Artificial Intelligence Project 1
Binary Image Denoising

Class 1, Team 6

November 18, 2014

1 Team Members

Name	Student ID	Job
Fan Ziyao	12330081	Team leader, implementation
Chen Yingcong	12330049	Implementation
Chen Xiongtao	12330040	Modeling, implementation
Huang Long	12330132	Implementation
Zhang Qiuyi	12330402	Implementation, documentation

2 Problem Description

Let the observed noisy image be described by an array of binary pixel values $y_i \in \{1, +1\}$, where the index $i = 1, ..., D$ runs over all pixels. We shall suppose that the image is obtained by taking an unknown noise-free image, described by binary pixel values $x_i \in \{1, +1\}$ and randomly flipping the sign of pixels with some small probability, say, 10%. Given the noisy image, our goal is to recover the original noise-free image.

3 Modeling

Knowing that the density of noise is small, there should be a strong correlation between x_i and y_i. Another prior knowledge is that neighboring pixels x_i and x_j in an image are strongly correlated. This knowledge implies that $\{\mathbf{x}, \mathbf{y}\}$ has the Markov property and can be described with a undirected graph [1], so we can use the Markov random field to model this problem.

There are two types of cliques in this graph. The first one $\{x_i, y_i\}$ has an associated energy function that expresses the correlation between these variables. Here we can use a simple energy function for them: $-\eta x_i y_i$, where η is a positive constant. This will produce a lower energy (thus encouraging a higher probability) when x_i and y_i have the same sign and a higher energy when they have the opposite sign.

The other type of cliques is $\{x_i, x_j\}$ where i and j are indices of neighboring pixels. Again, we want the energy to be lower when the pixels have the same sign than when they have the opposite sign, and so we choose an energy given by $-\beta x_i x_j$ where β is another positive constant.

Because a potential function is an arbitrary, non-negative function over a maximal clique, we can multiply it by any nonnegative functions of subsets of the clique, or equivalently we can add the corresponding energies. In this model, this allows us to add an extra term hx_i for each pixel i in the noise-free image. This would bias the model towards pixel values that have one particular sign in preference to the other. Hence the complete energy function for this model is:

$$E(\mathbf{x}, \mathbf{y}) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i$$

which defines a joint distribution over x and y given by:

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\{-E(\mathbf{x}, \mathbf{y})\}$$

where Z is the *partition function* [1]. Our goal is then defined as finding an $\mathbf{x}$ such that:

$$\mathbf{x} = \arg\min E(\mathbf{x}, \mathbf{y})$$

4 Algorithm and Implementation

4.1 Iterated Conditional Modes

We now fix the elements of y to the observed values given by the pixels of the noisy image, which implicitly defines a conditional distribution $p(x|y)$ over noise-free images. This is an example of the *Ising model*, which has been widely studied in statistical physics.

For this problem, we wish to find an image x having a high probability (ideally the maximum probability), which is equivalent to making E as small as possible. Since E can be viewed as a function of $M \times N$ variables, with each variable representing a pixel value, we can use the method of gradient descent to search for the minimizer of E. More precisely, for each pixel x_i we search by evaluating two possible states for $x_i = 1$ and $x_i = 1$, keeping all other pixel value fixed, and choose between the two states. The search will be repeated until some stopping criterion is satisfied.

A straightforward scheme would be to choose whichever state that leads to a lower energy at each step of the search. This greedy approach, described in Algorithm 1, is known as the *iterated conditional modes*, or ICM. Despite being a simple and effective strategy, ICM is prone to local optima. Therefore, we need another approach for a better result.

Algorithm 1 Binary image denoising with iterated conditional modes

```
function DENOISE(y, β, η, h)                         ▷ y is the noisy image
    Initialize x with x = y
    Initialize E_best with E_best = E(x, y)
    for k = 1 → k_max do
        for each pixel x_i do
            E_1 = E(x, y)
            x_i = −x_i                                ▷ flip the pixel
            E_2 = E(x, y)
            if E1 > E2 then
                if E_2 < E_best then
                    Record the best energy E_best = E_2
            else
                x_i = −x_i                            ▷ flip the pixel back
    return x
```

4.2 Simulated Annealing

It is easy to see that $E(x, y)$ is a non-convex function of x, which implies that there will be multiple local minima for E depending on the initial state. To search for the global optimun, we need a global optimization strategy. In this project, we use *simulated annealing*(SA), which can be easily integrated with method of gradient descent. By adding randomness to our searching strategy, we increase the probability of reaching a global optimum (and SA does approximate one in this model, as proven in [2]).The complete algorithm is described in Algorithm 2.

Algorithm 2 Binary image denoising with simulated annealing

function DENIOSE($\mathbf{y}$, β, η, h) ▷ $\mathbf{y}$ is the noisy image
  Initialize $\mathbf{x}$ with $\mathbf{x} = \mathbf{y}$
  Initialize $Ebest$ with $Ebest = E(\mathbf{x}, \mathbf{y})$
  for $k = 1 \to k_{max}$ **do**
    Compute the temperature $t = \text{temperature}(k, k_{max})$
    for each pixel x_i **do**
      $E_1 = E(\mathbf{x}, \mathbf{y})$
      $x_i = -x_i$ ▷ flip the pixel
      $E_2 = E(\mathbf{x}, \mathbf{y})$
      Compute the transition probability $p = prob(E_1, E_2, t)$
      if $p > q$ where q is a random number in $[0, 1]$ **then**
        if $E_2 < E_{best}$ **then**
          Record the best energy $Ebest = E_2$
      else
        $x_i = -x_i$ ▷ flip the pixel back
  return x

Remark 1.

The temperature function is a decreasing function of iterations. It must converge to 0 as $k \to k_{max}$. For this implementation we use

$$\text{temperature}(k, k_{max}) = \frac{1}{500}\left(\frac{1}{k} - \frac{1}{k_{max}}\right)$$

Remark 2.

The probability transition function used for this implementation is

$$prob(E_1, E_2, t) = \begin{cases} 1, & E_1 > E_2 \\ \mathbf{e}^{\frac{E_1 - E2}{t}}, & E_1 \leq E_2 \end{cases}$$

4.3 Local Optimization

With either ICM or SA, we only need to alter the E for values relavent to the flipped pixel at each step. Let x_i be the pixel to be flipped, we can just update E with the new $hx_i + \eta x_i y_i + \sum \beta x_i x_j$ where x_j are the neighboring pixels of x_i. Then we can localize the calculation of E_1 and E_2, making the implementation much faster.

5 Experiment Result

We first take a binary image(for simplicity we choose a black-and-white one), then flip its pixels with 10% probability to generate the noisy image. After that, we try to restore the original image by running both ICM and SA denoising, and compare their results. For the energy function, we choose $\beta = 1 \times 10^{-4}, \eta = 2.1 \times 10^{-4}, h = 0$. The maximum number of iteration k_{max} is 30 for both ICM and SA.

The experiment results are shown in Figure 1 - 4.

Figure 1: Original image

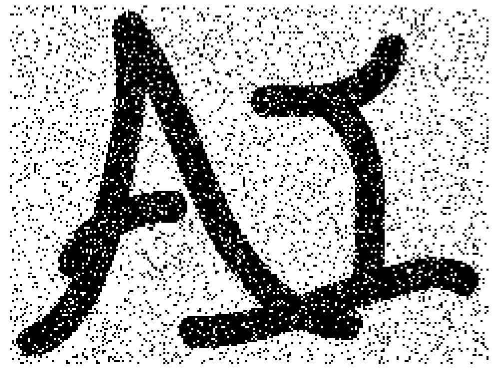

Figure 2: Noisy image with 10% pixels flipped

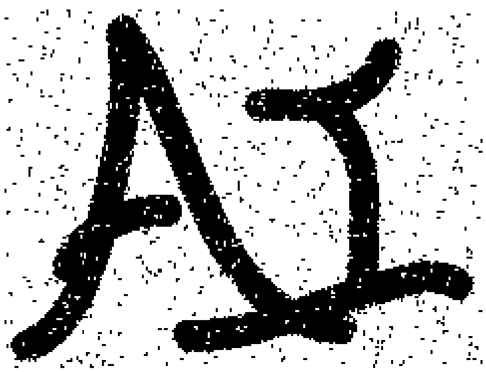

Figure 3: Denoised with ICM

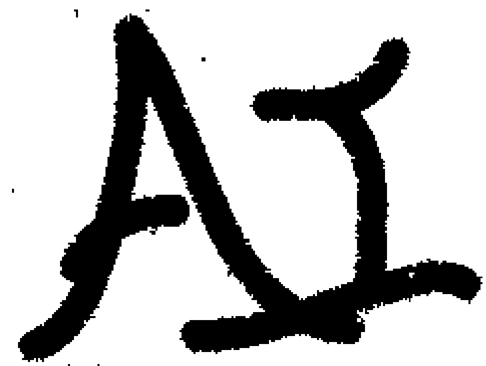

Figure 4: Denoised with SA

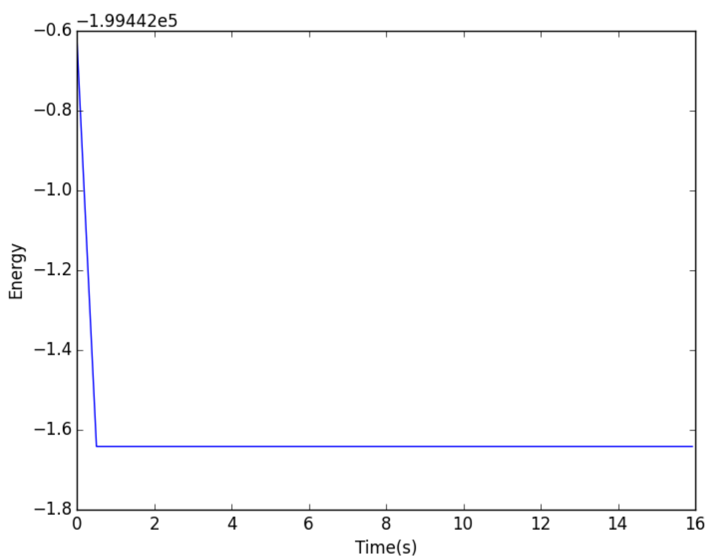

Figure 5: Time-Energy series of ICM

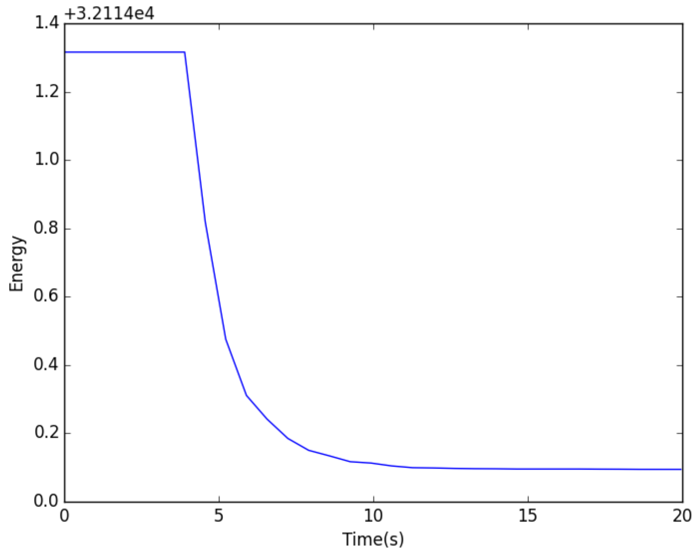

Figure 6: Time-Energy series of SA

To evaluate the denoised results, we count the non-zero elements in $\mathbf{x} - \mathbf{y}$. ICM produces an image with 96.21% of the pixels agree with the original one, while SA yeilds 99.19%.

6 Discussion

Both ICM and SA restored most of the original image, though SA is visually much better than ICM, which is not surprising – after all, SA approximates the global optimum while ICM tends to stuck in local optima.

From a visual inspection of the images, it is apparent that SA successfully removed the majority of the outlying noises, especially those in the uniform areas. Since the neighborhood defined in E is only 4-adjacent, SA can not capture the overall attributes of the image, such as the egdes. The randomness incorporated by SA also makes it uncertain about these noises. Therefore, in areas where the noises obscure the edges of the object, some pixels have been mistakenly flipped; in area with high density of noises, some of them have been clustered instead of being removed. Aside from these small artifacts, the result produced by SA is rather appealing.

References

[1] Bishop, C. M., et al. *Pattern recognition and machine learning*, vol. 1. springer New York, 2006.

[2] Geman, S., and Geman, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6 (1984), 721–741.