Some aspects of real-time image denoising influenced by shot noise and compound Poisson noise

Oleg Kobylin, Oleksandra Putiatina

Kharkiv National University of Radioelectronics, 14 Nauki Ave., Kharkiv, 61166, Ukraine

Abstract

This paper introduces an alternative image model for video data that is represented as a series of images affected by shot noise. This type of noise not only disrupts the current frame in the video sequence but also goes over to subsequent frames, gradually diminishing over time until it vanishes. The shot-noise process is characterized by a sequence of jumps that decay as time passes. Under specific conditions, this process converges to a Gaussian distribution. To tackle this issue, the Kalman filter is proposed as a solution for removing noise and restoring the compromised image sequence. Numerical experiments demonstrate the effectiveness of the proposed approach in denoising videos corrupted by shot noise. The results of the proposed method were compared to the results provided by spatial Wiener filter, median filter, bilateral filter and a multilayer perceptron model. PSNR was calculated for the above methods.

Keywords

video, sequence of images, Gaussian process, compound Poisson noise, denoising, Kalman filter

1. Introduction

In certain applications digital devices can introduce noise to the original image or sequence of images. While there are various image denoising techniques that are effective at restoring noisy images, shot noise still remains mostly unexplored. Currently, no satisfactory algorithm exists that can effectively denoise images sequence of images affected by shot noise.

We propose a novel method for restoring images affected by shot noise. Unlike noise that vanishes immediately, shot noise diminishes gradually over time. This type of noise arises from defects in the hardware of a device or issues within the camera sensor [1]. Shot noise, driven by a Poisson process, exhibits a specific pattern of gradual decay following each occurrence.

When the intensity of the underlying Poisson process is high, shot noise can be effectively approximated by Gaussian noise. Numerous algorithms have been proposed in the literature to address various types of noise. These include linear filters, such as median and mean filters, as well as non-linear filters, which are discussed in sources [2, 3, 4]. Analysis and comparison of different image denoising methods is discussed in [5]. Such methods can also be employed for mitigating shot noise. If the jump rate is significant and the individual jumps are relatively small, the resulting shot noise increasingly resembles Gaussian noise. There are several established techniques for filtering Gaussian noise, including the bilinear filter, anisotropic diffusion filter, and Kernel Regression filter, as detailed in [6, 7, 8, 9].

The goal is to create a method for restoring images affected by shot noise by utilizing a Gaussian approximation of the shot-noise process in combination with Kalman filtering.

doors-2025: 5th Edge Computing Workshop, April 4, 2025, Zhytomyr, Ukraine
oleg.kobylin@nure.ua (O. Kobylin); oleksandra.putiatina@nure.ua (O. Putiatina)
https://nure.ua/staff/oleg-anatoliyovich-kobilin (O. Kobylin); https://nure.ua/staff/oleksandra-ievgenivna-putjatina (O. Putiatina)
0000-0003-0834-0475 (O. Kobylin); 0000-0003-4853-7125 (O. Putiatina)

2. Jump regression analysis for modeling noisy image

A standard two-dimensional grayscale image can be represented using the following jump regression equation [10]:

$$I_{ij} = f(x_i, y_j) + \epsilon_{ij}, i = 1, ..., n; j = 1, ..., m., \tag{1}$$

where

- (x_i, y_j) is the $(i, j)^{th}$ pixel;
- $f(x_i, y_j)$ is the true image intensity level at (x_i, y_j);
- ϵ_{ij} is the pointwise noise;
- I_{ij} is the observed image intensity level at (x_i, y_j).

The sequence of 2-D images can be modeled in the following way:

$$I_{ijk} = f_k(x_{ik}, y_{jk}) + \epsilon_{ijk}, \tag{2}$$

where

- $i = 1, ..., n; j = 1, ..., m., k = 1, ..., l.$
- (x_{ik}, y_{jk}) is the $(i, j)^{th}$ pixel of the k^{th} image;
- $f_k(x_{ik}, y_{jk})$ is the true image intensity level at (x_{ik}, y_{jk}) of the k^{th} image;
- ϵ_{ijk} is the pointwise noise of the k^{th} image;
- I_{ijk} is the observed image intensity level at (x_{ik}, y_{jk}) of the k^{th} image.

Therefore, the sequences $f_k(x, y), k = 1, 2, \ldots, l$ are the 2-D profiles, and monitoring the image sequence is equivalent to monitoring the 2-D profile sequence.

2.1. Shot noise process and its approximation

Shot noise is a type of random noise marked by abrupt intensity fluctuations that dissipate over time. This phenomenon frequently appears in real-time image processing, where individual pixels may experience sudden intensity spikes caused by defects in image recording devices. The impact of shot noise diminishes with time, resulting in reduced interference on the same pixel in subsequent frames of the image sequence.

Shot noise λ_t is defined in the following way:

$$\lambda_t = \lambda_0 e^{-\delta t} + \sum_{i=1}^{M_t} Y_i e^{-\delta(t-s_i)} = \lambda_0 e^{-\delta t} + e^{-\delta t} \sum_{i=1}^{M_t} Y_i e^{\delta s_i},$$

where

- λ_0 is the initial value of λ_t;
- $\{Y_i\}_{i=1,2,\ldots}$ is the sequence of iid random variables with distribution function $F(y)$ and $E(Y_i) = \mu_1$;
- $\{s_i\}_{i=1,2,\ldots}$ is the sequence representing the event times of a Poisson process M_t with constant intensity ρ;
- δ is the rate of exponential decay.

The distribution of the random variables $\{Y_i\}$ can be arbitrary, for example normal distribution or beta distribution [11].

The expectation of the shot noise λ_t, assuming that λ_0 is known, is as follows from [11]:

$$E(\lambda_t) = \frac{\mu_1 \rho}{\delta} + \left(\lambda_0 - \frac{\mu_1 \rho}{\delta}\right) e^{-\delta t} \to \frac{\mu_1 \rho}{\delta}\ as\ t \to \infty, \tag{3}$$

and, moreover, if the initial value λ_0 equals $\mu_1\rho/\delta$, then we have a stationary case and the mean value $E(\lambda_t)$ will be equal to $\mu_1\rho/\delta$ and will not depend on time t.

The variance of the shot noise process λ_t is as in [11]:

$$Var(\lambda_t) = \frac{\mu_2\rho}{2\delta}(1 - e^{-2\delta t}) \to \frac{\mu_2\rho}{2\delta}\ as\ t \to \infty. \tag{4}$$

Consider the following linear transformation:

$$Z_t^{(p)} = \frac{\lambda_t - \mu_1\rho/\delta}{\sqrt{\mu_2\rho/2\delta}}. \tag{5}$$

The main result of the paper [[11]] is that $Z_t^{(p)}$ (the normalization of shot noise or linear transformation) converges to some Z_t that is normally distributed. Assume that $\rho \to \infty$ and that λ_0 is a random variable independent of everything else, such that $(\lambda_0 - (\mu_1\rho/\delta))(\mu_1\rho/2\delta)^{-1/2}$ converges in distribution to Z_0. Then $Z_t^{(p)}$ converges in law to Z_t, where

$$dZ_t = -\delta Z_t dt + \sqrt{2\delta} dB_t, \tag{6}$$

where B_t is standard Brownian motion.

This implies that Z_t is normally distributed with mean $E(Z_t) = Z_0 e^{-\delta t} \to 0\ as\ t \to \infty$ and variance $Var(Z_t) = 1 - e^{-2\delta t} \to 1\ as\ t \to \infty$. If $\lambda_0 = \mu_1\rho/\delta$, then $Z_0 = 0$ and, therefore, $E(Z_t) = 0$.

Following the linear transformation (5) the shot-noise process λ_t has the following form:

$$\lambda_t = \frac{\mu_1\rho}{\delta} + Z_t^{\rho}\sqrt{\frac{\mu_2\rho}{2\delta}}.$$

Define $\hat{\lambda}_t$ as Gaussian approximation of λ_t as follows:

$$\hat{\lambda}_t = \frac{\mu_1\rho}{\delta} + Z_t\sqrt{\frac{\mu_2\rho}{2\delta}}.$$

When the intensity of jumps is relatively high, shot noise can be effectively approximated as Gaussian noise. This allows for the application of Kalman filtering techniques in real-time image restoration processes. A series of 2-D images affected by shot noise can be represented through the following model:

$$I_{ijk} = f_k(x_{ik}, y_{jk}) + \lambda_{ijk}, \tag{7}$$

where

- $i = 1, ..., n; j = 1, ..., m., k = 1, ..., l.$;
- (x_{ik}, y_{jk}) is the $(i, j)^{th}$ pixel of the k^{th} image;
- $f_k(x_{ik}, y_{jk})$ is the true image intensity level at (x_{ik}, y_{jk}) of the k^{th} image;
- λ_{ijk} is the pointwise shot-noise of the k^{th} image;
- I_{ijk} is the observed image intensity level at (x_{ik}, y_{jk}) of the k^{th} image.

Applying the Gaussian approximation of the shot noise we obtain the following model of the sequence of the 2-D images:

$$I_{ijk} = f_k(x_{ik}, y_{jk}) + \hat{\lambda}_{ijk} = f_k(x_{ik}, y_{jk}) + \frac{\mu_1\rho}{\delta} + Z_t\sqrt{\frac{\mu_2\rho}{2\delta}}, \tag{8}$$

where

- Z_t is normally distributed with mean 0 and variance 1 as $t \to \infty$;
- μ_1 and μ_2 are first and second initial moment of the random variable Y;

- ρ is the intensity of the underlying Poisson process;
- δ is the rate of exponential decay.

There occurs a correction term $\frac{\mu_1\rho}{\delta}$ in equation (8), which is constant additive part to the image intensity. Therefore

$$I_{ijk} = f_k(x_{ik}, y_{jk}) + \hat{\lambda}_{ijk} = g_k(x_{ik}, y_{jk}) + Z_t\sqrt{\frac{\mu_2\rho}{2\delta}}, \tag{9}$$

where

$$g_k(x_{ik}, y_{jk}) = f_k(x_{ik}, y_{jk}) + \frac{\mu_1\rho}{\delta}. \tag{10}$$

2.2. Compound Poisson process and its approximation

Compound Poisson noise process is defined in the following way:

$$J_t = \sum_{i=1}^{M_t} Y_i.$$

This definition is a special case of the shot-noise process with $\delta = 0$.
We recall the result from previous subsection:

$$V_t^{(\rho)} = \frac{J_t - \mu_1\rho}{\sqrt{\mu_2\rho}} \to B_t,\ as\ \rho \to \infty,$$

where $J_t = \sum_{i=1}^{M_t} Y_i$, M_t is Poisson process with intensity ρ and B_t is Brownian motion.

This means that compound Poisson process

$$J_t = \sqrt{\mu_2\rho/}V_t^{\rho} + \mu_1\rho$$

can be approximated by

$$\hat{J}_t = B_t\sqrt{\mu_2\rho} + \mu_1\rho$$

.

Applying the Gaussian approximation of the compound Poisson noise process we obtain the following approximate model of the sequence of the 2-D images. Therefore

$$I_{ijk} = f_k(x_{ik}, y_{jk}) + \hat{J}_{ijk} = g_k(x_{ik}, y_{jk}) + B_t\sqrt{\mu_2\rho}, \tag{11}$$

where

$$g_k(x_{ik}, y_{jk}) = f_k(x_{ik}, y_{jk}) + \frac{\mu_1\rho}{\delta}. \tag{12}$$

It can be noticed that both shot-noise and compound Poisson noise process both have Gaussian approximations, but these noise processes have different impact on the images. Shot noise decays and disappears as time passes. But the Compound Poisson noise process does not disappear, its effect stays at that part of the image where it occurred.

Both types of noise can be filtered out using Kalman filter if the intensity of the underlying Poisson process is high and thus the Gaussian approximation holds.

2.3. Kalman filtering for image restoration

Kalman filtering is used to estimate the variables of the control system subject to stochastic disturbances caused by noisy measurements of the input variables [12, 13, 14, 15]. There are two kinds of equations in Kalman filter: time update equations and measurement update equations. The time update equations obtain a priory estimates of the state and covariance matrix. The measurement update equations improve the estimate of the state by using new measurement.

We assume that the image sequence is captured by several cameras and the noise interferes during the image acquisition process. Then the image sequence is processed for further analysis. The aim is to construct the filter that would efficiently denoise the image sequence and would produce acceptable results with such kind of noise. The representation of the image for the Kalman filter is pixelwise.

The noise free pixel value is f_k, which is assumed to be the first order autoregression model. The process looks as follows:

$$f_{k+1} = \alpha f_k + v_k, \tag{13}$$

where

- α is a constant and depends on signal parameters;
- v_k is Gaussian noise with zero mean and σ^2 variance.

Such a model is often used to represent pixel values in video signals.

The measured signal is given by the following equation:

$$I_k = f_k + w_k, \tag{14}$$

where w_k the independent of v_k additive zero mean Gaussian white noise with variance σ^2.
We need to construct a linear unbiased estimate $\hat{f}_k$ of f_k having the observations $I_1, I_2, ..., I_k$. The estimation error is denoted by $\tilde{f}_k = f_k - \hat{f}_k$. The variance of the error is denoted by $\tilde{p}_k$ and defined as follows:

$$\tilde{p}_k = E[(\tilde{f}_k)^2].$$

The discrete Kalman filter is assessed through the iterative application of the following equations:

- filter equations;

$$f_k^* = a^2 \hat{f}_{k-1}, \tag{15}$$

$$\hat{f}_k = f_k^* + K_k \{I_k - f_k^*\}; \tag{16}$$

- variance of the estimation error and the coefficient K;

$$p_k^* = a^2 \tilde{p}_{k-1} + \sigma_v^2, \tag{17}$$

$$K_k = p_k^* \{p_k^* + \sigma_w^2\}^{-1}, \tag{18}$$

$$\tilde{p}_k = p_k^* - K_k p_k^*. \tag{19}$$

Note that $p_{k+1}^* = E[(a^2 \tilde{f}_k + \sigma_v)^2]$. This algorithm is applied to each pixel and at each time instant.

The estimated pixel value $\hat{f}_k$ the output for each iteration k. It provides the filtered image for the further analysis.

We consider the following parameters for the shot-noise process:

- intensity of the Poisson process equals 1.04,
- jump size y is normally distributed N(0, 0.007).

The shot-noise and compound Poisson process look as shown in figures 1 and 2.

The example consists of a sequence of 2D RGB images, where certain pixels in the video are affected by a shot noise process, leading to corruption as noted in [16]. A Kalman filter is then employed to process the sequence of images corrupted by shot noise, effectively restoring them. Figures 3 and 4 illustrate the example.

Figure 3 shows one image taken from the original video sequence of images. The original image sequence is not corrupted by any noise. Figure 4(a) shows the same image as in figure 3 but corrupted by the additive shot-noise. One can see that not the whole image is corrupted but only a part of it (the upper part). This means that not every pixel is noisy but only the pixels on the upper side. In real life this may occur due to the problems with the camera sensor. Next, the shot-noise corrupted image

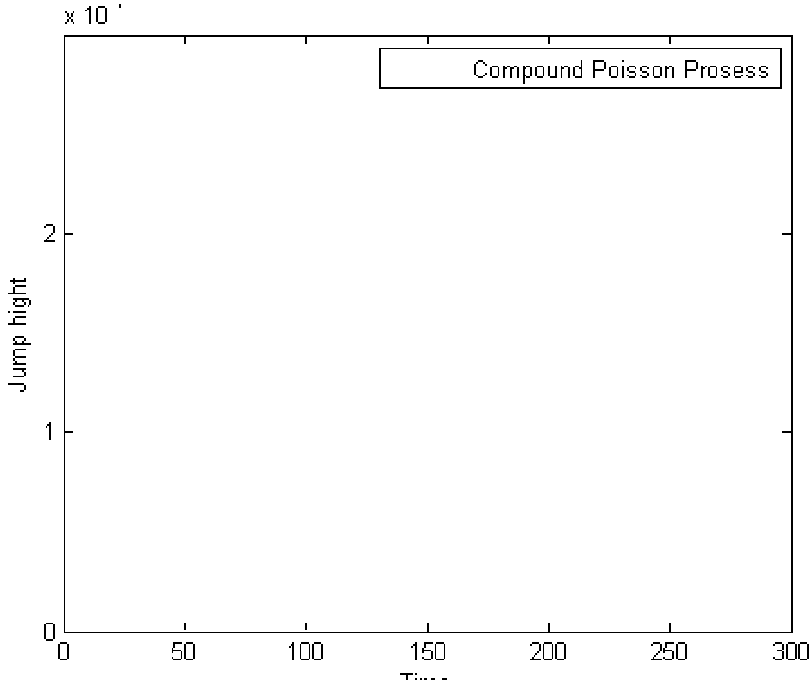

Figure 1: Compound Poisson process.

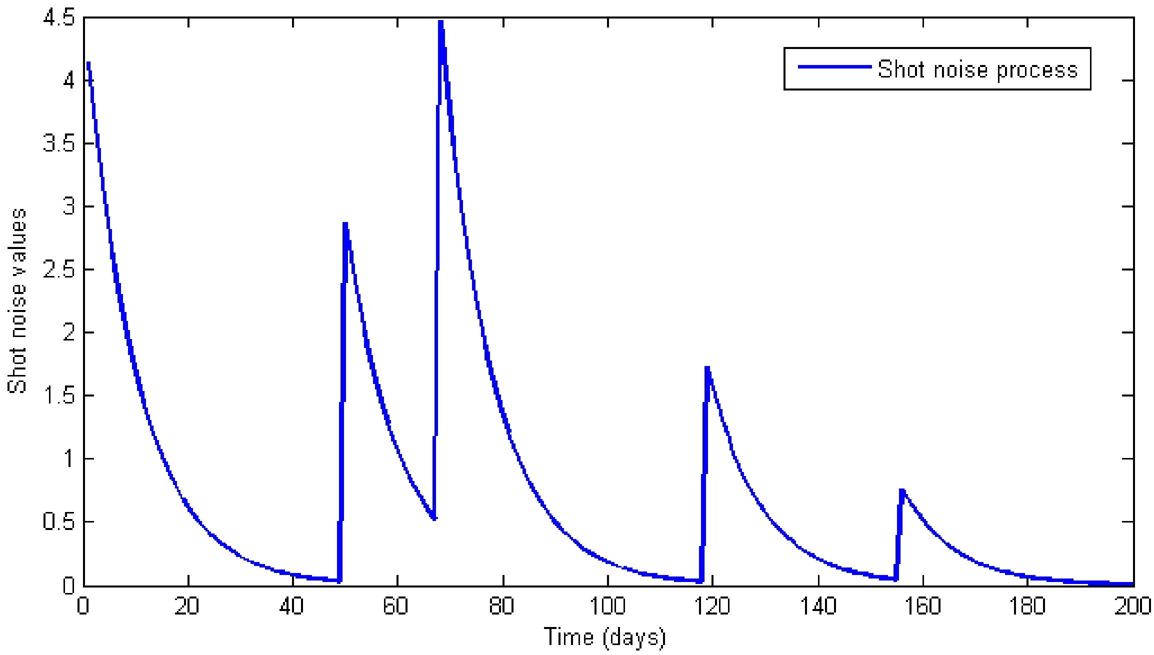

Figure 2: Shot noise process.

sequence is denoised using the Kalman filtering techniques. In figure 4(b) the corresponding denoised image is shown. By comparing figures 3 and 4(b) (calculating the MSE), one can evaluate the quality of restoration. Although, the noise is not removed completely, the quality of the image becomes much better. The similar results concerning compound Poisson noise are show in figures 4(c) and 4(d).

The results of the performance of other denoising techniques, such as spatial Wiener filter, median filter, bilateral filter, multilayer perceptron model, comparing to the proposed approach are provided in table 1.

The higher PSNR value value means the better image filtration quality, therefore, the use of proposed filtering approach (Kalman filtering) gave better results amoung other filters.

Figure 3: Original image.

(a) (b) (c) (d)

Figure 4: (a) shot noise corrupted image, (b) restored image without shot noise, compound Poisson noise corrupted image, (d) restored image without compound Poisson noise removed.

Table 1
Comparison of results in PSNR(dB).

Filters	**PSNR(dB)** value
The proposed filtering approach	39.36
Spatial Wiener filter	35.71
Median filter	37.93
Bilateral filter	36.55
Multilayer perceptron model	38.97

3. Conclusion

In this paper we have solved the problem of filtering the pixels of the video corrupted by shot-noise and compound Poisson noise. Shot noise brings the effect of instant jump of the pixel value that slowly decays as time passes. Compound Poisson noise gives the effect of the instant jump of the pixel intensity that does not decay as time passes. Therefore it is very important to filter this kind of noise. Standard filtering techniques would not work well for such type of noise. Therefore, we considered approximating shot-noise and compound Poisson noise with Gaussian process and applying Kalman filtering to remove shot noise from the sequence of images. Kalman filtering uses the filtered image from the previous step to denoise (filter) the image on the current step. The graphical illustration was shown on several figures. In average, the shot noise corrupted image sequence is 90% restored using Kalman filter comparing to the original image sequence. For compound Poisson process the filtering quality is 80% on average.

The central result of this paper allows to model the effects of the random shot-noise and compound Poisson jumps that corrupts the video data, to approximate the both kinds of noise by Gaussian noise and to apply the Kalman filtering techniques for the video restoration.

Declaration on Generative AI: The authors have not employed any generative AI tools.

References

[1] A. Awad, Denoising images corrupted with impulse, Gaussian, or a mixture of impulse and Gaussian noise, Engineering Science and Technology, an International Journal 22 (2019) 746–753. doi:10.1016/j.jestch.2019.01.012.
[2] S. Swamy, P. K. Kulkarni, A basic overview on image denoising techniques, International Research Journal of Engineering and Technology (IRJET) 7 (2020) 851–857. URL: https://www.irjet.net/archives/V7/i5/IRJET-V7I5166.pdf.
[3] R. Szeliski, Computer Vision: Algorithms and Applications, Texts in Computer Science, 2 ed., 2010. doi:10.1007/978-3-030-34372-9.
[4] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, Wiley, 2000.
[5] S. Swathi, V. Magudeeswaran, Analysis and Comparison of Different Image Denoising Techniques- Review, International Journal of Computer Science and Mobile Applications 2 (2014) 115–124. URL: https://www.ijcsma.com/articles/analysis-and-comparison-of-different-image-denoising-techniques-review.pdf.
[6] S. Ertürk, Real-Time Digital Image Stabilization Using Kalman Filters, Real-Time Imaging 8 (2002) 317–328. doi:10.1006/rtim.2001.0278.
[7] M. Piovoso, P. A. Laplante, Kalman filter recipes for real-time image processing, Real-Time Imaging 9 (2003) 433–439. doi:10.1016/j.rti.2003.09.005, Special Issue on Software Engineering of Real-time Imaging Systems.
[8] S. Citrin, M. Azimi-Sadjadi, A full-plane block Kalman filter for image restoration, IEEE Transactions on Image Processing 1 (1992) 488–495. doi:10.1109/83.199918.

[9] P. Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012.

[10] P. Qiu, Jump regression, image processing, and quality control, Quality Engineering 30 (2018) 137–153. doi:10.1080/08982112.2017.1357077.

[11] O. Putyatina, J. Sass, Approximation for portfolio optimization in a financial market with shot-noise jumps, Computational Management Science 15 (2018) 161–186. doi:10.1007/s10287-017-0294-5.

[12] F. Ababsa, Robust Extended Kalman Filtering for camera pose tracking using 2D to 3D lines correspondences, in: 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2009, pp. 1834–1838. doi:10.1109/AIM.2009.5229789.

[13] R. M. Tymchyshyn, O. Y. Volkov, O. Y. Gospodarchuk, Y. P. Bogachuk, Modern approaches to computer vision, Upravlâûŝie sistemy i mašiny (2018) 46–73. doi:10.15407/usim.2018.06.046.

[14] V. A. Gorokhovatsky, Compression of descriptions in the structural image recognition, Telecommunications and Radio Engineering 70 (2011) 1363–1371. doi:10.1615/telecomradeng.v70.i15.60.

[15] P. F. Alcantarilla, A. Bartoli, A. J. Davison, KAZE Features, in: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), Computer Vision – ECCV 2012, volume 7577 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 214–227. doi:10.1007/978-3-642-33783-3_16.

[16] C. M. Bishop, Pattern Recognition and Machine Learning, Information Science and Statistics, Springer, 2006. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf.