

MUTUAL INFORMATION
MUTUAL INFORMATION-BASED REGISTRATION OF
DIGITALLY RECONSTRUCTED RADIOGRAPHS AND
ELECTRONIC PORTAL IMAGES

by

Katherine Anne Bachman

Master of Basic Science, Mathematics, University of Colorado at Denver, 2002

Bachelor of Science, Chemistry, University of Colorado at Denver, 2000

A thesis submitted to the
University of Colorado at Denver
in partial fulfillment
of the requirements for the degree of
Master of Science
Applied Mathematics

2005

This thesis for the Master of Science

degree by

Katherine Anne Bachman

has been approved

by

Weldon A. Lodwick

William L. Briggs

William E. Cherowitzo

Date

Bachman, Katherine Anne (M.S., Applied Mathematics)

Mutual Information
Mutual Information-Based Registration of
Digitally Reconstructed Radiographs and
Electronic Portal Images

Thesis directed by Professor Weldon A. Lodwick

ABSTRACT

This study regards discrete mutual information and demonstrates the use of the theory with an example of radiological image registration with in-plane, two-dimensional images using various search strategies.

Image registration is the process of finding an optimal geometric transformation between corresponding image data. Although it has applications in many fields, the one that is addressed in this thesis is medical image registration. Medical image registration has a wide range of potential applications, but the emphasis is on radiological imaging.

In addition to having application in communication and computer vision, mutual information has proven robust and has resulted in fully automated registration algorithms that are currently in use. Mutual information, which is given by the difference between the sum of the entropies of individual overlapping images and the joint entropy of the combined images, is a measure of the reduction in uncertainty about one image due to knowledge of the other. This non-parametric technique makes no assumption of the functional form or

relationship between image intensities in the two images.

The thesis is organized as follows. Chapter 1 gives a broad overview of medical image registration as the context in which to present the subject of mutual information-based medical image registration. Chapter 2 regards the theory and mathematics of discrete mutual information and its origin in information theory. Chapter 3 looks at the implementation of the theory applied to image registration in general. Chapter 4 looks at an application of mutual information-based image registration in radiological imaging - registration of Digitally Reconstructed Radiographs (DRRs) and Electronic Portal Images (EPIs). Chapter 5 consists of concluding remarks. The Appendix includes information that is relevant, but not critical, to the understanding of the material presented in this thesis. Because probability theory is a major part of information theory and, consequently, mutual information theory, a brief overview of discrete probability theory is included in the Appendix as a quick reference. Examples are provided in the Appendix as well as throughout the body of the thesis.

This abstract accurately represents the content of the candidate's thesis. I recommend its publication.

Signed _____
Weldon A. Lodwick

DEDICATION

To my parents, Dr. and Mrs. Miles MacGran, and to my husband, Randy.

ACKNOWLEDGMENT

I would like to thank my advisor, Weldon A. Lodwick, for the means of support obtained for me, without which this paper might not have been possible. I would also like to thank Dr. Lodwick and Francis Newman for their meticulous review and critique of the thesis, and Francis Newman for supplying the images used for the DRR/EPI registration experiments and other related images. I would like to thank Weldon Lodwick, Bill Briggs, and Bill Cherowitzo for acting as my Graduate Committee, and also Mike Jacobson for attending the thesis presentation.

CONTENTS

| | |
|---|------|
| Figures | x |
| Tables | xiii |
| <u>Chapter</u> | |
| 1. Medical Image Registration Overview | 1 |
| 1.1 Taxonomy of Medical Image Registration Methodology | 2 |
| 1.1.1 Prospective and Retrospective Methods | 4 |
| 1.2 Types of Transformation | 7 |
| 1.3 Image Registration as an Optimization Problem | 9 |
| 2. Mutual Information Theory | 13 |
| 2.1 Information | 13 |
| 2.2 Entropy (uncertainty or complexity) | 15 |
| 2.2.1 Joint Entropy and Conditional Entropy | 25 |
| 2.2.2 Relative Entropy | 31 |
| 2.2.3 Concavity of Entropy | 36 |
| 2.2.4 Entropy of a Signaling System | 37 |
| 2.2.5 Entropy and Kolmogorov Complexity | 38 |
| 2.3 Mutual Information | 40 |
| 3. Mutual Information Theory Applied to Image Registration | 51 |
| 4. Mutual Information-Based Registration of Digitally Reconstructed Ra- diographs (DRRs) and Electronic Portal Images (EPIs) | 61 |

| | | |
|-----------------------------|---|-----|
| 4.1 | Experiments and Results | 65 |
| 4.1.1 | Greedy Algorithm | 65 |
| 4.1.2 | Genetic Algorithm | 76 |
| 4.2 | Nelder-Mead (MATLAB's fminsearch) | 81 |
| 4.3 | Simulated Annealing | 81 |
| 4.4 | Other Experiments | 83 |
| 5. | Conclusion | 86 |
| <u>Appendix</u> | | |
| A. | Brief Discrete Probability Theory | 88 |
| B. | Properties of Convex Functions | 95 |
| C. | Miscellaneous | 96 |
| D. | Sample Output | 98 |
| <u>References</u> | | 114 |

FIGURES

| | | |
|--------|---|----|
| Figure | | |
| 1.1 | Registration example with multiple solutions. The image B is to be transformed to image A. The rightmost image shows B registered to A by two translations and a rotation of 45 degrees. | 10 |
| 2.1 | Discrete noiseless binary channel. | 13 |
| 2.2 | Discrete binary symmetric channel. | 13 |
| 2.3 | The entropy function for base 2, e , and 10 logs. | 15 |
| 2.4 | Decomposition of a choice from three possibilities. | 16 |
| 2.5 | The entropy function (base 2 log) of two probabilities. | 21 |
| 2.6 | Bound on $\log_e(x)$ function. | 23 |
| 2.7 | Noisy communication channel, Example 2. | 28 |
| 2.8 | Relationship between entropy and mutual information. The mutual information $I(A, B)$ corresponds to the intersection of the information in A with the information in B | 46 |
| 3.1 | The plot on the left is the joint histogram of two randomly generated 150×150 matrices taking on values from 1 to 256. The plot on the right is the plot of the joint histogram of one matrix with itself. . . | 53 |
| 3.2 | Relationship between entropy and mutual information. The mutual information $I(A, B)$ corresponds to the intersection of the information in A with the information in B | 54 |

| | | |
|------|--|----|
| 3.3 | Venn diagram of the relationship between joint entropy and mutual information of totally unrelated images. | 54 |
| 3.4 | Venn diagram of the relationship between joint entropy and mutual information of totally related images. | 55 |
| 3.5 | The image data sets, A and B | 56 |
| 3.6 | The joint probability distribution of Example 4 as a surface. | 57 |
| 3.7 | The image data sets A , rotated -90 degrees, and B | 59 |
| 3.8 | The joint probability distribution as a surface of Example 4 with image A rotated -90° as in Figure 3.7. | 60 |
| 4.1 | A typical radiation treatment plan [11]. | 61 |
| 4.2 | Typical aligned pelvis DRR [11]. | 62 |
| 4.3 | Typical aligned pelvis EPI [11]. | 62 |
| 4.4 | A misaligned EPI and the DRR automatically rotated to the EPI position [11]. | 63 |
| 4.5 | Joint histogram of misaligned images. Referring to the rightmost plot, the x- and y- axes represent the range of intensities in the images. The z-axis represents the probability that an intensity in one image will occur with an intensity in the other. It is a surface plot of the probability values represented in the plot on the left. [11] | 63 |
| 4.6 | The result after the automated registration process [11]. | 64 |
| 4.7 | Greedy algorithm, run 1, 256×256 images. | 68 |
| 4.8 | Greedy algorithm, run 2, 256×256 images. | 70 |
| 4.9 | 64×64 DRR and EPI. | 73 |
| 4.10 | 128×128 DRR and EPI. | 73 |

| | | |
|------|---|----|
| 4.11 | 256×256 DRR and EPI. | 74 |
| 4.12 | Plot of MI versus Angle of Rotation from simulated annealing data. | 84 |
| 4.13 | Detail of Figure 4.12, plot of MI versus Angle of Rotation from simulated annealing data. | 85 |
| A.1 | Example 9. | 93 |

TABLES

Table

| | | |
|------|--|----|
| 4.1 | <i>Sample run (Figure 4.7) converges to the optimum transformation in 8.712 minutes.</i> | 69 |
| 4.2 | <i>Sample run (Figure 4.8) converges to a suboptimal solution in 5.0823 minutes.</i> | 71 |
| 4.3 | <i>Convergence data for 22 runs, 256×256 images.</i> | 72 |
| 4.4 | <i>64×64 run converges in 0.6504 minutes.</i> | 76 |
| 4.5 | <i>128×128 run converges in 1.2058 minutes.</i> | 77 |
| 4.6 | <i>256×256 run converges in 4.3367 minutes.</i> | 78 |
| 4.7 | <i>Summary of Tables 4.3, 4.4, and 4.5.</i> | 78 |
| 4.8 | <i>Genetic algorithm. 256×256 images.</i> | 81 |
| 4.9 | <i>Parameter list for genetic run pairs.</i> | 82 |
| 4.10 | <i>MATLAB's fminsearch (Nelder-Mead) algorithm. 256×256 images.</i> | 82 |

1. Medical Image Registration Overview

Image registration is the process of finding an optimal geometric transformation between corresponding image data. In other words, given a reference or model, image A , and a test, or floating image, image B , find a suitable transformation, T , such that the transformed test image becomes similar to the reference. The image registration problem typically occurs when two images represent essentially the same object, but there is no direct spatial correspondence between them. The images might be acquired with different sensors, or the same sensor at different times or from different perspectives.

Modern three-dimensional treatment radiation planning is based on sequences of tomographic images. Computed tomography (CT) has the potential to quantitatively characterize the physical properties of heterogeneous tissue in terms of electron densities. Magnetic resonance (MR), which looks at hydrogen atom densities, is very often superior to CT, especially for the task of differentiating between healthy tissue and tumor tissue. Positron emission tomography (PET), single photo emission tomography (SPECT), and MRS (magnetic resonance spectroscopy) imaging have the potential to include information on tumor metabolism. The various image modalities are non-competing. They have specific properties and deliver complementary information. The images supply important information for delineation of tumor and target volume, and for therapy monitoring.

There is usually a high degree of shared information between images of different modalities of the same structures. This is certainly the case with images of the same modality which have been acquired at different times or from different perspectives as in the example presented in Chapter 4. In this case, the modality is CT, or CT-derived, and the problem is rectification of orientation, or pose, of one CT image with respect the other by in-plane rotations and/or shifts.

In order to use image sequences of the same, or various, modalities simultaneously, a definite relation between the picture elements (pixels), or volume elements (voxels) of the various image sequences needs to be established. Methods which are able to calculate and establish these relations are called registration, matching, or image correlation techniques.

1.1 Taxonomy of Medical Image Registration Methodology

Image registration methods can be categorized by

- User interaction

- Manual

- The transformation is determined directly by user interaction.

- Semi-automatic

- The transformation is calculated automatically, but the user determines the image features used and the start-up parameters.

- Automatic

- No user interaction is required.

- Scope and elasticity of transformation

- Scope

- * Global

- Global transformations modify the data set as a whole, that is, not just the therapy relevant area, by applying a single function to all elements.

- * Local

- In local procedures the function can vary. Single voxels or pixels, single image slices, or single organs could be affected.

- Elasticity or plasticity (geometrical properties) of transformation

- * Rigid transformations

- The distance between any two points in the first image is preserved when these two points are mapped onto the second image. Rigid transformations can be decomposed into translation, rotation, and reflection. Rigid transformations are special cases of affine transformations. A transformation is called affine when any straight line in the first image is mapped onto a straight line in the second image while parallelism is preserved.

- * Non-rigid affine transformations

- Examples of non-rigid affine transformations are both uniform and non-uniform scaling and shearing.

- * Curved, or elastic, transformations

A curved or elastic transformation can map a straight line onto a curve, for example, transformations described by polynomial functions. Rigid transformations can be described as curved transformations with zero elasticity. Elastic transformations can be approximated using local affine, that is, piecewise linear, algorithms when using a small granularity, that is, a fine level of detail.

- Usage of auxiliary means

Methods are distinguished by the amount of auxiliary information which must be used before, or during, image acquisition to allow subsequent registration (for example, the use of external or internal markers).

1.1.1 Prospective and Retrospective Methods

The methods listed in the previous section can also be categorized under what are called prospective and retrospective image registration methods [13].

Prospective methods always register artificial, as opposed to anatomical, landmarks (fiducials), for example, objects attached to the patient that are detectable in all pertinent modalities. If the position of the landmarks relative to the patient remains constant, a high degree of accuracy is possible. The disadvantage is that, since the calculation of the transformation is based only on a restricted number of landmarks, the resulting transformations are always rigid. If organ movements occur between the acquisition of different image series, these shifts cannot be considered correctly. Also, this procedure is invasive and inconvenient for the patient and staff.

Retrospective methods do not need any setup steps and are not subject to the conditions of image acquisition. To derive the necessary transformation, only patient related references (anatomical landmarks, surface of structures, etc.) are used.

Retrospective methods include the following:

- Feature-based methods
 - Point matching (Procrustes method)

Identification of corresponding points in the image series requires user interaction. An advantage of this method is that landmarks can be defined just in the therapy relevant image area, or locally. Therefore, the registration results will not be influenced by possible distortions in other regions of the image. A disadvantage is that it is difficult to locate corresponding landmarks in complementary image modalities precisely. Thus, a large amount of anatomical knowledge is required of the user. The precision of the resulting transformation depends on the precision with which these corresponding landmarks can be identified and depends therefore on the resolution (pixel size) and slice distance/thickness of the images.

- Surface matching

Rather than single points, this method uses the entire surface of a structure, for example, the patient's contour. A transformation is used which minimizes the distance between the surface of

corresponding structures. An example is the head/hat analogy, where the contour of the head, segmented in the first image data set, is put over the head contour in the second data set. In this way, uncertainties in the definition of single points do not carry much weight. The disadvantage is that the structures must be segmented in an initial step. Since segmentation is a non-trivial task and normally only semi-automatic segmentation procedures deliver acceptable results, these methods are time-consuming.

- Metrics of similarity

This retrospective approach is a method where a value for the similarity of the two data sets is calculated. The image data set, an area or volume, to be registered is transformed by an optimization procedure, Section 1.3, until the similarity with the first data set achieves a maximum. Voxel similarity measure techniques can be fully automatic and have an accuracy comparable to bone-implanted markers, but they can also fail [5, 25]. A common cause of failure can be that the images are poorly aligned at the start.

- Sum of squared intensity differences, a correlation-based distance measure
- Mutual information (an information theoretic technique)

Mutual Information measurements consider the intensity distribution of both image data sets and are, therefore, well suited to the

registration of multimodal image sequences, since no presumptions about the intensities have to be made (for example, it is possible that regions with a high intensity in the reference image correspond to regions with low, medium or high intensity in the template image). The mutual information reaches a maximum if, for a given intensity in image A , a distinct intensity in image B is found. (In other words, two different intensities in image A don't correspond to the same intensity in image B .) There are no restrictions concerning the intensity combinations. An advantage of this method is that it does not depend on the localization of single landmarks. No user interaction is required. It can be fully automatic in that it makes no assumption of the functional form or relationship between image intensities in the image to be registered. A disadvantage is that the calculation effort is much higher compared to point matching methods. Also, since this method considers the global intensity distribution and looks for the optimum over the image sequence on the whole, it is possible that the precision of the superposition is suboptimal in the area of interest. The theory and an application of mutual information-based medical image registration are presented in the chapters that follow.

1.2 Types of Transformation

Since we are three-dimensional, moving beings, registration should be four-dimensional, that is, it should include the three spatial dimensions as well the

temporal dimension. However, in practice, assumptions and approximations are made so that the body can be represented in fewer dimensions.

- 2D-2D

2D images may be registered simply by a rotation and two orthogonal translations. Differences in scaling from the real object to each of the images to be registered may also be necessary. Controlling the geometry of image acquisition is usually very difficult, so clinically relevant examples of 2D-2D registration are rare. For this study, 2D images are used for the sake of simplicity and the fact that these were the images that were available. These images are registered by a rotation and vertical and horizontal translations.

- 3D-3D

In the case of 3D-3D image registration, three translations and three rotations bring the images into registration. It is assumed that the imaged part of the body behaves as a rigid body, that is, the internal anatomy of the patient has not distorted or changed. Careful calibration of scanning devices/s is required to determine image scaling.

- 2D-3D

When establishing correspondence between 3D and projection images, 2D-3D registration may be required. The main application of these methods is in image-guided interventions.

- Time

This class of registration problem concerns image sequences that follow

some process that changes with time.

[5]

1.3 Image Registration as an Optimization Problem

Algorithms that directly calculate the transformation to establish correspondence between two images can be used where the images to be registered have very similar intensities and the transformation required to establish correspondence is small. This is the case with the point matching, or Procrustes, method described above. In other cases, a process of optimization is required. Optimization algorithms compute a cost, or similarity, function relating to how well two images are registered. The goal is to minimize, or maximize, the associated cost function. The cost function can be expressed as

$$C = C_{transformation} - C_{similarity},$$

where the first term characterizes the cost associated with particular transformations (lateral translations, rotations, nonlinear, etc.) and the second term characterizes the similarity between the reference and test images. Mutual information and sum of squared intensity differences are examples of cost functions. For mutual information-based image registration, as will be seen in the following chapters, the cost function increases as the images to be registered come into alignment. Conversely, when the sum of squared intensity differences algorithm is used, the cost function decreases as alignment increases.

Image registration as an optimization problem is hard to solve in that the problem is ill-posed [9]. Small changes of the input images can lead to completely different registration results. Optimization algorithms can converge to

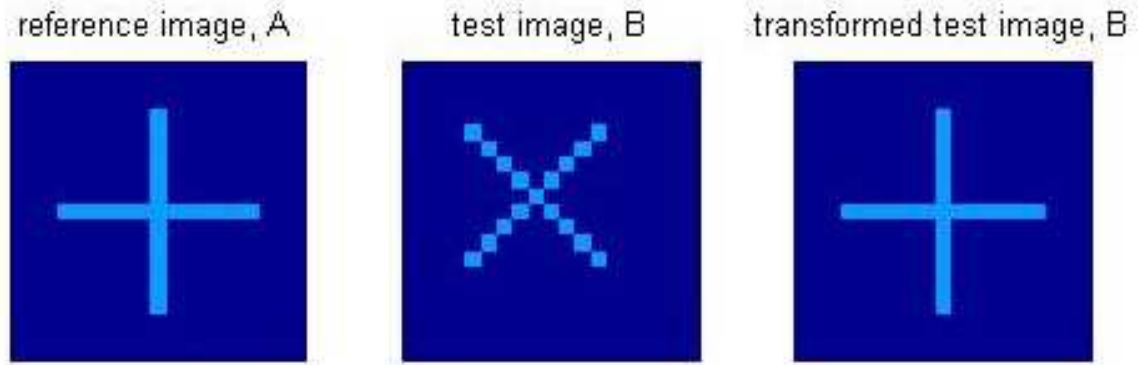


Figure 1.1: Registration example with multiple solutions. The image B is to be transformed to image A. The rightmost image shows B registered to A by two translations and a rotation of 45 degrees.

a suboptimal solution, a local optimum, which can cause registration to fail. Moreover, the solution may not be unique. This is illustrated in the simple example of Figure 1.1. In order to register the image labeled ‘B’ to the one labeled ‘A,’ several solutions are found: two pure translations followed by a rotation of 45 degrees, or equivalently, a rotation of 135 degrees followed by two pure translations, etc. The objective value at optimality is the same, but the solution path is different. Clearly, more information is required in order to determine which transformation should be used.

A property of the medical images that are registered is that they are discrete. That is, the object is sampled at a finite number of voxels or pixels. In this study, four discrete search algorithms are used with mutual information to find a solution to a medical image registration problem: a greedy algorithm, a genetic algorithm, and MATLAB’s `fminsearch` algorithm that uses the Nelder-Mead simplex (direct search) method.

Greedy algorithms are algorithms which follow the problem solving meta-heuristic of making the locally optimum choice at each stage with the hope of finding the global optimum. Greedy algorithms find the overall, or globally, optimal solution for some optimization problems, but may find suboptimal solutions for some instances of other problems. Greedy algorithms work in phases. In each phase, a decision is made that appears to be good, without regard for future consequences. Generally, this means that some local optimum is chosen. This ‘take what you can get now’ strategy is the source of the name for this class of algorithms. Hopefully, when the algorithm terminates, the local optimum is the global optimum. If this is the case, then the algorithm is correct. Otherwise, the algorithm has produced a suboptimal solution. In the case of medical image registration, it is probably safe to state that the global optimum is the required solution.

Genetic algorithms comprise a particular class of evolutionary algorithms inspired by the mechanisms of genetics, which has been applied to global optimization optimization problems. Genetic algorithms use biologically-derived techniques such as inheritance, mutation, natural selection, and recombination (or crossover). With genetic algorithms, each of a population consisting of a number of trial solutions, in this case image transformations, is evaluated to yield ‘fitness.’ A new generation is created, crossover being the dominate means of creating new members, from the better of them. The process is continued through a number of generations with the objective that the population should evolve to contain a global optimum or at least an acceptable solution.

The MATLAB program `fminsearch` uses the Nelder-Mead simplex (direct search) method of [7]. This is a direct search method that does not use numerical or analytic gradients. It is based on evaluating a function at the vertices of a simplex, then iteratively shrinking the simplex as better points are found until some desired bound is obtained [10].



Figure 2.1: Discrete noiseless binary channel.

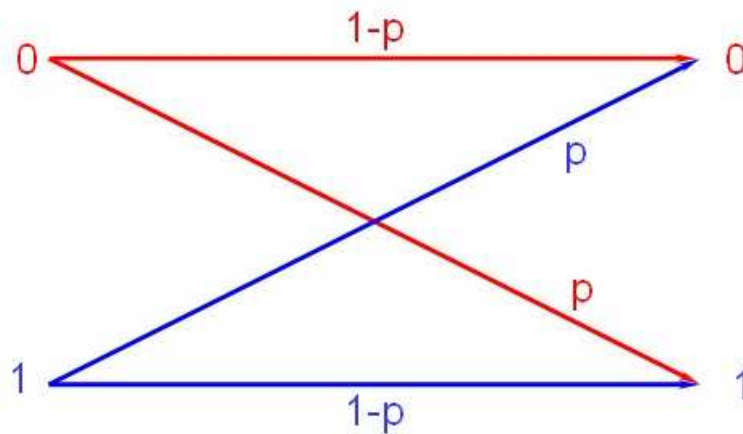


Figure 2.2: Discrete binary symmetric channel.

2. Mutual Information Theory

The derivation of mutual information from information theory and the associated mathematics are considered in this chapter.

2.1 Information

In 1928, R. V. L. Hartley [6] was the first to suggest the use of logarithms for the measure of information. The definition of information in the context of

information theory is an engineering definition based on probabilities, not on the meaning of symbols communicated to a human receiver. Therefore, it is somewhat counterintuitive.

Consider an alphabet of n symbols, a_1, a_2, \dots, a_n , each with its probability $p(a_1), p(a_2), \dots, p(a_n)$ of occurrence, communicated via some channel. If $p(a_1) = 1$, then a_1 will be received with certainty. This is the case of the noiseless binary communication channel pictured in Figure 2.1. Any transmission, 0 or 1, is received without error. There is no surprise in the occurrence of a_1 given $p(a_1)$, so no information is obtained. If a symbol with a low probability occurs, there is more surprise, more information. This might be the case for the binary channel pictured in Figure 2.2, where, for example, a 0 is sent and 1 is received. Thus, information is somewhat inversely related to the probability of occurrence.

The information from two independent symbols is the sum of the information from each separately (See Appendix A, Example 6). Since the probabilities of two independent choices are multiplied together to get the probability of the compound event, it is natural to define the amount of information as

Definition 2.1 *Information*

$$I(a_i) = \log \frac{1}{p(a_i)}.$$

[21]

Remark 2.2 *Information is inversely proportional to probability. Symbols with the least probability of occurring will provide the most information.*

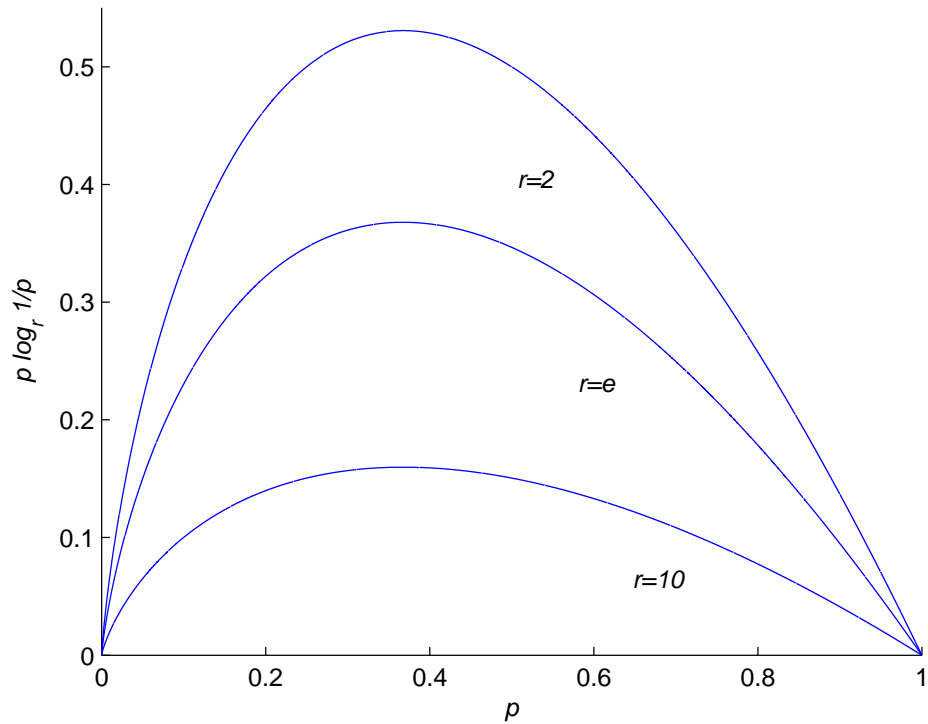


Figure 2.3: The entropy function for base 2, e , and 10 logs.

As a result,

$$I(a_1) + I(a_2) = \log \frac{1}{p(a_1)p(a_2)} = I(a_1, a_2).$$

Thus, given the two probabilities of events, the probability of both occurring, assuming independence, is a product. This product yields the amount of information as a sum.

2.2 Entropy (uncertainty or complexity)

The entropy function measures the amount of uncertainty, surprise, or information in the outcome of a situation, for example, the reception of a message or the

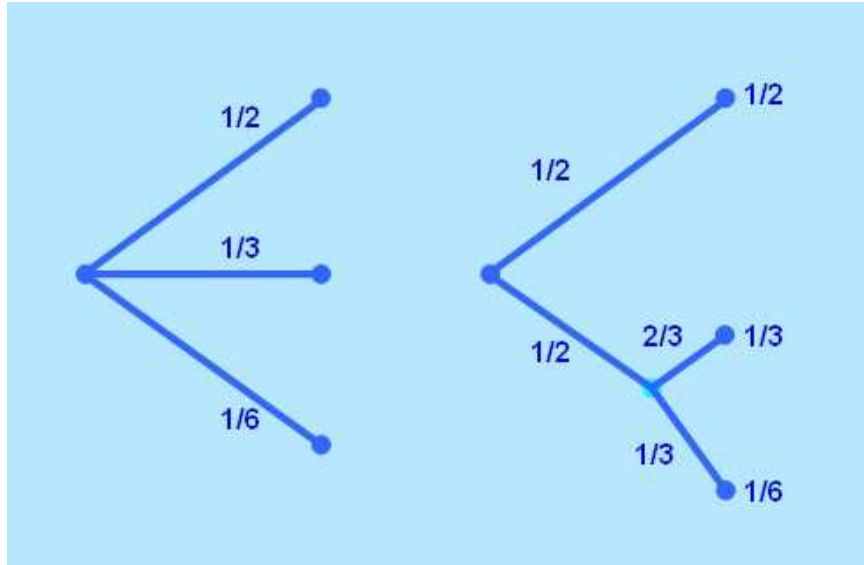


Figure 2.4: Decomposition of a choice from three possibilities.

outcome of some experiment. It involves only the distribution of probabilities. The Shannon-Wiener entropy measure H , originally developed as part of communication theory in the 1940s [14, 16], is the most commonly used measure of information in signal and image processing. This formula is derived from three conditions that a measure of uncertainty in a communication channel should satisfy.

1. The functional should be continuous in p .
2. If all p_i equal $\frac{1}{n}$, where n is the number of symbols, then H should be monotonically increasing in n .
3. If a choice is broken down into a sequence of choices, then the original value of H should be the weighted sum of the constituent H . That is,

$H(p_1, p_2, p_3) = H(p_1, p_2 + p_3) + (p_2 + p_3)H\left(\frac{p_2}{p_2+p_3}, \frac{p_3}{p_2+p_3}\right)$. The meaning of this condition is illustrated in Figure 2.4. On the left there are three possibilities: $p_1 = \frac{1}{2}$, $p_2 = \frac{1}{3}$, and $p_3 = \frac{1}{6}$. On the right a choice is made first between two possibilities each with probability $\frac{1}{2}$. If the second occurs, then a choice is made between probabilities $\frac{2}{3}$ and $\frac{1}{3}$. The final results have the same probabilities as before. In this special case it is required that

$$H\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right) = H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{2}H\left(\frac{2}{3}, \frac{1}{3}\right).$$

The coefficient $\frac{1}{2}$ is the weighting factor introduced because this second choice occurs only half the time.

Shannon proved that the $-\sum_{i=1}^n p_i \log p_i$ form was the only functional form satisfying all three conditions.

Theorem 2.3 *The only H satisfying all three assumptions is of the form:*

$$H = -K \sum_{i=1}^n p_i \log p_i,$$

where K is a positive constant.

Proof: Let $H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right) = A_n$. From condition (3) a choice can be decomposed from s^m equally likely possibilities into a series of m choices, each from s equally likely possibilities from which

$$A(s^m) = mA(s)$$

is obtained. Similarly

$$A(t^n) = nA(t)$$

n arbitrarily large can be chosen and m found to satisfy

$$s^m \leq t^n < s^{(m+1)}.$$

Taking the logarithms and dividing by $n \log s$ yields

$$\frac{m}{n} \leq \frac{\log t}{\log s} \leq \frac{m}{n} + \frac{1}{n} \text{ or } \left| \frac{m}{n} - \frac{\log t}{\log s} \right| < \epsilon,$$

where ϵ is arbitrarily small. From the monotonic property of $A(n)$,

$$A(s^m) \leq A(t^n) \leq A(s^{m+1})$$

$$mA(s) \leq nA(t) \leq (m+1)A(s).$$

Dividing by $nA(s)$,

$$\begin{aligned} \frac{m}{n} \leq \frac{A(t)}{A(s)} \leq \frac{m}{n} + \frac{1}{n} \text{ or } \left| \frac{m}{n} - \frac{A(t)}{A(s)} \right| < \epsilon \\ \left| \frac{A(t)}{A(s)} - \frac{\log t}{\log s} \right| \leq 2\epsilon \quad A(t) = K \log t, \end{aligned}$$

where K must be positive to satisfy (2).

Now suppose there is a choice from n possibilities with commensurable probabilities $p_i = \frac{n_i}{\sum n_i}$ where the n_i are integers. A choice can be broken down from $\sum n_i$ possibilities into a choice from n possibilities with probabilities p_1, \dots, p_n and then, if the i th were chosen, a choice from n_i with equal probabilities. Using condition (3) again, the total choice from $\sum n$ is equated as computed by two methods

$$K \log \sum n_i = H(p_1, \dots, p_n) + K \sum p_i \log n_i.$$

Hence

$$\begin{aligned} H &= K \left[\sum p_i \log \sum n_i - \sum p_i \log n_i \right] \\ &= -K \sum p_i \log \frac{n_i}{\sum n_i} = -K \sum p_i \log p_i. \end{aligned}$$

If the p_i are incommensurable, they may be approximated by rationals and the same expression must hold by the continuity assumption. Thus the expression holds in general. The choice of coefficient K is a matter of convenience and amounts to the choice of a unit of measure. ■

[15]

Entropy will have a maximum value if all symbols have equal probability of occurring (that is, $p_n = \frac{1}{n}$ for all i), and have a minimum value of zero if the probability of one symbol occurring is one, and the probability of all the others occurring is zero. [5]

Definition 2.4 *The **entropy** $H(X)$ of a discrete random variable X (Appendix A, Definition A.1) is a measure of the uncertainty of the random variable, and is defined by*

$$\begin{aligned} H(X) &= - \sum_{x \in X} p(x) \log p(x) \\ &= \sum_{x \in X} p(x) \log \frac{1}{p(x)}. \end{aligned}$$

The above quantity can also be expressed as $H(p)$. [2]

Remark 2.5 *The entropy of X can also be interpreted as the expected value (Appendix A, Definition A.3) of $\log \frac{1}{p(X)}$, where X is drawn according to the probability distribution function $p(x)$. Thus*

$$H(X) = E_p \log \frac{1}{p(X)}.$$

[2]

The following lemmas are immediate consequences of Definition 2.4.

Lemma 2.6 $H(X) \geq 0$.

Proof: $0 \leq p(x) \leq 1$ implies $\log(\frac{1}{p(x)}) \geq 0$. ■

Lemma 2.7

$$H_b(X) = (\log_b a)H_a(X).$$

(Please refer to Appendix C)

Proof:

$$\log_b p = \frac{\log_a p}{\log_a b} = \log_b a \log_a p.$$

■

[2]

Example 1. Entropy as a function of two probabilities (Figure 2.5).

Let

$$X = \begin{cases} 1 & \text{with probability } p, \\ 0 & \text{with probability } 1 - p. \end{cases}$$

Then

$$H(X) = -p \log p - (1 - p) \log(1 - p) \equiv H(p).$$

Using 2 as the base of the logarithm, $H(x) = 1$ when $p = \frac{1}{2}$. Figure 2.5 illustrates some of the basic properties of entropy. Entropy is a concave (Appendix B, Definition B.2) function of the distribution and equals 0 when p equals 0 or 1.

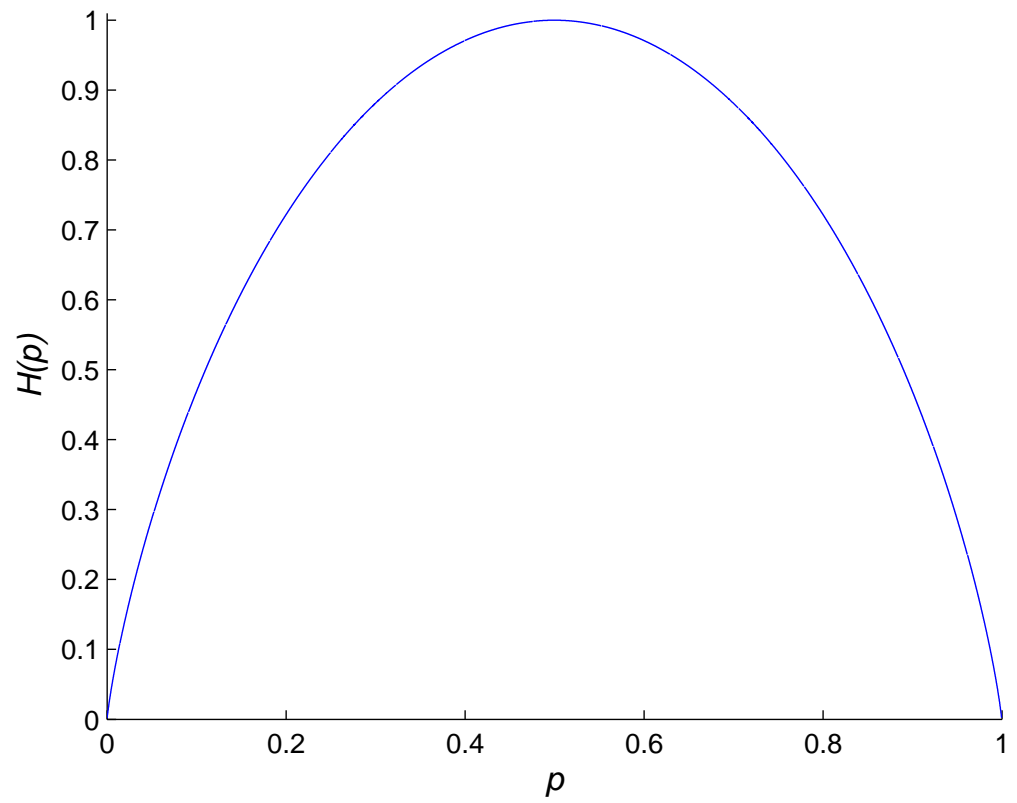


Figure 2.5: The entropy function (base 2 log) of two probabilities.

This makes sense since there is no uncertainty when p equals 0 or 1. Uncertainty is maximum when p equals $\frac{1}{2}$, which also corresponds to the maximum value of the entropy using base 2 logarithm.

Using Lemma 2.7 for the function $p \log_2 \frac{1}{p}$ (Figure 2.3, r=2) and deriving,

$$\begin{aligned} \frac{d}{dp} \left(p \log_2 \frac{1}{p} \right) &= \frac{d}{dp} \left(p \log_e \frac{1}{p} \right) \log_2 e \\ &= \log_2 \frac{1}{p} - \log_2 e. \end{aligned}$$

From the last equation, it can be seen that the slope at $p = 0$ is infinite and that the maximum of the entropy occurs at $p = \frac{1}{e}$.

Remark 2.8

$$\lim_{x \rightarrow 0} (x \log_e x) = 0.$$

$$\lim_{x \rightarrow 0} (x \log_e x) = \lim_{x \rightarrow 0} \left(\frac{\log_e x}{\frac{1}{x}} \right).$$

Application of l'Hospital's rule yields

$$\lim_{x \rightarrow 0} \left(\frac{\frac{1}{x}}{\frac{-1}{x^2}} \right) = \lim_{x \rightarrow 0} (-x) = 0.$$

Theorem 2.9 (*Fundamental Inequality*)

$$\sum_{i=1}^q x_i \log_2 \left(\frac{y_i}{x_i} \right) \leq 0$$

Proof: Fitting the tangent line at the point (0,1) on the $\log_e x$ function (Figure 2.6), the slope is

$$\frac{d(\log_e x)}{dx} \Big|_{x=1} = 1,$$

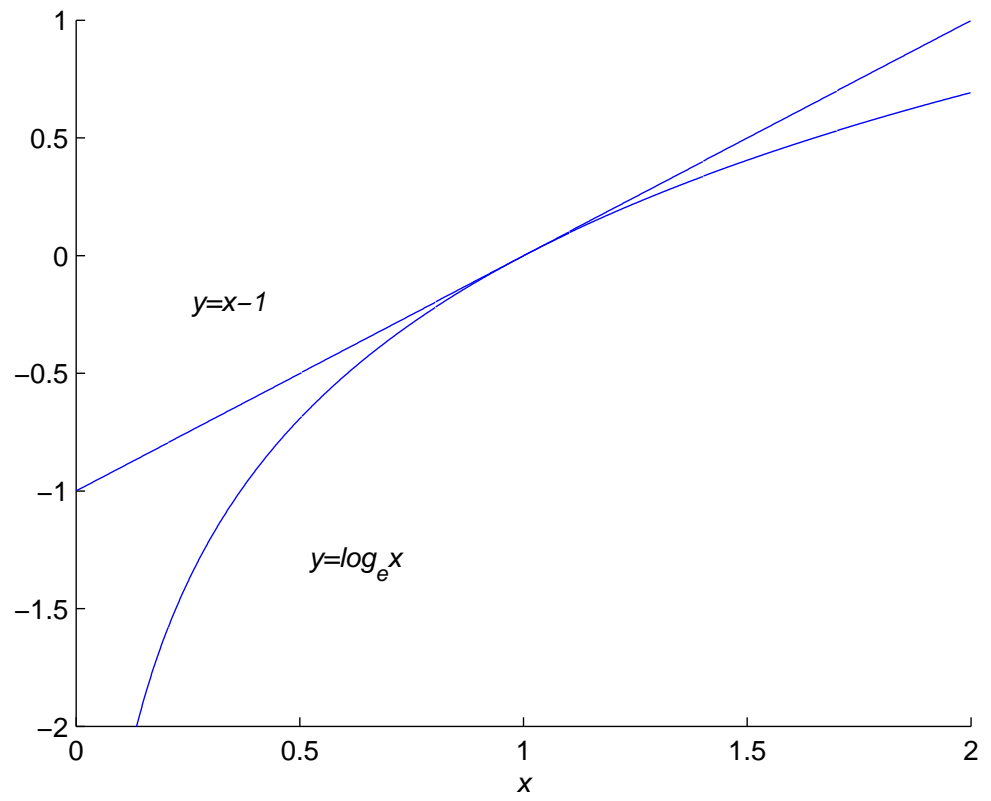


Figure 2.6: Bound on $\log_e(x)$ function.

so that the tangent line is

$$y - 0 = 1(x - 1)$$

or

$$y = x - 1.$$

Therefore, for all x greater than 0,

$$\log_e x \leq x - 1. \tag{2.1}$$

Let x_i and y_i be two probability distribution functions, where, of course, $\sum x_i = 1$ and $\sum y_i = 1$, $x_i, y_i \geq 0$. Now consider the following expression, using Lemma 2.7, involving both distributions.

$$\sum_{i=1}^q x_i \log_2 \left(\frac{y_i}{x_i} \right) = \frac{1}{\log_e 2} \sum_{i=1}^q x_i \log_e \left(\frac{y_i}{x_i} \right).$$

Using the relationship of Equation 2.1,

$$\begin{aligned} \frac{1}{\log_e 2} \sum_i x_i \log_e \left(\frac{y_i}{x_i} \right) &\leq \frac{1}{\log_e 2} \sum_i x_i \left(\frac{y_i}{x_i} - 1 \right) \\ &= \frac{1}{\log_e 2} \sum_i (y_i - x_i) \\ &= \frac{1}{\log_e 2} \left(\sum_i y_i - \sum_i x_i \right) \\ &= 0. \end{aligned}$$

Converting back to log base 2 yields the *Fundamental Inequality*,

$$\sum_{i=1}^q x_i \log_2 \left(\frac{y_i}{x_i} \right) \leq 0,$$

or, equivalently,

$$\sum_{i=1}^q x_i \log_2 \left(\frac{x_i}{y_i} \right) \geq 0.$$

For the last two equations, equality holds only when all the $x_i = y_i$. The lefthand sides of the last two equations are also called the *relative entropy* or *Kullback-Leibler distance* (See also Section 2.2.2). ■

[21]

2.2.1 Joint Entropy and Conditional Entropy

Definition 2.10 The *joint entropy* $H(X, Y)$ of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ is defined as

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y). \quad (2.2)$$

[2]

For the special case when X and Y are statistically independent (Appendix A, Definition A.5) for all i and j , the independence condition is expressed by

$$P(x_i, y_j) = p(x_i)p(y_j).$$

The joint entropy becomes

$$H(X, Y) = \sum_{i=1}^q \sum_{j=1}^s p(x_i)p(y_j) \left\{ \log \left[\frac{1}{p(x_i)} \right] + \log \left[\frac{1}{p(y_j)} \right] \right\}.$$

Since

$$\sum_{i=1}^q p(x_i) = 1 \text{ and } \sum_{j=1}^s p(y_j) = 1,$$

then

$$H(X, Y) = H(X) + H(Y).$$

Definition 2.11 If $(X, Y) \sim p(x, y)$, then the **conditional entropy** $H(Y|X)$ is defined as

$$\begin{aligned}
 H(Y|X) &= \sum_{x \in X} p(x) H(Y|X = x) \\
 &= - \sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log p(y|x) \\
 &= - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(y|x) \\
 &= - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)}. \tag{2.3}
 \end{aligned}$$

[2]

Remark 2.12 If X, Y independent, then

$$\begin{aligned}
 H(Y|X) &= - \sum_{x \in X} \sum_{y \in Y} p(x)p(y) \log p(y) \\
 &= \sum_{x \in X} \sum_{y \in Y} p(x)p(y) \log \frac{1}{p(y)} \\
 &= \sum_{y \in Y} p(y) \log \frac{1}{p(y)} \\
 &= H(Y).
 \end{aligned}$$

Theorem 2.13 (*Chain Rule for Entropy*)

Let X_1, X_2, \dots, X_n be drawn according to $p(x_1, x_2, \dots, x_n)$. Then

$$H(X_1, X_2, \dots, X_n) = \sum_{i=1}^n H(X_i | X_{i-1}, \dots, X_1).$$

Proof: By repeated application of the two-variable expansion rule for entropies,

$$\begin{aligned}
H(X_1, X_2) &= H(X_1) + H(X_2|X_1), \\
H(X_1, X_2, X_3) &= H(X_1) + H(X_2, X_3|X_1) \\
&= H(X_1) + H(X_2|X_1) + H(X_3|X_2, X_1), \\
&\vdots \\
H(X_1, \dots, X_n) &= H(X_1) + H(X_2|X_1) + \dots + H(X_n|X_{n-1}, \dots, X_1) \\
&= \sum_{i=1}^n H(X_i|X_{i-1}, \dots, X_1).
\end{aligned}$$

Alternative proof: Write $p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i|x_{i-1}, \dots, x_1)$, and evaluate

$$\begin{aligned}
H(X_1, X_2, \dots, X_n) &= - \sum_{x_1, x_2, \dots, x_n} p(x_1, x_2, \dots, x_n) \log p(x_1, x_2, \dots, x_n) \\
&= - \sum_{x_1, \dots, x_n} p(x_1, \dots, x_n) \log \prod_{i=1}^n p(x_i|x_{i-1}, \dots, x_1) \\
&= - \sum_{x_1, \dots, x_n} \sum_{i=1}^n p(x_1, \dots, x_n) \log p(x_i|x_{i-1}, \dots, x_1) \\
&= - \sum_{i=1}^n \sum_{x_1, \dots, x_n} p(x_1, \dots, x_n) \log p(x_i|x_{i-1}, \dots, x_1) \\
&= - \sum_{i=1}^n \sum_{x_1, \dots, x_i} p(x_1, \dots, x_i) \log p(x_i|x_{i-1}, \dots, x_1) \\
&= \sum_{i=1}^n H(X_i|X_{i-1}, \dots, X_1).
\end{aligned}$$

■

[2]

Theorem 2.14 (*Chain Rule for Conditional Entropy*)

$$H(X, Y) = H(X) + H(Y|X).$$

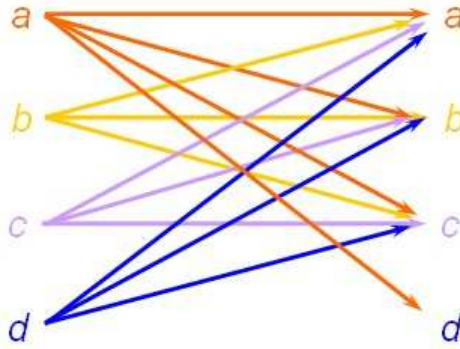


Figure 2.7: Noisy communication channel, Example 2.

Proof:

$$\begin{aligned}
 H(X, Y) &= - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y). \\
 &= - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x) p(y|x) \\
 &= - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x) - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(y|x) \\
 &= - \sum_{x \in X} p(x) \log p(x) - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(y|x) \\
 &= H(X) + H(Y|X).
 \end{aligned}$$

Equivalently,

$$\log p(X, Y) = \log p(X) + \log p(Y|X).$$

Taking the expectation of both sides of the equation yields the theorem. ■

[2]

Example 2. Noisy communication channel.

The input source to a noisy communication channel, Figure 2.7, is a random

variable X (Appendix A, Definition A.1) over the four symbols a, b, c, d . The output from the channel is a random variable Y over the same four symbols. The joint distribution (Appendix A, Definition A.4) of X and Y is given by

| | | | | | |
|---|-----|------|------|------|-----|
| | | X | | | |
| | | x=a | x=b | x=c | x=d |
| Y | y=a | 1/8 | 1/16 | 1/16 | 1/4 |
| | y=b | 1/16 | 1/8 | 1/16 | 0 |
| | y=c | 1/32 | 1/32 | 1/16 | 0 |
| | y=d | 1/32 | 1/32 | 1/16 | 0 |

The marginal distribution (Appendix A, Definition A.4) for X is $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$.

The marginal entropy of X , $H(X)$, is

$$\begin{aligned}
 - \sum_{i=1}^4 p(x_i) \log_2 p(x_i) &= 4 \left[-\frac{1}{4} (\log_2 1 - \log_2 4) \right] \\
 - \sum_{i=1}^4 p(x_i) \log_2 p(x_i) &= 4 \left(\frac{1}{2} \right) \\
 &= 2 \text{ bits.}
 \end{aligned}$$

Since base 2 log is used, the resulting unit of information is called a *bit* (binary digit). If base e is used, the unit of information is called a *nat*. If base 10 is used, the unit is called a *Hartley*, after R. V. L. Hartley, the first person to propose the use of the logarithmic measure of information.

The marginal distribution for Y is $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$.

The marginal entropy of Y , $H(Y)$, is

$$\begin{aligned} -\sum_{i=1}^4 p(y_i) \log_2 p(y_i) &= 2 \left[-\frac{1}{8} (\log_2 1 - \log_2 8) \right] + -\frac{1}{4} (\log_2 1 - \log_2 4) \\ &\quad + -\frac{1}{2} (\log_2 1 - \log_2 2) \\ &= 2 \left(\frac{3}{8} \right) + 2 \left(\frac{1}{2} \right) \\ &= \frac{7}{4} \text{ bits.} \end{aligned}$$

The joint entropy, $H(X, Y)$, of X and Y in bits is the sum of $-p \log p$ (Section 2.2.1, Equation 2.2) over all 16 probabilities in the joint distribution.

$$\begin{aligned} -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(x, y) &= -\frac{1}{4} (\log_2 1 - \log_2 4) + 2 \left[-\frac{1}{8} (\log_2 1 - \log_2 8) \right] \\ &\quad + 6 \left[-\frac{1}{16} (\log_2 1 - \log_2 16) \right] + 4 \left[-\frac{1}{32} (\log_2 1 - \log_2 32) \right] \\ &= \frac{1}{2} + 2 \left(\frac{3}{8} \right) + 6 \left(\frac{4}{16} \right) + 4 \left(\frac{5}{32} \right) \\ &= \frac{27}{8} \text{ bits.} \end{aligned}$$

The conditional entropy $H(Y|X)$ (Section 2.2.1, Equation 2.3) is

$$\begin{aligned}
-\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 \frac{p(x, y)}{p(x)} &= -\frac{1}{8} \log_2 \left(\frac{\frac{1}{8}}{\frac{1}{4}} \right) - \frac{1}{16} \log_2 \left(\frac{\frac{1}{16}}{\frac{1}{4}} \right) \\
&\quad - 2 \left[\frac{1}{32} \log_2 \left(\frac{\frac{1}{32}}{\frac{1}{4}} \right) \right] \\
&\quad - \frac{1}{16} \log_2 \left(\frac{\frac{1}{16}}{\frac{1}{4}} \right) - \frac{1}{8} \log_2 \left(\frac{\frac{1}{8}}{\frac{1}{4}} \right) \\
&\quad - 2 \left[\frac{1}{32} \log_2 \left(\frac{\frac{1}{32}}{\frac{1}{4}} \right) \right] \\
&\quad - 4 \left[\frac{1}{16} \log_2 \left(\frac{\frac{1}{16}}{\frac{1}{4}} \right) \right] - \frac{1}{4} \log_2 \left(\frac{\frac{1}{4}}{\frac{1}{4}} \right) \\
&= \frac{1}{8} + \frac{1}{8} + \frac{3}{16} + \frac{1}{8} + \frac{1}{8} + \frac{3}{16} + \frac{1}{2} + 0 \\
&= \frac{11}{8} \text{ bits.}
\end{aligned}$$

This example is continued in Section 2.3, where the mutual information for this example is calculated.

2.2.2 Relative Entropy

Definition 2.15 *The **relative entropy** or **Kullback-Leibler distance** between two probability mass functions $p(x)$ and $q(x)$ is defined as*

$$\begin{aligned}
D(p||q) &= \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \\
&= E_p \log \frac{p(X)}{q(X)},
\end{aligned} \tag{2.4}$$

where E denotes expectation.

Definition 2.16 *The **conditional relative entropy**, $D(p(y|x)||q(y|x))$ is the average of the relative entropies between the conditional probability mass functions $p(y|x)$ and $q(y|x)$ averaged over the probability mass function $p(x)$. More*

precisely,

$$\begin{aligned} D(p(y|x)||q(y|x)) &= \sum_x p(x) \sum_y p(y|x) \log \frac{p(y|x)}{q(y|x)} \\ &= E_{p(x,y)} \log \frac{p(Y|X)}{q(Y|X)}, \end{aligned}$$

where E denotes expectation.

Theorem 2.17 (*Chain Rule for Relative Entropy*)

$$D(p(x, y)||q(x, y)) = D(p(x)||q(x)) + D(p(y|x)||q(y|x)).$$

Proof:

$$\begin{aligned} D(p(x, y)||q(x, y)) &= \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{q(x, y)} \\ &= \sum_x \sum_y p(x, y) \log \frac{p(x)p(y|x)}{q(x)p(y|x)} \\ &= \sum_x \sum_y p(x, y) \log \frac{p(x)}{q(x)} + \sum_x \sum_y p(x, y) \log \frac{p(y|x)}{p(y|x)} \\ &= D(p(x)||q(x)) + D(p(y|x)||q(y|x)). \end{aligned}$$

■

[2]

Theorem 2.18 (*Jensen's Inequality*)

If f is a convex function (Appendix B, Definition B.1) and X is a random variable, then

$$Ef(X) \geq f(EX),$$

where E denotes expectation. If f is strictly convex, then equality implies that $X = EX$ with probability 1, that is, X is a constant.

Proof: For a two mass point distribution, the inequality becomes

$$p_1 f(x_1) + p_2 f(x_2) \geq f(p_1 x_1 + p_2 x_2),$$

which follows directly from the definition of convex functions. Suppose that the theorem is true for distributions with $k - 1$ mass points. Then writing $p'_i = \frac{p_i}{1 - p_k}$ for $i = 1, 2, \dots, k - 1$,

$$\begin{aligned} \sum_{i=1}^k p_i f(x_i) &= p_k f(x_k) + (1 - p_k) \sum_{i=1}^{k-1} p'_i f(x_i) \\ &\geq p_k f(x_k) + (1 - p_k) f\left(\sum_{i=1}^{k-1} p'_i x_i\right) \\ &\geq f\left(p_k x_k + (1 - p_k) \sum_{i=1}^{k-1} p'_i x_i\right) \\ &= f\left(\sum_{i=1}^k p_i x_i\right), \end{aligned}$$

where the first inequality follows from the induction hypothesis and the second follows from the definition of convexity. ■

[2]

Theorem 2.19 (*Log sum inequality*)

For non-negative numbers, a_1, a_2, \dots, a_n and b_1, b_2, \dots, b_n ,

$$\sum_i^n a_i \log \frac{a_i}{b_i} \geq \left(\sum_i^n a_i\right) \log \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i}$$

with equality if and only if $\frac{a_i}{b_i} = \text{constant}$.

Proof: Assume without loss of generality that $a_i \geq 0$ and $b_i \geq 0$ (If $a_i = b_i = 0$, then the sum is 0). The function $f(t) = t \log t$ is strictly convex, since

$f''(t) = \frac{1}{t} \log e > 0$ for all positive t . Hence, by Jensen's inequality (Theorem 2.18),

$$\sum a_i f(t_i) \geq f\left(\sum a_i t_i\right)$$

for $a_i \geq 0$, $\sum_i a_i = 1$. Setting $a_i = \frac{b_i}{\sum_{j=1}^n b_j}$ and $t_i = \frac{a_i}{b_i}$,

$$\sum \frac{a_i}{\sum b_j} \log \frac{a_i}{b_i} \geq \sum \frac{a_i}{\sum b_j} \log \sum \frac{a_i}{\sum b_j},$$

is obtained which is the log sum inequality. ■

[2]

Theorem 2.20 $D(p||q)$ is convex in the pair (p, q) , that is, if (p_1, q_1) and (p_2, q_2) are two pairs of probability mass functions, then

$$D(\lambda p_1 + (1 - \lambda)p_2 || \lambda q_1 + (1 - \lambda)q_2) \leq \lambda D(p_1 || q_1) + (1 - \lambda)D(p_2 || q_2) \quad (2.5)$$

for all $0 \leq \lambda \leq 1$.

Proof: The log sum inequality is applied to a term on the lefthand side of 2.5, that is,

$$\begin{aligned} & (\lambda p_1(x) + (1 - \lambda)p_2(x)) \log \frac{\lambda p_1(x) + (1 - \lambda)p_2(x)}{\lambda q_1(x) + (1 - \lambda)q_2(x)} \\ & \leq \lambda p_1(x) \log \frac{\lambda p_1(x)}{\lambda q_1(x)} + (1 - \lambda)p_2(x) \log \frac{(1 - \lambda)p_2(x)}{(1 - \lambda)q_2(x)}. \end{aligned}$$

Summing this over all x , the desired property is obtained. ■

[2]

Theorem 2.21 (*Information Inequality*)

Let $p(x)$, $q(x)$, $x \in \mathcal{H}$, be two probability distribution functions. Then

$$D(p||q) \geq 0$$

with equality if and only if

$$p(x) = q(x) \text{ for all } x.$$

Proof: Let $A = \{x : p(x) > 0\}$ be the support of $p(x)$. Then

$$\begin{aligned}
-D(p||q) &= -\sum_{x \in A} p(x) \log \frac{p(x)}{q(x)} \\
&= \sum_{x \in A} p(x) \log \frac{q(x)}{p(x)} \\
&\leq \log \sum_{x \in A} p(x) \frac{q(x)}{p(x)} && (2.6) \\
&= \log \sum_{x \in A} q(x) \\
&\leq \log \sum_{x \in \mathcal{X}} q(x) \\
&= \log 1 \\
&= 0,
\end{aligned}$$

where 2.6 follows from Theorem 2.18. Since $\log t$ is a strictly concave function of t , there is equality in 2.6 if and only if $\frac{q(x)}{p(x)} = 1$ everywhere, that is, $p(x) = q(x)$. Hence $D(p||q) = 0$ if and only if $p(x) = q(x)$ for all x . ■

[2]

Theorem 2.22 $H(X) \leq \log |\mathcal{H}|$, where $|\mathcal{H}|$ denotes the number of elements in the range of X , with equality if and only if X has a uniform distribution over \mathcal{H} .

Proof: Let $u(x) = \frac{1}{|\mathcal{H}|}$ be the uniform probability mass function over \mathcal{H} , and let $p(x)$ be the probability mass function (Definition A.2, Appendix A) for X .

Then

$$D(p||u) = \sum p(x) \log \frac{p(x)}{u(x)} = \log |\mathcal{H}| - H(X).$$

By the non-negativity of relative entropy,

$$0 \leq D(p||u) = \log |\mathcal{H}| - H(X).$$

■

Corollary 2.23

$$D(p(y|x)||q(y|x)) \geq 0,$$

with equality if and only if $p(y|x) = q(y|x)$ for all y and x with $p(x) > 0$.

[2]

2.2.3 Concavity of Entropy

Theorem 2.24 (*Concavity of entropy*)

$H(p)$ is a concave (Appendix B, Theorem B.1) function of p .

Proof:

$$H(p) = \log |\mathcal{H}| - D(p||u),$$

where u is the uniform distribution on $|\mathcal{H}|$ outcomes. The concavity of H then follows directly from the convexity of D .

Alternative Proof: Let X_1 be a random variable with distribution p_1 taking on values in a set A . Let X_2 be another random variable with distribution p_2 on the same set. Let

$$\theta = \begin{cases} 1 & \text{with probability } \lambda \\ 2 & \text{with probability } 1 - \lambda \end{cases}$$

Let $Z = X_\theta$. Then the distribution of Z is $\lambda p_1 + (1 - \lambda)p_2$. Since conditioning reduces entropy (Theorem 2.31), we have

$$H(Z) \geq H(Z|\theta),$$

or equivalently,

$$H(\lambda p_1 + (1 - \lambda)p_2) \geq \lambda H(p_1) + (1 - \lambda)H(p_2),$$

which proves the concavity of the entropy as a function of the distribution. ■

[2]

2.2.4 Entropy of a Signaling System

Consider again the communication channel of Section 2.1. If $I(a_i)$ units of information are obtained when the symbol a_i is received, then, on the average, since $p(a_i)$ is the probability of getting the information, $I(a_i)$, for each symbol a_i ,

$$p(a_i)I(a_i) = p(a_i) \log \frac{1}{p(a_i)}.$$

From this it follows that, on the average, over the entire alphabet of symbols a_i ,

$$\sum_{i=1}^n p(a_i) \log \frac{1}{p(a_i)}$$

will be received. This important quantity is the entropy of the signaling system, A , having symbols a_i with probabilities $p(a_i)$.

$$H(A) = \sum_{i=1}^n p(a_i) \log \frac{1}{p(a_i)}$$

2.2.5 Entropy and Kolmogorov Complexity

The Kolmogorov Complexity is approximately equal to the Shannon entropy H . Entropy is often called *information* in that it is a bound length on the code that is required to transmit a message.

Kolmogorov complexity (also called minimum description length principle, Chaitin complexity, descriptive complexity, shortest program length, and algorithmic entropy) is related to communication, an application of information theory. If the sender and receiver agree upon a specification method, such as an encoding or compression technique, then a message a can be transmitted as b and decoded given some fixed method L , denoted $L(b) = a$. The cost of the transmission of a is the length of the transmitted message, b . The least cost is the minimum length of such a message. The minimum length is the entropy of a given the method of transmission. The complexity of the minimum description length is universal, that is, computer independent.

Definition 2.25 *The **Kolmogorov complexity** of a binary string a , $K(a)$, is defined as the length of the shortest binary program for computing the string*

$$K(a) = \min[U(b) = a],$$

where U represents the abstract universal Turing computer that can implement any algorithm and compute any computable function.

[2, 3]

Remark 2.26 *Kolmogorov Complexity is reminiscent of Ockham's Razor, a principle formulated by William of Ockham (1300s) stating that terms, concepts, assumption, etc., must not be multiplied beyond necessity. According to*

[2], Einstein allegedly said “Everything should be made as simple as possible, but no simpler.”

Theorem 2.27 (*The Fundamental Theorem for a Noiseless Channel*) *Let a source have entropy H (bits per symbol) and a channel have a capacity C (bits per second). Then it is possible to encode the output of the source in such a way as to transmit at the average rate $\frac{C}{H} - \epsilon$ symbols per second over the channel where ϵ is arbitrarily small. It is not possible to transmit at an average rate greater than $\frac{C}{H}$.*

Proof: The converse part of the theorem, that $\frac{C}{H}$ cannot be exceeded, may be proven by noting that the entropy of the channel input per second is equal to that of the source, since the transmitter must be non-singular, and also this entropy cannot exceed the channel capacity. Hence $H' \leq C$ and the number of symbols per second $= \frac{H'}{H} \leq \frac{C}{H}$.

To prove the first part of the theorem, consider the set of all sequences of N symbols produced by the source. For large N , the symbols can be divided into two groups, one containing less than $2^{(H+\eta)N}$ members and the second containing less than 2^{RN} members (where R is the logarithm of the number of different symbols) and having a total probability less than μ . As N increases and η and μ approach zero. The number of signals of duration T in the channel is greater than $2^{(C-\theta)T}$ with θ small when T is large. If

$$T = \left(\frac{H}{C} + \lambda \right) N$$

is chosen, then there will be a sufficient number of sequences of channel symbols for the high probability group when N and T are sufficiently large (however small

λ) and also some additional ones. The high probability group is coded in an arbitrary one-to-one way into this set. The remaining sequences are represented by larger sequences, starting and ending with one of the sequences not used for the high probability group. This special sequence acts as a start and stop signal for a different code. In between, a sufficient time is allowed to give enough different sequences for all the low probability messages. This will require

$$T_1 = \left(\frac{R}{C} + \varphi \right) N$$

where φ is small. The mean rate of transmission in message symbols per second will then be greater than

$$\left[(1 - \delta) \frac{T}{N} + \delta \frac{T_1}{N} \right]^{-1} = \left[(1 - \delta) \left(\frac{H}{C} + \lambda \right) + \delta \left(\frac{R}{C} + \varphi \right) \right]^{-1}.$$

As N increases, δ , λ and φ approach zero and the rate approaches $\frac{C}{H}$. ■

Remark 2.28 *If a source can only produce one particular message, its entropy is zero and no channel is required.*

[15]

2.3 Mutual Information

Mutual information is a measure of the amount of information that one random variable contains about another random variable. It is the reduction in the uncertainty of one random variable due to the knowledge of the other.

C. E. Shannon first presented the functional form of mutual information in 1948. He defined it as the “rate of transmission” in a noisy communication channel between source and receiver. [14]

Consider a communication channel where the input symbols are a_i and the output are b_j . The channel is defined by the conditional probabilities $P_{i,j}$. Prior to reception, the probability of the input symbol a_i was $p(a_i)$. This is the *a priori* probability of a_i . After reception of b_j , the probability that the input symbol was a_i becomes $P(a_i|b_j)$, the conditional probability that we sent a_i given that we received b_j . This is an *a posteriori* probability of a_i . The change in the probability measures how much the receiver learned from the reception of the b_j . In an ideal channel with no noise, the *a posteriori* probability is 1, since we are certain from the received b_j exactly what was sent. In practical systems there are finite nonzero probabilities that errors will occur and the receiver cannot be absolutely sure what was sent. The difference between the information uncertainty before (the *a priori* probabilities) and after (the *a posteriori* probabilities) reception of a b_j measures the gain in information due to the reception of the b_j . This information is called the mutual information and is defined as

Definition 2.29 (*Mutual Information*)

$$I(a_i, b_j) = \log \left[\frac{1}{p(a_i)} \right] - \log \left[\frac{1}{P(a_i|b_j)} \right] = \log \left[\frac{P(a_i|b_j)}{p(a_i)} \right].$$

[21]

If $p(a_i)$ equals $P(a_i|b_j)$, then no information has been gained and the mutual information is zero, that is, no information has been transmitted. Only when something new has been learned about the probabilities of the a_i from the received b_j is the mutual information positive.

Multiplying the numerator and denominator of the rightmost log term by $p(b_j)$ yields

$$\frac{P(a_i|b_j)}{p(a_i)} = \frac{P(a_i|b_j)p(b_j)}{p(a_i)p(b_j)} = \frac{P(a_i, b_j)}{p(a_i)p(b_j)} = \frac{P(b_j|a_i)p(a_i)}{p(a_i)p(b_j)} = \frac{P(b_j|a_i)}{p(b_j)}.$$

Therefore,

$$I(a_i, b_j) = \log \left[\frac{P(a_i, b_j)}{p(a_i)p(b_j)} \right] = I(b_j, a_i).$$

From the symmetry of the a_i and b_j ,

$$\begin{aligned} I(a_i, b_j) &= \log \left[\frac{P(a_i|b_j)}{p(a_i)} \right], \\ I(b_j, a_i) &= \log \left[\frac{P(b_j|a_i)}{p(b_j)} \right], \end{aligned}$$

and

$$I(a_i, b_j) = I(b_j, a_i).$$

Additionally,

$$I(a_i, b_j) \leq I(a_i).$$

This follows from Definition 2.1,

$$I(a_i) = \log \left[\frac{1}{p(a_i)} \right],$$

and

$$\begin{aligned} I(a_i, b_j) &= \log P(a_i|b_j) + I(a_i) \\ &= \log P(a_i|b_j) + \log \frac{1}{p(a_i)} \\ &= \log P(a_i|b_j) - \log p(a_i) \end{aligned} \tag{2.7}$$

$$= \log \frac{P(a_i|b_j)}{p(a_i)}. \tag{2.8}$$

Since the maximum for the $P(a_i|b_j)$ is 1,

$$I(a_i, b_j) \leq I(a_i),$$

where equality occurs when $P(a_i|b_j) = 1$.

If a_i and b_j are independent, that is, if

$$P(a_i|b_j) = p(a_i), \text{ or equivalently}$$

$$P(a_i, b_j) = p(a_i)p(b_j), \text{ then}$$

$$I(a_i, b_j) = 0. \tag{2.9}$$

This follows from 2.7 and 2.8 above.

Because of noise, the behavior of a channel can be understood only on the average. Therefore, the mutual information must be averaged over the alphabets, A and B , using the appropriate probabilities.

$$\begin{aligned} I(A, b_j) &= \sum_i P(a_i|b_j) I(a_i, b_j) \\ &= \sum_i P(a_i|b_j) \log \left[\frac{P(a_i|b_j)}{p(a_i)} \right]. \end{aligned}$$

Similarly,

$$I(a_i, B) = \sum_j P(b_j|a_i) \log \left[\frac{P(b_j|a_i)}{p(b_j)} \right].$$

These two equations are called the *conditional mutual information*. $I(A, b_j)$ measures the information gain about the alphabet A provided by the reception of b_j . $I(a_i, B)$ is the information gain about the alphabet B given that a_i was

sent. Finally,

$$\begin{aligned}
 I(A, B) &= \sum_i P(a_i) I(a_i, B) \\
 &= \sum_i \sum_j P(a_i, b_j) \log \left[\frac{P(a_i, b_j)}{p(a_i)p(b_j)} \right] \\
 &= I(B, A).
 \end{aligned} \tag{2.10}$$

by symmetry. This equation, the system mutual information, provides a measure of the information gain of the whole system and does not depend on the individual input and output symbols but only on their frequencies. From Equation 2.10, it can be seen that the mutual information $I(B, A)$ is the relative entropy (Equation 2.4) between the joint distribution, $P(a_i, b_j)$, and the product $p(a_i)p(b_j)$.

The system mutual information has the properties

$$I(A, B) \geq 0 \tag{2.11}$$

$I(A, B) = 0$ if and only if A and B are independent. (Equation 2.9)

$I(A, B) = I(B, A)$ (Equation 2.10)

[21]

The proof of Equation 2.11 lies in the following corollary to Theorem 2.21.

Corollary 2.30 (*Non-negativity of mutual information*)

For any two random variables (alphabets, for example), A, B ,

$$I(A, B) \geq 0,$$

with equality if and only if A and B are independent.

Proof: $I(A, B) = D(p(a, b) || p(a)p(b)) \geq 0$, with equality if and only if $p(a, b) = p(a)p(b)$, that is, A and B are independent. ■

Theorem 2.31 (*Conditioning reduces entropy*)

$$H(A|B) \leq H(A)$$

with equality if and only if A and B are independent.

Proof:

$$0 \leq I(A, B) = H(A) - H(A|B).$$

■

Remark 2.32 *Intuitively, the theorem says that knowing another random variable B can only reduce the uncertainty in A . This is true only on average. Specifically, $H(A|B = b)$ may be greater than or less than or equal to $H(A)$, but on the average $H(A|B) = \sum_y p(y)H(A|B = b) \leq H(A)$. For example, in a court case, specific new evidence might increase uncertainty, but on the average evidence decreases uncertainty.*

[2]

The various entropies can be related to each other by the following algebraic

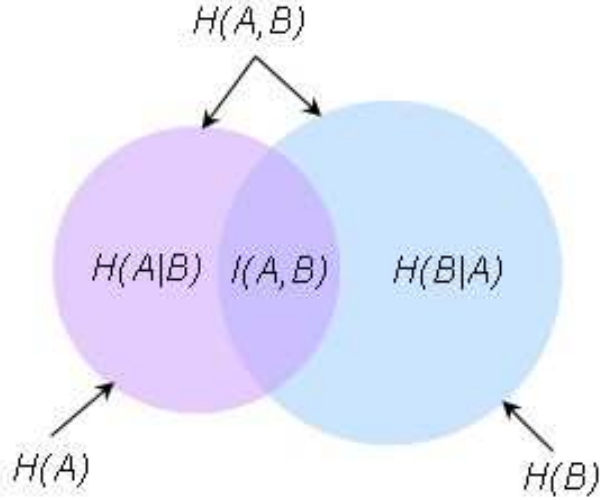


Figure 2.8: Relationship between entropy and mutual information. The mutual information $I(A, B)$ corresponds to the intersection of the information in A with the information in B .

manipulations:

$$\begin{aligned}
 I(A, B) &= \sum_i^q \sum_j^s P(a_i, b_j) \log \left[\frac{P(a_i, b_j)}{p(a_i)p(b_j)} \right] \\
 &= \sum_i^q \sum_j^s P(a_i, b_j) [\log P(a_i, b_j) - \log p(a_i) - \log p(b_j)] \\
 &= - \sum_i^q \sum_j^s P(a_i, b_j) \log \left[\frac{1}{P(a_i, b_j)} \right] \tag{2.12}
 \end{aligned}$$

$$\begin{aligned}
 &+ \sum_i^q p(a_i) \log \left[\frac{1}{p(a_i)} \right] + \sum_j^s p(b_j) \log \left[\frac{1}{p(b_j)} \right] \\
 &= H(A) + H(B) - H(A, B) \geq 0. \tag{2.13}
 \end{aligned}$$

From Theorem 2.14,

$$\begin{aligned}
 H(A, B) &= H(A) + H(B|A) \\
 &= H(B) + H(A|B),
 \end{aligned}$$

and two results are obtained:

$$I(A, B) = H(A) - H(A|B) \geq 0 \quad (2.14)$$

$$= H(B) - H(B|A) \geq 0. \quad (2.15)$$

Therefore,

$$0 \leq H(A|B) \leq H(A)$$

$$0 \leq H(B|A) \leq H(B)$$

and

$$H(A, B) \leq H(A) + H(B).$$

[21]

Theorem 2.33 (*Concavity of Mutual Information*)

Let $(A, B) \sim p(a, b) = p(a)p(b|a)$. The mutual information $I(A, B)$ is a concave function of $p(a)$ for fixed $p(b|a)$ and a convex function of $p(b|a)$ for fixed $p(a)$.

Proof: To prove the first part, expand the mutual information

$$I(A, B) = H(B) - H(B|A) = H(B) - \sum p(a)H(B|A = a).$$

If $p(b|a)$ is fixed, then $p(b)$ is a linear function of $p(a)$ (Appendix A, Equation A.3). Hence $H(B)$, which is a concave function of $p(b)$, is a concave function of $p(a)$. The second term is a linear function of $p(a)$. Hence the difference is a concave function of $p(a)$.

To prove the second part, fix $p(a)$ and consider two different conditional distributions $p_1(b|a)$ and $p_2(b|a)$. The corresponding joint distributions are $p_1(a, b) = p(a)p_1(b|a)$ and $p_2(a, b) = p(a)p_2(b|a)$, and their respective marginals are $p(a)$, $p_1(b)$, and $p(a)$, $p_2(b)$. Consider a conditional distribution

$$p_\lambda(b|a) = \lambda p_1(b|a) + (l - \lambda)p_2(b|a)$$

that is a mixture of $p_1(b|a)$ and $p_2(b|a)$. The corresponding joint distribution is also a mixture of the corresponding joint distributions,

$$p_\lambda(a, b) = \lambda p_1(a, b) + (l - \lambda)p_2(a, b),$$

and the distribution of B is also a mixture

$$p_\lambda(b) = \lambda p_1(b) + (l - \lambda)p_2(b).$$

Hence, let $q_\lambda(a, b) = p(a)p_\lambda(b)$ be the product of the marginal distributions, to yield

$$q_\lambda(a, b) = \lambda q_1(a, b) + (l - \lambda)q_2(a, b).$$

Since the mutual information is the relative entropy between the joint distribution and the product of the marginals, that is,

$$I(A, B) = D(p_\lambda \| q_\lambda),$$

and relative entropy $D(p \| q)$ is a convex function of (p, q) , it follows that the mutual information is a convex function of the conditional distribution. ■

[2]

Definition 2.34 The *conditional mutual information* of random variables X and Y given Z is defined by

$$\begin{aligned} I(X, Y|Z) &= H(X|Z) - H(X|Y, Z) \\ &= E_{p(x,y,z)} \log \frac{p(X, Y|Z)}{p(X|Z)p(Y|Z)}. \end{aligned}$$

[2]

Mutual information also satisfies a chain rule.

Theorem 2.35 (*Chain Rule for Information*)

$$I(X_1, X_2, \dots, X_n, Y) = \sum_{i=1}^n I(X_i, Y|X_{i-1}, X_{i-2}, \dots, X_1).$$

Proof:

$$\begin{aligned} I(X_1, X_2, \dots, X_n, Y) &= H(X_1, X_2, \dots, X_n) - H(X_1, X_2, \dots, X_n|Y) \\ &= \sum_{i=1}^n H(X_i|X_{i-1}, \dots, X_1) - \sum_{i=1}^n H(X_i|X_{i-1}, \dots, X_1, Y) \\ &= \sum_{i=1}^n I(X_i, Y|X_1, X_2, \dots, X_{i-1}). \end{aligned}$$

■

[2]

Example 3. Example 2, Section 2.2.1, continued.

The mutual information between the two random variables, X and Y , in bits (Appendix C), using some of the answers obtained in Example 2, can be found several ways (Equations 2.13, 2.14, and 2.15):

$$\begin{aligned} I(X, Y) &= H(Y) - H(Y|X) \\ &= \frac{7}{4} - \frac{11}{8} \\ &= \frac{3}{8} \text{ bits.} \end{aligned}$$

$$\begin{aligned} I(X, Y) &= H(X) - H(X|Y) \\ &= 2 - \frac{13}{8} \\ &= \frac{3}{8} \text{ bits.} \end{aligned}$$

$$\begin{aligned} I(X, Y) &= H(Y) + H(X|Y) - H(X, Y) \\ &= 2 + \frac{7}{4} - \frac{27}{8} \\ &= \frac{3}{8} \text{ bits.} \end{aligned}$$

3. Mutual Information Theory Applied to Image Registration

The mutual information measure for image registration has proven to be very robust and has resulted in fully automated 3D-to-3D rigid-body registration algorithms that are now in use [5, 18, 19, 20].

The use of mutual information as a basis for a registration metric was proposed independently by Collignon et al [1] and the MIT group [24] in 1995. Mutual information for image registration is defined (Equations 2.13, 2.14, and 2.15) as:

$$\begin{aligned} I(A(x), T(B(x))) &= H(A(x)) + H(T(B(x))) - H(A(x), T(B(x))) \\ &= H(A(x)) - H(A(x)|T(B(x))) \\ &= H(B(x)) - H(T(B(x))|A(x)), \end{aligned}$$

where $A(x)$ is a model, or reference, image and $T(B(x))$ is a transformation of the test image. The images, A and B , are matrices of pixel (picture element) values, x .

Mutual information measures the joint entropy, $H(A(x), T(B(x)))$, of the model image, A , and test image, B . At optimal alignment, the joint entropy is minimized with respect to the entropy of the overlapping part of the individual images, so that the mutual information is maximized. Minimizing the joint entropy minimizes the conditional entropy,

$$H(A(x)|T(B(x))) = H(A(x), T(B(x))) - H(B(x)),$$

since the entropy of the model is fixed.

The image B can be considered as having two parts - the part that is similar to the model, A , B_m , and the part that cannot be related to the model, B' .

$$B_m \equiv \{B(x) \ni T^{-1}(A(x)) \text{ is defined}\},$$

and,

$$B' \equiv B - B_m.$$

$$\begin{aligned} I(A, B) &= H(A) + H(B) - H(A, B) \\ &= H(A) + H(B_m, B') - H(A, B_m, B') \\ &= H(A) + H(B_m) + H(B') - H(A, B_m) - H(B') \\ &= H(A) + H(B_m) - H(A, B_m) \end{aligned} \tag{3.1}$$

The assumption is that B' is independent of A and B_m . Therefore, discarding pixels such as those representing background, removes them from the analysis and maximizes the mutual information. Equation 3.1 shows that maximizing the entropy of the modeled part of the image, $H(B_m)$, and minimizing the joint entropy, $H(A, B_m)$, maximizes the mutual information $I(A, B)$. [22]

For image registration, A and B are two image data sets. Entropy is calculated using the images' histograms or probability density functions (Appendix A, Definition A.2). Each point in one image, and its associated intensity, will correspond to a point, with its respective intensity, in the other. Scatter plots of these image intensities can be generated point by point. These are two-dimensional plots of image intensity of one image against corresponding image

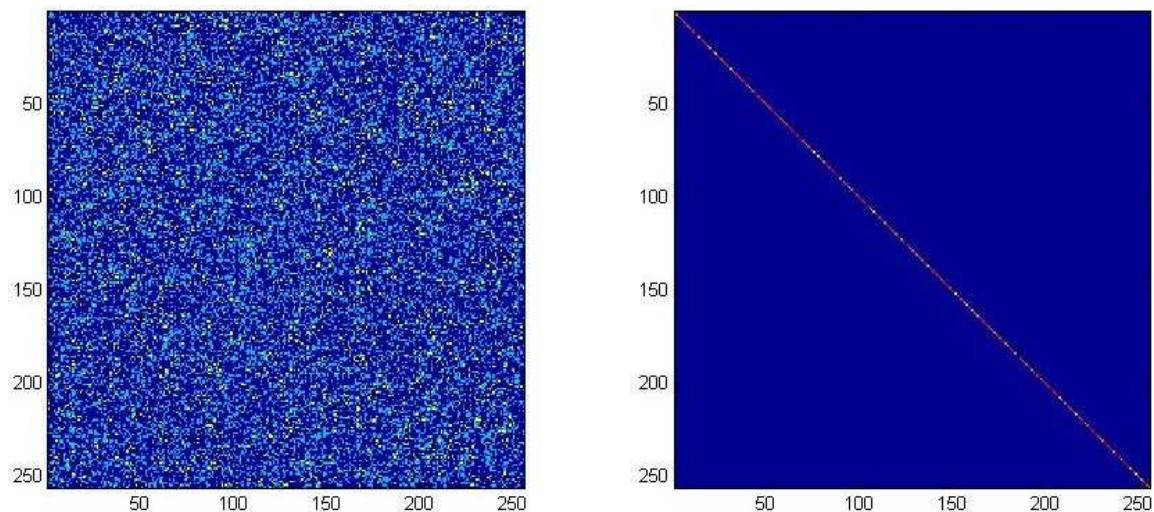


Figure 3.1: The plot on the left is the joint histogram of two randomly generated 150×150 matrices taking on values from 1 to 256. The plot on the right is the plot of the joint histogram of one matrix with itself.

intensity of the other. The resulting plot, a two-dimensional histogram, is called a joint intensity histogram. When divided by the number of contributing pixels, it is equal to the joint probability distribution (Appendix A, Definition A.4). For each pair of image intensities in the two images, the joint probability distribution provides a number equal to the probability that those intensities occur together at corresponding locations in the two images. To illustrate the idea of a joint histogram, refer to Figure 3.1. The plot on the left is the joint histogram of two randomly generated 150×150 matrices taking on values from 1 to 256. The plot on the right is the plot of the joint histogram of one matrix with itself. Therefore, it is diagonal as should be expected for identical matrices.

Joint entropy measures the amount of information in the combined images. If A and B are totally unrelated, then the joint entropy will be the sum of the entropies of the individual images (refer to Equation 2.12 and Figures 3.2 and

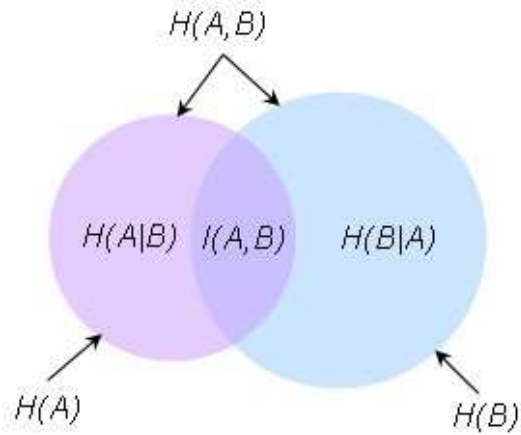


Figure 3.2: Relationship between entropy and mutual information. The mutual information $I(A, B)$ corresponds to the intersection of the information in A with the information in B .

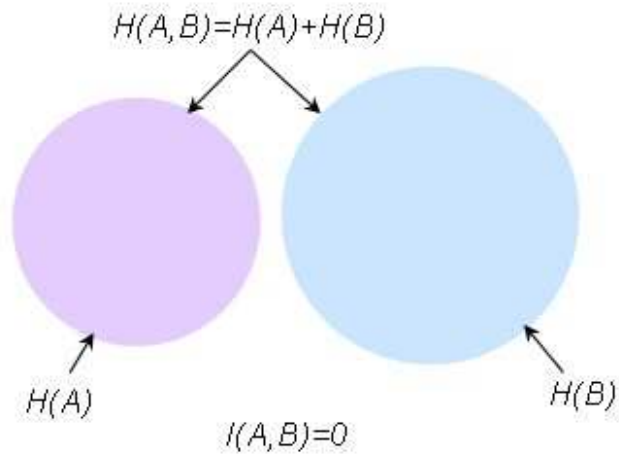


Figure 3.3: Venn diagram of the relationship between joint entropy and mutual information of totally unrelated images.

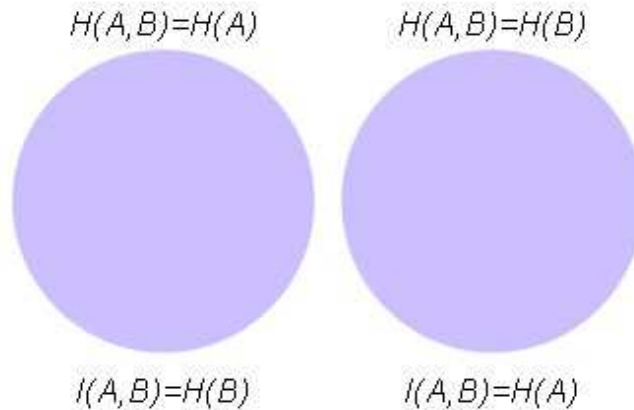


Figure 3.4: Venn diagram of the relationship between joint entropy and mutual information of totally related images.

3.3). Therefore, there will be no mutual information. Conversely, if A and B are completely dependent, that is, knowledge of the outcome of an event in A gives exact knowledge of the outcome of an event in B , then the joint entropy of the two images is equivalent to the entropy of the image A , so the mutual information is $H(B)$. If knowledge of the outcome of an event in B gives exact knowledge of the outcome of an event in A , then the joint entropy of the two images is equivalent to the entropy of the image B , so the mutual information is $H(A)$ (refer to Figure 3.4). The more closely related, that is, less independent, the images are, the smaller in absolute value the joint entropy compared to the sum of the individual entropies. That is,

$$I(A, B) = H(A) + H(B) - H(A, B).$$

This equation shows that maximizing the entropy of the test image, $H(B)$, and minimizing the joint entropy, $H(A, B)$, maximizes the mutual information $I(A, B)$.

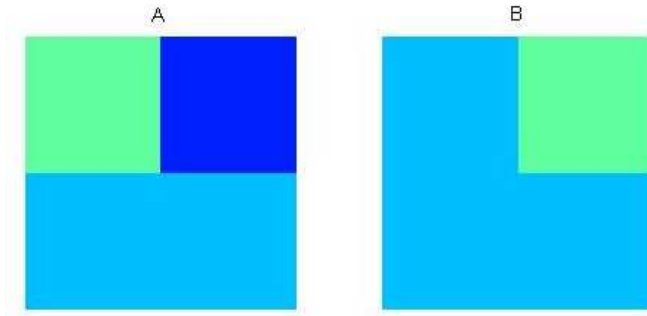


Figure 3.5: The image data sets, A and B .

Minimizing the joint entropy, calculated from the joint intensity histogram, was proposed by Studholme et al [17] and Collignon [1] as a basis for a registration method. However, joint entropy on its own does not provide a robust measure of image alignment, as it is often possible to find alternative alignments that result in much lower joint entropy. For example, mutual information, which is given by the difference between the sum of the entropies of the individual images at overlap and the joint entropy of the combined images, works better than simply joint entropy in regions of image background (low contrast of neighboring pixels) where there will be low joint entropy, but this is offset by low individual entropies as well, so that the overall mutual information will be low.

Example 4. A sample calculation of the mutual information between two matrices, image data sets, Figure 3.5.

Let

$$A = \begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 2 & 3 \\ 2 & 2 \end{bmatrix} .$$

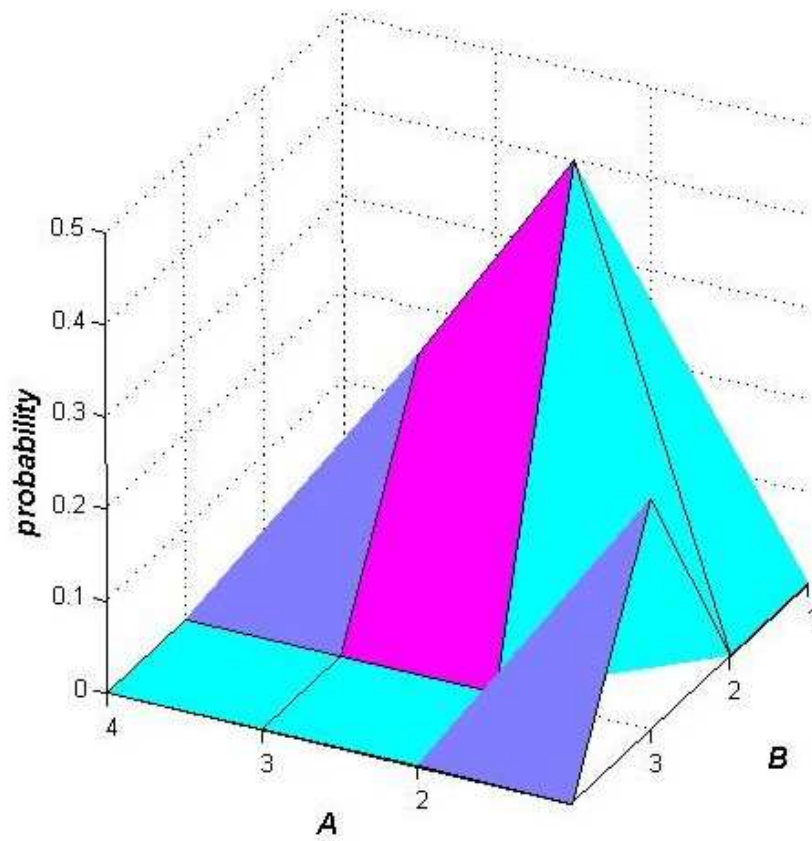


Figure 3.6: The joint probability distribution of Example 4 as a surface.

Determine the mutual information between the matrices A and B . That is, find

$$I(A, B) = \sum_i^q \sum_j^s P(a_i, b_j) \log \left[\frac{P(a_i, b_j)}{p(a_i)p(b_j)} \right], \text{ (Equation 2.10).}$$

Then

| | | | | | |
|----------|--|---------------|---------------|---------------|---------------|
| | | A | | | |
| | | 1 | 2 | 3 | $p(b_j)$ |
| 1 | | 0 | 0 | 0 | 0 |
| B 2 | | 0 | $\frac{1}{2}$ | $\frac{1}{4}$ | $\frac{3}{4}$ |
| 3 | | $\frac{1}{4}$ | 0 | 0 | $\frac{1}{4}$ |
| $p(a_i)$ | | $\frac{1}{4}$ | $\frac{1}{2}$ | $\frac{1}{4}$ | * |

is the joint probability distribution, where the rightmost column and bottom row consist of the marginal probabilities, $p(b_j)$ and $p(a_i)$, for the values b and a , respectively, and the nonzero values are the $P(a_i, b_j)$. Calculation of the mutual information yields

$$\begin{aligned} I(A, B) &= \frac{1}{2} \log \frac{\frac{1}{2}}{\frac{1}{8}} + \frac{1}{4} \log \frac{\frac{1}{4}}{\frac{3}{16}} + \frac{1}{4} \log \frac{\frac{1}{4}}{\frac{1}{16}} \\ &= \frac{1}{2} \log \frac{4}{3} + \frac{1}{4} \log \frac{4}{3} + \frac{1}{4} \log 4 \\ &\cong .5(.2877) + .25(.2877) + .25(1.3863) \\ &\cong .5623 \text{ nats.} \end{aligned}$$

It might be tempting to think that if image A were rotated -90° as in (Figure 3.7), then a greater value for mutual information would result since 3 out of 4 pixel values would match in intensity. However, that would not be the case. The mutual information would be the same. This technique makes no assumption about the relationship between image intensities in the images to be registered.

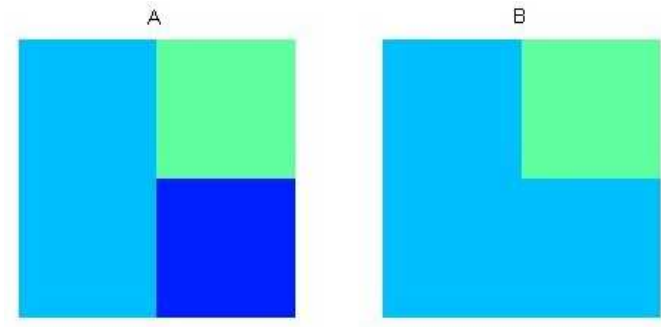


Figure 3.7: The image data sets A , rotated -90 degrees, and B .

Rotating $A -90^\circ$ yields

$$A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \quad \text{and} \quad B \text{ (unchanged)} = \begin{bmatrix} 2 & 3 \\ 2 & 2 \end{bmatrix}.$$

The joint probability distribution becomes

| | | A | | | $p(b_j)$ |
|----------|---|---------------|---------------|---------------|---------------|
| | | 1 | 2 | 3 | |
| B | 1 | 0 | 0 | 0 | 0 |
| | 2 | $\frac{1}{4}$ | $\frac{1}{2}$ | 0 | $\frac{3}{4}$ |
| | 3 | 0 | 0 | $\frac{1}{4}$ | $\frac{1}{4}$ |
| $p(a_i)$ | | $\frac{1}{4}$ | $\frac{1}{2}$ | $\frac{1}{4}$ | * |

and the calculation of the mutual information yields .5623 nats as in the case prior to the rotation of the image A .

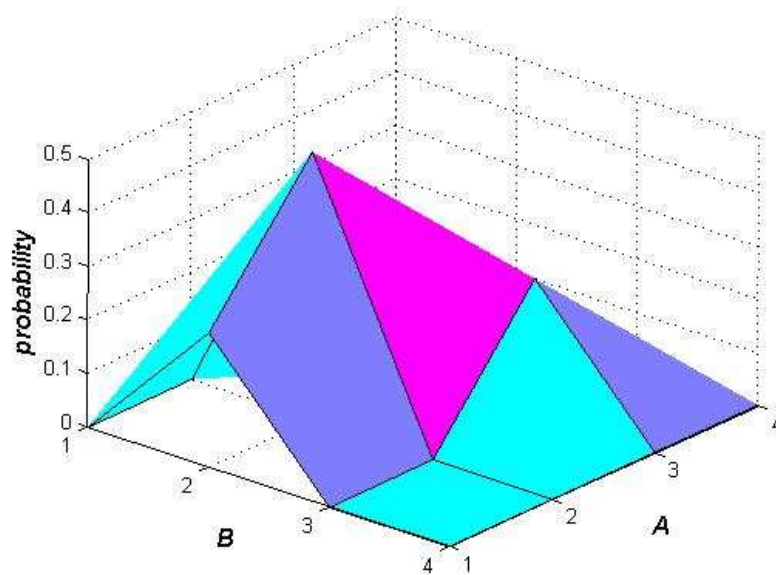


Figure 3.8: The joint probability distribution as a surface of Example 4 with image A rotated -90° as in Figure 3.7.

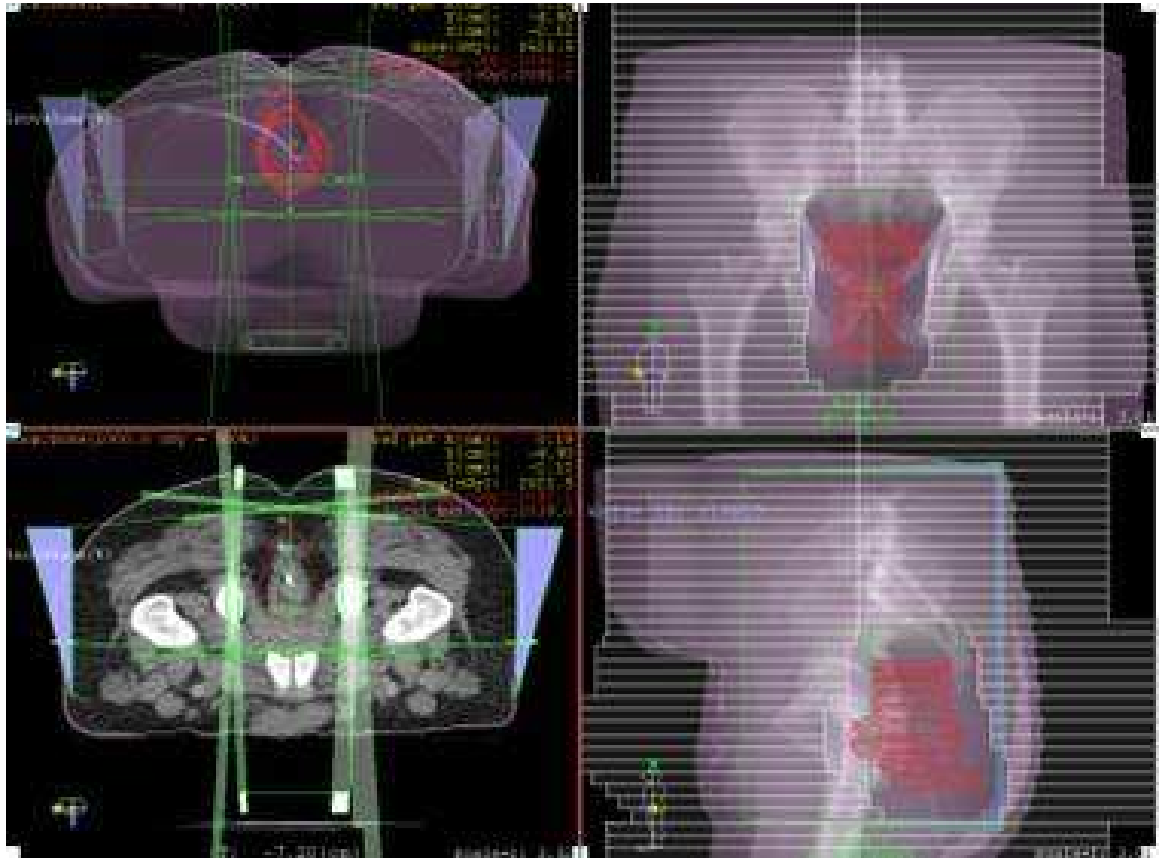


Figure 4.1: A typical radiation treatment plan [11].

4. Mutual Information-Based Registration of Digitally Reconstructed Radiographs (DRRs) and Electronic Portal Images (EPIs)

Electronic portal imaging devices (EPIDs) provide real-time digital images that allow for a time-efficient patient repositioning before radiation treatment. A

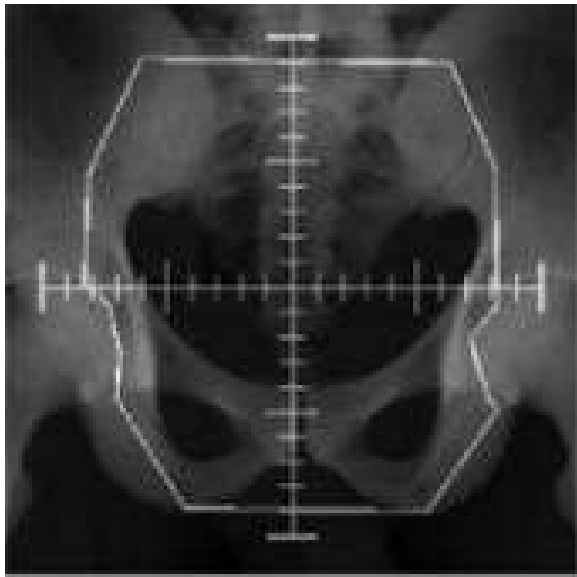


Figure 4.2: Typical aligned pelvis DRR [11].



Figure 4.3: Typical aligned pelvis EPI [11].

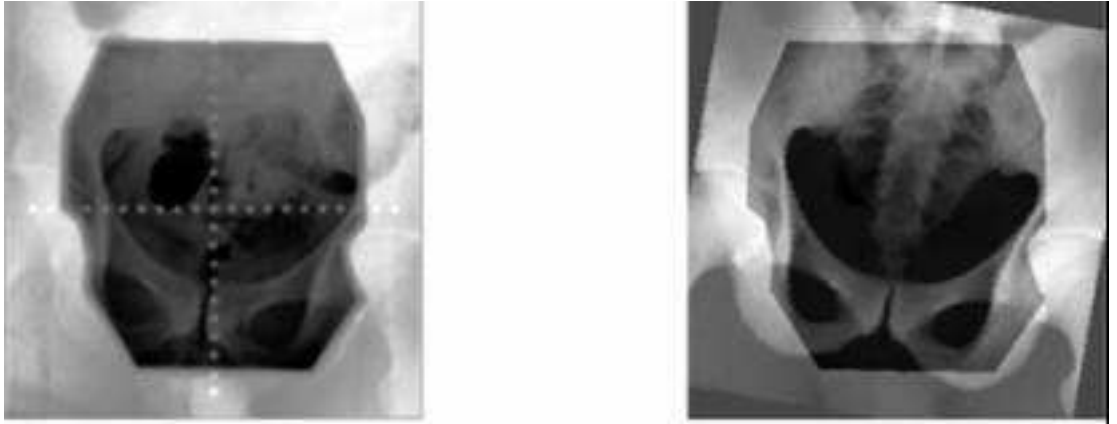


Figure 4.4: A misaligned EPI and the DRR automatically rotated to the EPI position [11].

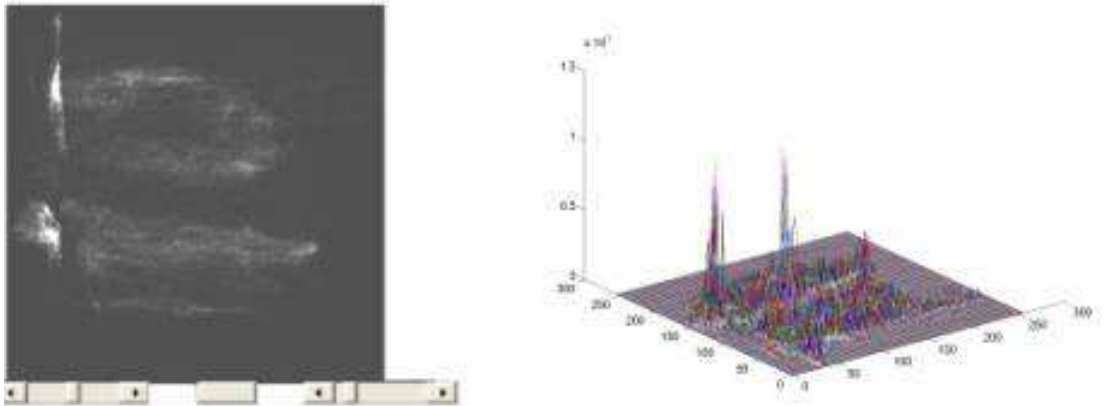


Figure 4.5: Joint histogram of misaligned images. Referring to the rightmost plot, the x- and y- axes represent the range of intensities in the images. The z-axis represents the probability that an intensity in one image will occur with an intensity in the other. It is a surface plot of the probability values represented in the plot on the left. [11]

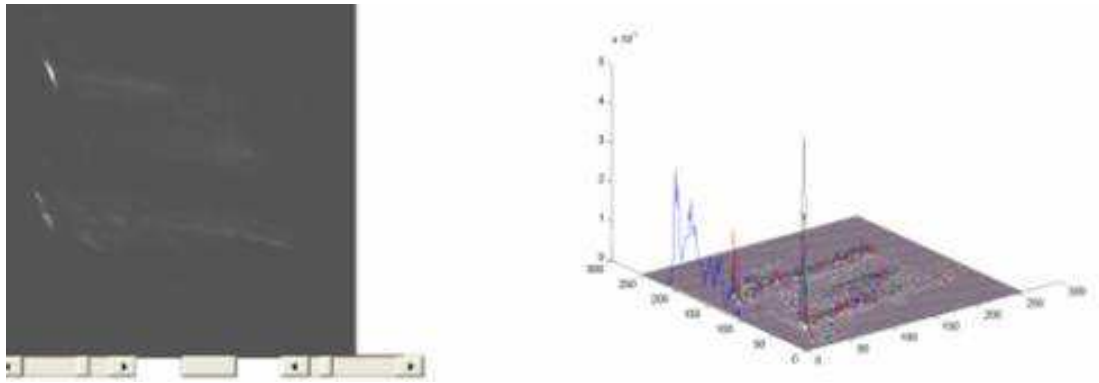


Figure 4.6: The result after the automated registration process [11].

physician can decide, based on these portal images, if the treatment field is placed correctly relative to the patient. A preliminary patient positioning can be carried out based on external markers, either on the patient's skin or the mask systems. The position of the target region (tumor) and the organs at risks (spinal column, brain, etc.) is not fixed relative to these markers. Therefore, an image of the patient's anatomy prior to treatment is needed. A possible misalignment can then be corrected by repositioning the patient on the treatment couch. For example, portal images can be compared to digitally reconstructed radiographs (DRRs), generated from the planning CT (computed tomograph). DRRs are computed by integrating (summing intensities) along rays through the CT volume to simulate the process of perspective projection in x-ray imaging. [13]

To formulate DRR/EPI registration as a mutual information-based image registration problem, the joint image histogram is used as the joint probability density function. The transformation, T , that represents in-plane rotations

and shifts, or some other type of transformation, represents the communication channel. The model for DRR/EPI registration becomes:

$$I(A(x), T(B(x))) = H(A(x)) + H(T(B(x))) - H(A(x), T(B(x))),$$

where $A(x)$ is the model, or reference, image, and $T(B(x))$ is the transformation of a test image. The objective is to find the transformation, T , that maximizes the mutual information of the system. The problem can be stated:

Find T such that $I(A(x), T(B(x)))$ is maximized.

Figure 4.1 shows a typical treatment plan, where the parallel lines on the right two images represent the leaves of a multileaf beam collimator, a beam-shaping device. Figures 4.2 and 4.3 show a pelvic region aligned in the DRR and the EPI. Figure 4.4 shows the pelvic region misaligned in the EPI (left) and the DRR (right) automatically rotated, using mutual information-based image registration, to the EPI position. Figures 4.5 and 4.6 each shows 2D and 3D representations of the joint histogram before and after alignment.

4.1 Experiments and Results

Registration was performed on a set of images provided by [11] consisting of one DRR and nineteen EPIs. Setup errors were unknown. Based on registration results and visual inspection of the results, the setup errors ranged from translations of 0 to ± 7 pixels and rotations of 0 to ± 10 degrees.

4.1.1 Greedy Algorithm

For this part of the study/research, a greedy algorithm [12] developed in MATLAB (The MathWorks, Inc.) was used. It is a greedy algorithm in that it follows the problem solving metaheuristic of making a locally optimum choice

at each stage with the hope of finding the global optimum. The algorithm is like the heuristic of a tabu search in that moves are recorded, so that a move won't be unnecessarily repeated. In this respect, the record functions as long term memory in that the history of the entire exploration process is maintained. In some applications of a tabu search, repeating a move can be advantageous so that such a list functions as short term memory in recording the most recent movements. However, in the current application, a particular move always results in the same outcome so that repeating a move is inefficient.

The transformation, T , for this example is a series of translations and rotations for two-dimensional images of resolution 256×256 pixels. An initial three-dimensional vector consisting of the extent of a move left, right, up, and down, an x -shift and a y -shift, in pixels, and angle of rotation, in degrees, is given. The test image is transformed according to these parameters. The initial x - and y -shifts and angle of rotation are added componentwise to each of 52 rows in a preset matrix. The first two elements of each row of the matrix consists of two elements from the set $\{-2,-1,0,1,2\}$ to be added to the initial values for the x -shift and y -shift. The third and final element in each row is added to the initial value for angle of rotation and is from the set $\{-1,-.5,0,.5,1\}$. This results in the generation of 52 new transformation vectors. The mutual information is then calculated and stored for each of these row vectors. The set of parameters, x - and y -shifts and angle of rotation, from the solution set that corresponds to the maximum value of mutual information is used as the starting vector for the next iteration. If there is a return to a previous transformation, the program terminates.

The capture range of the correct optimum is the portion of the parameter space in which an algorithm is more likely to converge to the correct optimum [5]. The greedy algorithm does not work well if the initial transformation vector is not in the capture range of the correct optimum in that there is divergence from the optimal solution. A local optimum is eventually attained and the program terminates. For the following examples, the transformation that results in the optimal solution (for $0 \pm .5$ or integer $\pm .5$ rotation values for the 256×256 case) is an x -shift of 1 pixel, a y -shift of 0 pixels, and a rotation of -10 degrees, denoted (1 0 -10).

Table 4.1 and Figure 4.7 show convergence to the approximate optimum resulting from a starting transformation of (11 11 10). Table 4.2 and Figure 4.8 show convergence to an incorrect solution from the starting transformation of (12 12 12). A rough estimate of the capture range for this algorithm can be derived from Table 4.3 which gives the starting and final transformation vectors for 22 trials, where there is convergence to the correct transformation, (1 0 -10), for initial transformations of (-10 -10 -18) and (25 25 11). Therefore, the capture range is roughly within 35 pixels in the horizontal direction, 36 pixels in the vertical direction, and 20 degrees of rotation. Unfortunately, the size of the capture range depends on the features in the images and cannot be known *a priori* [5]. However, visual inspection of the registered images can reveal convergence outside the capture range.

The problem of the patient positioning has to be solved using an algorithm that is not only sufficiently accurate, but one that is also fast enough to fulfill the requirements of daily clinical use. For this application, the requirement that the



Figure 4.7: Greedy algorithm, run 1, 256×256 images.

| iteration | x-shift | y-shift | rotation | MI |
|-----------|---------|---------|----------|--------|
| 1 | 11 | 11 | 10 | 0.7638 |
| 2 | 13 | 9 | 9 | 0.7780 |
| 3 | 11 | 7 | 8 | 0.7871 |
| 4 | 9 | 5 | 7 | 0.7969 |
| 5 | 9 | 3 | 6 | 0.8043 |
| 6 | 9 | 3 | 5 | 0.8087 |
| 7 | 9 | 3 | 4 | 0.8149 |
| 8 | 9 | 3 | 3 | 0.8177 |
| 9 | 7 | 1 | 2 | 0.8203 |
| 10 | 5 | 1 | 1 | 0.8253 |
| 11 | 4 | 1 | 0.5 | 0.8311 |
| 12 | 2 | 1 | -0.5 | 0.8349 |
| 13 | 2 | -1 | -1.5 | 0.8419 |
| 14 | 0 | -1 | -2.5 | 0.8542 |
| 15 | -2 | 1 | -3.5 | 0.8845 |
| 16 | -4 | -1 | -4.5 | 0.9112 |
| 17 | -4 | 1 | -5.5 | 0.9466 |
| 18 | -2 | 1 | -6.5 | 0.9869 |
| 19 | -2 | 1 | -7.5 | 1.0420 |
| 20 | 0 | 1 | -8.5 | 1.1204 |
| 21 | 0 | 1 | -9.5 | 1.1916 |
| 22 | 1 | 0 | -10 | 1.2176 |
| 23 | 0 | 0 | -9.5 | 1.2021 |
| 24 | 1 | 0 | -10 | 1.2176 |

Table 4.1: *Sample run (Figure 4.7) converges to the optimum transformation in 8.712 minutes.*

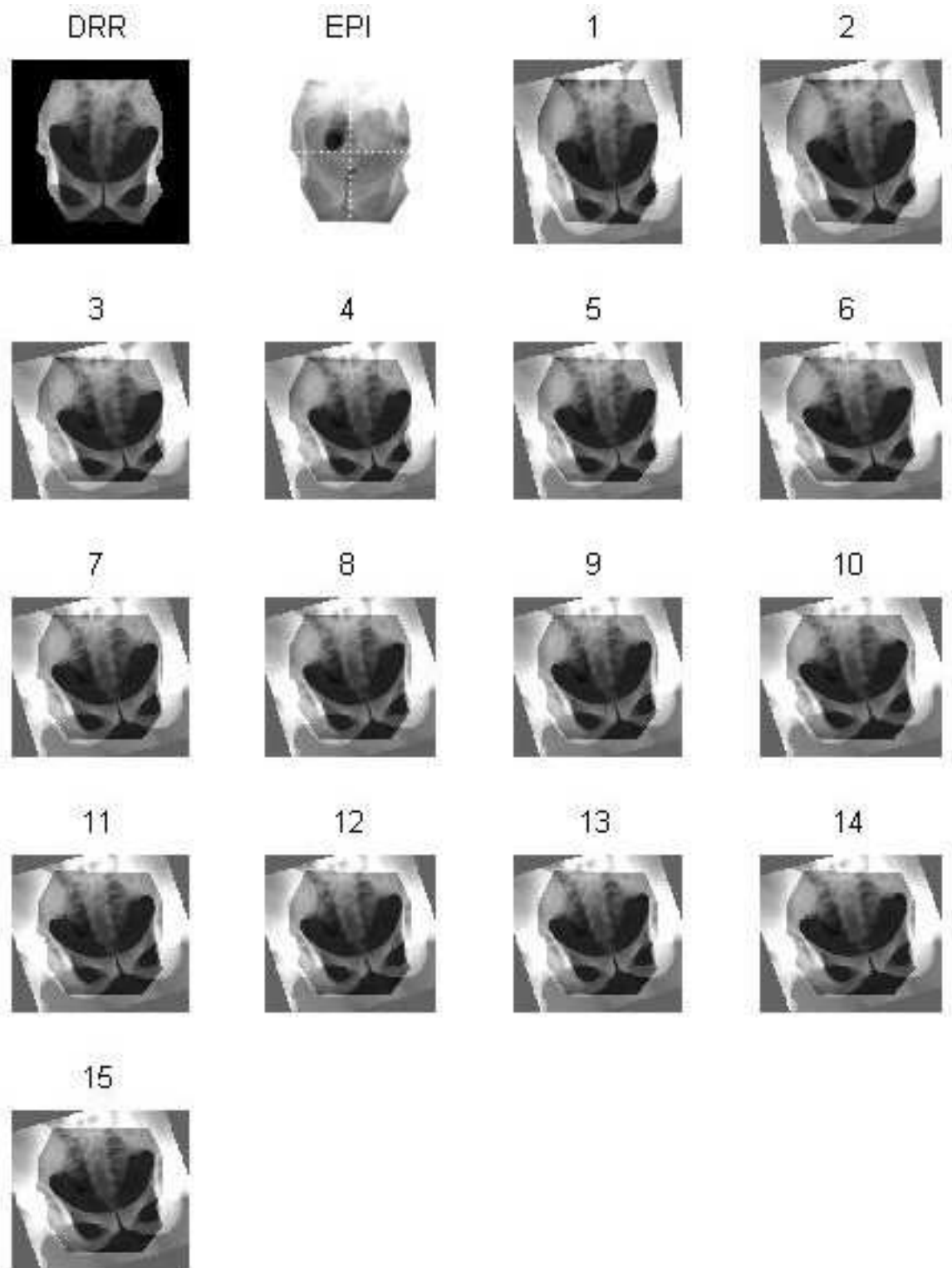


Figure 4.8: Greedy algorithm, run 2, 256×256 images.

| iteration | x-shift | y-shift | rotation | MI |
|-----------|---------|---------|----------|--------|
| 1 | 12 | 12 | 12 | 0.7663 |
| 2 | 14 | 10 | 13 | 0.7734 |
| 3 | 14 | 8 | 14 | 0.7821 |
| 4 | 14 | 6 | 14 | 0.7890 |
| 5 | 12 | 4 | 14 | 0.7922 |
| 6 | 10 | 2 | 15 | 0.7982 |
| 7 | 8 | 2 | 16 | 0.8029 |
| 8 | 8 | 2 | 16.5 | 0.8052 |
| 9 | 8 | 0 | 16.5 | 0.8058 |
| 10 | 6 | -2 | 16.5 | 0.8067 |
| 11 | 6 | -4 | 17.5 | 0.8086 |
| 12 | 6 | -6 | 17.5 | 0.8135 |
| 13 | 7 | -6 | 18 | 0.8150 |
| 14 | 8 | -6 | 18 | 0.8152 |
| 15 | 7 | -6 | 18 | 0.8150 |

Table 4.2: *Sample run (Figure 4.8) converges to a suboptimal solution in 5.0823 minutes.*

| $T_0(x, y, \text{angle})$ | $T_{final}(x, y, \text{angle})$ | MI | time(min) |
|---------------------------|---------------------------------|----------|-----------|
| (11 11 10) | (1 0 -10) | 1.217601 | 8.788317 |
| (12 12 10) | (1 0 -10) | 1.217601 | 8.709550 |
| (13 13 10) | (1 0 -10) | 1.217601 | 8.498200 |
| (16 16 10) | (1 0 -10) | 1.217601 | 9.150317 |
| (20 20 10) | (1 0 -10) | 1.217601 | 10.609733 |
| (30 30 10) | (42 46 7.5) | 0.750654 | 4.397467 |
| (11 11 11) | (8 -6 18) | 0.815214 | 4.548517 |
| (20 20 11) | (1 0 -10) | 1.217601 | 10.912367 |
| (25 25 11) | (1 0 -10) | 1.217601 | 11.816650 |
| (26 26 11) | (8 -6 18) | 0.815214 | 8.016017 |
| (29 29 11) | (51 2 15.5) | 0.829537 | 7.151950 |
| (00 00 12) | (-14 -9 18) | 0.833644 | 4.187017 |
| (11 11 12) | (8 -6 18) | 0.815214 | 4.644167 |
| (-10 -10 -19) | (6 -25 -15.5) | 0.883105 | 5.377883 |
| (-10 -10 -18) | (1 0 -10) | 1.217601 | 4.953967 |
| (-11 -11 -18) | (6 -25 -15.5) | 0.883105 | 5.203483 |
| (-12 -12 -18) | (6 -25 -15.5) | 0.883105 | 5.360033 |
| (-13 -13 -18) | (4 -50 -20.5) | 0.916436 | 9.069867 |
| (-17 -17 -18) | (-11 -33 -26) | 0.879511 | 4.321567 |
| (-18 -18 -18) | (-3 -35 -27) | 0.888810 | 5.214000 |
| (-19 -19 -18) | (-23 -44 -19) | 0.909333 | 6.070050 |
| (-20 -20 -18) | (-23 -44 -19) | 0.909333 | 5.909833 |

Table 4.3: *Convergence data for 22 runs, 256×256 images.*

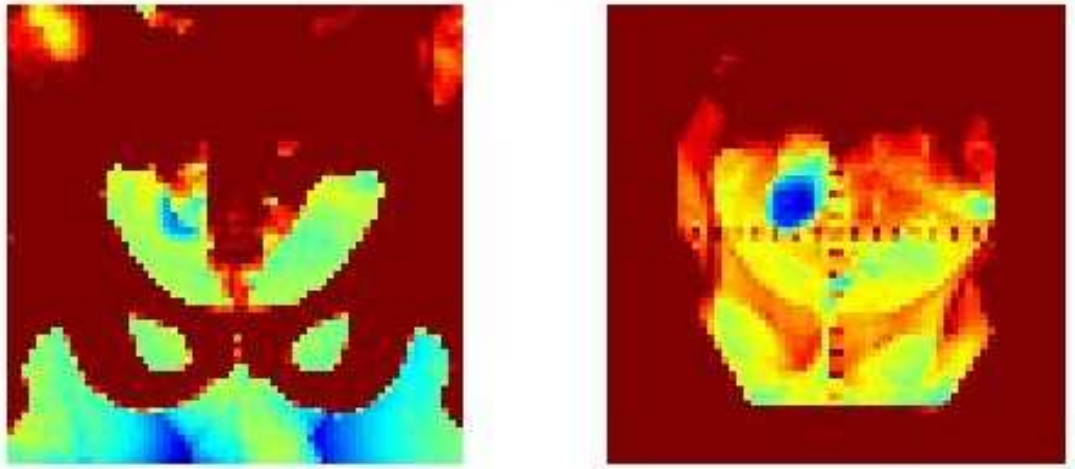


Figure 4.9: 64×64 DRR and EPI.

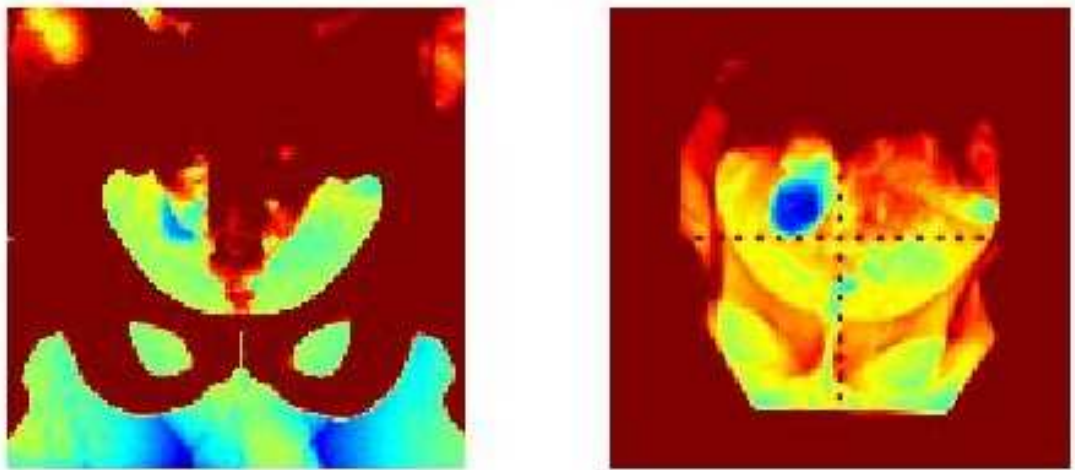


Figure 4.10: 128×128 DRR and EPI.

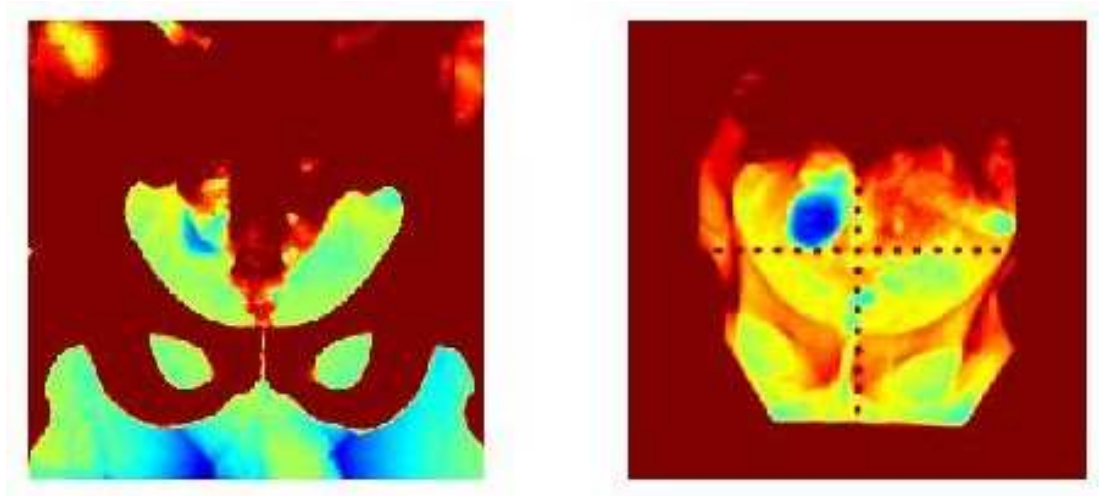


Figure 4.11: 256×256 DRR and EPI.

initial set of parameters be in a relatively narrow capture range of the optimum does not seem unreasonable assuming that external markers are used for preliminary patient positioning. However, the speed of convergence can be improved by finding the optimal transformation for low resolution versions of the images to be registered. Figures 4.9, 4.10, and 4.11 show the sets of images used in these examples at resolutions of 64×64 , 128×128 , and 256×256 , respectively. Tables 4.4, 4.5, and 4.6 respectively, show the progression toward convergence for these sets. Table 4.7 summarizes the data. As expected, it requires the least amount of time for the lowest, 64×64 , resolution. Also, the 64×64 and 128×128 resolutions converge to a slightly different transformation solution, $(0 \ 0 \ -9.5)$, than that of the 256×256 resolution. However, it is very close and can be used as the starting transformation for a higher resolution case. From Table 4.6, iterations 11 and 12, it can be seen that this would require only two iterations compared to

12, a significant reduction in computation time. So, a multi-resolution approach, that is, starting with a series of relatively coarse resolutions with the objective being to find one or more approximate solutions, or candidate local maxima, as suitable starting solutions for refinement at progressively higher resolutions, is one way to speed convergence to the global optimum. Of course, whether or not the global optimum, or something reasonably close to it, has been attained, has to be determined, for the type of registration problem described here, by visual inspection.

The multi-resolution method was implemented using first the 64×64 resolution, starting with $(0\ 0\ 0)$, which converged to the solution $(0\ 0\ -9.5)$. This solution was then used as the starting transformation vector with the 256×256 resolution. The algorithm converged to $(1\ 0\ -10)$ in two iterations with total time for both resolutions equal to 1.5311 minutes.

The program can be improved for the case of the possibility of a poorly chosen initial starting point. For example, the algorithm can be called repeatedly while a record of the transformation vectors that have been checked is maintained. A new, not previously checked, transformation vector can be created for a new iteration of the algorithm. The maximum mutual information might then be retrieved from the stored list of values. This is basically ‘multistart’ optimization and was implemented and apparently works well. Using a set of 64×64 resolution images, with an initial transformation vector of $(-50\ 50\ -50)$, and the specification that the greedy algorithm be called 10 times, the program converged to the expected optimum of $(0\ 0\ -9.5)$ for the 64×64 resolution. This occurred at iteration 63 of 104 total iterations and in approximately 5.7 minutes.

| iteration | x-shift | y-shift | rot(degrees) | MI |
|-----------|---------|---------|--------------|----------|
| 1 | 0 | 0 | 0 | 1.904037 |
| 2 | -2 | 0 | -1 | 1.932668 |
| 3 | -2 | 0 | -2 | 1.946724 |
| 4 | -2 | 0 | -3 | 1.947779 |
| 5 | -2 | 0 | -4 | 1.974409 |
| 6 | 0 | 0 | -5 | 1.981745 |
| 7 | 0 | 0 | -6 | 2.010966 |
| 8 | 0 | 0 | -7 | 2.027811 |
| 9 | 0 | 0 | -8 | 2.059447 |
| 10 | 0 | 0 | -9 | 2.101459 |
| 11 | 0 | 0 | -9.5 | 2.110144 |

Table 4.4: 64×64 run converges in 0.6504 minutes.

4.1.2 Genetic Algorithm

Unlike the greedy algorithm, the genetic algorithm, created in MATLAB for this part of the study, begins with a random set of solutions, or a population, consisting of a number of trial solutions. A trial solution consists of a transformation vector as described in the previous section, that is, a lateral shift, a vertical shift, and a rotation. The mutual information of each is evaluated. The transformations that yield a mutual information values that meets or exceed a

| iteration | x-shift | y-shift | rot(degrees) | MI |
|-----------|---------|---------|--------------|----------|
| 1 | 0 | 0 | 0 | 1.200664 |
| 2 | -2 | -2 | -1 | 1.236596 |
| 3 | -4 | 0 | -2 | 1.252961 |
| 4 | -4 | 0 | -3 | 1.272571 |
| 5 | -4 | 0 | -4 | 1.291002 |
| 6 | -2 | 0 | -5 | 1.314651 |
| 7 | -2 | 0 | -6 | 1.353761 |
| 8 | -2 | 0 | -7 | 1.393065 |
| 9 | 0 | 0 | -8 | 1.440195 |
| 10 | 0 | 0 | -9 | 1.527838 |
| 11 | 0 | 0 | -9.5 | 1.549610 |

Table 4.5: *128×128 run converges in 1.2058 minutes.*

| iteration | x-shift | y-shift | rot(degrees) | MI |
|-----------|---------|---------|--------------|----------|
| 1 | 0 | 0 | 0 | 0.795863 |
| 2 | 2 | -2 | -1 | 0.842388 |
| 3 | 0 | -2 | -2 | 0.850167 |
| 4 | -2 | -2 | -3 | 0.872867 |
| 5 | -4 | -2 | -4 | 0.899841 |
| 6 | -4 | 0 | -5 | 0.929021 |
| 7 | -4 | 0 | -6 | 0.965123 |
| 8 | -2 | 0 | -7 | 1.013159 |
| 9 | -2 | 0 | -8 | 1.076260 |
| 10 | 0 | 0 | -9 | 1.170099 |
| 11 | 0 | 0 | -9.5 | 1.202078 |
| 12 | 1 | 0 | -10 | 1.217601 |

Table 4.6: *256×256 run converges in 4.3367 minutes.*

| resolution | $T_0(x, y, \text{angle})$ | $T_{final}(x, y, \text{angle})$ | MI | time(min) |
|------------|---------------------------|---------------------------------|--------|-----------|
| 64× 64 | (0 0 0) | (0 0 -9.5) | 2.1101 | 0.6504 |
| 128× 128 | (0 0 0) | (0 0 -9.5) | 1.5496 | 1.2058 |
| 256× 256 | (0 0 0) | (1 0 -10) | 1.2176 | 4.3367 |

Table 4.7: *Summary of Tables 4.3, 4.4, and 4.5.*

user-specified value above the average value are selected. Each of these is randomly paired with another member in this set. A new set of vectors is created by randomly exchanging vector elements between the pairs. Additionally, for each iteration, another random set of trials half the size of the original set is added, an ‘immigration’ step. Every trial is recorded so that none is repeated. After a user-specified number of iterations, the program terminates and the vector that yields the maximum value for mutual information is considered to be the solution. Obviously, the larger the population and the larger the number of iterations, the greater the chance that this will be the optimal solution.

An experiment was run two times with the 256×256 DRR and EPI image shown in Figure 4.11 and yielded the results shown in Table 4.8. The approximate optimal solution was found in run 2. Each run began with a population of 100 and an initial transformation vector (0 0 0). Forty iterations were specified as well as a maximum shift, in absolute value, of 11 pixels, and maximum rotation, also in absolute value, of 10 degrees. Clearly, more iterations are required to guarantee an approximately optimal solution.

Repeating the same experiment, but using maximum shifts, in absolute value, of 2 pixels, and maximum rotation, also in absolute value, of 10 degrees yielded the results shown in Table 4.8. The logic here is to assume that the actual transformation lies within, in absolute value, the range of values determined by these maxima. Note the shorter runtime and that both runs yielded an approximately optimal solution. This is no doubt due to the fact that the respective vector values were from a relatively narrow range and close to the optimum.

If, say, 10 ± 2 had been designated as the range of values to be considered for runs 3 and 4, then one could expect a shorter runtime. For these runs, rotation values from 0 to 10 were possible, so a wider range of values generated resulted in longer runtime.

The algorithm was adjusted to account for a range of values between user-specified minimum and maximum values for each parameter. The results for runs 5 and 6 are displayed in Table 4.8. The only changes in the experimental parameters (Table 4.9) were the addition of a minimum transformation vector of values, $(0 \ 0 \ 9)$, and a change from 10 to 11 for the upper bound on rotation giving a maximum transformation vector of $(2 \ 2 \ 11)$. Not only was the runtime significantly reduced, but mutual information values closer to optimal were achieved. Note that where the greedy search allows additions to integer rotation values from the set $\{-1, -.5, 0, .5, 1\}$, the genetic algorithm allows for a continuous range, that is, randomly generated decimal values plus or minus an integer or zero, of values for rotation.

In all of these experiments, fitness was defined as having a mutual information value 1.1 and above times the average value for an iteration. Keeping all parameters the same and increasing this factor to 1.2, yielded the results shown in Table 4.8. As expected, mutual information values near optimum were achieved and there was a significant reduction in runtime apparently due to the fact that fewer vectors deemed as fit can be chosen from a narrower range of values that are also within the capture range of the optimum.

The parameters used for these experiments are tabulated in Table 4.9.

4.2 Nelder-Mead (MATLAB's `fminsearch`)

| run | x-shift | y-shift | rot(degrees) | MI | time(min) |
|-----|---------|---------|--------------|--------|-----------|
| 1 | 0 | 0 | -9 | 1.1701 | 19.7487 |
| 2 | 1 | 0 | -10 | 1.2176 | 19.0459 |
| 3 | 1 | 0 | -10 | 1.2176 | 6.2782 |
| 4 | 1 | 0 | -10 | 1.2176 | 6.5528 |
| 5 | 1 | 0 | -9.8635 | 1.2211 | 4.2006 |
| 6 | 1 | 0 | -9.8969 | 1.2206 | 3.4533 |
| 7 | 1 | 0 | -9.8888 | 1.2208 | 2.7907 |
| 8 | 1 | 0 | -9.8249 | 1.2185 | 2.6258 |

Table 4.8: *Genetic algorithm. 256×256 images.*

The MATLAB program `fminsearch` uses the Nelder-Mead simplex (direct search) method of [7]. This is a direct search method that does not use numerical or analytic gradients. It is based on evaluating a function at the vertices of a simplex, then iteratively shrinking the simplex as better points are found until some desired bound is obtained.

The transformation, $(1\ 0\ -9.8635)$, of the best result, 1.2211, from the experiments above (Table 4.8, Run 5), was used as the starting transformation with `fminsearch`. The refined result, $(1\ 0\ -9.8673)$ yielding 1.2213, is shown in Table 4.10.

4.3 Simulated Annealing

To confirm and further refine the result of the previous section, a simulated annealing program created in MATLAB was used to vary the decimal values

| <i>parameters</i> | <i>run pairs</i> | | | |
|---------------------------|------------------|---------|---------|---------|
| | 1,2 | 3,4 | 5,6 | 7,8 |
| $T_0(x, y, \text{angle})$ | (0 0 0) | (0 0 0) | (0 0 0) | (0 0 0) |
| initial population | 100 | 100 | 100 | 100 |
| iterations | 40 | 40 | 40 | 40 |
| above average factor | 1.1 | 1.1 | 1.1 | 1.2 |
| min x-shift | - | - | 0 | 0 |
| min y-shift | - | - | 0 | 0 |
| minimum rotation | - | - | 9 | 9 |
| max x-shift | 11 | 2 | 2 | 2 |
| max y-shift | 11 | 2 | 2 | 2 |
| max rot | 10 | 10 | 11 | 11 |

Table 4.9: *Parameter list for genetic run pairs.*

| x-shift | y-shift | rot(degrees) | MI | time(min) |
|---------|---------|--------------|--------|-----------|
| 1 | 0 | -9.8673 | 1.2213 | 1.1455 |

Table 4.10: *MATLAB's fminsearch (Nelder-Mead) algorithm. 256×256 images.*

of the angle of rotation of -9. The integral part of the value was not changed, since, from the previously described experiments and visual inspection, it was established that the solution for rotation is $-9 \pm$ a decimal value. The results from over 7500 decimal values generated are shown in Figures 4.12 and 4.13. A value of -9.867333 as the angle of rotation with a mutual information value of 1.221330 was obtained. Figure 4.13 shows that a range of angle values gives the same mutual information value.

A major advantage of simulated annealing is its ability to avoid becoming trapped at local minima. The algorithm employs a random search that can accept changes that decrease objective function as well as those that increase it. However, for this application, acceptance of a transformation that results in a smaller value for mutual information makes no sense, so only those that result in a higher mutual information value are accepted.

4.4 Other Experiments

In addition to the EPI example used in this chapter, 18 other EPIs of resolution 64×64 were tested using the greedy algorithm which was modified to limit the search space. In each case, the search space included x - and y -shifts of 0 to ± 8 pixels, and rotations of 0 to ± 10 degrees. Additionally, in every case, the greedy algorithm was called 20 times with 50 iterations maximum specified per run. It was determined, by the data and visual inspection of the output graphics, that the resultant transformations were successful in all cases.

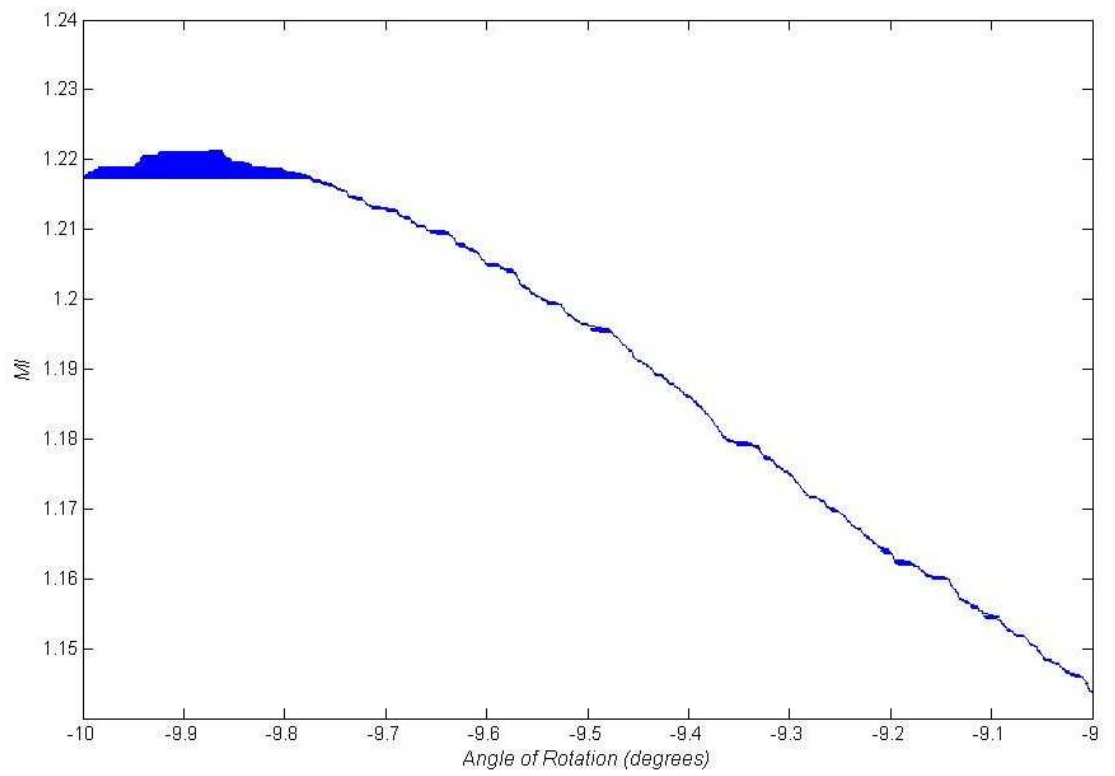


Figure 4.12: Plot of MI versus Angle of Rotation from simulated annealing data.

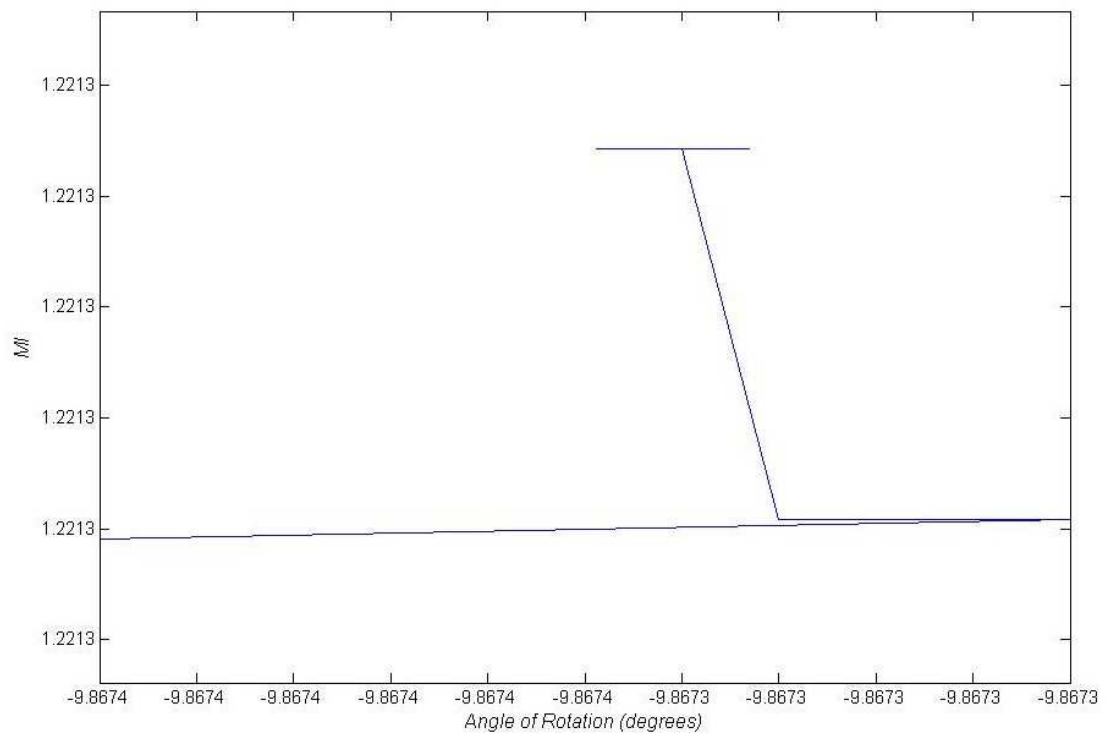


Figure 4.13: Detail of Figure 4.12, plot of MI versus Angle of Rotation from simulated annealing data.

5. Conclusion

In the experiments described in Chapter 4, mutual information-based image registration was found to be a robust and easily implemented technique. In every case the entire, unprocessed, image was registered. Because the setup errors were unknown, the registrations were judged to be successful or not based on visual inspection. In all nineteen cases, the use of the mutual information technique resulted in successful transformations as determined by visual inspection.

It appears that the greedy and genetic algorithms perform well, and in a reasonable amount of time from a clinical perspective, given that there is assumed to be a good estimate of the general location of the optimum. If such an estimate could not be made, then the genetic algorithm would probably outperform the greedy algorithm since the greedy algorithm tends to terminate at a local optimum. Of course, this problem is resolved by multistart optimization, that is, doing multiple searches starting with a variety of transformations resulting in multiple solutions, and then choosing the solution which yields the highest value of mutual information.

Given a full range of possible transformation values, the genetic algorithm can find the global optimum given enough time. The time required may be more than is practicable, but the problem can be made practicable, as was demonstrated in this experiment, by constraining the search area to within the capture range of the optimum. As mentioned in Section 4.1.1, the size of the capture range depends on the features in the images and cannot be known *a*

priori, but visual inspection of the registered images can reveal convergence outside of the capture range.

The advantage of mutual information-based image registration is that it can be fully automatic in that it makes no assumption of the functional form or relationship between image intensities in the image to be registered. However, it is clearly important that a method of quality assurance be used to ensure that only well registered images are used for clinical decision making.

Appendix A. Brief Discrete Probability Theory

Definition A.1 *Given an experiment whose outcome is unpredictable, such an outcome, say X , is called a **random variable**, or **trial**. The **sample space** of the experiment is the set of all possible trials. If the sample space is either finite or countably infinite, the random variable is said to be **discrete**.*

Example 5. Roll of a die.

Suppose that a die is rolled once, and let X denote the outcome of the experiment. The sample space for this experiment is the 6-element set

$$\Omega = \{1, 2, 3, 4, 5, 6\},$$

where each outcome i , for $i = 1, \dots, 6$, corresponds to the number of dots on the face that turns up. The event

$$E = \{1, 3, 5\}$$

corresponds to the statement that the result of the roll is an odd number. Assuming that the die is fair, or unloaded, the assumption is that every outcome is equally likely. Therefore, a probability of $\frac{1}{6}$ is assigned to each of the six outcomes in Ω , that is, $m(i) = \frac{1}{6}$, for $1 \leq i \leq 6$.

Definition A.2 *Let X be a random variable which denotes the outcome, of finitely many possible outcomes, of an experiment. Let Ω be the sample space of the experiment. A **probability distribution function**, or **probability mass***

function, for X is a real-valued function m whose domain is Ω and which satisfies:

1. $m(\omega) \geq 0$, for all $\omega \in \Omega$, and
2. $\sum_{\omega \in \Omega} m(\omega) = 1$.

For any $E \subset \Omega$, the **probability** of E is defined as the number $P(E)$ given by

$$P(E) = \sum_{\omega \in E} m(\omega).$$

Example 6. Example 5 continued.

The 6-element set $\Omega = \{1, 2, 3, 4, 5, 6\}$ is the sample space for the experiment in which a die is rolled. The die is assumed to be fair, and the distribution function is defined by

$$m(i) = \frac{1}{6}, \text{ for } i = 1, \dots, 6.$$

If E is the event that the result of the roll is an odd number, then $E = \{1, 3, 5\}$ and

$$\begin{aligned} P(E) &= m(1) + m(3) + m(5) \\ &= \frac{1}{6} + \frac{1}{6} + \frac{1}{6} \\ &= \frac{1}{2}. \end{aligned}$$

Definition A.3 Let X be a numerically-valued discrete random variable with sample space Ω and distribution $m(x)$. The **expected value** $E(X)$ is defined by

$$E(X) = \sum_{x \in \Omega} xm(x),$$

provided that this sum converges absolutely. The expected value is also referred to as the **mean**, and $E(X)$ is denoted μ for short.

Example 7. Coin toss.

Suppose that a fair coin is tossed three times. Let X denote the number of heads that appear. The sample space is

$$\Omega = \{\text{HHH HHT HTH THH TTH HTT THT TTT}\}.$$

The possible values of X are 0, 1, 2, and 3. The corresponding probabilities are $\frac{1}{8}$, $\frac{3}{8}$, $\frac{3}{8}$, and $\frac{1}{8}$. Therefore, the expected value of X equals

$$0 \left(\frac{1}{8}\right) + 1 \left(\frac{3}{8}\right) + 2 \left(\frac{3}{8}\right) + 3 \left(\frac{1}{8}\right) = \frac{3}{2}.$$

Definition A.4 Let x and y be random variables which can take on values in $X = v_1, v_2, \dots, v_m$, and $Y = w_1, w_2, \dots, w_n$, respectively. (x, y) can be thought of as a vector or point in the product space of x and y . There is a **joint probability** $p_{ij} = P(x = v_i, y = w_j)$ for each possible pair of values (v_i, w_j) . The **joint probability distribution function** $P(x, y)$ satisfies:

$$P(x, y) \geq 0$$

$$\sum_{x \in X} \sum_{y \in Y} P(x, y) = 1.$$

The joint probability distribution function is a complete characterization of the pair of random variables (x, y) . Everything that can be computed about x and y , individually or together, can be computed from $P(x, y)$. The separate **marginal distributions** for x and y can be obtained by summing over the unwanted variable.

$$P_x(x) = \sum_{y \in Y} P(x, y)$$

$$P_y(y) = \sum_{x \in X} P(x, y).$$

Definition A.5 Two variables are statistically **independent** if

$$P(X, Y) = P(X) P(Y).$$

Definition A.6 Two variables, X and Y , are statistically **dependent** if knowing the value of one of them results in a better estimate of the value of the other.

This is the **conditional probability** (Bayes' probability) of x given y :

$$P(x = v_i | y = w_j) = \frac{P(x = v_i, y = w_j)}{P(y = w_j)},$$

or, in terms of distribution, or mass, functions,

$$P(x|y) = \frac{P(x, y)}{P(y)}.$$

If x and y are statistically independent (Definition A.5), then

$$P(x|y) = P(x), \text{ and}$$

$$P(y|x) = P(y).$$

Example 8. Roll of a die.

An experiment consists of rolling a die once. Let X be the outcome. Let F be the event $X = 6$, and let E be the event $X > 4$. The distribution function $m(\omega) = \frac{1}{6}$ for $\omega = 1, 2, \dots, 6$. Thus, $P(F) = \frac{1}{6}$. Suppose that it happens that the event E has occurred. This leaves only two possible outcomes: 5 and 6. In the absence of any other information, these outcomes would still be regarded as equally likely, so the probability of F is $\frac{1}{2}$, so that $P(F|E) = \frac{1}{2}$.

Definition A.7 (General Bayes' Formula) Suppose there is a set of events, H_1, H_2, \dots, H_m , **hypotheses**, that are pairwise disjoint and such that the sample

space Ω satisfies

$$\Omega = H_1 \cup H_2 \cup \dots \cup H_m.$$

Suppose there is an event, E (**evidence**), which gives some information about which hypothesis is correct.

Before the evidence is received, there exists the set of **prior probabilities**, $P(H_1), P(H_2), \dots, P(H_m)$, for the hypotheses. If the correct hypothesis is known, then so is the probability for the evidence. That is, $P(E|H_i)$ is known for all i . To find the probabilities for the hypotheses given the evidence, that is, the conditional probabilities $P(H_i|E)$, or **posterior probabilities**, write

$$P(H_i|E) = \frac{P(H_i \cap E)}{P(E)}. \quad (\text{A.1})$$

The numerator can be calculated with the information given by

$$P(H_i \cap E) = P(H_i)P(E|H_i). \quad (\text{A.2})$$

Since only one of the events H_1, H_2, \dots, H_m can occur, the probability of E can be written as

$$P(E) = P(H_1 \cap E) + P(H_2 \cap E) + \dots + P(H_m \cap E).$$

Using Equation A.2, the righthand side can be seen to equal

$$P(H_1)P(E|H_1) + P(H_2)P(E|H_2) + \dots + P(H_m)P(E|H_m). \quad (\text{A.3})$$

Using Equations A.1, A.2, and A.3, yields **Bayes' Formula**

$$P(H_i|E) = \frac{P(H_i)P(E|H_i)}{\sum_{k=1}^m P(H_k)P(E|H_k)}. \quad (\text{A.4})$$

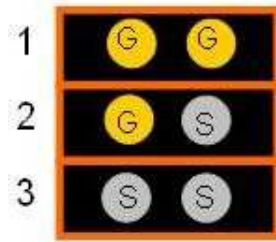


Figure A.1: Example 9.

Example 9. Drawers containing gold and silver coins.

Consider a set of three drawers, each containing coins (refer to Figure A.1). The probabilities of a drawer containing gold coins given that a particular drawer is chosen are as follows:

$$P(G|1) = 1$$

$$P(G|2) = \frac{1}{2}$$

$$P(G|3) = 0.$$

Suppose that a blindfolded person chooses a drawer and it is found to contain gold. To find the probability that drawer 1 was chosen given that gold is found, apply Bayes' Rule (Equation A.4) using the fact that each drawer has probability

$\frac{1}{3}$ of being chosen.

$$\begin{aligned} P(1|G) &= \frac{P(G|1)P(1)}{P(G|1)P(1) + P(G|2)P(2) + P(G|3)P(3)} \\ &= \frac{1 \left(\frac{1}{3}\right)}{1 \left(\frac{1}{3}\right) + \frac{1}{2} \left(\frac{1}{3}\right) + 0 \left(\frac{1}{3}\right)} \\ &= \frac{\frac{1}{3}}{\frac{1}{3} + \frac{1}{6}} \\ &= \frac{2}{3}. \end{aligned}$$

Appendix B. Properties of Convex Functions

Definition B.1 A function $f(x)$ is said to be **convex** over an interval (a, b) if for every $x_1, x_2 \in (a, b)$ and $0 \leq \lambda \leq 1$,

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2).$$

A function f is said to be **strictly convex** if equality holds only if $\lambda = 0$ or $\lambda = 1$.

Definition B.2 A function f is **concave** if $-f$ is convex.

Theorem B.3 If the function f has a second derivative which is non-negative (positive) everywhere, then the function is convex (strictly convex).

Proof: The Taylor series expansion of the function around x_0 , that is,

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x^*)}{2}(x - x_0)^2,$$

where x^* lies between x_0 and x , is used. By hypothesis, $f''(x^*) \geq 0$, and thus the last term is always non-negative for all x .

Let $x_0 = \lambda x_1 + (1 - \lambda)x_2$ and $x = x_1$ to obtain

$$f(x_1) \geq f(x_0) + f'(x_0) [(1 - \lambda)(x_1 - x_2)].$$

similarly, taking $x = x_2$,

$$f(x_2) \geq f(x_0) + f'(x_0) [\lambda(x_2 - x_1)],$$

is obtained. ■

Appendix C. Miscellaneous

Any set of logs is proportional to any other set. This follows from the fundamental relationship

$$\log_a x = \frac{\log_b x}{\log_b a} = (\log_a b) \log_b x,$$

which is derived as follows.

Begin with the equivalent equations

$$v = \log_a x \text{ and } a^v = x.$$

Taking the logarithm, base b , of both sides of the second equation yields

$$\log_b a^v = \log_b x = v \log_b a.$$

Solving for v yields

$$\log_a x = \frac{\log_b x}{\log_b a}.$$

The resulting unit of information from the use of base 2 logs is called a *bit*. When base 10 is used, the unit is called a *Hartley*, after R. V. L. Hartley, who first proposed the use of the logarithmic measure of information. A *nat* is the unit of information when base e is used.

Example 10.

$$\begin{aligned}\log_{10} 2 &= \frac{\log_2 2}{\log_2 10} \\ &= (\log_{10} 2) \log_2 2 \\ &= 0.3010299956 \text{ Hartleys}\end{aligned}$$

Appendix D. Sample Output

Sample output for Greedy algorithm

x

x =

| | | | | | | | | | |
|---|---|----|----|----|----|----|----|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 17 | 17 | 0 | 0 | 17 | 0 | 0 | 0 |
| 0 | 0 | 17 | 17 | 17 | 17 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 17 | 17 | 17 | 17 | 0 | 0 |
| 0 | 0 | 17 | 17 | 0 | 0 | 17 | 17 | 0 | 0 |
| 0 | 0 | 17 | 17 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

y

y =

| | | | | | | | | | |
|---|---|---|---|----|----|----|----|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 17 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 17 | 17 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 17 | 17 | 17 | 17 | 0 | 0 |

| | | | | | | | | | |
|---|---|---|---|---|---|----|----|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 17 | 17 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

greedy_kate(x,y1,50,[0 0 0]);

Starting values are:

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 0 | 0 | 0 | 0.03923 |

Completed iteration #1 (of 50).

This iteration took 0 minutes and 2.714 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 1 | 0 | -0.5 | 0.13087 |

Completed iteration #2 (of 50).

This iteration took 0 minutes and 1.212 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 1 | 0 | -1.5 | 0.13087 |

Completed iteration #3 (of 50).

This iteration took 0 minutes and 0.761 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 1 | 0 | -2.5 | 0.13087 |

Completed iteration #4 (of 50).

This iteration took 0 minutes and 0.731 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 1 | 0 | -3.5 | 0.13087 |

Completed iteration #5 (of 50).

This iteration took 0 minutes and 0.671 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 1 | 0 | -4.5 | 0.13087 |

Completed iteration #6 (of 50).

This iteration took 0 minutes and 0.681 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 1 | 0 | -5.5 | 0.13087 |

Completed iteration #7 (of 50).

This iteration took 0 minutes and 0.731 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 1 | 0 | -6.5 | 0.13087 |

Completed iteration #8 (of 50).

This iteration took 0 minutes and 0.671 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 1 | 0 | -7.5 | 0.13087 |

Completed iteration #9 (of 50).
This iteration took 0 minutes and 0.781 seconds.
X-shift Y-shift rot(degrees) MI
1 0 0 0.13087

Completed iteration #10 (of 50).
This iteration took 0 minutes and 0.671 seconds.
X-shift Y-shift rot(degrees) MI
1 0 1 0.13087

Completed iteration #11 (of 50).
This iteration took 0 minutes and 0.631 seconds.
X-shift Y-shift rot(degrees) MI
1 0 1.5 0.13087

Completed iteration #12 (of 50).
This iteration took 0 minutes and 0.651 seconds.
X-shift Y-shift rot(degrees) MI
1 0 2.5 0.13087

Completed iteration #13 (of 50).
This iteration took 0 minutes and 0.732 seconds.
X-shift Y-shift rot(degrees) MI

1 0 3 0.13087

Completed iteration #14 (of 50).

This iteration took 0 minutes and 0.651 seconds.

X-shift Y-shift rot(degrees) MI

1 0 4 0.13087

Completed iteration #15 (of 50).

This iteration took 0 minutes and 0.681 seconds.

X-shift Y-shift rot(degrees) MI

1 0 4.5 0.13087

Completed iteration #16 (of 50).

This iteration took 0 minutes and 0.741 seconds.

X-shift Y-shift rot(degrees) MI

1 0 5.5 0.13087

Completed iteration #17 (of 50).

This iteration took 0 minutes and 0.681 seconds.

X-shift Y-shift rot(degrees) MI

1 0 6 0.13087

Completed iteration #18 (of 50).

This iteration took 0 minutes and 0.681 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 1 | 0 | 7 | 0.13087 |

Completed iteration #19 (of 50).

This iteration took 0 minutes and 0.761 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|---------|
| 1 | 0 | 7.5 | 0.13087 |

Completed iteration #20 (of 50).

This iteration took 0 minutes and 0.671 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|-------|
| 1 | 0 | 8.5 | 0.185 |

Completed iteration #21 (of 50).

This iteration took 0 minutes and 0.961 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|-------|
| 1 | 0 | 9 | 0.185 |

Completed iteration #22 (of 50).

This iteration took 0 minutes and 0.811 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|----------|
| 1 | 0 | 10 | 0.184997 |

Completed iteration #23 (of 50).

This iteration took 0 minutes and 0.782 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|----------|
| 1 | 0 | 10.5 | 0.184997 |

Completed iteration #24 (of 50).

This iteration took 0 minutes and 0.781 seconds.

| X-shift | Y-shift | rot(degrees) | MI |
|---------|---------|--------------|-------|
| 1 | 0 | 8.5 | 0.185 |

REPEAT VISIT. STOPPED AT ITERATION #24

The time for this run was 0 minutes and 20.791 seconds.

Sample output for Genetic algorithm

```
geneticMI3(x,y1,40,[.4097 .4097 0],[.4097 .4097 .4097 0],[0 0  
0],0,0,0,3,3,10,400,1.1,'gami10')
```

Elapsed time is 28.361000

seconds.

time =

28.3610

location of maxima:

u =

376

377

```

v =
    4
    4
max_row =
    -1.0000      0    9.0000    0.1929
    -1.0000      0   10.0000    0.1929
ans =
    1028 (number of rows in truncated list below)
ans =
    0.4097    0.4097    0.4097         0
   -3.0000    2.0000  -10.0000    0.0000
   -3.0000    2.0000    9.0000    0.0000
   -3.0000    2.0000   10.0000    0.0000
   -3.0000    1.0000  -10.0000    0.0000
   -3.0000    1.0000    7.0000    0.0000
   -3.0000    2.0000   -9.0000    0.0000
    ⋮
   -1.0000      0    6.0000    0.1850
   -1.0000      0    7.0000    0.1850
   -1.0000      0    8.0000    0.1850
   -1.0000      0    9.0000    0.1929
   -1.0000      0   10.0000    0.1929

```

Sample output for Simulated Annealing algorithm

```
simanne1([50 50 .1],[50 50 .1 0],x,y1,[0 0  
0],0,0,0,3,3,10,'simanne1')
```

```
time =
```

```
0.4880333333333333
```

```
u =
```

```
24
```

```
156
```

```
241
```

```
282
```

```
312
```

```
380
```

```
382
```

```
397
```

```
485
```

```
553
```

```
763
```

```
796
```

```
818
```

```
860
```

```
930
```

```
980
```

```
v =
```

```
4
```

```
4
```

4
4
4
4
4
4
4
4
4
4
4
4
4
4
4

max_row =

| | |
|-------|---|
| (1,1) | 1 |
| (2,1) | 1 |
| (3,1) | 1 |
| (4,1) | 1 |
| (5,1) | 1 |
| (6,1) | 1 |
| (7,1) | 1 |
| (8,1) | 1 |
| (9,1) | 1 |

| | |
|--------|------------------|
| (10,1) | 1 |
| (11,1) | 1 |
| (12,1) | 1 |
| (13,1) | 1 |
| (14,1) | 1 |
| (15,1) | 1 |
| (16,1) | 1 |
| (1,3) | 9.25877147290670 |
| (2,3) | 8.49723539040031 |
| (3,3) | 9.15356313362336 |
| (4,3) | 9.54948895952625 |
| (5,3) | 9.20309876318741 |
| (6,3) | 8.82152895422397 |
| (7,3) | 9.34367845763154 |
| (8,3) | 9.31054863126581 |
| (9,3) | 9.21674279078533 |
| (10,3) | 9.73489190849639 |
| (11,3) | 9.96486102463224 |
| (12,3) | 8.60833789062264 |
| (13,3) | 9.43464291635121 |
| (14,3) | 9.80718158604694 |
| (15,3) | 9.81079208395498 |
| (16,3) | 8.89107255365213 |
| (1,4) | 0.18499708990259 |

```

(2,4)      0.18499708990259
(3,4)      0.18499708990259
(4,4)      0.18499708990259
(5,4)      0.18499708990259
(6,4)      0.18499708990259
(7,4)      0.18499708990259
(8,4)      0.18499708990259
(9,4)      0.18499708990259
(10,4)     0.18499708990259
(11,4)     0.18499708990259
(12,4)     0.18499708990259
(13,4)     0.18499708990259
(14,4)     0.18499708990259
(15,4)     0.18499708990259
(16,4)     0.18499708990259

```

ans =

1002 (number of rows in truncated list below)

ans =

```

1 2  9.54488116126639  0.00095770862808
0 2  9.63540372426218  0.00095770862808
1 2  9.15734939198735  0.00095770862808
1 2  9.94007518447298  0.00095770862808
0 2  9.81569914903702  0.00095770862808
0 2  9.45049423187438  0.00095770862808

```

| | | | |
|---|---|------------------|------------------|
| 1 | 2 | 9.45407042521757 | 0.00095770862808 |
| 1 | 2 | 8.10628360705867 | 0.00095770862808 |
| 1 | 2 | 8.00844673858309 | 0.00095770862808 |
| 0 | 2 | 9.02264441938358 | 0.00095770862808 |
| 0 | 2 | 9.37606174283209 | 0.00095770862808 |
| 1 | 2 | 9.18904731947569 | 0.00095770862808 |
| 0 | 2 | 8.98184102431525 | 0.00095770862808 |
| 0 | 2 | 9.90925402780049 | 0.00095770862808 |
| 1 | 2 | 9.38875756851503 | 0.00095770862808 |
| 1 | 2 | 9.67475367960192 | 0.00095770862808 |
| 1 | 2 | 9.54109363339149 | 0.00095770862808 |
| 0 | 2 | 9.60511687683569 | 0.00095770862808 |
| 1 | 2 | 9.35015741190421 | 0.00095770862808 |
| 1 | 2 | 9.77038116862139 | 0.00095770862808 |
| 0 | 2 | 9.55887884395221 | 0.00095770862808 |
| 1 | 2 | 8.93050126301304 | 0.00095770862808 |
| 0 | 2 | 9.29439576406895 | 0.00095770862808 |
| 0 | 2 | 9.84980029201121 | 0.00095770862808 |
| 0 | 2 | 9.04063742033726 | 0.00095770862808 |
| 0 | 2 | 9.98186868186102 | 0.00095770862808 |
| 1 | 2 | 8.12959020607492 | 0.00095770862808 |
| 1 | 2 | 0.79037100346942 | 0.00133945418175 |
| 1 | 2 | 2.62896546774062 | 0.00133945418175 |
| 1 | 2 | 8.68056010829237 | 0.00133945418175 |

| | | | |
|---|---|------------------|------------------|
| 1 | 2 | 5.16512954032670 | 0.00133945418175 |
| 0 | 2 | 1.42137250518467 | 0.00133945418175 |
| 0 | 2 | 3.48822983115181 | 0.00133945418175 |
| 1 | 2 | 5.60426411656511 | 0.00133945418175 |
| 1 | 2 | 3.55963825049321 | 0.00133945418175 |
| 1 | 2 | 7.15555203069753 | 0.00133945418175 |
| 0 | 2 | 3.25184057031221 | 0.00133945418175 |
| 0 | 2 | 2.89980696049129 | 0.00133945418175 |
| 1 | 2 | 5.48385100408593 | 0.00133945418175 |
| 0 | 2 | 6.24388657826624 | 0.00133945418175 |
| 1 | 2 | 4.29106980239575 | 0.00133945418175 |
| 1 | 2 | 2.52245955895128 | 0.00133945418175 |
| 0 | 2 | 3.99818346936935 | 0.00133945418175 |
| 0 | 2 | 4.96810355888851 | 0.00133945418175 |

⋮

| | | | |
|---|---|------------------|------------------|
| 1 | 0 | 6.16967289136000 | 0.13086916818810 |
| 1 | 0 | 4.36484151939276 | 0.13086916818810 |
| 1 | 0 | 5.46553184636482 | 0.13086916818810 |
| 1 | 0 | 4.96328660944760 | 0.13086916818810 |
| 1 | 0 | 4.06357651283500 | 0.13086916818810 |
| 1 | 0 | 6.91831625658650 | 0.13086916818810 |
| 1 | 0 | 0.14247435980314 | 0.13086916818810 |
| 1 | 0 | 3.71589377794519 | 0.13086916818810 |

| | | | |
|---|---|------------------|------------------|
| 1 | 0 | 6.60444255487201 | 0.13086916818810 |
| 1 | 0 | 1.53686617793864 | 0.13086916818810 |
| 1 | 0 | 1.30529054135547 | 0.13086916818810 |
| 1 | 0 | 4.52465927140714 | 0.13086916818810 |
| 1 | 0 | 5.41068222932806 | 0.13086916818810 |
| 1 | 0 | 1.11016922231647 | 0.13086916818810 |
| 1 | 0 | 4.88238425361932 | 0.13086916818810 |
| 1 | 0 | 3.00114412189347 | 0.13086916818810 |
| 1 | 0 | 7.35932220769292 | 0.13086916818810 |
| 1 | 0 | 1.40035841931849 | 0.13086916818810 |
| 1 | 0 | 0.87490312374116 | 0.13086916818810 |
| 1 | 0 | 5.93693158561465 | 0.13086916818810 |
| 1 | 0 | 7.46208399878517 | 0.13086916818810 |
| 1 | 0 | 0.45033920245829 | 0.13086916818810 |
| 1 | 0 | 8.14361151319716 | 0.17777133059671 |
| 1 | 0 | 8.20454341696700 | 0.17777133059671 |
| 1 | 0 | 8.30275319119587 | 0.17777133059671 |
| 1 | 0 | 8.23136282287183 | 0.17777133059671 |
| 1 | 0 | 8.40206241858286 | 0.17777133059671 |
| 1 | 0 | 8.24182433767606 | 0.17777133059671 |
| 1 | 0 | 9.25877147290670 | 0.18499708990259 |
| 1 | 0 | 8.49723539040031 | 0.18499708990259 |
| 1 | 0 | 9.15356313362336 | 0.18499708990259 |
| 1 | 0 | 9.54948895952625 | 0.18499708990259 |

| | | | |
|---|---|------------------|------------------|
| 1 | 0 | 9.20309876318741 | 0.18499708990259 |
| 1 | 0 | 8.82152895422397 | 0.18499708990259 |
| 1 | 0 | 9.34367845763154 | 0.18499708990259 |
| 1 | 0 | 9.31054863126581 | 0.18499708990259 |
| 1 | 0 | 9.21674279078533 | 0.18499708990259 |
| 1 | 0 | 9.73489190849639 | 0.18499708990259 |
| 1 | 0 | 9.96486102463224 | 0.18499708990259 |
| 1 | 0 | 8.60833789062264 | 0.18499708990259 |
| 1 | 0 | 9.43464291635121 | 0.18499708990259 |
| 1 | 0 | 9.80718158604694 | 0.18499708990259 |
| 1 | 0 | 9.81079208395498 | 0.18499708990259 |
| 1 | 0 | 8.89107255365213 | 0.18499708990259 |

REFERENCES

- [1] A. COLLIGNON, F. MAES, D. DELAERE, D. VANDERMEULEN, P. SUETENS, AND G. MARCHAL, Automated multimodality image registration using information theory, in *Information Processing in Medical Imaging (IPMI'95)* (Y. Bizais, C. Barillot, and R. Di Paola, Eds.), pp. 263-274. Dordrecht: Kluwer, 1995.
- [2] THOMAS M. COVER, JOY A. THOMAS, *Elements of Information Theory*, Wiley-Interscience, August 12, 1991, ISBN 0-471-06259-6.
- [3] RICHARD O. DUDA, PETER E. HART, DAVID G. STORK, *Pattern Classification*, 2nd Edition, John Wiley & Sons, Inc., New York, 2001, ISBN 0-471-05669-3.
- [4] CHARLES M. GRINSTEAD, J. LAURIE SNELL, *Introduction to Probability*, 2nd Rev Edition, American Mathematical Society, July 1, 1997, ISBN 0-8218-0749-8.
- [5] JOSEPH V. HAJNAL, DEREK L. G. HILL, AND DAVID J. HAWKES, Eds., *Medical Image Registration*, The Biomedical Engineering Series, CRC Press, Boca Raton, FL, 2001, ISBN 0-8493-0064-9.
- [6] R. V. HARTLEY, Transmission of information, *Bell Sys. Tech. Journal*, pp. 7-535, 1928.
- [7] J.C. LAGARIAS, J. A. REEDS, M. H. WRIGHT, AND P. E. WRIGHT, Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal of Optimization*, vol. 9, Number 1, pp. 112-147, 1998.
- [8] JOHN H. MATHEWS AND KURTIS K. FINK, *Numerical Methods Using Matlab*, 4th Edition, 2004, Prentice-Hall Inc., Upper Saddle River, New Jersey, ISBN: 0-13-065248-2.
- [9] JAN MODERSITZKI, *Numerical Methods for Image Registration*, Numerical Mathematics and Scientific Computation, Oxford University Press, Inc., New York, 2004, ISBN 0-19-852841-8.

- [10] J. A. NELDER AND R. MEAD, A Simplex Method for Function Minimization. *Comput. J.* 7, pp. 308-313, 1965.
- [11] FRANCIS NEWMAN, Department of Radiation Oncology, University of Colorado Health Sciences Center.
- [12] FRANCIS NEWMAN¹, STEVE HUMPHRIES², D.C. HANSELMAN³, ET AL, 1. Department of Radiation Oncology, University of Colorado Health Sciences Center; 2. Stryker Leibinger, MI; 3. University of Maine, Orono ME. Mutual information-based image registration software using a greedy algorithm.
- [13] WOLFGANG SCHLEGEL, AND ANDREAS MAHR, EDS., *3D Conformal Radiation Therapy: A multimedia introduction to methods and techniques*, 1st edition, CD-ROM, Springer-Verlag (Birkhauser), Berlin, Heidelberg, New York, December 15, 2001, ISBN: 3540148841.
- [14] CLAUDE E. SHANNON, The mathematical theory of communication (parts 1 and 2). *Bell Syst. Tech J.* vol. 27, pp. 379-423 and 623-656, 1948.
- [15] CLAUDE E. SHANNON AND WARREN WEAVER, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana and Chicago, 1998, ISBN 0-252-72548-4.
- [16] CLAUDE E. SHANNON, Communication in the presence of noise. *Proc. IRE*, vol. 37, pp. 10-21, 1949. Reprinted in *Proc. IEEE* pp. 86, 447-457, 1998.
- [17] C. STUDHOLME, D. L. G. HILL, AND D. J. HAWKES, Multiresolution voxel similarity measures for MR-PET registration, in *Information Processing in Medical Imaging (IPMI'95)* (Y. Bizais, C. Barillot, and R. Di Paola, Eds.), pp. 287-298. Dordrecht: Kluwer, 1995.
- [18] C. STUDHOLME, D. L. G. HILL, AND D. J. HAWKES, An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recogn.* vol. 32, pp. 71-86, 1999.
- [19] C. STUDHOLME, D. L. G. HILL, AND D. J. HAWKES, Automated 3D registration of MR and CT images of the head. *Med. Image Anal.*, vol. 1, pp. 163-175, 1996.
- [20] C. STUDHOLME, D. L. G. HILL, AND D. J. HAWKES, Automated 3D registration of MR and PET brain images by multi-resolution optimization of voxel similarity measures. *Med. Physics*, vol. 24, pp. 25-36, 1997.
- [21] Unidentified textbook.

- [22] PAUL VIOLA, *Alignment by Maximization of Mutual Information*, Ph.D. thesis, Massachusetts Institute of Technology, June, 1995.
- [23] FREDERICK H. WALTERS, LLOYD R. PARKER, JR., STEPHEN L. MORGAN, STANLEY N. DEMING, AUTHORS; STEVEN D. BROWN, ED., *Sequential Simplex Optimization: a technique for improving quality and productivity in research, development, and manufacturing*, Chemometrics Series, CRC Press, Boca Raton, FL, 1991, ISBN 0-8493-5894-9.
- [24] W. M. WELLS, P. VIOLA, H. ATSUMI, S. NAKAJIMA, AND R. KIKINIS, Multi-modal volume registration by maximization of mutual information. *Med. Image Anal.*, vol. 1, pp. 35-51, 1996.
- [25] J.B. WEST, J.M. FITZPATRICK, M.Y. WANG, B.M. DAWANT, C.R. MAURER, JR., R.M. KESSLER, R.J. MACIUNAS, C. BARILLOT, D. LEMOINE, A. COLLIGNON, F. MAES, P. SUETENS, D. VANDERMEULEN, P.A. VAN DEN ELSEN, S. NAPEL, T.S. SUMANAWEEERA, B. HARKNESS, P.F. HEMLER, D.L.G. HILL, D.J. HAWKES, C. STUDHOLME, J.B.A. MAINTZ, M.A. VIERGEVER, G. MALANDAIN, X. PENNEC, M.E. NOZ, G.Q. MAGUIRE, JR., M. POLLACK, C.A. PELIZZARI, R.A. ROBB, D. HANSON, AND R.P. WOODS, Comparison and Evaluation of retrospective intermodality brain image registration techniques, *J. Comput. Assist. Tomogr.*, vol. 21, pp. 554-566, 1997.