance stabilization for noisy+ estimate combination in iterative poisson denoising. *IEEE signal processing letters*, 23(8):1086–1090, 2016.

Buades, A., Coll, B., and Morel, J.-M. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8): 2080–2095, 2007.

Enright, P. L., McBurnie, M. A., Bittner, V., Tracy, R. P., McNamara, R., Arnold, A., Newman, A. B., et al. The 6-min walk test: a quick measure of functional status in elderly adults. *Chest*, 123(2):387–398, 2003.

Foi, A. Optimization of variance-stabilizing transformations. *Preprint, 2009b*, 94:1809–1814, 2009.

Foi, A., Trimeche, M., Katkovnik, V., and Egiazarian, K. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

Guo, J., Huang, W., and Williams, B. M. Integrated heteroscedasticity test for vehicular traffic condition series. *Journal of transportation engineering*, 138(9):1161–1170, 2012.

Huber, W., Von Heydebreck, A., Sültmann, H., Poustka, A., and Vingron, M. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18(suppl_1):S96–S104, 2002.

Ittner, C. D. and Larcker, D. F. Assessing empirical research in managerial accounting: a value-based management perspective. *Journal of accounting and economics*, 32 (1-3):349–410, 2001.

Poor, H. V. *An introduction to signal detection and estimation*. Springer Science & Business Media, 2013.

Sakia, R. M. The box-cox transformation technique: a review. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 41(2):169–178, 1992.

Starck, J.-L., Murtagh, F. D., and Bijaoui, A. *Image processing and data analysis: the multiscale approach*. Cambridge University Press, 1998.

Tibshirani, R. Estimating transformations for regression via additivity and variance stabilization. *Journal of the American Statistical Association*, 83(402):394–405, 1988.

Tippett, L. 2—statistical methods in textile research. part 2—uses of the binomial and poisson distributions. *Journal of the Textile Institute Transactions*, 26(1):T13–T50, 1935.

Yan, X. and Su, X. *Linear regression analysis: theory and computing*. World Scientific, 2009.

Zhang, M., Zhang, F., Liu, Q., and Wang, S. Vst-net: variance-stabilizing transformation inspired network for poisson denoising. *Journal of Visual Communication and Image Representation*, 62:12–22, 2019a.

Zhang, Y., Zhu, Y., Nichols, E., Wang, Q., Zhang, S., Smith, C., and Howard, S. A poisson-gaussian denoising dataset with real fluorescence microscopy images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11710–11718, 2019b.