Statistical Dependence Estimation for Object Interaction and Matching

by

Kinh Tieu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2006

Author .....
Department of Electrical Engineering and Computer Science
September 1, 2006

Certified by .....
W. Eric L. Grimson
Professor, Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Statistical Dependence Estimation for Object Interaction and Matching

by
Kinh Tieu

Submitted to the Department of Electrical Engineering and Computer Science
on September 1, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This dissertation shows how statistical dependence estimation underlies two key problems in visual surveillance and wide-area tracking. The first problem is to detect and describe interactions between moving objects. The goal is to measure the influence objects exert on one another. The second problem is to match objects between non-overlapping cameras. There, the goal is to pair the departures in one camera with the arrivals in a different camera so that the resulting distribution of relationships best models the data. Both problems have become important for scaling up surveillance systems to larger areas and expanding the monitoring to more interesting behaviors. We show how statistical dependence estimation generalizes previous work and may have applications in other areas. The two problems represent different applications of our thesis that statistical dependence estimation underlies the learning of the structure of probabilistic models.

First, we analyze the relationship between Bayesian, information-theoretic, and classical statistical methods for statistical dependence estimation. Then, we apply these ideas to formulate object interaction in terms of dependency structure model selection. We describe experiments on simulated and real interaction data to validate our approach. Second, we formulate the matching problem in terms of maximizing statistical dependence. This allows us to generalize previous work on matching, and we show improved results on simulated and real data for non-overlapping cameras. We also prove an intractability result on exact maximally dependent matching.

Thesis Supervisor: W. Eric L. Grimson
Title: Professor, Electrical Engineering and Computer Science

Acknowledgments

I would like to thank my advisor Eric Grimson for many years of generous support and advice. Eric has allowed me to choose my own path and research, while steering me towards interesting excursions, culminating in this thesis. I also thank Leslie Kaelbling and Whitman Richards for their flexibility and understanding while serving on my thesis committee. Their advice has helped to shape and greatly improve this dissertation.

The Computer Science and Artificial Intelligence Laboratory and MIT has been a great place to learn and work. It is a place with some of the smartest yet different people around. I am fortunate to have been able to learn in such an environment and to have met such good friends. There are too many people to thank, mostly I thank the students for listening, asking, and sharing.

Besides my committee, the people who most influenced my work have also been collaborators: Paul Viola, John Fisher, Erik Learned-Miller, Chris Stauffer, Michael Siracusa.

I thank my friends who kept asking me when I would finish. I am grateful to my family for their patience and unwavering belief in me. Finally, to my dearest Tracy for putting up with all of it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This dissertation shows how *statistical dependence estimation* underlies two key problems in visual surveillance and wide-area tracking. The first problem is to detect and describe *interactions* between moving objects, such as one pedestrian following another. The second problem is to match objects between *non-overlapping cameras*, such as inferring that a pedestrian first seen at an entrance, and later at an exit, is the same person. Both problems have become important for scaling up surveillance systems to larger areas and expanding the monitoring to more interesting behaviors. We show how statistical dependence estimation generalizes previous work and may have applications in other areas. The two problems represent different applications of our thesis: statistical dependence estimation underlies the learning of the structure of probabilistic models.

The goal of surveillance is to monitor the behavior of objects [80, 11]. Behavior, consisting of actions, manifests itself through the motion of an object. Technological advances in tracking and classifying moving objects have dramatically increased the interest in automated surveillance. Machine surveillance has the potential to relieve the burden of human operators. Furthermore, machines are less susceptible to psychological factors such as limited attention and inconsistency. Although many aspects of behavior are intuitively clear and well-studied, the theory and practice of automated monitoring is far from mature.

Most of the research to date has focused on single-object behaviors, such as vehicles performing U-turns or pedestrians loitering [65, 27, 28, 2, 6, 26, 69, 15, 57, 19, 8, 9, 56, 7, 91]. However, it is also important to understand the interaction between objects. Previous related work [63, 41, 66, 35, 36] includes training classifiers to detect behaviors and adapting natural language-based grammars for describing behavior. The task was often to develop a computational framework for known interactions. Our goal is to understand the nature of interaction itself. We show how statistical dependence is a natural basis for a quantitative theory of object interaction. More specifically, the amount of statistical dependence measures the strength of interaction, while the form of dependence reflects the type of interaction. This makes intuitive notions about interaction precise, yet remains general enough to explain a variety of interactions.

Matching is the process of arranging the elements of two sets into a one-to-one corre-

spondence. In motion analysis [88], it is the problem of finding corresponding features in an image sequence. In target tracking, matching tracks and measurements is called data association, and is a mature field in the single camera case [76, 13]. We study a new case, where the problem is to pair observations of the same object in different cameras. In general, matching is a difficult problem because of the one-to-one constraints and the large feasible set of possible matchings.

Clearly, multiple cameras are required for wide-area surveillance. In some cases, the cameras have overlapping fields-of-view, so that both cameras see the same object at the same time. This dramatically reduces the ambiguity in matching, and we can apply techniques from multi-view geometry to obtain a relative calibration between the cameras [82, 52, 37]. In general, however, it is unrealistic to assume overlapping fields-of-view because cameras may be aimed in a wide variety of directions and often have limited viewing angles because of obstructions such as buildings. Furthermore, it is often desirable to monitor only a few areas with narrow fields-of-view, but which are spatially dispersed, such as entrances and exits. The problem then is matching across non-overlapping cameras. This is a much more difficult problem for two reasons: first, there is no geometric relationship between the camera views because, by definition, they image different parts of the scene; second, unlike the case of overlapping views, here the cameras never see the same object at the same time, which increases the match ambiguity. We show how statistical dependence can be used as a general measure of match quality, thus providing an optimization criterion for finding the best match. Our work generalizes previous work on inferring the topology of non-overlapping cameras [55] and also explicitly addresses the matching problem. We can regard inferring the topology of a network of cameras as a weaker form of relative calibration for the case of non-overlapping cameras. It helps us to determine, when an object leaves the field-of-view of one camera, where is it expected to be seen again, if at all, and when. Matching is the crucial component in this inference because a match across cameras implies that the field-of-views are topologically connected.

1.1 Object Interaction

> **interaction, *n.***
> Reciprocal action; action or influence of persons or things on each other.
> *Oxford English Dictionary* [78]

The problem of object interaction involves analyzing the influence of objects on each other. What kinds of interaction are there, and what is their structure? For example, consider the two pedestrians shown in Figure 1-1: X (center of first image) traces out some path, and Y (lower-left corner of first image) follows X. Intuitively, Y takes a path very similar to X's, but stays behind X. We can deconstruct the interaction as X's motion influencing Y's. Or, to put it another way, Y's motion is dependent on X's. More specifically, "following" behavior reflects a particular type of dependence. Thus, the qualitative notion of interaction can be quantified by measuring the amount of statistical dependence. Con-

Figure 1-1: Image sequence of one pedestrian following another. Images are approximately six seconds apart.

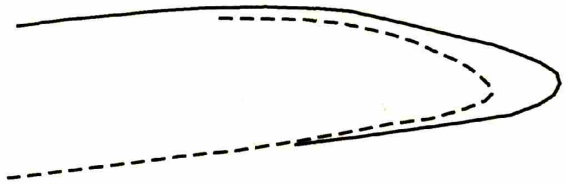


Figure 1-2: Trajectories for "Y following X".

versely, the lack of statistical dependence implies non-interaction, or independently moving objects. The kind of interaction is represented by the type of dependence. Unlike previous work, we do not simply fit observed data to predefined dependency structures, but aim to infer the model structure from the data.

The actions of an object are perceived through its motion. The time sequence of positions or states captures the motion history and is called a trajectory. Figure 1-2 shows the corresponding position trajectories for the example from Figure 1-1. The trajectory of an object can be modeled as a Markov process so that its present state y_t is dependent on only its previous state y_{t-1}:

$$p(y_t|y_1, ..., y_{t-1}) = p(y_t|y_{t-1}). \tag{1.1}$$

This model is inspired by the classical mechanics of particles where the state consists of position, velocity, and acceleration. The Markov assumption is commonly used in target tracking and other statistical applications because it simplifies the theory and makes computations efficient.

When there is no interaction, the trajectories of two objects can be described by two independent Markov processes. However, when there is an interaction (X influences Y),

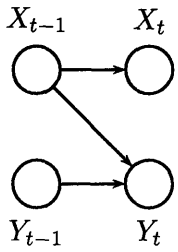


Figure 1-3: Dependency graph for "Y following X".

the model should reflect this influence:

$$p(y_t|y_{t-1}, x_{t-1}). \tag{1.2}$$

Relating back to our "Y follows X" example, the state of Y depends on not only its previous state, but also the previous state of X. Thus when X turns, Y must react to continue following X. Figure 1-3 is a graphical representation of this dependency structure. Nodes represent states and arcs denote influence. In general, the Markov assumption places arcs only between the previous and next states within a trajectory, while influence between objects requires arcs between trajectories. The details about the kind of interaction are further encoded by the specifics of conditional distribution $p(y_t|y_{t-1}, x_{t-1})$.

In the object interaction problem, the model for X and Y are unknown. Thus, detecting whether there is any interaction means determining the influences or structure of the model (that is, the arcs in the dependency graph). Furthermore, the exact form of the conditional probability distribution $p(y_t|y_{t-1}, x_{t-1})$ is unknown and also reflects the kind of interaction. We will show how statistical dependence estimation allows us to infer the structure of the interaction model for the observed data. The strength of this approach is that it is intuitive yet quantitative, and allows us to relax the assumptions on the nature of possible interactions.

As an example, consider again the image sequence in Figure 1-1. In the first image, pedestrian Y (lower-left corner) begins to follow pedestrian X (center). A third pedestrian Z, independent of X and Y, is seen in the third image. The trajectories for X and Y are shown in Figure 1-2, while the trajectories for X and Z are shown in Figure 1-4. The estimated statistical dependence is 1.08 for "Y following X" compared to 0.09 for "Z and X moving independently"[1]. In addition, the estimated statistical dependence between Z and Y is 0.13, which is consistent because if Z were dependent on Y, then because Y and X are dependent, there would be a dependency chain from Z to X, contradicting the previous result. In summary, our approach would infer the "Y following X" dependency model shown in Figure 1-3 along with a separate independent Markov chain for Z.

[1]See chapter three for details.

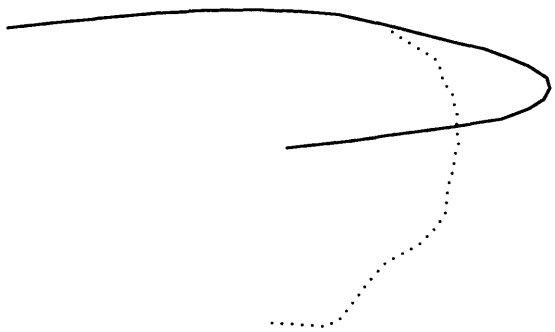


Figure 1-4: Trajectories for "Z and X moving independently".

1.2 Matching between Non-Overlapping Cameras

> **matching, *a.***
> That matches or corresponds; being a suitable counterpart; forming one of a pair or set.
> **matching, *n.***
> *Math.* A subset of the set of edges of a (usually bipartite) graph in which no two edges share a common vertex; a subgraph of a graph which may be constructed by selecting a subset of the set of vertices and assigning a distinct adjacent vertex to each member of this subset.
> *Oxford English Dictionary* [78]

In the previous section we discussed how statistical dependence estimation can be used to analyze the interaction between objects. Here we describe how statistical dependence can also be used to match objects across non-overlapping cameras. Consider the simplest case of two cameras (Figure 1-5), and assume that we can detect and track objects within each camera. Let $x_1, ..., x_n$ represent departures from the first camera, and $y_1, ..., y_n$ represent arrivals in the second camera[2]. A matching M is a permutation of the indices $1, ..., n$ such that the corresponding pairs are $(x_i, y_{M(i)})$. In other words, departure x_i is matched with arrival $y_{M(i)}$. A matching must be a one-to-one correspondence so that each x_i is matched with only a single $y_{M(i)}$ and *vice versa*. If the arrival times are in-order with respect to the departures times, then clearly the true matching (pairing departures and arrivals of the

[2]Refer to chapter four for discussion of the case of unequal numbers of arrivals and departures and the possibility of missing matches.

Camera 1 Camera 2

Figure 1-5: Two camera views of two portions of the same road.

same object) is simply the identity permutation ($M(i) = i$). However in many cases, such as vehicles and pedestrians moving with different speeds, arrival times will be out-of-order, so the true matching will be unknown.

How do we find the true matching? First, we must have some way of knowing that a proposed matching is correct. The probability of the data given a matching is

$$p(\{(x_i, y_{M(i)})\}|M) = p(\{y_{M(i)}\}|\{x_i\}, M)p(\{x_i\}). \tag{1.3}$$

The problem can be solved by finding the most likely matching:

$$\arg\max_M p(\{y_{M(i)}\}|p(\{x_i\})|M). \tag{1.4}$$

This is the matching that makes the matched pairs most probable. Still, searching for the true matching is difficult because there are an exponential number of possible matchings[3].

Previous related work simplified the problem by making various assumptions. The simplest case is when $p(y|x)$ is known exactly. The problem then turns out to be an instance of the classic assignment problem which, surprisingly, can be solved in polynomial time [67, 64, 48]. An example of such a case is when all objects have the same known distribution of transition times. When $p(y|x)$ is unknown but belongs to a parametric family such as a Gamma(a, b), gradient descent techniques can be used to find locally optimal solutions [25]. This case still assumes a single fixed relationship between x and y (subsuming the previous case), but does not assume that the relationship is known exactly (that is, the parameters a, b of the Gamma distribution are unknown).

What happens when the conditional distribution $p(y|x)$ is not only unknown, but may vary depending on the particular object? For example, vehicles and pedestrians have different transition times because they travel at different speeds. In other words, the relationship between x and y is potentially one of $p_1(y|x), ..., p_m(y|x)$, where m is unknown. A par-

[3]The number of permutations is $n!$, which by Stirling's approximation is $\approx \sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n}$ [18].

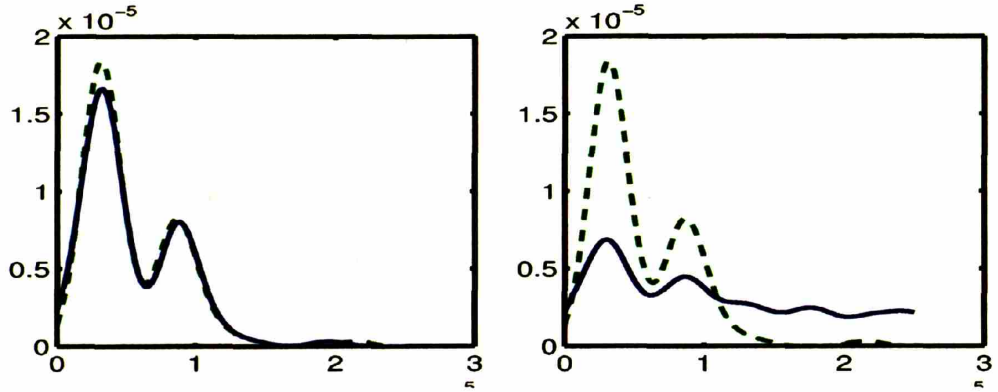
Figure 1-6: Example of objects between the camera views of Figure 1-5 matched by our approach. The top and bottom rows are objects from cameras 1 and 2, respectively. The second object from the right in the bottom row was considered an outlier.

ticularly simple way to formulate the problem is to model $p(y|x)$ as non-parametric. This is the most general case and includes the more restrictive cases previously described. The situation is that there are many possible relationships between each x_i and $y_{M(i)}$, none of which can, *a priori*, be assumed known. We would still like to find a best matching, but now we regard best as the matching that leads to the best model of the data. We are no longer searching over real valued parameters, but over explanatory models. This problem is unique because we must simultaneously solve for both the matching and the model. As an analogy, consider trying to solve a special kind of jigsaw puzzle where most of the pieces can be fit together and what the final image should look like is unknown. Our approach to matching corresponds to fitting the puzzle pieces together so that the resulting image looks typical or realistic. The idea is that typical, real images possess a great deal of structure, even if that structure can vary widely.

The best model of observed data can be quantified by its minimum description length [75]. The quality of a matching is based on how well the corresponding relationships explain the data and how compactly the relationships can be encoded. The more structure there is in the relationships, the more it can be compressed. It turns out that maximizing statistical dependency minimizes description length because dependency between x and y induces structure. Maximum dependence also implies maximum average probability because short code length corresponds to large average probability. Unfortunately, we will prove that exact non-parametric matching is intractable. Thus, we resort to a Markov chain Monte Carlo approximation algorithm as was done in related work [16, 71, 21].

As an example, consider the two cameras views of two portions of the same road in Figure 1-5. Both vehicles and pedestrians move from one camera to the other and do so with different, unknown transition times. Our approach searches for the matching with maximum statistical dependence. It is able to find many correct matches (86% in total), some of which are shown in Figure 1-6. In contrast, a naive approach using raw image appearance similarity had only 15% correct matches as shown in Figure 1-7. Given a matching, we can compute the corresponding travel times between cameras for the matched

Figure 1-7: Examples of objects matched by naive raw pixel appearance.



Our approach Previous approach

Figure 1-8: Our approach finds a travel time distribution that is closer to the true one (dotted) because we use a more general measure of statistical dependence and explicitly addresses the matching problem.

objects. In this case, the distribution of travel times should be roughly bi-modal because pedestrians and vehicles move at different speeds. A random matching generally produces a set of randomly distributed travel times. Figure 1-8 shows how our approach is better able to estimate the true travel time distribution than previous work.

1.3 Contributions

The primary contributions of this dissertation are two-fold:

1. Formulate object interaction in terms of dependency structure model selection,
   (a) Analyze the relationship between Bayesian, information theoretic/geometric, and classical methods for statistical dependence estimation,
   (b) Empirical validation on simulated and real interaction data,

2. Formulate matching problem in terms of maximizing statistical dependence,
   (a) Recast previous matching methods in our formulation,
   (b) Prove intractability of exact maximally dependent matching,
   (c) Generalize previous non-overlapping camera matching, and show improved results on simulated and real data.

1.4 Outline of the Dissertation

In retrospect, it is no surprise that statistical dependence estimation underlies both the problem of object interaction and matching across non-overlapping cameras because both problems involve estimating the structure of relationships between objects. Statistical dependence enables quantitative comparisons between dependency structures. In object interaction, we are interested in measuring the influence objects exert on one another. In matching, the goal is to pair the departures in one camera with the arrivals in a different camera so that the resulting distribution of relationships best models the data. The primary contribution of this dissertation is showing how statistical dependence estimation underlies these two problems and generalizes previous work. Our thesis is that statistical dependence estimation is the key to learning the structure of probabilistic models.

The next chapter introduces the statistical background necessary for a precise definition of statistical dependence. It also discusses how statistical dependence estimation relates to the machine learning problem of model selection. Chapter Three discusses the object interaction problem in depth. There we review previous work and show how statistical dependence naturally captures the idea of interaction. We show how the dependency model reflects the type of interaction, while the amount of statistical dependence measures the strength of interaction. The fourth chapter reviews the matching problem and studies the case of non-overlapping cameras. We show how statistical dependence, as a generic measure of match quality, can be used for optimizing the match between observations in non-overlapping cameras. We prove the problem to be intractable, and thus use a Markov chain Monte Carlo approximation to find a good match. Experiments validating our statistical dependence estimation approach to the problems of object interactions and matching across non-overlapping cameras are presented in their respective chapters. The concluding chapter summarizes the dissertation and discusses future work.

Chapter 2

Statistical Dependence

Our thesis is that statistical dependence is the key to learning the structure of probabilistic models. In the opening chapter we discussed how estimating dependency structure underlies the problems of object interaction and matching. In this chapter we show how to measure statistical dependence and how to estimate it from observed data. These general statistical techniques are the basis for applying our ideas to any problem involving the structure of probabilistic dependencies.

Information theory [77, 12, 54] provides the quantitative tools for measuring statistical dependence. The original motivation for a theory of information came from problems in communication engineering, such as transmitting messages across telephone wires. Kullback [49] then explored the intimate connection between information theory and statistical inference. We continue along these lines and study the relationship between information theory, statistical dependence and model structure.

Readers with a background in information theory can skip the introduction in Section 2.2 and Section 2.2.1. The material on information theory in this chapter is intended as a detailed but quick overview so proofs may be omitted. For a more rigorous treatment, the reader may refer to a standard textbook[12, 54].

2.1 Statistical Dependence

Our starting point is the abstract notion of statistical *independence* [14, 68, 18]. The non-intuitive, mathematical definition of statistical independence is that the joint distribution is a product of the marginal distributions:

$$p(x, y) = p(x)p(y). \tag{2.1}$$

We are illustrating the case of two random variables (RVs) X and Y; the extension to an arbitrary number of variables will become clear. Independence is more easily understood

$p(x,y)$		Y	
		0	1
X	0	$(1-p)(1-q)$	$(1-p)q$
	1	$p(1-q)$	pq

Table 2.1: Joint distribution of two independent binary RVs.

by factoring the joint distribution:

$$p(x,y) = p(x)p(y|x). \tag{2.2}$$

This leads to equality between the conditional and marginal distributions:

$$p(y|x) = p(y), \tag{2.3}$$

that is, knowing X does not change our knowledge about Y[1]. In other words, independent RVs do not interact.

As an example, consider tossing two fair coins so that the outcome of the second coin Y is independent of the outcome of the first coin X. The probability of each pair (x,y) has an equal probability of 0.25. In general, for $p(X=1)=p$ and $p(Y=1)=q$, the independent joint distribution is of the form shown in Table 2.1. There are only two instead of three degrees of freedom for the four possible outcomes.

With independence well-defined, we can naturally define dependence as the *absence* of independence, or that the RVs do interact. For example, consider the joint distribution of two binary RVs shown in Table 2.2. Clearly X and Y are dependent because $p(y|x)$ depends on the value of X. This is manifest in a higher probability for pairs with equal values, namely $(0,0)$ and $(1,1)$. In fact, these two pairs account for 90% of the probability. A physical explanation for this joint distribution is: first, toss a fair coin X, where $p(X=1)=0.5$; if $X=0$, toss a coin Y with bias $p(Y=1|X=0)=0.1$, otherwise, when $X=1$, toss a coin Y with bias $p(Y=1|X=1)=0.9$. The statistical dependence comes from the fact that the bias of the second coin Y depends on the result of the first toss X. Another way to look at it is to notice that it is impossible to put this dependent distribution into the form of the joint distribution for independent RVs as shown in Table 2.1.

In data modeling, assuming independence is equivalent to choosing a model structure *a priori*. This is a good idea when we have strong prior beliefs about the dependency structure of the data. However, in our case, the problems themselves, that is object interactions and matching, are explicitly about determining dependency structure. The key difference is that this type of learning must decide between different representations of the data, rather than just different values of the parameter of a single, fixed representation. In summary, our problems are truly about knowledge discovery and acquisition, an arguably higher level

[1]By symmetry we also have $p(x,y) = p(y)p(x|y)$, so knowing Y also does not affect our knowledge of X.

$p(x,y)$		Y	
		0	1
X	0	0.45	0.05
	1	0.05	0.45

Table 2.2: The RVs X and Y are dependent because the conditional distribution $p(y|x)$ depends on the value of X.

of learning than simply choosing parameter values.

Based on our *indirect* definition of statistical dependence, it is clear that any absence of independence is dependence. However this is only a *qualitative* definition, and we may wish to distinguish between joint distributions that are nearly independent versus ones that are far from independent (or very dependent). For example, if the joint probabilities in Table 2.2 are all close to 0.25, then the distribution would be similar to two tosses of a fair coin[2]. This naturally leads us to consider a *quantitative* measure of dependence (and independence) so that we could, for example, order the set of all distributions on two binary RVs by the amount of statistical dependence. Before we study such a measure, we first need some background in information theory.

2.2 Information Theory

If communication is the transmission of information, then information theory begins by quantifying the amount of information transmitted. Abstractly, we have a sender, a receiver, and a channel for communication between the two. Information is transmitted when the receiver learns something new from the sender. For example when the sender transmits a binary answer to a question, the receiver has obtained information. For the receiver to gain information or learn something new, the answer must obviously, at the outset, be *uncertain.* This leads to the idea of a probabilistic source because with a known deterministic source, there can be no uncertainty, and hence no information. In fact, the more uncertain the answer is at first, the more information the answer gives when it is received. It is important to keep in mind that probability distributions are the objects of interest in the algebra of information theory.

Perhaps the simplest probabilistic source is the outcome of a coin toss. If the coin is heavily biased towards heads (that is, $p(\text{heads}) \approx 1$), then when we learn the outcome, we are not very surprised because we would have bet on heads as the result. So the information content here is low. On the other hand, if the coin is fair, then we go from complete uncertainty before the toss, to complete certainty after the toss, so the information received is high. So we see that information is proportional to prior uncertainty.

[2]One meaning of similar could be that both distributions would generate samples in similar proportions, namely similar amounts of each pair $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$.

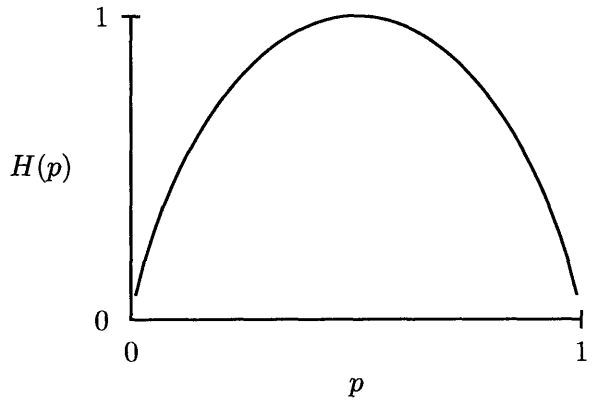


Figure 2-1: The entropy of a Bernoulli distribution decreases symmetrically to zero as the probability moves from 0.5 towards zero and one.

2.2.1 Entropy

How would we go about measuring information or uncertainty? One way is to specify a few intuitive axioms on any measure of uncertainty as a function[3] of a probability distribution, such as requiring that uncertainty increase with the number of equally likely outcomes. As a consequence, we can derive entropy,

$$H(p) = -\sum_i p_i \log p_i, \tag{2.4}$$

as the unique measure of information [77]. We see that entropy grows as $\log n$ for n equiprobable outcomes. We are illustrating the case of discrete probability distributions; we can extend to continuous RVs by replacing summations with integrals, with some caveats that we will point out when important.

As an example, a fair coin, which can be modeled as a Bernoulli(0.5) distribution, has a maximum entropy of one bit (in $\log_2$ units), and is, of course, also the coin with the most uncertain outcome. Figure 2-1 shows how the entropy of a Bernoulli distribution decreases symmetrically to zero as the probability moves from 0.5 towards zero and one, consistent with our intuition about the uncertainty of a Bernoulli distribution.

We can extend entropy to two RVs,

$$H(X,Y) = -\sum_{x,y} p(x,y) \log p(x,y), \tag{2.5}$$

[3]For notational convenience, a function f of a probability distribution is sometimes written as $f(p)$, and other times as $f(X)$, where RV X has probability distribution p.

and define conditional entropy,

$$H(Y|X) = -\sum_x p(x)H(Y|x). \tag{2.6}$$

Straightforward calculations with entropy yield a rich algebraic structure, which greatly aids reasoning with information sources [77, 12, 54]. We have the chain rule:

$$H(X,Y) = H(X) + H(Y|X), \tag{2.7}$$

so that the uncertainty of the pair (X,Y) is the sum of the uncertainty in X and the uncertainty in Y, conditioned on knowing X. We also have the fact that conditioning reduces entropy:

$$H(Y|X) \leq H(Y), \tag{2.8}$$

which intuitively says that knowing X, or having more information, can only reduce the uncertainty in Y. Thus we see that entropy, which resulted from a set of axioms on any measure of uncertainty, has a natural and intuitive set of algebraic properties. We can think of entropy as measuring the randomness of an RV. Because structure is by definition less random than chaos, entropy will prove crucial to characterizing dependency structure.

2.2.2 Mutual Information

How is entropy related to statistical dependence? If two RVs are independent, $p(y|x) = p(y)$, and the conditional entropy $H(Y|X)$ is equal to the marginal entropy $H(Y)$. Then the joint entropy $H(X,Y) = H(X) + H(Y)$, so intuitively, the uncertainty in two independent RVs is simply the sum of their individual uncertainties. Once again we can naturally regard dependence as the absence of these conditions. In particular we can use the mutual information (MI),

$$I(X;Y) = H(Y) - H(Y|X), \tag{2.9}$$

to measure exactly the extent to which X and Y are dependent [77, 12, 54]. MI is simply the reduction in uncertainty of one RV given another. Two RVs are independent if they have zero mutual information because conditioning does not reduce entropy in that case. Conversely, two RVs are dependent if knowing one tells you something about the other. It can be shown that MI is symmetric,

$$I(Y;X) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(X;Y), \tag{2.10}$$

which is a desirable property of a measure of statistical dependence.

MI is always non-negative because $H(Y) \geq H(Y|X)$. Statistical dependence is highest when $H(Y|X) = 0$, or when knowing X removes all of the uncertainty in Y (that is, Y is a deterministic function of X)[4]. MI is a function of both unconditional and conditional uncertainties. Thus, for Y to be strongly dependent on X, not only must the conditional

[4]For continuous Y, a deterministic function leads to infinite information.

entropy $H(Y|X)$ be small, but the prior entropy $H(Y)$ must also be large. This is important because, for example, in object interactions, we are more sure that pedestrian Y is following X when Y stays behind X even as X makes many turns, than if X's trajectory was a straight path.

As an example, consider again the distribution of two binary RVs shown in Table 2.2. The marginal entropies $H(X)$ and $H(Y)$ are both 1 bit, making the RVs, on their own, maximally uncertain. However, the conditional entropy, $H(Y|X)$ is 0.3251 bits which says that X tells us something about Y. Indeed, $I(X;Y) = 1 - 0.3251 = 0.6749$ bits, consistent with the fact that X and Y are statistically dependent. The MI reaches its maximum of one bit when both $p(0,0) = p(1,1) = 0.5$, or Y is a deterministic function of X.

We have now reinterpreted statistical dependence in terms of mutual information. The advantage of this quantitative definition is that it allows us to measure the amount of statistical dependence. For two RVs, statistical dependence ranges from zero to $H(Y)$[5]. We can now answer to what extent are two RVs dependent.

2.2.3 Kullback-Leibler Divergence

The Kullback-Leibler (KL) divergence [49, 12, 54],

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}, \tag{2.11}$$

is the average log-likelihood ratio in a *hypothesis test* [14, 68] between p and q, when x is distributed according to p. It can be interpreted as a pseudo-distance[6] between probability distributions because when p and q are far apart, they are easily distinguished by sampling. For example the KL divergence between two Gaussian distributions increases with the distance between their means.

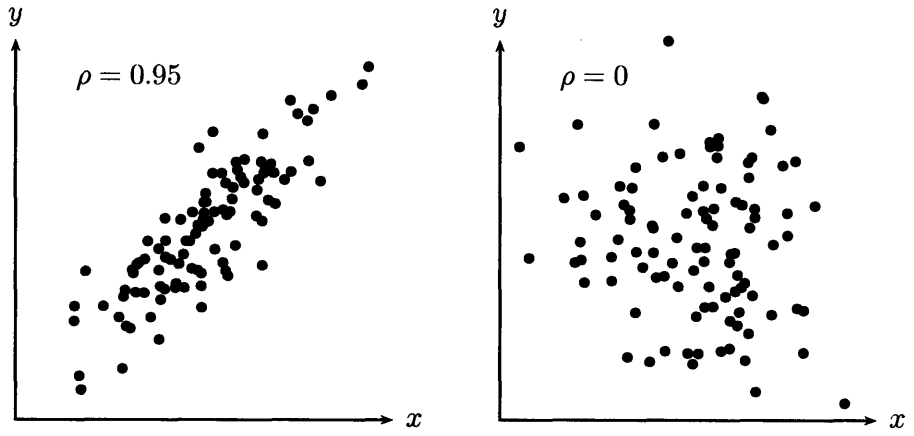
KL divergence generalizes the information theoretic quantities defined previously and makes the connection to statistical inference explicit. It enables us to re-interpret MI as the divergence between the joint distribution and the product of marginals:

$$I(X;Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{2.12}$$

KL divergence makes clear the close connection between MI and statistical dependence by way of deciding between dependent and independent hypotheses of the data. In summary, when statistical dependence is high, conditional entropy $H(Y|X)$ is low, MI is high, and KL divergence between the joint $p(x,y)$ and the product of marginals $p(x)p(y)$ is high. All this means is that X and Y are strongly related in the sense that although highly uncertain by themselves, one can be well predicted with knowledge of the other.

[5]For continuous RVs, statistical dependence ranges from zero to the logarithm of the ratio of the volumes of Y and $Y|X$. The volume is the effective alphabet size of coding the RV.

[6]Unlike a true distance, KL divergence is asymmetric and does not satisfy the triangle inequality.



Dependent Independent

Figure 2-2: When Y is dependent on X, Y is better predicted given X. When Y is independent of X, predicting Y is no easier when given X.

As an example, consider jointly Gaussian RVs X and Y. Statistical dependence can be measured by computing the KL divergence between the joint Gaussian and the product of marginal Gaussians:

$$\int_{x,y} G(x,y;\mu,\Sigma) \log \frac{G(x,y;\mu,\Sigma)}{G(x;\mu_X,\sigma_X)G(y;\mu_Y,\sigma_Y)} = -\frac{1}{2}\log(1-\rho^2), \tag{2.13}$$

where G is the Gaussian distribution and μ_X, σ_X, μ_Y, σ_Y are the marginal elements of the mean μ and the covariance matrix Σ. The correlation coefficient is

$$\rho = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}, \tag{2.14}$$

where σ_{XY} is the off-diagonal element of Σ. This result is consistent with the fact that ρ measures correlation, which is a commonly used measure for statistical dependence. We see here that this is valid for Gaussians. When $\rho = 0$, the RVs are uncorrelated (independent); otherwise, larger ρ means more correlated (dependent). Figure 2-2 shows how with highly correlated Gaussians, Y is well predicted by X; on the contrary, with uncorrelated Gaussians, predicting Y given X is no better than without X. By measuring the distance between the dependent joint distribution and independent product of marginals, the KL divergence is exactly the amount of statistical dependence.

We can also easily generalize to more than two RVs, so that for example the statistical dependence between X, Y, and Z can simply be measured with

$$D(p(x,y,z)||p(x)p(y)p(z)). \tag{2.15}$$

It should not be surprising that information theory is a useful background from which

to analyze statistical dependence because both dependence and information quantify the probabilistic relationship between two RVs. Information is about predicting one RV from another, and statistical dependence is exactly how one variable influences the other. By using information to measure dependence, we are measuring influence by its effect on prediction (or uncertainty).

2.2.4 Information Geometry

To gain further insight into the hypothesis testing formulation of statistical dependence, we look at an information geometric interpretation of KL divergence [49, 1, 12]. Let M_1 be the manifold of dependent distributions $p(x, y)$, and M_0 be the manifold of independent distributions $p(x)p(y)$. The space M_1 contains all distributions on two RVs, while M_0 is a subspace of M_1. Clearly, a probability distribution p is independent if it is a member of M_0. As p moves away from M_0 it becomes more dependent. The distance of p from M_0 is related to the KL divergence between p and its projection onto M_0, $p^{\perp}$. This is the distribution in M_0 that minimizes the KL divergence to p as shown in the following theorem.

Theorem 1. *The KL divergence between a dependent joint distribution and an independent product of marginal distributions is minimized by the corresponding marginals $p(x)$ and $p(y)$ of the joint:*

$$D(p(x,y)||p(x)p(y)) \leq D(p(x,y)||f(x)g(y)), \tag{2.16}$$

where f and g are distributions for X and Y, respectively.

Proof.

$$D(p(x,y)||f(x)g(y)) = \int p(x,y) \log \frac{p(x,y)}{f(x)g(y)} dxdy \tag{2.17}$$

$$= \int p(x,y) \log p(x,y) dxdy - \int p(x,y) \log f(x) dxdy - \int p(x,y) \log g(y) dxdy \tag{2.18}$$

$$= \int p(x,y) \log p(x,y) dxdy - \int p(x) \log f(x) dx - \int p(y) \log g(y) dy. \tag{2.19}$$

$$\tag{2.20}$$

Because $\int p(x) \log p(x) dx \geq \int p(x) \log p'(x) dx$ for any p', $p(x)$ and $p(y)$ are the minimizing marginal distributions. □

Consider again the example of two binary RVs. Here, M_1 is a three-dimensional manifold determined by $p(0,0)$, $p(0,1)$, and $p(1,0)$. Under M_0, the joint distribution must be of the form shown in Table 2.1, so M_0 is a sub-manifold with only two degrees of freedom, $p(X = 1)$, and $p(Y = 1)$. The statistical dependence of a distribution p is the distance

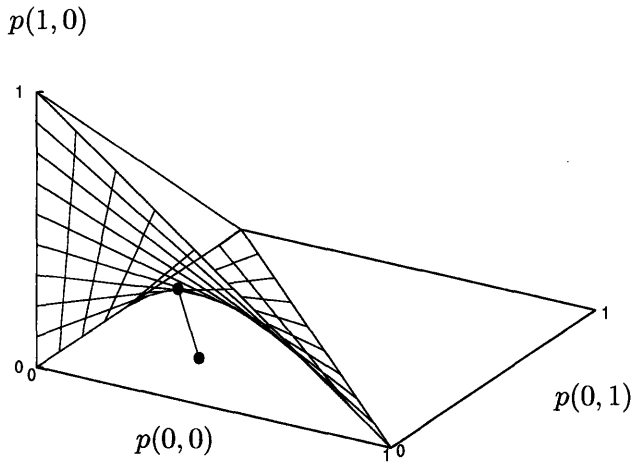


Figure 2-3: The dependent manifold M_2 (tetrahedral solid) is the set of all probability distributions on two binary RVs. The independent manifold M_0 (mesh) is a two-dimensional sub-manifold of M_1. The length of the line between the two points represents the KL divergence between a distribution and its projection onto M_0, and measures the amount of statistical dependence.

from p to its best-approximating distribution in M_0 in terms of KL divergence as shown in Figure 2-3.

The information geometry analogy also allows us to interpret our results in the jointly Gaussian case. There, we can parameterize the dependent manifold with (μ, Σ), where $\mu \in R^2$ and $\Sigma \in R^{2\times 2}$: positive definite, and the independent manifold with $(\mu_X, \sigma_X, \mu_Y, \sigma_Y)$, where $\mu_x, \mu_y, \sigma_X, \sigma_Y \in R$. It then becomes clear that the independent space is a sub-manifold of the dependent space because $\mu = (\mu_X, \mu_Y)$, and (σ_X, σ_Y) are the diagonal terms of Σ. Indeed, the structure of the covariance matrix naturally governs the dependency structure of jointly Gaussian RVs.

2.3 Dependency Structure

We have seen how information theory provides a quantitative and intuitive background for studying statistical dependence. We will now investigate how information and dependence is related to the structure of probabilistic models. A probabilistic model is designed to explain observed data by accounting for its *regularities*. This is done by quantifying the

$p(x, y)$ Dependent

$p(x)p(y)$ Independent

Figure 2-4: Dependent and independent dependency graphs for two RVs and their corresponding factorizations.

relationship between the RVs. A simple example of a model is a linear relationship:

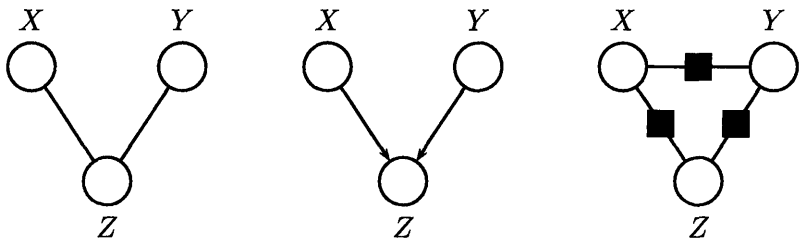
$$Y = \alpha X + N, \tag{2.21}$$

with additive Gaussian noise $N \sim G(0, \sigma)$. The regularity is expressed as a noisy linear relationship. Indeed our earlier example of correlated jointly Gaussian RVs is captured exactly by this type of model.

By dependency structure, we do not mean the value of α, or even the linearity, but the dependence between X and Y. In this case, the alternative dependency structure would be simply treating X independent of Y. The same dependency structure can instantiate different probability distributions by varying the functional form and/or the parameter values. In this sense, the dependency structure is the most basic and stable property shared by all of the probability distributions. Statistical dependency structure is a useful concept because it provides a general yet compact way to characterize probabilistic models. It captures the representational structure of any model. In addition, the sparsity of a dependency structure has a profound on the efficiency of the storage and computational requirements of probabilistic inference. In short, explicitly modeling dependency structure is both computationally practical and cognitively plausible. For us, determining the dependency structure is exactly the question posed by the problems of object interaction and matching.

2.3.1 Dependency Graphs

Dependency graphs [72, 50, 42] are a convenient way to represent dependency structure. The idea is to borrow the syntax from graph theory but use probabilistic semantics. Each node is a RV and arcs represent dependency information. The overall joint probability of the variables can then be written as a product of appropriate functions of subgraphs. The advantage of dependency graphs (as with graphs for other representing other types of knowledge) is their ability to decompose the overall structure into the relationship between local structures based on the topology of the graph. In the simplest cases, *separation* in a graph corresponds to independence conditioned on the separating nodes. For example, the case of two variables discussed previously is represented simply as either a connected (dependent) or disconnected (independent) graph of two nodes as show in Figure 2-4.



$\phi_{XY}(x,y)\phi_{YZ}(y,z)$ Markov network | $p(x)p(y)p(z|x,y)$ Bayes net | $f(x,y)g(y,z)h(z,x)$ Factor graph

Figure 2-5: Different types of dependency graphs and their corresponding factorizations.

For three variables, there are many more possibilities, corresponding to different factorizations of $p(x,y,z)$. For example, a *Markov network* [72, 23] can be used to represent the factorization

$$p(x,y,z) \propto \phi_{XY}(x,y)\phi_{YZ}(y,z), \tag{2.22}$$

where the potential functions ϕ represent compatibilities whose normalized product gives probabilities. The corresponding graph is the Markov network shown in Figure 2-5. The semantics encoded by the chain is that X and Z are conditionally independent given Y. This type of dependency models, for example, the direct physical analogy of a kinematic chain, where knowing Y determines X irrespective of Z.

Bayes nets [72, 30] introduce a notion of causality by using directed edges. For example, the Bayes net in Figure 2-5 represents the factorization

$$p(x,y,z) = p(x)p(y)p(z|x,y), \tag{2.23}$$

which can describe a decision Z based on two independent coin tosses X and Y. Causally we think of X and Y as independent, but from a probabilistic standpoint, knowing Z renders X and Y dependent. For example if Z is the function that indicates whether X and Y are the same or different, then knowing Z and X clearly tells us something about Y.

What if we want the joint to be a product of every pairwise interaction:

$$p(x,y,z) \propto f(x,y)g(y,z)h(z,x)? \tag{2.24}$$

There is no way to consistently represent this with either a Bayes or Markov network. The more general dependency graph we can use is a *factor graph* [47] shown in Figure 2-5. As the name implies, factor graphs are designed to represent the factorization of a function by adding function nodes, which makes the graph bipartite (edges only occur between variable and function nodes) and edges undirected.

To recapitulate, first we formulated the abstract notion of statistical dependence in terms of MI and KL divergence so that it could be measured quantitatively. Then, we showed how statistical dependence is related to dependency structure, and introduced dependency

graphs as a convenient representation of structure. Up to now, we have assumed knowledge of the probability distributions discussed. In practice, we are only given samples from the distributions. Thus the next section discusses how to estimate our items of interest from observed data.

2.4 Inference and Estimation

Given data, the most common form of statistical inference is to first choose a model, and then estimate the parameters of that model by maximizing the likelihood of the data under the model [14, 5, 54]. In our previous example of linear-Gaussian relationships, this means estimating the value of α. The key limitation is that the dependency structure of the model, the direct linear-Gaussian relationship, is assumed instead of inferred from the data. Thus the form of statistical dependence between the variables is already determined when the model is chosen because it is the structure of the model that determines the nature of statistical dependence. What we infer, namely the value of α, is not the statistical dependence, but the exact form of the linear-Gaussian dependency. The fact that the dependency is linear is already determined by the model, and the strength of dependence is also known if we fix σ.

Assuming a dependency structure is tantamount to assuming that we know the type of object interaction or that we know the dependence between two non-overlapping cameras. Of course, these are exactly the questions posed by our motivating problems. In other words, the dependency structure is unknown, and the goal is to infer this structure. Unfortunately, with fewer assumptions the inference is made more difficult.

2.4.1 Model Selection

The problem of inferring dependency structure falls under the general problem of model selection [5, 54, 75]. The goal is to choose the model that best explains the data. The best model is understood as the one that generalizes the best in terms of accurate predictions on future unseen data. By model we mean a set of probability distributions, such as the set of all bivariate Gaussians.

In principle, the solution to the model selection problem is clear: simply choose the model that is *a posteriori* most probable. We illustrate the case for two models; the extension to multiple models is straightforward. Let the two models be M_1 and M_0. The quantity of interest is the log posterior odds of the models given data D,

$$\log \frac{p(M_1|D)}{p(M_0|D)} = \log \frac{p(D|M_1)p(M_1)}{p(D|M_0)p(M_0)} \tag{2.25}$$

$$= \log \frac{\int p(D|\theta_1, M_1)p(\theta_1|M_1)d\theta_1}{\int p(D|\theta_0, M_0)p(\theta_0|M_0)d\theta_0} + \log \frac{p(M_1)}{p(M_0)}. \tag{2.26}$$

Notice that this integrates over all possible probability distributions in each model. This is required because we do not know which distribution (that is, the exact value of θ_i)

the data actually came from. The log prior odds of the models provides the appropriate threshold for choosing between the models; when the priors are equal, the threshold is zero. The advantage of this Bayesian model selection over traditional hypothesis and goodness-of-fit testing is that it directly computes the quantities of interest. In addition, because the computed quantities are probabilities, they are automatically calibrated. Furthermore, Bayesian model selection is conceptually simpler. The primary disadvantage is that the computations are usually more difficult.

An interesting feature of Bayesian model selection is that it automatically applies *Occam's Razor* [5, 54, 53] to penalize more complex models. Consider the space of all possible data sets. A simple model assigns probability to only a few data sets, while a complex model distributes probability to more data sets. Because probability must integrate to unity, the probability assigned to some data sets will be larger in the simple model than the complex one. Thus, if the observed data happens to be one of these data sets, the simple model will provide a better explanation. Another way to look at it is to assume that the data is only explained well by a single $\hat{\theta}_i$ in each model. Because the more complex model has to distribute probability over more parameter values, the weight $p(\hat{\theta}_i|M_i)$ is smaller for the more complex model.

As a particularly simple example, consider two models for coin tosses: M_0 is a fair coin and M_1 is a biased coin. Let the observed data be 52 heads out of 100 tosses. Now M_0 contains a single probability distribution $\theta_0 = 0.5$, giving $\log p(D|M_0) = \log 0.5^{100} = -69.31$. The best distribution in M_1 is $\hat{\theta}_1 = 0.52$, giving $\log p(D|\hat{\theta}_1, M_1) = \log(0.52^{52}0.48^{48}) = -69.23$. However, M_1 also contains other distributions, one for each value of θ_1. A reasonable prior for θ_1 is the Jeffreys prior [39], which is designed to be invariant to transformations of the parameter. For a binomial, it is a Beta(1/2,1/2) distribution which favors more biased coins as shown in Figure 2-6. With a Jeffreys prior, $\log p(D|M_1) = \log \int p(D|\theta_1, M_1)p(\theta_1|M_1)d\theta_1 = -71.77$. Thus, with equal prior odds for the models, we would decide M_0, the fair coin, in accord with intuition. In this case of 100 tosses, we need either 62 or more heads, or 38 or fewer heads for M_1 to be the more probable model. If the number of tosses is 10, then 8 or more heads, or 2 or fewer heads is required for M_1 to be more probable. In general, if the observed proportion of heads deviates from 0.5, then as the number of tosses increases, the biased coin model becomes more probable; again, agreeing with intuition.

Notice that the log posterior odds in Bayesian model selection has a similar form to the KL divergence. This enables us to draw a connection to classical hypothesis testing, except here we average over all probability distributions in a model. If we approximate the integrals based on a single distribution θ_i, then we get exactly approximations to KL divergences. The next section will show this in detail when testing different dependency models.

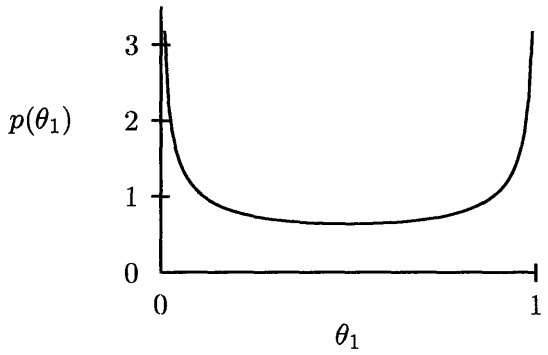


Figure 2-6: Jeffreys prior for the binomial distribution. Higher probability is assigned to more biased distributions.

2.4.2 Dependency Structure Selection

Dependency structure selection is a special case of model selection where the dependency structure is what differentiates the models. In other words, we want to infer the most probable dependency graph given observed data. Naturally the inference should integrate over all possible parameter values.

Consider testing the statistical dependence between two RVs as illustrated in Figure 2-4. The *Bayes factor* [4, 43] is

$$\frac{p(x,y|M_1)}{p(x,y|M_0)} = \frac{\int p(x,y|\theta_1,M_1)p(\theta_1|M_1)d\theta_1}{\int p(x|\theta_X,M_0)p(\theta_X|M_0)d\theta_X \int p(y|\theta_Y,M_0)p(\theta_Y|M_0)d\theta_Y}, \tag{2.27}$$

which simply compares a dependent versus independent model. To derive the connection to KL divergence, we take the logarithm of a Laplace (saddle-point) approximation of the integrals around the posterior modes $\hat{\theta}_i$:

$$\log \frac{p(x,y|M_1)}{p(x,y|M_0)} \approx \log \frac{p(x,y|\hat{\theta}_1,M_1)p(\hat{\theta}_1|M_1)\Delta_{\theta_1}}{p(X|\hat{\theta}_X,M_0)p(\hat{\theta}_X|M_0)\Delta_{\theta_X}p(Y|\hat{\theta}_Y,M_0)p(\hat{\theta}_Y|M_0)\Delta_{\theta_Y}} \tag{2.28}$$

$$= \log \frac{p(x,y|\hat{\theta}_1,M_1)}{p(x|\hat{\theta}_X,M_0)p(y|\hat{\theta}_Y,M_0)} + \log \frac{p(\hat{\theta}_1|M_1)\Delta_{\theta_1}}{p(\hat{\theta}_X|M_0)\Delta_{\theta_X}p(\hat{\theta}_Y|M_0)\Delta_{\theta_Y}} \tag{2.29}$$

$$= \sum_i \log \frac{p(x_i,y_i|\hat{\theta}_1,M_1)}{p(x_i|\hat{\theta}_X,M_0)p(y_i|\hat{\theta}_Y,M_0)} + O \tag{2.30}$$

where O is the Occam factor [5, 54, 53] which penalizes more complex models. If the data actually came from the M_1 distribution (dependent), then the log Bayes factor is approximately

$$nD(p(x_i,y_i|\hat{\theta}_1,M_1)||p(x_i|\hat{\theta}_X,M_0)p(y_i|\hat{\theta}_Y,M_0)). \tag{2.31}$$

On the other hand, if the data came from the M_0 distribution (independent), then we have

$$-nD(p(x_i|\hat{\theta}_X, M_0)p(y_i|\hat{\theta}_Y, M_0)||p(x_i, y_i|\hat{\theta}_1, M_1)) = 0, \tag{2.32}$$

because a dependent joint distribution can always explain an independent product of marginals distribution. This is easily seen by recalling that $p(x, y) = p(x)p(y|x)$, so that the dependent distribution can mimic the independent one by simply letting $p(y|x) = p(y)$. In general this occurs when the models are nested, meaning one model is a subset of the other. Related analysis can be found in [12, 92, 34]. Learning structure from data has increasing become an active research area [10, 72, 30].

As an example, consider again the space of probability distributions on two binary RVs as shown in Figure 2-3. The dependent model is a Multinomial(θ) while the independent model is a product of a Multinomial(θ_X) and a Multinomial(θ_Y). By using a conjugate Dirichlet prior we can compute the evidence analytically [92]:

$$\int p(D|\theta)p(\theta)d\theta = \int_{\theta_i>0;\sum_i \theta_i=1} \prod_i \theta_i^{n_i} \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1} \tag{2.33}$$

$$= \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int_{\theta_i>0;\sum_i \theta_i=1} \prod_i \theta_i^{n_i + \alpha_i - 1} \tag{2.34}$$

$$= \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \frac{\prod_i \Gamma(n_i + \alpha_i)}{\Gamma(n + \sum_i \alpha_i)}, \tag{2.35}$$

where n_i are the multinomial counts, and α are the parameters for the Dirichlet prior. We can then compute the Bayes factor:

$$\frac{p(x, y|M_1)}{p(x, y|M_0)} = \frac{\frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \frac{\prod_i \Gamma(n_i + \alpha_i)}{\Gamma(n + \sum_i \alpha_i)}}{\frac{\Gamma(\sum_i \alpha_{xi})}{\prod_i \Gamma(\alpha_{xi})} \frac{\prod_i \Gamma(n_{xi} + \alpha_{xi})}{\Gamma(n_x + \sum_i \alpha_{xi})} \frac{\Gamma(\sum_i \alpha_{yi})}{\prod_i \Gamma(\alpha_{yi})} \frac{\prod_i \Gamma(n_{yi} + \alpha_{yi})}{\Gamma(n_y + \sum_i \alpha_{yi})}} \tag{2.36}$$

$$= \frac{\prod_i \Gamma(n_i + \alpha_i)}{\prod_i \Gamma(n_{xi} + \alpha_{xi}) \prod_i \Gamma(n_{yi} + \alpha_{yi})} \frac{\Gamma(n_x + \sum_i \alpha_{xi})\Gamma(n_y + \sum_i \alpha_{yi})}{\Gamma(n + \sum_i \alpha_i)} \tag{2.37}$$

$$\frac{\Gamma(\sum_i \alpha_i)}{\Gamma(\sum_i \alpha_{xi})\Gamma(\sum_i \alpha_{yi})} \frac{\prod_i \Gamma(\alpha_{xi}) \prod_i \Gamma(\alpha_{yi})}{\prod_i \Gamma(\alpha_i)} \tag{2.38}$$

For multinomials, the classical χ^2 statistic [14, 68, 49] can be used to test for statistical dependence:

$$\chi^2 = \sum_i \frac{(p_i - q_i)^2}{q_i}, \tag{2.39}$$

where p and q are maximum likelihood estimates of the dependent and independent distributions, respectively. The statistic is a goodness-of-fit criterion that measures how much the counts from the two distributions are likely to differ. In fact, we can derive χ^2 as an approximation to the KL divergence between p and q by taking a Taylor series expansion

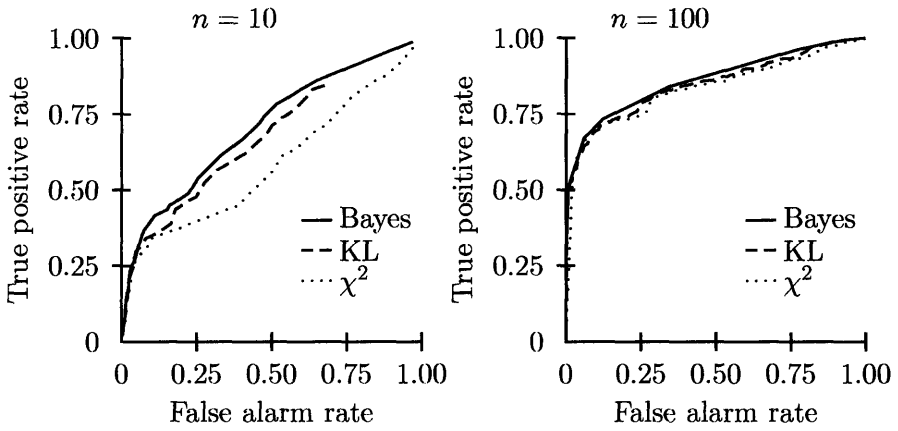


Figure 2-7: The ROC curve for deciding between dependence and independence. Bayes performs the best on average, especially when the number of data points is small. The KL approximation also performs well when the number of data points is large.

[12, 49]:

$$D(p||q) = \sum_i p_i \log \frac{p_i}{q_i} \tag{2.40}$$

$$\approx \sum_i (p_i - q_i) + \frac{1}{2}\frac{(p_i - q_i)^2}{q_i} \tag{2.41}$$

$$= \frac{1}{2}\chi^2. \tag{2.42}$$

We see that the χ^2 statistic is a quadratic approximation to the KL divergence.

As a simple demonstration of these ideas, we performed Monte Carlo simulations for the two binary RVs case. We randomly generated 1000 dependent and independent distributions. We then sampled points from each distribution and performed model selection. Figure 2-7 shows the receiver operating characteristic (ROC) for the Bayes, KL approximation, and a traditional χ^2 test for dependence. Truly dependent distributions generally have high estimated KL divergence. Some of the randomly sampled dependent distributions are very close to independent distributions and thus have low KL divergence. This also pushes the ROC curve down because those distributions have low statistical dependence. On average, Bayes performs the best, especially when the number of data points is low. When there is a lot of data, the KL approximation also performs well. As a further illustration, Figure 2-8 shows the area under the ROC curve as a function of n. For all n, Bayes performs the best, followed by KL, and finally χ^2, with the performance difference decreasing with n, as expected.

In summary, statistical dependence estimation involves comparing a dependent versus an independent explanatory model for the observed data. This model selection can be approx-

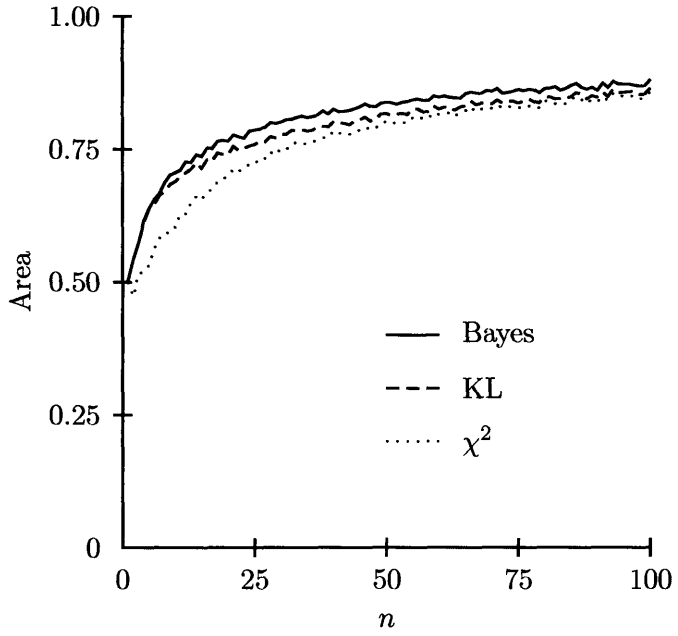


Figure 2-8: The area under the ROC curve as a function of n for deciding between dependence and independence. Bayes out-performs KL which out-performs χ^2, with the differences decreasing with n.

imated as computing the KL divergence between the posterior most probable probability distributions in each model.

2.5 Order, Regularity, and Structure

Recall that we started by defining statistical dependence abstractly and then related it to entropy, mutual information, and KL divergence. Model selection tied these concepts to hypothesis testing of dependency structures. Another view of these ideas is in terms of *complexity* and organization. Perception can be thought of as the identification of patterns in data. By pattern we mean order as in some sort of regularity. This regularity is captured by the underlying structure of the pattern, and for probabilistic patterns, the structure is exactly the statistical dependency structure. As an analogy, in graphics we want variability from known structure, while in vision and pattern recognition we want the underlying structure despite the variability in the data. The key idea is that dependency relates variables so that from an unstructured set of variables we get systems of structured probabilistic models. Measuring statistical dependence is a way of measuring structure and is therefore a way to infer structure. We believe this to be an important aspect of perception and learning which has received less attention than it deserves.

2.6 Summary

We discussed the connections between statistical dependence, information theory, and model structure. Entropy, mutual information, and KL divergence turn out to be the key quantities for measuring dependence. Model selection compares model dependency structures and thus also measures dependence by comparing dependent versus independent models.

We have shown details for the simple cases of two RVs and linear-Gaussian models, but the approach generalizes in a straightforward manner to multiple variables and arbitrary distributions. In the next chapter we apply these ideas to the problem of object interactions.

Chapter 3

Object Interaction

In our introductory chapter, we briefly discussed how statistical dependence could be used to study object interaction. With the technical background from the previous chapter, we can now study the problem in earnest. This chapter will show how statistical dependence is the key concept for understanding object interaction. After presenting the framework, we describe experiments demonstrating the performance of our approach.

We will begin by describing the nature of object interaction and show how statistical dependence allows us to formulate a *quantitative theory*. The theory is simple and intuitive, yet allows for the detailed measurement of object interaction in a wide range of scenarios.

For our purposes, a good example of object interaction is the cartoon video of Heider and Simmel [31] shown in Figure 3-1. Although the objects were simple geometric figures, humans tended to explain what they saw in terms of interactions such as pursuit. Thus, it is clear that it is the *motion* of objects that cause us to perceive interactions. In the extreme case, even the motion of points is enough, such as the recognition of human motion from point light displays[1] [40]. Furthermore, in far-field surveillance, an object occupies a small number of pixels, so we can reliably track only its gross motion.

The reason that humans interpret the motion of objects in terms of interactions is because an interaction model provides a better explanation of the observed data. By better, we mean that the interaction model improves our understanding of the object motions by offering more accurate predictions. For example, by knowing that Y is in pursuit of X, we can more accurately predict Y's motion with knowledge of X's.

3.1 Interaction and Statistical Dependence

For simplicity, consider the case of two objects X and Y. Recall that the important feature is the motion of the objects, so we can think of the objects as simply points. Motion can be represented as a *trajectory* or time series of states $X(t)$. A state usually consists of position, velocity, and possibly acceleration. Intuitively, an interaction between X and Y means that

[1] Perception with minimal stimuli is also apparent in depth perception from random dot stereo-grams.

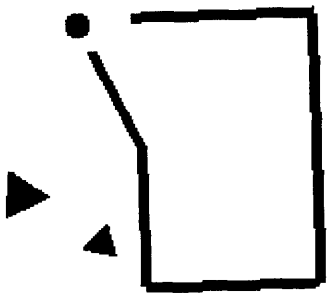


Figure 3-1: A frame from Heider and Simmel's cartoon video, which humans interpret in terms of interactions such as pursuit.

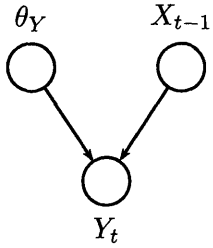


Figure 3-2: Dependency graph for X influencing Y.

there is some influence of the objects on each other. We can formalize this by defining the probability distributions:

$$p(x_t|\theta_X) \text{ and } p(y_t|x_{t-1}, \theta_Y). \tag{3.1}$$

This simply states that Y depends of X, along with other factors θ_Y. It is clear then that our theory simply states that interaction implies statistical dependence. We can then borrow all of the properties associated with statistical dependence, and apply them to interaction. The corresponding dependency graph is shown in Figure 3-2.

In particular, when there is no interaction between X and Y, the arc from X_{t-1} in the dependency graph disappears, and the conditional distribution for Y becomes

$$p(y_t|x_{t-1}, \theta_Y) = p(y_t|\theta_Y), \tag{3.2}$$

indicating that X and Y and independent. When X and Y do interact, the strength of

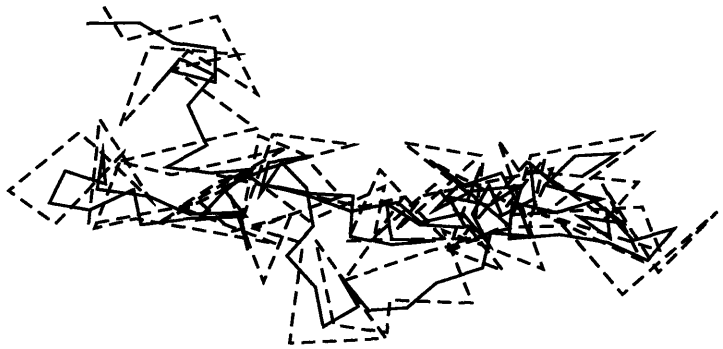


Figure 3-3: X (solid) moves randomly, and Y (dotted) follows X.

interaction is exactly the amount of statistical dependence. This is important in a case such as "Y follows X" because intuitively we are more confident that this type of interaction is occurring if X makes many turns, or in general moves in a complicated fashion. For example, if both X and Y move in a straight line, then it is possible that Y is following X, but we are not so confident. On the other hand, if X moves in a random fashion and Y remains near and behind X, then we are very confident that the interaction truly exists. Now recall that statistical dependence can be measured by the mutual information between X and Y:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = H(Y) - H(Y|X). \tag{3.3}$$

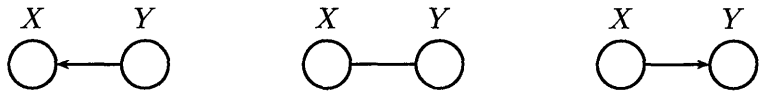
Thus when X and Y move in a straight line, each entropy term is low and the difference in entropies is also low. However, when X moves randomly, $H(X)$ and $H(Y)$ is high, while $H(X,Y)$ and $H(Y|X)$ are low because Y follows X, and so statistical dependence is high. This is an illustration of why a quantitative theory of object interaction is more useful than a purely qualitative one.

As an example, consider two scenarios: X moves at roughly the same speed along a straight line versus X moves at roughly the same speed but randomly (Figure 3-3). In both cases, Y would appear to be following X. In a straight line case, it is difficult to be certain because Y may happen to be taking the same path as X, such as two pedestrian on the same sidewalk. In the case where X moves randomly while Y follows a very similar path, it is easy to decide that "Y follows X" because it is highly improbable that the paths are so similar by chance, such as two pedestrian making exactly the same turns over a long period of time. The decomposition of the statistical dependence is shown in Table 3.1. The higher unconditional entropy $h(Y)$ in the case where X moves randomly is the primary reason for the correspondingly larger statistical dependence. Basically, in the straight line case, Y is well explained by a simple straight line model without knowledge of X. In other case, the random turns that Y makes appear unstructured until we realize that conditioned on X, the random turns are simply a result of trying to follow X's random turning.

In summary, our basic theory of object interaction is simply that interaction implies

	Random	Straight
$h(Y)$	3.5470	-0.4959
$h(Y\|X)$	0.3335	-1.5654
$I(X;Y)$	3.2135	1.0695

Table 3.1: The larger mutual information $I(X;Y)$ in the random case is primarily a result of higher unconditional entropy $h(Y)$.



$p(x)p(y|x)$ $p(x|y)p(y)$ $p(x,y)$

Figure 3-4: Dependency graphs corresponding to "Y follows X," "X follows Y," and symmetric influence, respectively from left to right.

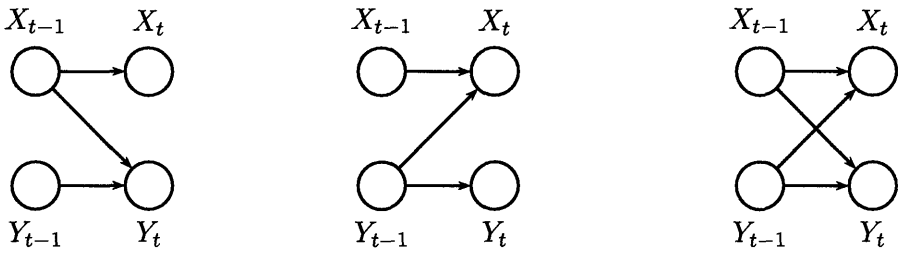
statistical dependence. Object motions are random processes and interact if and only if they are statistically dependent. The theory is general enough to account for any interaction, and yet allows one to precisely measure the strength of interaction by the amount of statistical dependence. This and other properties of statistical dependence are fortuitously in accord with intuition about object interaction. Below, we explore further details of our basic theory.

3.2 Causal Structure and Interaction Roles

In the previous section, we formulated object interaction in terms of statistical dependence. This enables us to decide whether objects interact by measuring the strength of statistical dependence. In this section we take a deeper look into object interaction by relating the roles of the objects in an interaction to the causal structure of their dependency. This will allows us to not only decide whether two objects are interacting, but how they interact.

Consider again our example of "Y follows X." The dependency graph for this particular interaction was shown in Figure 3-2. However, saying that X and Y are statistically dependent does not unambiguously determine the dependency graph. In fact, any of the dependency graphs shown in 3-4 is possible. What differentiates the graphs is their causal structure. For our example, it could be that "Y follows X," "X follows Y," or that they both influence each other (for example, in the case that they move symmetrically as a pair).

How can we determine the causal dependency structure from data? In general, learning causal structure is difficult [73]. In our case, each dependency graph has an arc and hence the same number of parameters. Thus each graph has the same complexity and it is unclear how to choose the directionality of the arc. Indeed, we can see this more simply by recalling



$p(x_t|x_{t-1})p(y_t|y_{t-1},x_{t-1})$ $p(x_t|x_{t-1},y_{t-1})p(y_t|y_{t-1})$ $p(x_t|x_{t-1},y_{t-1})p(y_t|y_{t-1},x_{t-1})$

Figure 3-5: Causal dependency graphs corresponding to "Y follows X," "X follows Y," and symmetric influence, respectively from left to right.

that we can always write the joint distribution as a product of conditionals:

$$p(x,y) = \begin{cases} p(x)p(y|x) \\ p(x|y)p(y) \end{cases} . \tag{3.4}$$

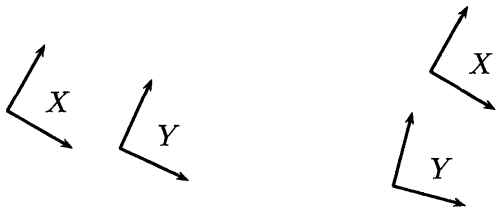
Fortunately we are working with trajectories, so we need only concern ourselves with *temporal causality.* This leads us to consider the causal dependency graphs shown in Figure 3-5. Now each dependency graph has a different set of arcs. Also, the directionality of arcs is always causal, from previous to next time steps.

As an example, if we analyze Figure 3-3 with the roles of X and Y reversed, then we obtain a much smaller mutual information of 0.0399. Intuitively, Y_{t-1} does not predict X_t well because, causally, Y reacts to X instead of the other way around. In summary, we can properly conclude that there is an interaction, and that "Y follows X" instead of *vice versa.*

3.3 Form of Interaction

Up to now, we have studied the causal dependency structure of interactions at a general level. This enables us to determine whether an interaction exists, and the roles of objects in the interaction. In certain applications, such as simply grouping the set of interacting objects, general causal dependency structure may be sufficient. Now consider the problem of deciding between "X and Y move together" versus "Y pursues, X evades." In both cases, the general dependency structure is the fully-connected, causal model in Figure 3-5, because both X and Y exert influences on each other. However, in the case of "X and Y move together," the objects try to stay near each other, while in the case of "Y pursues, X evades," exactly the opposite is true.

Our discussion of dependency structure has deliberately omitted any statement about the exact form of the probability distributions related to the graph. This is in accord with properly treating the details of the distributions as nuisance parameters. However, as we have seen in the example above, general dependency structure alone may be insufficient in



"X and Y move together" "Y pursues, X evades"

Figure 3-6: Moving local coordinate system representations of objects.

some scenarios. To answer the questions about what particular type of object interaction is occurring, we must step back from measuring statistical dependence to characterizing the form of dependence. In other words, we need to model the fact that X and Y not only interact, but interact in a particular way.

Based on our model, the information for the type of interaction is found in the parameters θ_Y for the conditional distribution $p(y_t|y_{t-1}, x_{t-1}, \theta_Y)$. For a particular causal dependency structure, θ_Y is an element of the space of all interaction types Θ. Different regions of Θ then correspond to different interaction types. Overall the conditional distribution $p(y_t|\cdot)$ consists of two factors: (1) its arguments determined by the causal dependency graph, (2) its form determined by the value of θ_Y. Thus, both "X and Y move together" and "Y pursues, X evades" have the same dependency graph, but different parameter values for the type of interaction.

For example, imagine that we represent each object as a *moving local coordinate system* as shown in Figure 3-6. For "X and Y move together," the conditional distributions would tend to keep the local coordinate systems aligned and next to each other. Conversely, for "Y pursues, X evades," X might turn to move away from the forward direction of Y, and Y would move towards X and match X's turns. In particular, we could model the relationship between the coordinate systems with a Euclidean *transformation*:

$$Y = RX + t, \tag{3.5}$$

where R is a rotation and t a vector. Interesting this use of transformations captures the notion of *prepositions* such as "in front," "behind," and "side." This kind of simple relative representation not only illustrates different forms for the conditional distributions, but could be the basis of a more *ad-hoc* system for categorizing object interactions [85]. The advantage of our theory is that it makes each factor in the representation of object interaction explicit and allows for quantitative measurements. To recap, it is important to consider the precise form of the conditional probability distribution $p(y_i|\cdot)$ in order to distinguish between different object interactions that have the identical dependency structures.

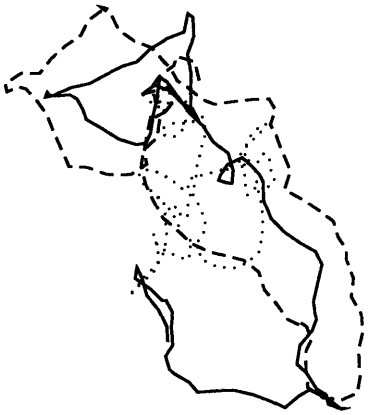


Figure 3-7: Trajectories for Z (dotted) stays in between X (solid) and Y (dashed) moving independently and randomly.

3.4 Beyond Two Objects

For simplicity and clarity, we have presented our general theory of object interaction in the common case of two objects. Now consider the interaction between three or more objects. An interesting example is when one object Y stays in between two others X and Z moving independently as shown in Figure 3-7. In this case, Y is dependent on X and Z corresponding to the causal dependency graph shown in Figure 3-8. Our theory is able to handle the general case of n objects with an n node dependency graph. For the example above, if we ignore causality, the dependency graph is a *star* with Y at the center. A star graph represents any interaction involving one object influenced by many others. The other objects are independent but become dependent through Y.

A second common dependency architecture is a *tree*. This could represent, for example, a convoy where vehicles are dependent on their immediate predecessors. The extreme case of a tree is a *chain* where each node follows its parent, such as in a line formation. In general the dependency graph captures the interaction between multiple objects and all of our operations and measurements that we discussed for the two node case applies.

This section concludes the general discussion of our theory of object interaction. In summary, by defining interaction in terms of statistical dependence, we are able to measure the strength of object interactions. Causal dependency graphs and the form of the associated conditional probability distributions allow us to decompose an interaction in its general dependency structure, and its detailed probabilistic form. In the next sections we present details of our approach and experimental results.

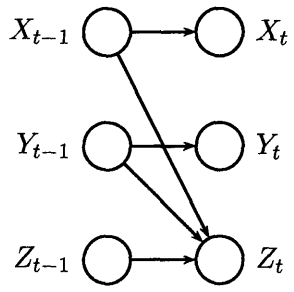


Figure 3-8: Causal dependency graph for "Y stays in between X and Z."

3.5 Modeling Details

To implement our general theory of object interaction, we need specific models for trajectories and dependencies. Recall that interaction is manifest through object motion. Therefore our observations are fundamentally a time series of object locations because motion is a trajectory in space. In particular for terrestrial[2] objects such as vehicles and pedestrians, the plane R^2 is often adequate.

3.5.1 Stochastic Processes and Entropy Rate

Recall that statistical dependence is measured by mutual information which is computed from entropies. Because trajectories are stochastic processes (that is, sequences of possibly non-independent, non-identically distributed RVs), we need a new measure of uncertainty for stochastic processes. Clearly, we cannot assume that the individual observations in a trajectory are independent because the next position is dependent on the current one. Fortunately, entropy can be generalized to the entropy rate [77, 12, 68]

$$H(\mathcal{Y}) = \lim_{T \to \infty} H(Y_1, ..., Y_T), \tag{3.6}$$

when the limit exists.

A common and reasonable assumption is to consider only *stationary* processes [68]:

$$\forall k \, p(y_1, ..., y_T) = p(y_{1+k}, ..., y_{T+k}), \tag{3.7}$$

so that the joint distribution of the sequence of random variables is invariant with respect to time shifts. This translates into an assumption that the motion dynamics do not vary with respect to time. For stationary processes, the entropy rate exists, and can be computed as

[2]Our theory works for aerial objects and higher dimensions as well.

[12, 68]

$$H(\mathcal{Y}) = \lim_{T \to \infty} H(Y_T | Y_{T-1}, ..., Y_1). \tag{3.8}$$

The next simplification is to assume *Markov* dynamics as alluded to in the first chapter. This is a common assumption in the tracking literature [76] and is motivated by the physics of motion. This allows us to represent a motion trajectory as

$$p(y_t | y_1, ..., y_{t-1}) = p(y_t | y_{t-1}), \tag{3.9}$$

so that only memory of the previous state is required. Essentially, we ignore long-term dependencies. This simplifies the entropy rate to

$$H(\mathcal{Y}) = H(Y_t | Y_{t-1}). \tag{3.10}$$

Intuitively, entropy rate measures how uncertain or predictable a process is. For stationary, Markov processes, this is simply the uncertainty of the next state given the current. Thus to measure statistical dependence of stationary, Markov trajectories, we extend mutual information to

$$I(\mathcal{X}; \mathcal{Y}) = H(\mathcal{Y}) - H(\mathcal{Y}|\mathcal{X}) \tag{3.11}$$

$$= H(Y_t | Y_{t-1}) - H(Y_t | Y_{t-1}, X_{t-1}). \tag{3.12}$$

Basically, this is the conditional mutual information of Y_t and X_{t-1} given Y_{t-1}.

3.5.2 Auto-Regressive Process

One way to compute entropy rate is to directly estimate the conditional distribution $p(y_t|y_{t-1})$. However, it is important to realize that trajectories, although embedded in the two-dimensional plane, are inherently one-dimensional curves. Even if we assume *ergodicity* [68], we would need a very long trajectory to accurately estimate conditional probabilities. A typical observation of a motion trajectory might visit a particular location only once. Thus to ensure that the direct estimates are useful, either the area of motion has to be very small or the trajectories have to be very long.

Our earlier discussion on relative local coordinate systems reminds us that the absolute coordinate system is not important. In other words, the starting position or orientation should not affect the statistical dependence. It is really the *accelerations* of speeding up and turning that determine the predictability of a motion trajectory. The absolute positions are an artifact of the observation process. In general, we should analyze relative

A simple but effective model for Markov trajectories is an auto-regressive (AR) process [68]. An AR(1) model is a linear first-order Markov process:

$$y_t = A y_{t-1} + w, \tag{3.13}$$

where A is the Markov dynamics matrix and w is zero-mean Gaussian noise with covariance

Behavior	Algorithm
Follow	desired velocity = position behind predicted position of target − current position
Together	desired velocity = (position beside predicted position of target - current position + target velocity)/2
Pursuit	desired velocity = predicted position of target - current position
Evade	desired velocity = - (predicted position of target - current position)
Mirror	desired velocity = - (estimated velocity of target)

Table 3.2: Summary of behavioral algorithms for simulating interactions.

matrix Σ. The parameters $\theta_Y = (A, \Sigma)$ can be estimated using least-squares methods. The entropy rate can then be estimated as the entropy of the Gaussian noise:

$$H(\mathcal{Y}) = \frac{1}{2}\log(2\pi e)^d|\Sigma|, \tag{3.14}$$

for $y \in R^d$. To capture statistical dependence between processes, we AR models with inputs from the other variable:

$$y_t = Ay_{t-1} + Bx_{t-1} + w. \tag{3.15}$$

The AR parameters A and B captures the form of statistical dependency and is useful for categorizing the type of object interaction as discussed in Section 3.3. For object motion, a second-order AR(2) model is used to capture velocity information.

3.6 Experiments

We assume that the data are motion trajectories. We focus on two-dimensional data because most objects are terrestrial, although our approach generalizes to higher dimension in a straightforward manner. In general, we will be measuring statistical dependence as mutual information. Large mutual information between variables in the dependency graph suggests that the corresponding arc exists.

3.6.1 Simulations

We simulated a variety of interactions including objects moving independently. Objects were modeled as particles with a maximum speed on one. Random motion was generated by adding a random acceleration drawn from a zero-mean spherical Gaussian distribution with variance 0.25. Velocities were updated by averaging the current and desired velocities. Interactive behaviors were implemented with the simple algorithms summarized in Table 3.2. To make some behaviors more interesting, turns were added to the trajectories by injecting random motion. Examples of the interactions are shown in Figure 3-9.

In Table 3.3, we report the estimated mutual information for the simulated interactions.

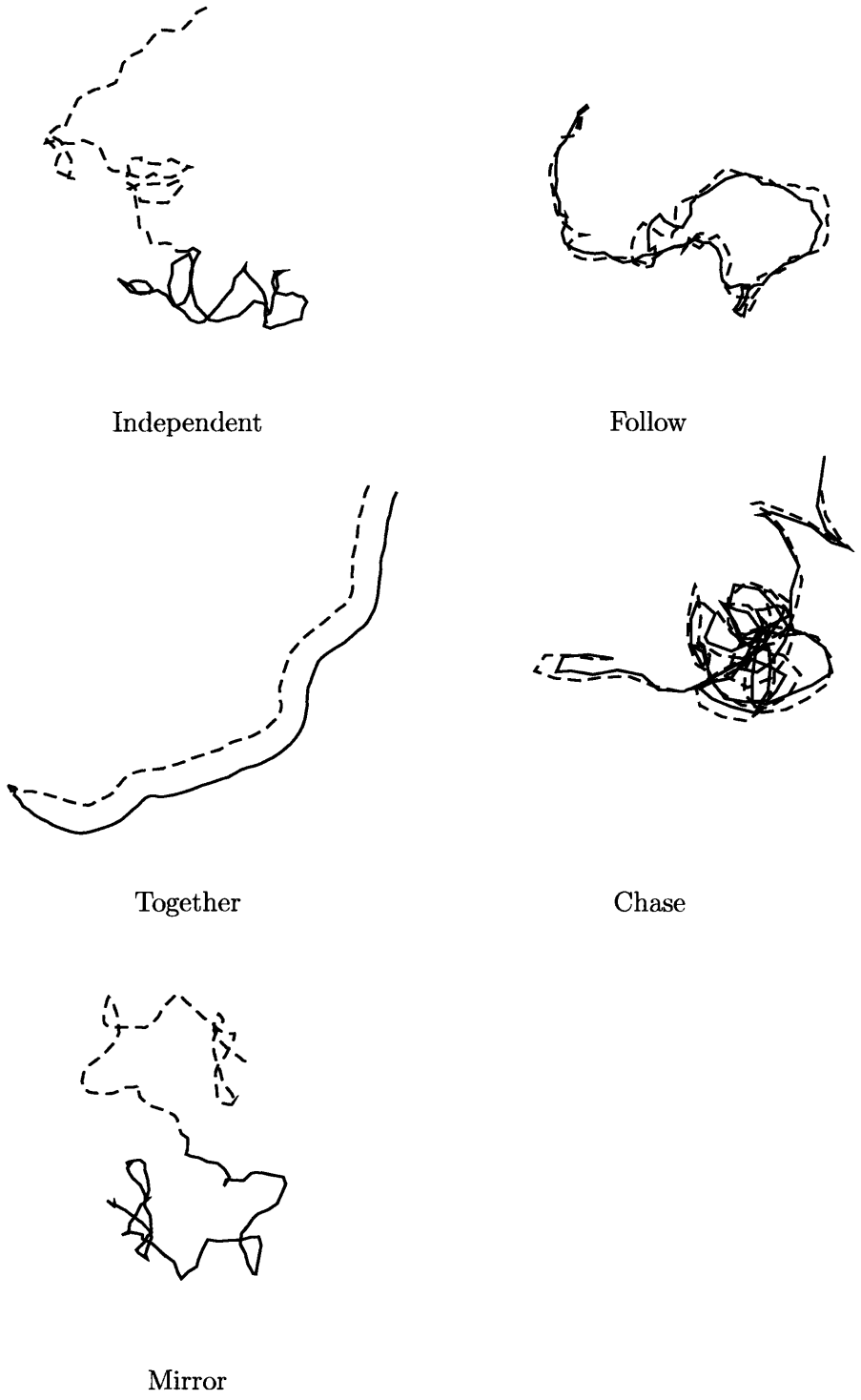


Figure 3-9: Trajectories of simulated interactions between X (solid line) and Y (dashed line).

Trial	Independent	Follow	Together	Chase	Mirror
1	0.01/0.04	0.03/1.03	0.52/0.51	0.23/2.49	0.04/62.22
2	0.03/0.05	0.02/1.05	0.77/0.85	0.20/2.62	0.01/59.45
3	0.02/0.03	0.01/1.05	0.32/0.25	0.27/2.78	0.02/59.48
4	0.06/0.08	0.01/1.02	0.61/0.59	0.21/2.64	0.02/54.57
5	0.03/0.01	0.02/1.01	0.40/0.37	0.23/2.56	0.03/54.31
6	0.03/0.01	0.03/0.95	0.58/0.48	0.21/2.43	0.03/53.64
7	0.03/0.08	0.02/1.00	0.69/0.70	0.24/2.58	0.01/57.73
8	0.01/0.04	0.04/1.03	0.49/0.36	0.22/2.47	0.01/59.24
9	0.02/0.05	0.01/1.05	0.56/0.60	0.16/2.41	0.04/55.32
10	0.04/0.05	0.01/1.14	0.50/0.56	0.23/2.61	0.01/55.42
$\bar{I}$	0.03/0.04	0.02/1.03	0.54/0.53	0.22/2.56	0.02/57.14
σ_I	0.01/0.02	0.01/0.05	0.12/0.17	0.03/0.11	0.01/2.88

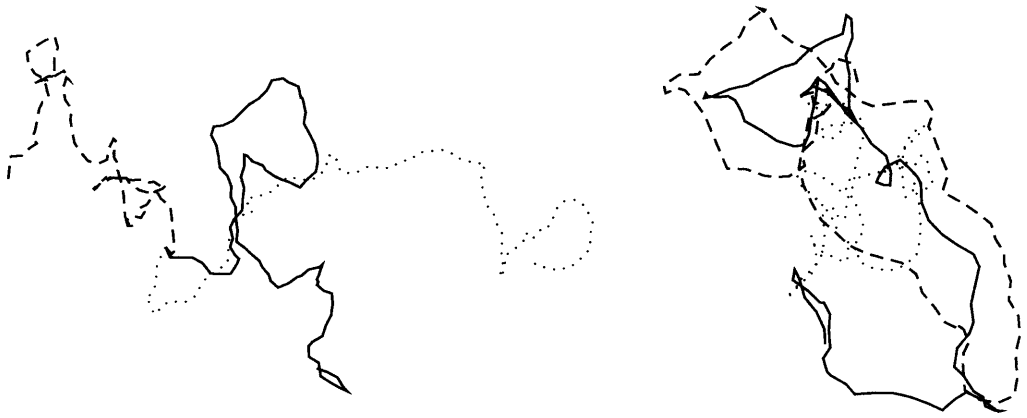
Table 3.3: Estimated $I(X_{t-1}; Y_t)/I(X_t; Y_{t-1})$ for ten trials of simulated interactions along with average $\hat{I}$ and standard deviation σ_I.

Ten trials were run with sample trajectories of length 200. As expected, objects moving independently had low values of MI. For "Y follows X", $I(X_{t-1}; Y_t)$ is large while $I(X_t; Y_{t-1})$ is neglible since X moves randomly and independent of Y. When objects move together, both MI terms are significant and roughly equal because X and Y influence each other symmetrically. In the chase interaction, $I(X_{t-1}; Y_t)$ is larger than $I(X_t; Y_{t-1})$ because Y always pursues X, while X makes random turns while evading so its future position is less predictable even when given information about Y. This is not too surprisingly because, intuitively, the pursuer and evader have asymmetric roles. The large MI values associated with the "mirror" interaction are due to the fact that Y exactly mirrors the motion of X. From the sample trajectory, we can see that Y's motion can be very precisely predicted from X's. The MI is asymmetric because, Y mirrors X is causally similar to "Y follows X".

The estimated MI for each arc in the dependency graph allows us to discriminate between independent, asymmetric and symmetric interactions. As a simple illustration, we built a nearest neighbor classifier [17, 5] with a single example from each of the interactions listed in Table 3.3. The feature vector was simply the estimated MI values $I(X_{t-1}; Y_t)$ and $I(X_t; Y_{t-1})$. We then computed the confusion matrix for classifying the other examples. The cross-validated results in Table 3.4 show how dependency structure features are useful for classifying the interactions. In general, fine-grained discrimination of interactions within a dependency structure equivalence class requires inspection of the form of the conditional probability distributions. With AR models, this is detemined by the parameters matrices A and B of Equation 3.15.

	Independent	Follow	Together	Chase	Mirror
Independent	9/1	0	0	0	0
Follow	0	9/1	0	0	0
Together	0.70/0.08	0	8.3/0.92	0	0
Chase	0	0	0	9/1	0
Mirror	0	0	0	0	9/1

Table 3.4: The cross-validated confusion matrix ($n/\%$) for classifying interactions based on dependency structure as represented by estimated MI values.



Independent Between

Figure 3-10: Trajectories of simulated independently moving objects and Z (dotted) between X (solid) and Y (dashed).

Trial	Independent	Between
1	0.11/0.07/0.05	1.29/0.32/0.21
2	0.05/0.02/0.01	1.40/0.29/0.30
3	0.04/0.01/0.03	1.56/0.40/0.31
4	0.06/0.02/0.02	1.40/0.33/0.25
5	0.13/0.04/0.08	1.48/0.40/0.23
6	0.13/0.06/0.03	1.26/0.36/0.20
7	0.08/0.02/0.05	1.41/0.27/0.30
8	0.12/0.05/0.04	1.41/0.34/0.22
9	0.08/0.04/0.01	1.37/0.21/0.29
10	0.12/0.07/0.04	1.38/0.36/0.19
$\bar{I}$	0.09/0.04/0.04	1.40/0.33/0.25
σ_I	0.03/0.02/0.02	0.08/0.06/0.04

Table 3.5: Estimated $I(Z_t; X_{t-1}, Y_{t-1}|Z_{t-1})/I(Z_t; X_{t-1}|Z_{t-1})/I(Z_t; Y_{t-1}|Z_{t-1})$ for ten trials of simulated interactions along with average $\hat{I}$ and standard deviation σ_I.

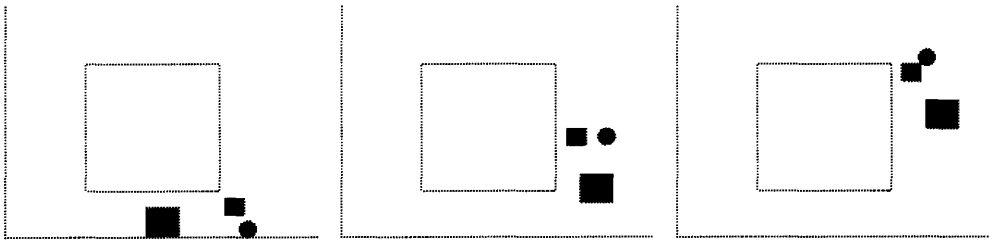


Figure 3-11: Three frames from a chase sequence similar to that of Heider and Simmel's cartoon video.

3.6.2 Heider and Simmel

In the introduction to this chapter, we referred to the study by Heider and Simmel [31] that demonstrated how humans interpreted motions as interactions. Figure 3-11 shows three frames from a chase sequence similar to that of Heider and Simmel's cartoon video. The large square Z chases both the small square X and the circle Y. The largest statistical dependency we found was for Z dependent on X and Y with a mutual information of 0.58. The next most significant values were 0.41 and 0.43, for $I(X_t; Z_{t-1})$ and $I(Y_t; Z_{t-1})$ respectively. This corresponds naturally to the fact that Z pursues X and Y, who evade Z. We also find that $I(X_t; Y_{t-1})$ and $I(Y_t; X_{t-1})$ have the next largest values of 0.27 and 0.20 respectively, because X and Y evade together.

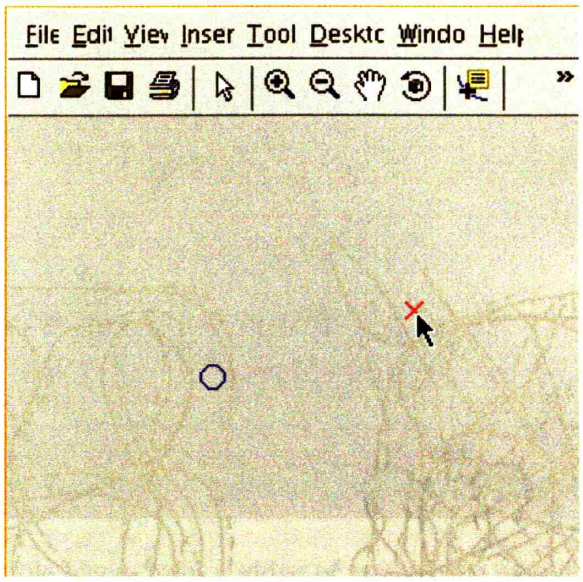


Figure 3-12: The "Interaction Game" window. Players use the mouse pointer to move the objects.

Independent	Follow	Together	Chase
0.02/0.03	0.06/0.40	0.16/0.28	0.52/0.06
0.02/0.01	0.01/0.38	0.35/0.27	0.31/0.02
0.03/0.03	0.07/0.79	0.09/0.32	1.01/0.04

Table 3.6: Estimated $I(X_{t-1}; Y_t)/I(X_t; Y_{t-1})$ for the "Interaction Game" data.

3.6.3 Interaction Game

To obtain real data while sidestepping low-level tracking issues, we created an "Interaction Game" user data collection tool. Players used a computer mouse to move points in a display window shown in Figure 3-12. They were told to either move independently, or to engage in some type of interaction. Samples of the trajectories are shown in Figure 3-13. Statistical dependence estimation results are shown in Table 3.6 for trajectories of length 500. In the chase interaction, one player chases the other, however the evader often did not move until the pursuer was close. This resulted in lower MI for the "evader depending on pursuer" arc. In contrast, in the simulated chase, the objects were in a constant pursuit and evade.

3.6.4 Video Data

We collected video of two people moving in a small area. The camera was located about 6 meters above the ground plane. An example of a frame from the video is shown in Figure 3-14. For a video sequence, trajectories can be obtained with a blob-based tracker [80].

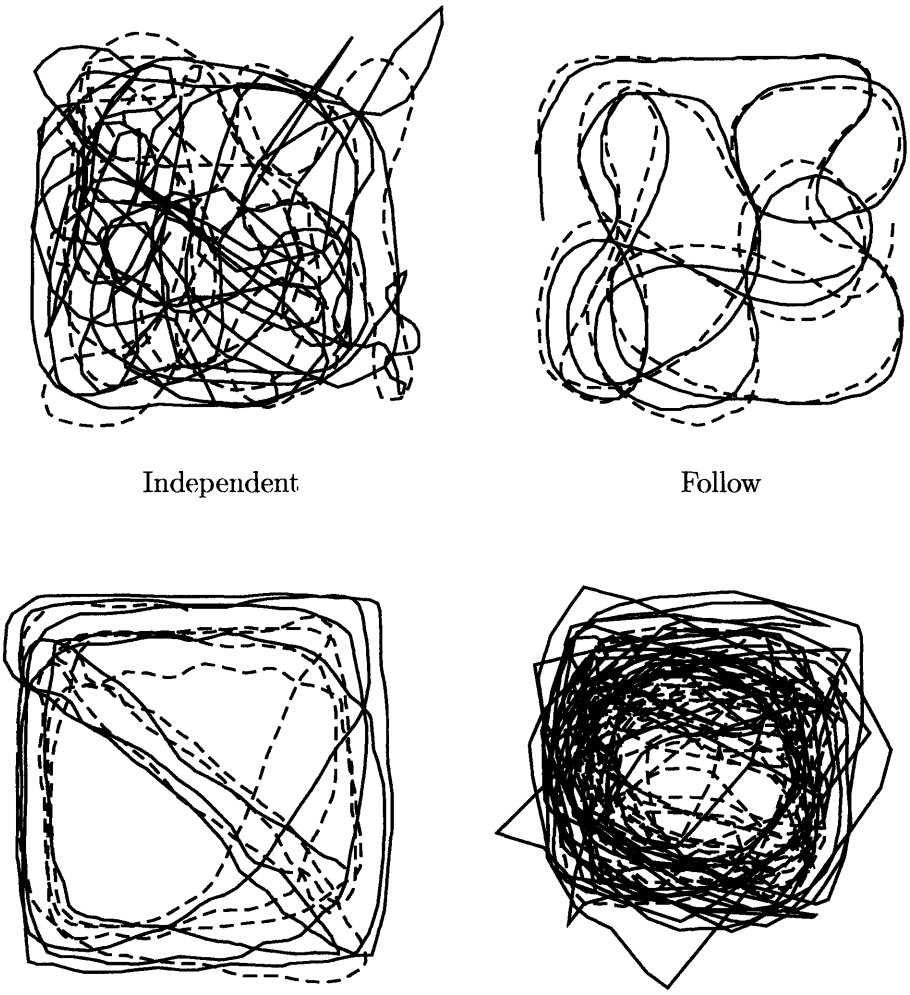


Figure 3-13: Trajectories from players in the "Interaction Game," X (solid) and Y (dashed).

Figure 3-14: Sample frame from a video of two people moving in a small area.

Independent	Follow	Together	Chase
0.03/0.02	0.14/0.24	0.19/0.09	0.42/0.12

Table 3.7: Estimated $I(X_{t-1}; Y_t)/I(X_t; Y_{t-1})$ for video data.

When objects are close together such as in a "together" or "chase" interaction, automated trackers have difficulty discriminating the motions, so we hand-corrected those tracks.

The results here are consistent with those from simulation and the "Interaction Game." Two people moving independently produced very low MI, while all other interactions increased statistical dependence between the motions. The asymmetric MI the people moving together was due to the fact that one person primarily dictated the turns other even though they were instructed to "move together." One way to disambigute this interaction from "Y follows X" is to simply check whether the other person is behind or to the side of the other. Note that our positive results are despite the fact that for our video data, the tracks undergo projective distortion, unlike the simulated and "Interaction Game" data.

We also collected a second video of two people interacting in a different environment (see Figure 3-17). The results (see Table 3.8) are similar to those of the first video.

Independent	Follow	Together	Chase
0.17/0.17	0.54/1.62	0.46/0.38	0.21/0.20

Table 3.8: Estimated $I(X_{t-1}; Y_t)/I(X_t; Y_{t-1})$ for video data.

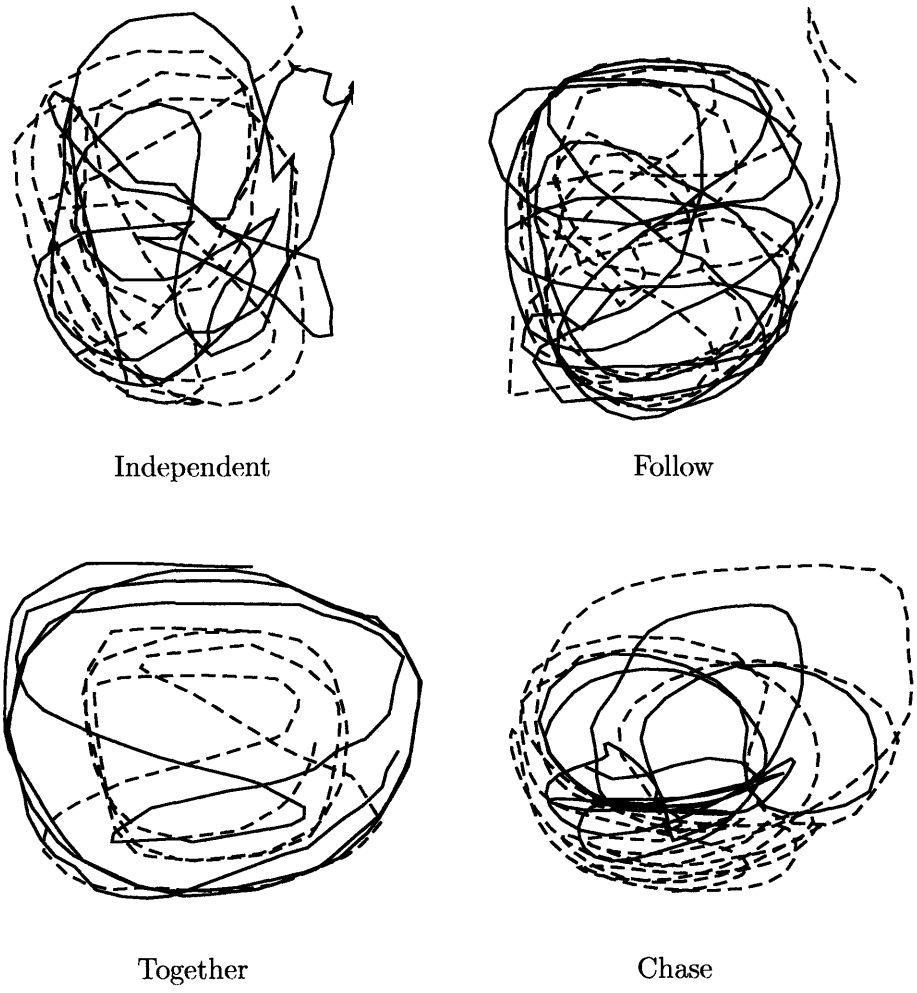


Figure 3-15: Trajectories from video data, X (solid) and Y (dashed).

Figure 3-16: Sample frame from a video of two people moving in a small area.

3.7 Related Work

Recently, analyzing the behavior of objects has become a hot topic because of increased interest in automated surveillance. However, work on automated analysis of behavior dates back to the seminal work of Nagel [65]. The literature on activity perception is both large and diverse [27, 28, 2, 6, 26, 69, 15, 57, 19, 8, 9, 56, 7, 91]. Much of the work has focused on finding good representations for activity and training classifiers for detecting particular behaviors. Many approaches try to leverage ideas designed for analying images in computer vision and pattern recognition for activity analysis. Most of the approaches have, as of yet, focused on single object activities.

To our knowledge, our formulation of object interaction explicitly in terms of statistical dependence and model selection is novel. Of course, our work has been inspired by a growing body of research. Below, we review prior important contributions to understanding object interaction. In general, previous work has primarily focused on specific applications of interaction analysis, such as training a detector of anomalous activity. In contrast, we have tried to develop a general theory of object interaction from first principles.

A particularly simple approach for detecting *anomalous* interactions was presented by Morris and Hogg [63]. They consider a pedestrian interacting with parked vehicles. Features such as walking speed and distance to a vehicle are computed, and the associated probability distribution fit to non-anomalous data. Anomalies are then detected as simply any observation with low probability under the learned distribution. Johnson et al. [41] also developed a system for human hand-shaking based on learning the joint distribution of human silhouette features. They were interested in acquiring an interaction model for *synthesizing* a virtual hand-shaker. The primary limitations of these approaches are their narrow definitions of interaction, which are specialized to scenarios such as parking lots

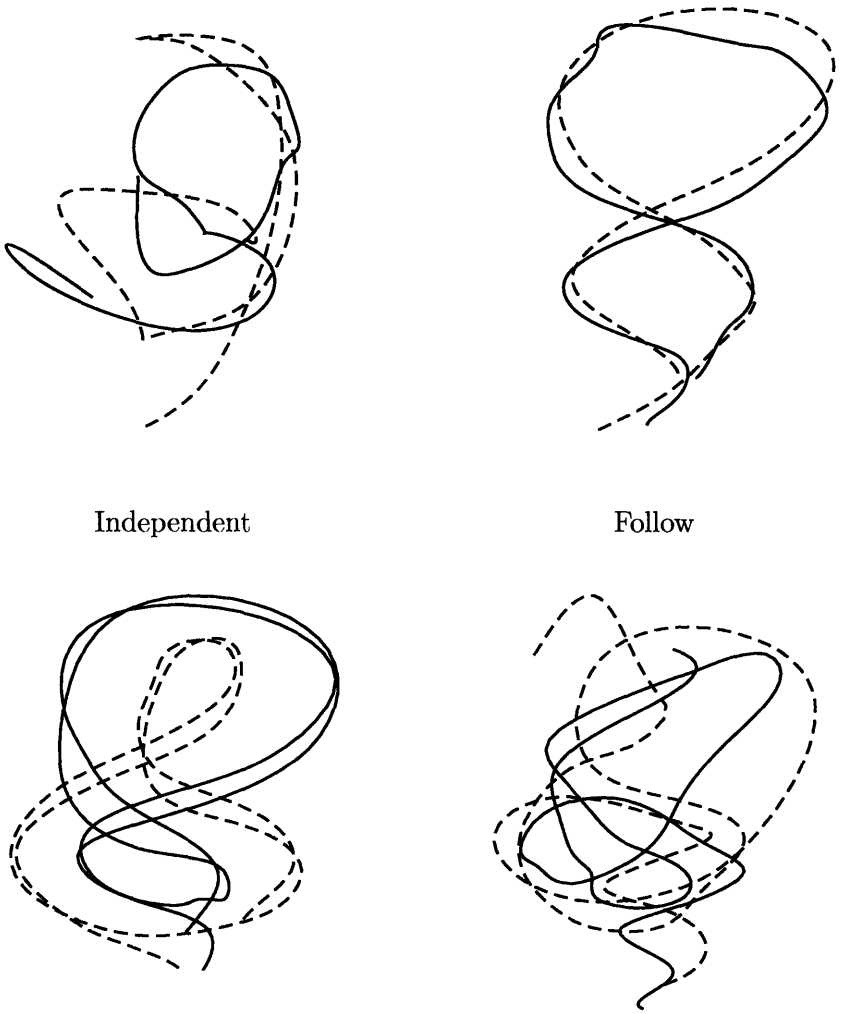


Figure 3-17: Trajectories from video data, X (solid) and Y (dashed).

and hand-shaking. Nevertheless, these approaches are notable for their use of probabilistic representations of behavior.

A different section of work is well-represented by the *grammar*-based approaches of Intille and Bobick [35] and Ivanov and Bobick [36]. The goal here was to examine language-inspired frameworks for representing complicated sequences of interactions. Intille and Bobick [35] analyze football plays with temporal constraint graphs on objects and goals. Ivanov and Bobick [36] use stochastic context free grammars to parse the separate behavior of multiple objects into coordinated multi-object behaviors. Grammar-based approaches operate on *symbolic* representations of behavior and focus on parsing individual behaviors into known interactive behaviors. These approaches are complementary to our method of discovering the structure of interactions based on statistical dependence.

Finally, the approach of Oliver and Pentland [66] is closest in spirit to our ideas. Their goal was to train models for a known set of interactive behaviors. To this end, they used *coupled hidden Markov models* on features such as distance and relative angle. The coupled model structure is fixed and assumes full dependency for two objects. Thus the important difference with our approach is that we do not assume a fixed dependency structure, and instead, infer it from the observed data.

In summary, our primary contribution over previous approaches is two-fold: (1) an explicit examination of the link between statistical dependence and object interaction, (2) learning the structure of dependencies from observed data rather than assuming a fixed model. Overall, our approach combined with the previous work reviewed here will be useful to engineers interested in developing a recognition system for a particular application and scientists studying the statistical nature of object interaction.

3.8 Summary

In this chapter we presented our theory of object interactions based on statistical dependence. The theory essentially equates interaction with dependence and shows how the various properties of statistical dependence have intuitive meanings in terms of object interactions. In particular, the theory enables us to do three things: (1) decide if there is an interaction, (2) explain how the objects are interacting, (3) describe what kind of interaction is occurring.

We then described computational details of our model and the results of experiments validating our approach. We also compared our approach to prior related work. Our primary contribution is drawing the connection between object interaction, statistical dependence, and model structure which enables the discovery of interaction models from observed data.

Chapter 4

Matching

In the previous chapter, we presented our theory of object interaction based on statistical dependence. In this chapter, we present the second part of our thesis that statistical dependence estimation also underlies the matching problem. We will begin by reviewing the problem of matching in general and explain the connection to statistical dependence. We then prove an intractability result for exact maximally dependent matching and suggest a Markov chain Monte Carlo (MCMC) approximation.

The second half of this chapter applies our ideas to the problem of matching objects between non-overlapping cameras. In particular, we show how our theory generalizes previous work and describe experiments demonstrating improved performance with our approach.

We defined statistical dependence in the second chapter, and explained how it could be estimated from observed data. We have tacitly assumed that observations are in the form of matching pairs (x, y). However in many cases, the low-level problem is often to find these matching pairs in the first place. For example, photogrammetry [32] requires matching points in two images, and motion analysis [88] looks for corresponding features across video frames. In general, the matching problem arises whenever corresponding observations are acquired by different sensors or are separated in space or time.

To gain intuition into the matching problem, consider a toy example with two cameras looking at different parts of a road. Four vehicles traveling at different speeds depart camera one and later arrive at camera two. The departure, travel, and arrival times are shown in Table 4.1. From the perspective of camera two, arrivals happen at times 3, 5, 6, 8, so the vehicles actually arrive in the order (1,3,2,4), which is the correct matching of departures and arrivals between the cameras. In general, we can represent the matching as a permutation of indices. Here, the matching problem is to decide the best correspondence between the departing and arriving vehicles. As with any optimization problem, we need to examine the cost function and the feasible set. The space of possible matchings is the set of all permutations[1] of (1,2,3,4). Thus we have a combinatorial (discrete) optimization problem [67], where the size of the feasible set is exponential. Our first concern, however, is, "what should be the cost function?"

[1]Some permutations are infeasible because arrival times cannot occur before departure times.

Departure	Travel	Arrival
1	2	3
2	4	6
3	2	5
4	4	8

Table 4.1: Departure, travel, and arrival times for a toy example of four vehicles moving between two cameras.

Clearly, we cannot match the vehicles in-order, nor minimize the average travel time because both would lead to the incorrect matching (1,2,3,4). If we knew the exact vehicle travel times or the distribution of travel times, the problem would be simpler. However, we could have many different vehicles traveling at various speeds, not to mention other objects such as bicycles and pedestrians. In general, it is unrealistic to assume we know the distribution of travel times *a priori*. At this point, the reader may ask, "why not use other features such as appearance?" We agree that appearance can and should be used. Often, the cost function for appearance assumes that it should not change. But appearance may change between cameras because of lighting, pose and other geometric and photic parameters. Furthermore, the type of appearance change may vary depending on the object. For example, although color change may be approximated well by a linear model, different colors often require different linear transformations [87].

Given the uncertainty described above, how can we formulate a well-defined cost function for matching? We can get some inspiration from the problem of clustering [17] a set of points. Clustering is often considered an *unsupervised* learning problem because the proper cost function is debatable. One way to make it well-defined is to cast it as a problem of grouping the data so that the grouping has maximum likelihood under a certain mixture distribution. Each component distribution of the mixture is then identified as a cluster. In the same way, we can define the best matching as that which has maximum likelihood.

To make the connection between matching and statistical dependence, we first reinterpret entropy as negative mean log probability:

$$H(X) = -\sum_{x} p(x) \log p(x). \tag{4.1}$$

In the finite sample case, we have an approximation:

$$\sum_{i=1}^{n} \log p(x_i) \approx -nH(X). \tag{4.2}$$

Thus maximizing likelihood is the same as minimizing entropy [90]. For matching this is the joint entropy $H(X, Y)$. Recall that mutual information can be written as a difference

between the marginal and joint entropies:

$$I(X;Y) = H(X) + H(Y) - H(X,Y). \tag{4.3}$$

Thus, minimizing $H(X,Y)$, maximizes $I(X;Y)$ which is our measure of statistical dependence. Matching is akin to pairwise clustering to maximize the likelihood of the matches.

In our toy example, consider the matching $M_1 = (1,3,2,4)$ versus $M_2 = (1,2,3,4)$. We want to examine the probability distribution of the pairs induced by each matching. The probability of a pair can be written as $p(x,y) = p(x)p(y|x)$, so that matching only affects the conditional distribution. Here, the conditional distribution is intimately tied to the distribution of travel times as we will show later. The difference between the matchings is then reflected in the entropy of the travel time distribution. The pairs for M_1 are (1,1), (2,3), (3,2), (4,4), leading to travel times of 2, 4, 2, 4, while the travel times for M_2 are 2, 3, 3, 4. Assuming possible transition times of 1, 2, 3, 4, the corresponding travel time entropies are 0.69 for M_1 versus 1.04 for M_2. Thus M_1 induces more statistical dependence than M_0 because its travel time distribution has lower entropy. In particular, M_1 hypothesizes a bi-modal travel time distribution compared to the more uniform one associated with M_0. Our assumption is that we favor a lower entropy distribution because it explains more of the regularity in the data. From a modeling perspective, maximizing statistical dependence is the same as maximizing the regularity in the data. Regularity is a generic guiding principle in modeling when few assumptions can be made of the data.

We can also look at the matching problem from the perspective of estimating dependency structure as we did with object interactions. A matching induces pairs (x,y), from which we can estimate the amount of statistical dependence. The matching problem then is to find pairs such that the overall dependence between X and Y is maximized. Thus, the dependency structure is assumed to be simply that X and Y are dependent. The problem is that we do not have direct access to samples (x,y) from this dependency. A matching generates samples, and we assume that the best matching corresponds to samples from a maximally dependent distribution. This distribution gives the largest mean probability of the data, which corresponds to maximizing the regularity in the data. A variant of this problem occurs in the tracking literature under the name of data association [76, 13]. There, the problem is to match targets with measurements, and the joint distribution of targets and measurements is assumed known. The main focus is on tackling the combinatorial challenge of efficiently finding the best matching among an exponential number of them.

The subsequent sections will discuss matching and statistical dependence in more detail. Finally, we apply our approach to matching between non-overlapping cameras and describe experiments to evaluate its performance. First, we will start with a review of the general matching problem.

4.1 Problem Formulation

For simplicity we describe the bi-partite matching scenario; conceptually, the extension to more than two sets is straightforward, although the associated properties of the problem and the algorithmic challenges can change dramatically [67]. We are given two sets of objects $\mathcal{X} = \{x_1, ...x_n\}$ and $\mathcal{Y} = \{y_1, ..., y_n\}$. A matching is a one-to-one correspondence between the elements of $\mathcal{X}$ and $\mathcal{Y}$. The one-to-one constraints mean that each x_i must match only one other y_j and *vice versa*. As stated earlier, the feasible set is the set of permutations on the indices $1, ..., n$. The matching problem then is to

$$\min_{\pi} c(\pi; \mathcal{X}, \mathcal{Y}), \tag{4.4}$$

where π is a permutation and c is some cost function. The pairs induced by a matching are $(x_i, y_{\pi(i)})$. We will show that, in the general case, c is the statistical dependence between X and Y.

4.1.1 Transformations

One way to motivate the use of statistical dependence as the optimization criterion for matching is to consider transformations between the matched pairs. A common way to model the relationship between two variables is with a transformation:

$$Y = T(X), \tag{4.5}$$

where T is the transformation. The idea is not new; for example, coordinate transformations were used in a theory for comparing natural shapes [84]. Commonly, T is assume fixed for all X, such as a single rigid body transformation for all points on an object. In our case, we generalize the model so that T is drawn from a probability distribution of transformations such as travel times between two cameras. In this case, the relationship becomes a simple translation:

$$Y = X + T, \tag{4.6}$$

where X and T are assumed independent.

Recall that mutual information can be written as a difference of entropies:

$$I(X;Y) = h(Y) - h(Y|X). \tag{4.7}$$

A matching only affects the second term, which we can simplify:

$$\begin{aligned} h(Y|X) &= h(X + T|X) && (4.8)\\ &= h(T|X) && (4.9)\\ &= h(T) && (4.10) \end{aligned}$$

because T is independent of X. This makes intuitive sense because once we know X, the

uncertainty in Y should be exactly the randomness of the transformation. We see now that maximizing statistical dependence is the same as minimizing the entropy of the distribution of transformations:

$$\max_{\pi} I(X;Y|\mathcal{X},\mathcal{Y},\pi) = \max_{\pi} h(Y) - h(Y|X;\mathcal{X},\mathcal{Y},\pi) \tag{4.11}$$

$$= \min_{\pi} h(T;\mathcal{X},\mathcal{Y},\pi). \tag{4.12}$$

Entropy then becomes the cost function c that we minimize over matchings. Below we show how other cost functions can be shown to be special cases of this generic criterion.

4.1.2 Fixed, Known Cost

A very special case is when the distribution of transformations $p(T)$ is known. An example is the identity transformation with noise so that Y is Gaussian distributed around X. Basically this says that the Y is most likely to be the same as X. In this case, $h(Y|X) = h(W)$, and the cost function becomes a least-squares log likelihood. We can encode all possible match costs in a matrix with elements

$$c_{ij} = L(x_i, y_j), \tag{4.13}$$

where L is a least-squares loss function. The cost function then becomes

$$c(\pi) = \sum_i c_{i\pi(i)} = \sum_{i,j} c_{ij} z_{ij}, \tag{4.14}$$

with one-to-one match constraints:

$$\sum_j z_{ij} = 1 \text{ and } \sum_i z_{ij} = 1, z_{ij} \in \{0,1\}. \tag{4.15}$$

This is the assignment problem [67, 64, 48], an integer programming problem because of the constraints on z_{ij}, but which can be solved as a standard linear programming problem because of unimodularity. Indeed the nonlinear constraints are what make the problem seemingly difficult. However, the feasible set can be be made convex with a theorem by Birkhoff [83] which shows that any doubly stochastic matrix can be written as a convex combination of permutation matrices. Because the cost function and doubly stochastic constraints are linear, the optimal solutions are at the corners of the convex polytope, namely the permutation matrices. The Hungarian algorithm solves the assignment problem in $O(n^3)$ [67, 64, 48].

4.1.3 Parametric Model

The Gaussian model of the previous section can be extended to allow a parameterized set of transformations:

$$Y = f(X;\theta) + W, \tag{4.16}$$

so that Y is some function of X parameterized by θ plus Gaussian noise. In image alignment we may allow an affine warp between the coordinate systems of the X and Y pixels [90, 61]. The transformation implicitly encodes the matching constraints. In translation, each X is guaranteed to match a single Y and vice versa. For affine warps, the transformation from X to Y may be many-to-one. The optimization problem is then to minimize

$$c(\theta) = c((f(x_1;\theta), y_1), ..., (f(x_n;\theta), y_n)). \tag{4.17}$$

Note that here, all X's share a single transformation. The cost function may take into account both the matched X's and Y's and the transformation θ. This is often solved using local search such as steepest descent. Note that by representing the matching with a transformation, the one-to-one matching constraints are implicitly satisfied so that the optimization is effectively unconstrained in the feasible space of transformations. As an example, computing correlation between X and Y with a time lag is a simple time shift transformation.

What if some of the X's use a transformation specified by θ_1, while other X's use a different transformation specified by θ_2? For this type of finite mixture distribution of transformations, one may then use the expectation-maximization algorithm with the single parameterized transformation cost problem as an inner loop. Much of the registration and alignment work in computer vision is based on this and related ideas [25, 90, 60].

4.1.4 Non-parametric Cost

In many problems, we do not expect the distribution of transformations $p(T)$ to be known nor Gaussian, such as in the case of pedestrian and vehicle transition times between cameras. We thus retain the simple additive model

$$Y = X + T, \tag{4.18}$$

but allow $p(T)$ to be non-Gaussian. This type of model makes fewer restrictive assumptions. The price we pay for this generality and flexibility is that matching becomes a hard optimization problem. Basically, the optimization will be difficult because of the nonlinearity of both the constraints and the cost function.

Recall that our cost function for a matching is the entropy of the transformation distribution. As shown at the beginning of this chapter, entropy is simply the negative average log likelihood. So in a sense it measures how well the distribution models its own samples. Given a sample x, an estimate of its entropy is then an estimate of its log likelihood in a general sense:

$$\int p(x) \log p(x) \approx \sum_i \log p(x_i). \tag{4.19}$$

Therefor the maximally dependent matching is also the most likely matching in this general sense.

4.2 Maximally Dependent Matching

We are given two sets of measurements $\{x_1, x_2, ..., x_n\}$ and $\{y_1, y_2, ..., y_n\}$ with unknown correspondence. If we knew the distribution of matching pairs $p(x, y)$, we could find the maximum likelihood matching efficiently using an algorithm for assignment. Without this knowledge, we set as our goal the matching with maximum dependence. Based on our previous discussion, the matching with maximum statistical dependence will also have the lowest entropy, and hence maximum likelihood. The description length of data is proportional its entropy. This allows us to identify maximal dependence with minimum description length (MDL) [75]. Note that this does not presuppose a true model of the data, but only attempts to find a good or compact model.

Entropy, in turn, can be viewed as negative average log probability. Because we do not know the distribution of the data, we must estimate it from the data as well. One way to estimate entropy given only a sample is to compute the leave-one-out probability with a non-parametric kernel density estimate:

$$-\int p(z)\log p(z) \approx -\sum_i \log p(z_i) \tag{4.20}$$

$$= -\sum_i \log \sum_{j \neq i} \Phi(z_i, z_j), \tag{4.21}$$

where Φ is a kernel. From the matching perspective, we are favoring matchings such that the resulting matched pairs have maximum probability without assuming a known distribution. Let t_{ij} be the transformation relating the matched pair $y_j = x_i + t_{ij}$. We then seek a matching that maximizes the probability of the resulting set of transformations. In summary, our optimization criterion is based on the following principle:

Proposition 1. *Maximally Dependent Matching (MDM): The best matching maximizes statistical dependence.*

A short description length for t_{ij} means that given X, we can describe Y with a short description of the transformation. This is the interpretation of statistical dependence in terms of description length and complexity. The best matching makes the data most dependent. This maximizes likelihood or minimizes entropy and hence description length. Our model selection problem is to choose a matching that gives the most compact description of the data. This description automatically maximizes the probability of the data without assuming any known model.

The corresponding decision problem for maximally dependent matching optimization is

MAXIMALLY DEPENDENT MATCHING (MDM)
INSTANCE: A bipartite graph $G = (V, E)$, values $x(e)$, $\forall e \in E$, kernel Φ, a positive integer K, and a number L.
QUESTION: Is there a matching of size K with likelihood $\sum_e \log \sum_{e'} \Phi(x(e), x(e')) \geq L$?

4.2.1 MDM is NP-complete

We will show MDM is NP-complete [22] by transforming the weighted clique problem to it. Intuitively, MDM is a hard problem because the likelihood of each match depends on the other chosen matches. The deceptively similar problem of maximum weight matching where the weights are fixed and independent of other weights is actually in P, and can be solved in $O(|V|^3)$ as previously mentioned. First we show that the weighted clique problem is NP-complete.

WEIGHTED CLIQUE (WC)
INSTANCE: A complete graph $G = (V, E)$, weights $w(e)$, $\forall e \in E$, a positive integer $K \leq |V|$, and a number W.
QUESTION: Does G contain a clique V' of size K with weight $\sum_{v \in V'} \log w(v) \geq W$, where $w(v) = \sum_{u \in V', u \neq v} w((v, u))$.

Theorem 2. *WC is NP-complete.*

Proof. WC $\in$ NP because given a clique, clearly we can compute its weight and check that it is larger than W in polynomial time.

We transform CLIQUE [22], which is known to be NP-complete to WC. Given a graph G' for CLIQUE, we build a complete G for WC by giving each edge in G' weight 2. All edges not originally in G' are given weight 1. Let $K = J$ and $W = J(J-1)$, where J is the clique size in CLIQUE.

If there is a clique of size J in CLIQUE, then there is a clique in WC of size J by construction. Each edge in the clique has weight 2, so each vertex has weight $J-1$. Thus the weight of the clique is $J(J-1)$.

If there is a clique in WC of size J and weight $J(J-1)$, then let that be the clique for CLIQUE as well. Each edge in the clique must have weight 0 or 1 by construction, so each edge must have weight 2 in order for the clique weight to be $J(J-1)$. Then by construction each edge in the clique must exist in G', so the clique for WC is a clique of size J for CLIQUE.

The transformation can be done in polynomial time because there are only a polynomial number of edges to add to G' to make it complete. □

Theorem 3. *MDM is NP-complete.*

Proof. MDM $\in$ NP because given a matching, clearly we can compute the likelihood and check if it is greater than L in polynomial time.

We transform WC to MDM. Given an instance of WC, we construct a bipartite graph of size $|V|$ so that each match is a vertex in the graph of WC, and only edges in WC are present in MDM. This can be done by only adding one-to-one edges in the bipartite graph. The kernel is chosen so that $\Phi(x(e), x(e')) = w(e'')$, where e'' is the weight of the corresponding edge in WC. We set $L = W$.

If there is a clique of size K and weight W in WC, then clearly the corresponding matches in MDM will be a matching of size K with likelihood $W \geq L$ by construction.

If there is a matching of size K with likelihood L, then clearly the corresponding vertices in WC are a clique with weight W.

Clearly, the transformation can be done in polynomial time because the number of edges added is $|V|$, and the kernel can be computed in $O(|V|^2)$. □

So although we have a flexible and general formulation of the matching problem, it is unlikely that there is an efficient algorithm for the optimization problem. This is not unlike other problems in unsupervised learning [17] such as clustering which often lead to hard combinatorial optimization problems. This naturally leads to approximation algorithms which we will present later in the chapter.

4.2.2 MDM Criterion Revisited

Recall that our optimization criterion was to choose the matching with maximal dependence. This could be equated with finding the maximum likelihood matching and minimizing entropy. Given samples from p, the cross-entropy [5] of modeling with q is[2]

$$h_p(q) = -\int p(z) \log q(z) \tag{4.22}$$

$$= -\int p(z) \log \frac{p(z)q(z)}{p(z)} \tag{4.23}$$

$$= -\int p(z) \log p(z) - \int p(z) \log \frac{q(z)}{p(z)} \tag{4.24}$$

$$= h(p) + D(p||q) \tag{4.25}$$

$$\geq h(p). \tag{4.26}$$

The inequality follows because KL divergence is non-negative [12]. The KL divergence $D(p||q)$ is the description length penalty for using an incorrect model. This shows how minimizing the entropy finds the true model that generated the samples.

However in our case, the matching determines the sample itself, before we can even estimate p with q. In other words, both the $p(z)$ and $q(z)$ terms in the cross-entropy depend on the matching because z depends on the matching. So we cannot guaranteed that the MDM matching will be the true matching. Nevertheless, it is a good matching because it explains the data well. Without some sort of side information or assumptions, it is unreasonable to expect any more than this. It maybe possible to analyze the probability that a false matching will have minimum entropy, but again this requires assumptions on the distribution p.

[2]Here we are using the entropy functional h in its proper form as taking a probability distribution argument.

4.3 Markov Chain Monte Carlo Approximation

We have defined the maximum likelihood matching problem that results from analyzing the statistical dependence between two random variables when the matching pairs of variables is unknown. This occurs in applications such as our non-overlapping camera network topology problem because observations from one camera to another arrive out-of-order. We have also shown how dependence is related to mutual information, entropy and likelihood. Our formulation allows us to generalize previous work to handle a larger variety of situations by eliminating restrictive assumptions. Unfortunately, the price for this generality is that the problem is NP-hard, as was shown in the previous section.

In such a situation we naturally look to approximation algorithms. We have chosen the Markov Chain Monte Carlo (MCMC) [24] framework for our problem because it has asymptotic performance guarantees, is simple, and easy to implement. Also it gives promising results for our data. Similar versions of MCMC have been used for related problems [71, 16, 21]. Briefly, MCMC is a way to draw samples from the posterior distribution of matchings given the data. It does this by cleverly using a Markov chain whose stationary distribution is the posterior distribution. In a sense, it is a smart random search with convergence guarantees.

4.3.1 Metropolis-Hastings

The Metropolis-Hastings sampler [24, 29, 59, 58] (see Algorithm 1) is the most general MCMC algorithm. The initial sample is a random matching or permutation. New samples are obtained by conditionally sampling a new matching given the current one via a proposal distribution $q(\pi'|\pi_j)$. The new sample is accepted with probability proportional to the relative likelihood of the new sample versus the current one. The likelihood of a matching is proportional to the log probability of the corresponding transformations, which we compute as $-h(T)$. The acceptance probability has the form of an energy function:

$$\exp[-(c(\pi') - c(\pi))]. \tag{4.27}$$

It is easy to see that lower cost matchings are always accepted. In addition, high cost matchings are accepted with exponentially decreasing probability. This is what differentiates MCMC from simple greedy local search. By allowing movements to higher cost feasible matchings, the algorithm gives itself a chance to explore more of the feasible set.

The algorithm repeats the sampling process for the desired number of samples. We can compute $I(X;Y_{\pi_j})$ for each sample π_j and take the average as the expected posterior MI, or choose the maximally dependent sample. The algorithm is very simple and easy to implement. The approximation improves with increasing number of samples at the cost of more computation.

Indeed Metropolis-Hastings sampling is very generic and can be applied to other optimization problems. It has roots in statistical mechanics especially with respect to simulating Ising models [45]. The advantages of the algorithm are convergence guarantees based on

Algorithm 1 Metropolis-Hastings

Initialize π_0; $j = 0$.
loop
  Sample π' from $q(\cdot|\pi_j)$.
  Sample U from $U(0,1)$.
  Let $\alpha(\pi_j, \pi') = \min\left(1, \frac{p(\pi')q(\pi_j|\pi')}{p(\pi)q(\pi'|\pi_j)}\right)$.
  if $U \leq \alpha(\pi_j, \pi')$ **then**
    $\pi_{j+1} = \pi'$.
  else
    $\pi_{j+1} = \pi_j$.
  end if
  $j \leftarrow j + 1$.
end loop

ergodic Markov Chain theory [24], and that the nonlinear matching constraints are always explicitly satisfied.

4.3.2 Proposal Distribution

The key to the efficiency of an MCMC algorithm is the choice of proposal distribution. Indeed, if the proposal distribution was the true posterior distribution, then clearly every sample is accepted and we have already converged. In practice, it seems effective to use proposals which make both local and more global changes. The local changes allow for fine tuning while the more global ones help avoid local minima. As stated before, MCMC is simply a more principled way of doing random search.

We use three different types of proposals for sampling matches:

1. Add,
2. Delete,
3. Swap.

Swapping matches is related to augmenting paths [67] used in algorithms for assignment such as the Hungarian algorithm. For an x_i, the idea is to choose a new match y_j and check if we violate the one-to-one constraints. If so, we look for a new match for the $x_{i'}$ which was previously matched to y_j. This process is repeated until either no match constraints are violated or failure. If we fail, we can choose a different starting x_i. This allows the algorithm to sample a new matching with large changes in the matches.

The ability to add and delete matches enables matches to be swapped without generating samples with highly improbable matches in the process. This might occur if we try to match everything at once because even with augmenting paths, new samples are generated in a very particular way. By allowing the number of matched objects to change, we effectively create

opportunities for fine tuning the matching. Of course we must prevent convergence to no matches at all. Furthermore our problem is formulated as matching all n observations. We can handle this by considered non-matches as matching a missing corresponding observation. In fact this will also enable us to deal with actual missing matches in real data. To make the cost function non-degenerate, we impose a penalty for missing matches. This is equivalent to setting these matches to some nominal probability. In summary, the proposals are simple to implement and work well in our experience.

4.3.3 Simulated Annealing

In optimization uses of MCMC sampling, the method of simulated annealing [46] is often used to speed up convergence and to avoid local minima. Once again the analogy is a physical one of cooling of a metal. A temperature β is added to the energy function resulting in

$$\exp\left[-\frac{(c(\pi') - c(\pi))}{\beta}\right]. \tag{4.28}$$

It is clear that starting at high temperature allows more transitions to higher cost matchings, while at low temperature only transitions to matchings which improve the cost are allowed. Essentially a tuning parameter is added to control the exploration of the feasible set. The heuristic of early exploration and late exploitation seems to work well in practice. An annealing schedule requires choosing an initial temperature and a rule for decreasing the temperature. We use the standard exponential cooling scheme

$$\beta \leftarrow k\beta, \tag{4.29}$$

where $0 < k < 1$. The initial temperature can be calibrated by choosing one that results in a certain average probability of acceptance, such as 0.8. We have found that clamping the temperature until a certain number of swaps have been made improves performance in our problems.

4.3.4 Entropy Estimation

The MCMC sampler requires a likelihood given a matching. We have shown how this likelihood can be computed as the entropy of the transformation distribution. The differential entropy of a random variable is defined as

$$h(T) = -E[\log p(t)]. \tag{4.30}$$

It is an average of the log probability of the samples, not their values as in simpler quantities such as moments. Thus entropy cannot be calculated directly from samples as simply as moments.

There are many ways to estimate the entropy given a sample [3]. Because entropy is a function of the density, an approach to calculate it is to first estimate the density. Indeed

all of the approaches do this, if only implicitly. The kernel density estimator [70] is a simple way to estimate the density given samples. It is defined as

$$\hat{p}(t) = \frac{1}{n}\sum_{i=1}^{n} \Phi\left(\frac{t - t_i}{\sigma}\right) \tag{4.31}$$

where σ is the bandwidth, Φ is the kernel, and $t_1, ..., t_n$ are independent and identically distributed samples called centers. The Gaussian function makes a convenient kernel. Kernel density estimators are simple yet flexible enough to model densities with multiple modes. In addition they can clearly be fit easily to data.

We approximate the entropy as

$$h(T) = -E[\log p(t)] \approx -\frac{1}{n}\sum_{i=1}^{n} \log \hat{p}(t_i). \tag{4.32}$$

Given a single set of samples, we use a leave-one-out estimate that evaluates the probability of a sample using all other samples but itself. This gives an estimate with smaller bias. The computation is $O(n^2)$. In fact, the bandwidth σ is optimized to minimize the entropy which maximizes likelihood as previously shown.

For one-dimensional data we can use the faster m-spacings estimate [89]. The estimate is

$$h(t) \approx \frac{1}{n}\sum_{i=1}^{n-m} \log\left(\frac{n}{m}(z_{i+m} - z_i)\right) \tag{4.33}$$

where z_i are the order statistics for t_i. The primary computation is sorting the data to obtain order statistics, which is $O(n \log n)$. The spacings estimate implicitly uses a piecewise uniform density estimate. This type of entropy estimator has been used in other problems such as independent components analysis [51].

4.4 Missing Matches

As previously mentioned, real data often contains missing observations which results in missing matches X and Y. For example, this can occur in a camera network if an object is undetected in one camera or simply enters a building or does not transition to the other camera at all. Thus some x_i's may not have corresponding $y_{\pi(i)}$'s. We consider missing data as outliers, and model the distribution of transformations as a mixture of the true and outlier distributions. To use the spacings estimate of entropy, we can minimize an upper bound on the mixture entropy. Let $p_1 = p(T|\omega = \text{missing})$, $p_2 = p(T|\omega = \text{present})$ and $\lambda = p(\omega = \text{missing})$. The joint entropy can be written as,

$$h(T, \omega) = H(\omega) + h(T|\omega) \tag{4.34}$$

$$= h(T) + H(\omega|T). \tag{4.35}$$

Thus,

$$h(T) = H(\omega) + h(T|\omega) - H(\omega|T) \tag{4.36}$$
$$= H(\lambda) + \lambda h(p_1) + (1-\lambda)h(p_2) - H(\omega|T) \tag{4.37}$$
$$\leq H(\lambda) + \lambda h(p_1) + (1-\lambda)h(p_2) \tag{4.38}$$

because $H(\omega|T) \geq 0$. Often a uniform outlier distribution is used both for simplicity and because it makes minimal assumptions on the outlier distribution.

Recall that we wanted to compute the maximally dependent matching which was shown to be NP-hard. Thus we used an MCMC algorithm with a proposal distribution that added, deleted, and swapped matches. Essentially this gives us a guided random search or local replacement algorithm. The Markov chain dynamics ensure convergence to the true posterior distribution. Overall the algorithm is quite simple and easy to implement.

4.5 Non-overlapping Cameras

We now return to our motivating problem of matching in a network of non-overlapping cameras described in chapter one. Recall that this problem is what led us to consider the idea of maximally dependent matching. In this chapter we apply the general reasoning from the previous chapters to our specific problem. An earlier version of our work is described in [86].

Consider the problem of wide-area surveillance, such as traffic monitoring and activity classification around critical assets (e.g. an embassy, a troop base, critical infrastructure facilities such as oil depots, port facilities, airfield tarmacs). We want to monitor the flow of movement in such a setting from a large number of cameras, typically with non-overlapping fields of view. To coordinate observations in these distributed cameras, we need to know the connectivity of movement between fields of view (i.e. when an object leaves one camera, it is likely to appear in a small number of other cameras with some probability). A simple example with two cameras imaging the upstream and downstream sections of a road is shown in Figure 4-1. We want to infer that objects leaving the upstream view are likely to transition to the downstream view. We also want to infer the distribution of transition times between the two views.

In some instances, one can carefully site and calibrate the cameras so that the observations are more easily coordinated. However even with calibrated cameras, the departure/arrival locations, connectivity, and transition time distribution still have to be learned. In many cases, cameras must be rapidly deployed and may not last for long periods of time. Hence we seek a passive way of determining the topology of the camera network. That is, we want to determine the structure of the set of cameras, and the typical transitions between cameras, based on noisy observations of moving objects in the cameras.

Departure and arrival locations in each camera view are nodes in the network. An arc between a departure node and an arrival node denotes connectivity (transition). We want to infer both the topology (that is, which arcs exist) and the transition distribution.

camera 1 (upstream) camera 2 (downstream)

Figure 4-1: Upstream and downstream camera views of two portions of the same road.

For the example in Figure 4-1 we want to infer that the views are connected and estimate the distribution of transition times. Simply put, we are given observations in a set of non-overlapping cameras and must infer how the cameras are related to each other. The problem can be viewed as system identification in that we are given only a myopic view of the system, namely the inputs and outputs, and must infer the dynamics or inner workings.

4.5.1 Related Work

Previous work on tracking across multiple cameras generally either assumed known camera topology or known correspondence. Methods which use assignment algorithms for tracking across multiple cameras assume the transition models are known or fit them with hand-labeled correspondences [33, 44, 38]. Other work for calibration also assume known correspondence [20, 74].

Makris et al. [55] have tackled the problem of estimating a multi-camera topology from observations. They assume a single mode transition distribution and exhaustively search for the location of the mode. Their method assumes all departure and arrival pairs within a time window are implicitly corresponding. The distribution of transition times obtained from this correspondence is examined for a peak by thresholding based on the mean and standard deviation. Essentially the correlation between arrival and departure times is computed using a loose, implicit notion of correspondence. They show promising results using this method.

Correlation is effective for monotonic relationships in general, but is not flexible enough to handle multi-modal distributions. Makris and Ellis [55] have acknowledged this fact, which can occur when both cars and pedestrians are part of the observations. Their approach essentially assumes a Gaussian transition distribution and implicit true correspondences within a chosen time window. However for a given departure observation, the true correspondence is a single arrival observation. So for all observations within a time window, the true and false correspondences generate a mixture of the true and false transition distribution. The time window size and distribution of observations determines the number

of false correspondences versus the single true correspondence. In general the more dense the observations and the longer the transition time, the more false correspondences. Thus their method suffers from assuming a unimodal transition distribution, and only implicitly dealing with the matching problem.

Our method generalizes their approach to more flexible, multi-modal transition distributions, and explicitly handles the matching problem. This is accomplished by using our information theoretic formulation of statistical dependence described in the previous chapters. Our approach makes very few assumptions and does not require supervision.

4.5.2 Limitations of Correlation

Given two stochastic processes $x(t)$ and $y(t)$, the classic way to measure their dependence is via the cross-correlation $E[x(t)y(t-\tau)]$ (assuming zero mean processes). The strength of dependence is proportional to correlation and the nature of dependence time lag τ is searched over some suitable range. The canonical example is a fixed linear dependence with Gaussian noise. There are two major limitations of this approach. The first is that correlation only correctly measures linear dependence, and at best can approximate monotonic dependence. This leads to weak values of dependence when the nature of dependency is nonlinear. For our application, arrivals may arrive out of order with respect to their departure times. This is a simple consequence of objects moving at different speeds, for example vehicles versus pedestrians. In this case, there is no single time lag shared by all objects. In addition, other transformations such as appearance change may be highly nonlinear.

The second limitation is not so much a property of correlation as to how it is usually applied for problems such as ours. Arrivals and departures are more appropriately treated as point processes or stochastic events on the time axis. In this case, rather than just computing correlation strictly at a fixed time lag, a time lag window is chosen to calculate $p(y \text{ at } t+h | x \text{ at } t)$. The window is used to sidestep sampling problems resulting from the discretization of time and to smooth out values to combat the sparsity of data. The problem is apparent when we look at this from the point of correspondence. Effectively, all arrivals and departures with the time window are considered matching. We call this a pseudo-matching. Already we see that this violates the natural one-to-one matching constraints of a permutation of indices. We are thus hampered by a distorted measure of dependence and less precise characterization of the nature of dependence.

Thus although correlation is simple and leads to simple algorithms, the formulation is inherently limited and inappropriate for the matching problem at hand. Indeed we can study how correlation performs in different situations. From the above discussion, the resulting distribution of transformations obtained is a mixture of the true distribution from a correct matching and the false distribution from incorrect matches:

$$p(\delta) = \lambda p_1(\delta) + (1-\lambda)p_0(\delta), \tag{4.39}$$

where λ is the proportion of true matches, and p_1 and p_0 are the distributions of true and false matches, respectively. Assuming linear dependence, correlation will perform well when

λ is close to one so that the pseudo-matching contains mostly true matches. This is the case if objects for example all shared the same time lag and the time between arrivals/departures is larger than the pseudo-matching time window. However, when the process is dense in observations, within a time window, the pseudo-matching will consist of a single true match and all other matches false. In addition if the time lag is large relative to the density of observations, then this ratio of true to false matches worsens. More formally, a non-parametric estimate of the densities is

$$p(\delta) = \frac{1}{n}\hat{p}_1(\delta) + \frac{n-1}{n}\hat{p}_0(\delta). \tag{4.40}$$

This shows how either the number of matches in a time lag window has to be small, so that n is small, or the distribution of true lags has to have very low entropy, so that $\hat{p}_1(\delta)$ has much larger values than $\hat{p}_0(\delta)$. One way to visualize this is with the true distribution being a narrow peak while the false distribution is low and wide, or nearly uniform. Note that even when a strong enough dependence is detected, the nature of this distribution as represented by the transformation distribution is almost always corrupted by false matches except in the case when the time lag window is large enough to accommodate all true matches, but the time between arrivals/departures is larger than this window. In contrast our approach of directly searching for valid matchings while maximizing dependency avoids these problems.

4.5.3 Camera Networks

Figure 4-1 shows two portions of a road that both vehicles and pedestrians move between. The corresponding camera network is shown in Figure 4-2. Nodes are camera arrival and departure locations while arcs represent the transition of objects locations. Each node can also contain information about the camera such as arrival and departure rates. Associated with an arc is information about how objects move from the source to destination node. For example, in the simplest case, the arc encodes the distribution of transition times for an object to move between the nodes. It can also encode other types of transformations such as changes in appearance. Arcs within a camera are assumed known from tracking in a single camera. We only assume that within a camera we can detect arrival and departure locations. Note that two cameras can be considered connected if there exists some arc between any of their respective nodes.

If we could identify the same object in different cameras (for example, using a license plate reader or face recognition system), then learning the topology and transitions would be easy. In practice, in wide-area surveillance, the matching between observations in different cameras is difficult to obtain because cameras may be widely separated and the observations may occupy only a few pixels. A key feature we can exploit is the time of arrival and departure. It can be measured fairly accurately by tracking in individual cameras. Other features such as image color can also be used.

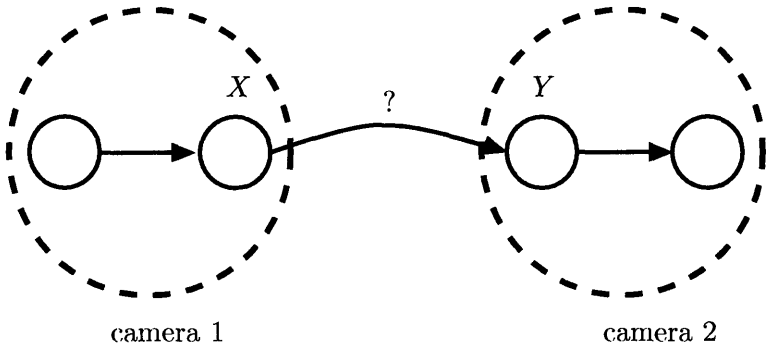


Figure 4-2: Camera network of Figure 4-1. Nodes correspond to arrival and departure locations in the camera view. Within-camera arcs are known via within-camera tracking.

4.5.4 Problem Formulation

To infer the topology of a camera network, assume we have identified arrival and departure locations and observations in each camera. For example, this can be done with a blob-based tracker in each camera separately [80]. For each pair of cameras, we want to infer whether they are connected and the distribution of transition times. Recall that this is made more difficult because the matching between observations in different cameras is unknown.

Suppose we are given observations of departure $x_1, .., x_n$ and arrival $y_1, .., y_n$ times in two connected cameras, respectively. Also, assume that the correspondence between the observations is given by a permutation π of the indices such that $(x_i, y_{\pi(i)})$ is a corresponding pair. We formalize this by writing

$$y_{\pi(i)} = t_i(x_i), \tag{4.41}$$

where the distribution of transformation t_i is parameterized by θ.

For departure and arrival time observations X and Y, the transformation T is an additive transition time between cameras: $Y = X + T$. Our formulation also captures other transformations such as color variations between cameras. We will show this in the experiments. Based on our formulation, both the strength and nature of dependence is determined by the distribution of T. Basically T tells us how X and Y are related, and the randomness in $p(T)$ indicates strength of dependence. In particular, strong dependence means that observations y are highly predictable given x. This will be reflected by low entropy in the distribution of T. So the strength of statistical dependence measures how connected two cameras are, and the nature of this dependency is encoded in the corresponding distribution of transformations.

4.5.5 Optimization

The combinatorial nature of permutations makes computation by direct enumeration intractable. We use the approximate MCMC algorithm described previously to find the maxi-

mally dependent matching. These matchings give us correspondences between observations, from which we can infer the strength and type of connectivity between cameras.

In-order Case

Assume, say for the non-overlapping camera network problem, the observations arrive in order. In other words, (x_i, y_i) is the true match. Obviously this makes finding the true matching trivial by simply matching in order. Now it may be the case that many of the observations do in fact arrive in order although we do not know this in advance. Thus it is useful to check that our method works in the simplest case.

If the in-order condition is met, then it must be the case that the maximum transition time is smaller than the time between observations. Even if the true transition distribution makes out-of-order possible, the fact that the data is in-order means that we can effectively truncate the transition distribution to the one actually observed. Perhaps the simplest transition distribution is a uniform $U(0, a]$, where a is the maximum transition time. Then we must have the time between observations $b > a$. A true match (x_i, y_i) will give transition time

$$y_i - x_i \leq a, \tag{4.42}$$

while a false match $j \neq i$ will give

$$\begin{aligned} y_j - x_i &= x_j + w_j - x_i && (4.43)\\ &> x_j - x_i && (4.44)\\ &> b. && (4.45) \end{aligned}$$

Clearly then false matches will increase the range of transition times, thereby increasing the transition distribution entropy and decreasing mutual information and statistical dependence. So for the simple in-order case, our optimization criterion gives the true matching.

Out-of-order Case

The general out-of-order case may occur if the time between observations can be smaller than the maximum transition time. As previously noted, because of the complex dependency between the samples and the entropy estimate, this case is difficult to analyze. It is possible that given the true matching, we might be able to find some set of matches such that swapping them would reduce the entropy. One way this can happen is to take true matches which result in points at the tail of the transition distribution, and swap them so that they fall closer to the mode of the distribution. This corresponds to a tightening of the transition distribution that would tend to give an overly optimistic value of dependence. Nevertheless it would preserve the form of the true distribution. Of course it is possible that false matches can give rise to an entirely different distribution which has even lower entropy.

We can write the result of combining true and false matches as a mixture

$$\bar{p}(w) = \lambda p(w) + (1 - \lambda)q(w), \tag{4.46}$$

where $0 \leq \lambda \leq 1$ controls the relative proportions of true vs. false matches. By the convexity of entropy, we have

$$h(\bar{p}) \geq \lambda h(p) + (1 - \lambda)h(q) \tag{4.47}$$

$$\geq h(p) \quad \text{if } h(q) \geq h(p). \tag{4.48}$$

Thus if the distribution resulting from false matches has higher entropy than the true distribution, then all is well and we are guaranteed to find the true matching. Otherwise the generality of our approach might choose a false matching that exhibits higher statistical dependency. Note that all of our analysis is asymptotic because n needs to be large for the estimates to converge to their true values. We leave any finite sample size analysis for future work.

4.5.6 Experiments

First, we show detailed results for a simulated and real road. In both cases, two cameras are positioned at two non-overlapping portions of the road. Finally, we show results for a simulated and real traffic network of cameras.

Simulated Road

To study the differences between our approach and previous work we simulated a data set of 100 points from a Poisson(0.1) departure process. The transition distribution is a mixture of Gamma(16.67, 0.33) and Gamma(266.67, 1.33). This generates a dense arrival process and two transition time modes with different means and identical variance. Real objects such as pedestrians and vehicles often exhibit this type of process.

Recall that the correlation method matches all observations within a transition time window. These assumed correspondences are used to estimate the distribution of transition times. Figure 4-3 shows the transition distributions estimated using the correlation method with various time windows. The number of false correspondences causes the transition distribution to differ greatly from the true distribution. It is difficult to choose a best correlation time window. Also, correlation weakens with increasing distance between the means of the mixture component distributions because it assumes unimodality. Although the transition distribution has low entropy, correlation fails to capture this.

Figure 4-4 shows our approach on the same data. Although we do not recover the transition distribution exactly, it is much closer in shape than the ones obtained from the correlation method. The estimated MI of 2.47 is close to the true value of 2.12. In general it is difficult to recover the true transition distribution, however our algorithm does find distributions that are qualitatively similar in structure (multi-modal) and quantitatively

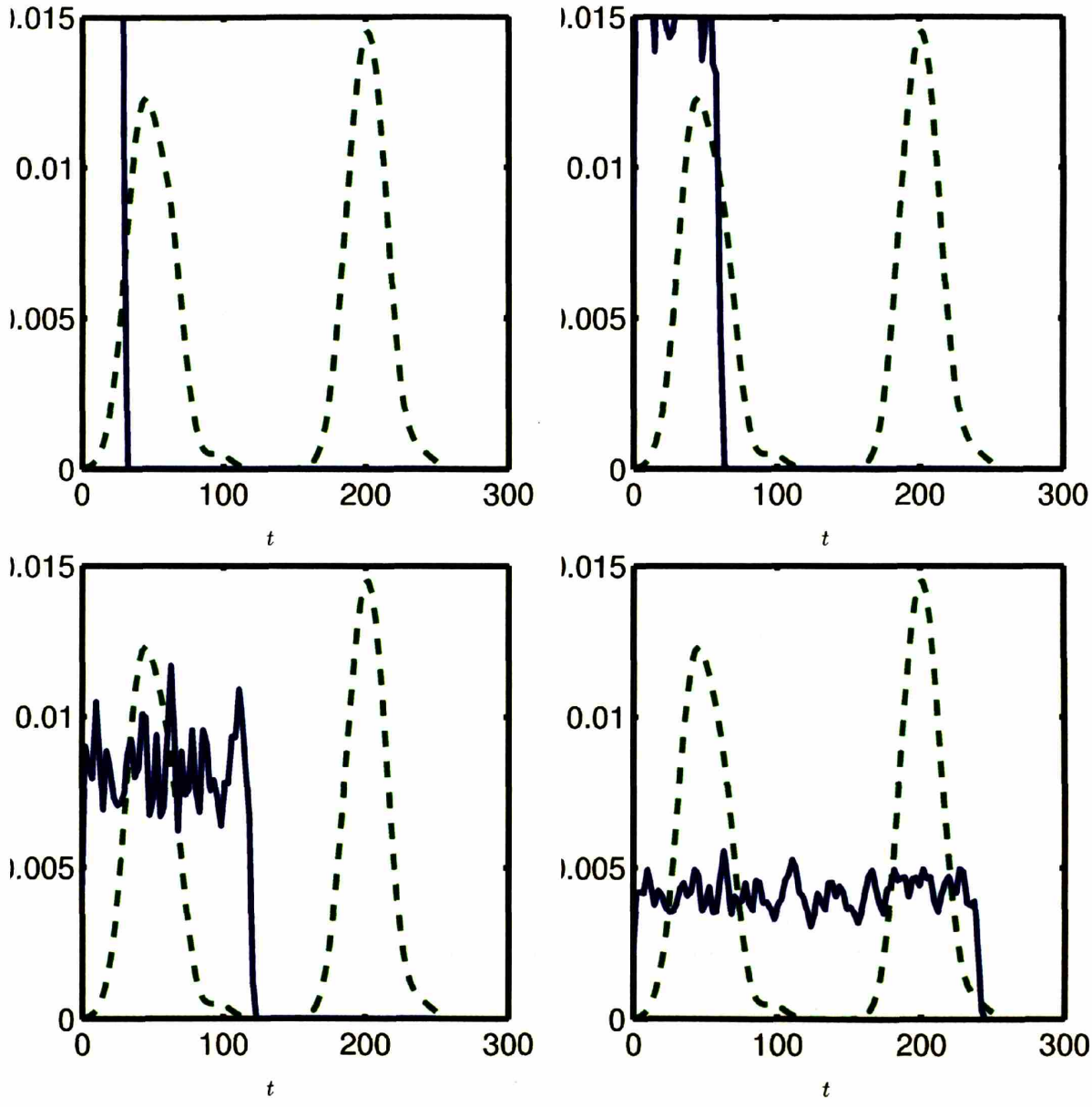


Figure 4-3: Transition distributions obtained using correlation with different time windows all fail to match the simulated multi-modal distribution (dashed plot). In addition, there is no clear maximum peak indicating statistical dependence.

similar in MI. Departure and arrival times alone may not be able to resolve the ambiguity that can occur by correspondences which shift the modes of the transition distribution.

Single Road

Consider the two views shown in Figure 4-1 of upstream and downstream portions of the same road. Cars and pedestrians passing through the scene will appear in one view and subsequently in the other. We hand-labeled 100 matches in one day of tracking data obtained from a blob-tracker [80]. The data also contained about 25% unmatched outliers.

Figure 4-5 shows the transition distributions estimated using the correlation method. As in the case for the simulated data, correlation cannot accurately recover the multimodal nature of the transitions. The also results in a higher entropy distribution and less statistical dependence. Figure 4-6 shows the results of our approach on this data. Note how the number of matches changes rapidly initially but eventually converges. Our recovered transition distribution matches the true distribution fairly well. The sharpness of the posterior correspondences point to why we can recover the transition distribution fairly accurately. Figure 4-7 shows a sample of the correspondences we obtain from our method. In total, 86% of the matches were valid. This is using temporal information alone. As a comparison, a naive approach which matches objects based on raw image appearance and the Hungarian algorithm results in only 15% valid matches as shown in Figure 4-8.

Simulated Traffic Network

We built a traffic simulator to generate data for a simulated network of cameras at intersections. The simulator was based on a real road network, and took into account real traffic patterns and vehicle dynamics. An example network is shown in Figure 4-9. Nodes represent traffic intersections with cameras, while arcs are roads. We simulated 1000 car trips using shortest paths from start to end node with some noise in the path. Departure and arrival times were recorded.

We computed MI for each pair of cameras. For each camera, edges with MI values above a fixed percentage of all values were added. For a Markov chain $X \rightarrow Y \rightarrow Z$, the data processing inequality guarantees $I(X;Y) \geq I(X;Z)$ and $I(Z;Y) \geq I(Z;X)$ [12]. Thus directly connected cameras have higher statistical dependence (assuming roughly equal unconditional entropies). Examples of learned graphs based on different percentage thresholds are shown in Figure 4-9. In our experiments greedy selection closely approximates the correct topology.

Real Traffic Network

We obtained data from a real traffic network of five cameras. Examples of the tracked vehicles in this network are shown in Figure 4-10. Once again we applied our method as for the simulated traffic network. For this experiment we also added color transformations from one camera to another. This is commonplace because cameras often have different

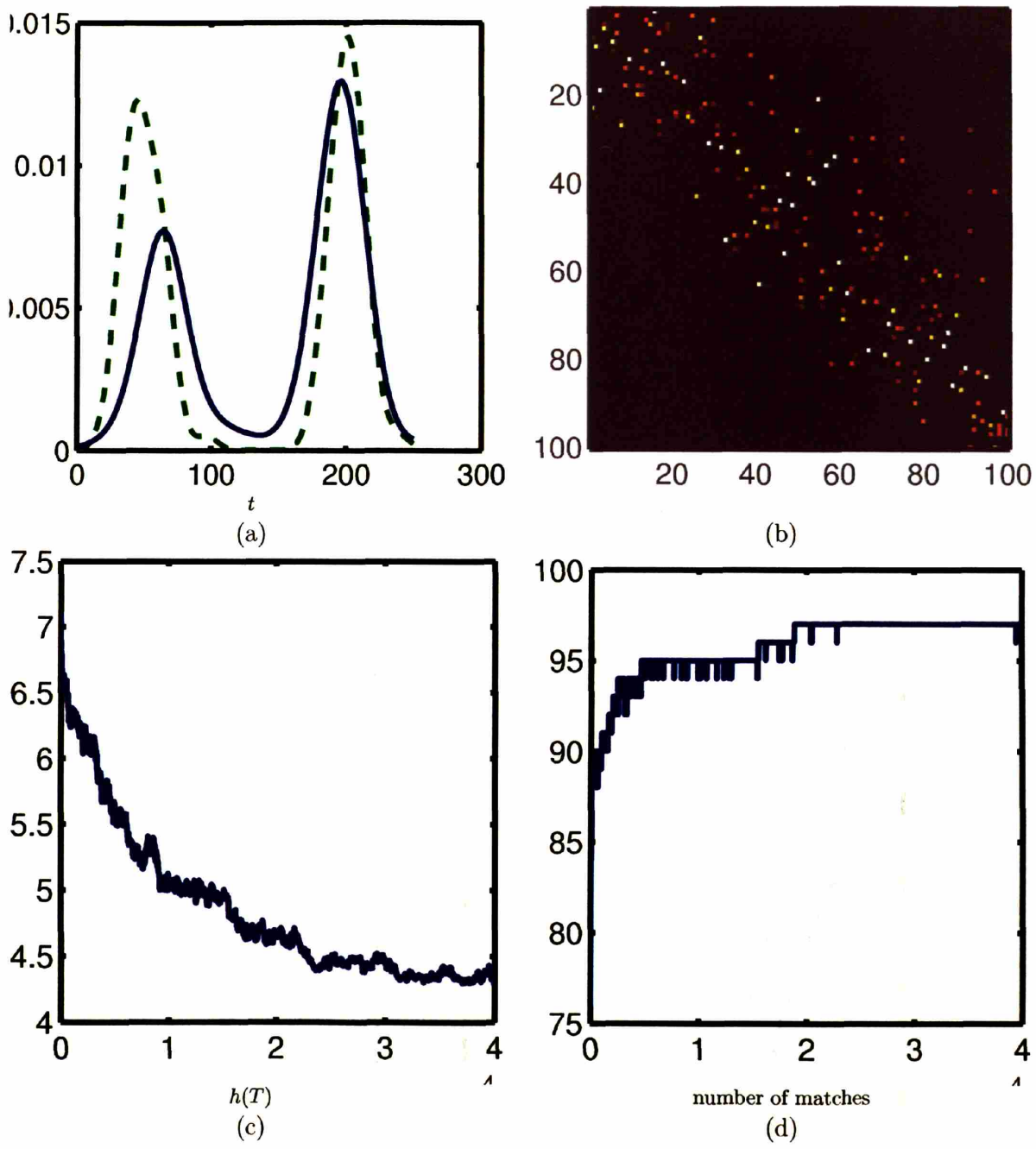


Figure 4-4: Our method on the simulated road. (a) Estimated transition distribution. (b) Samples from the posterior distribution of correspondences $p(\pi)$ (true correspondence along the diagonal). (c) Entropy of the transition distribution vs. MCMC iteration. (d) Number of correspondences vs. MCMC iteration.

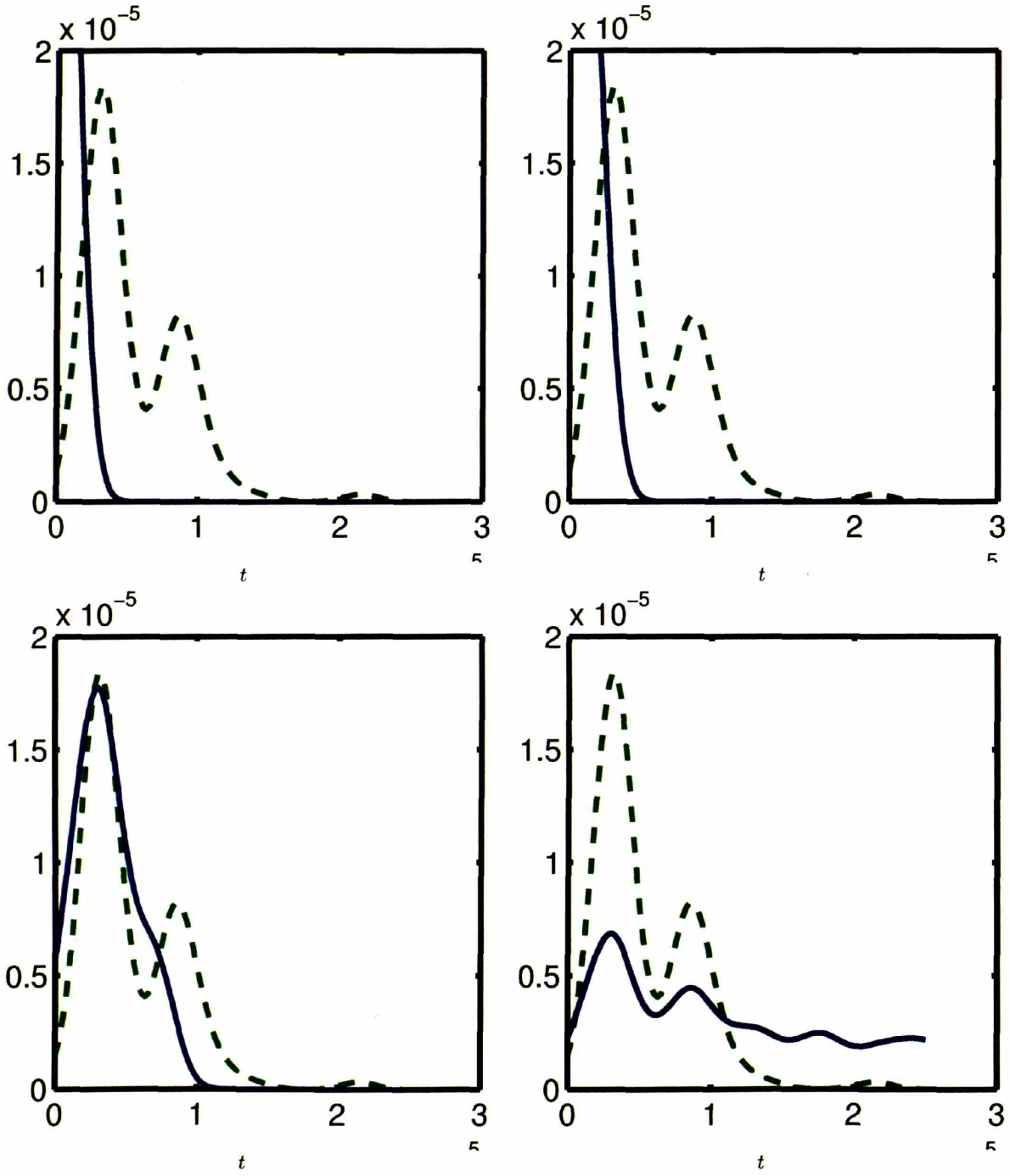


Figure 4-5: Transition distributions obtained using correlation with different time windows on the road data. The dotted distribution is the true one. The results vary widely for different time windows.

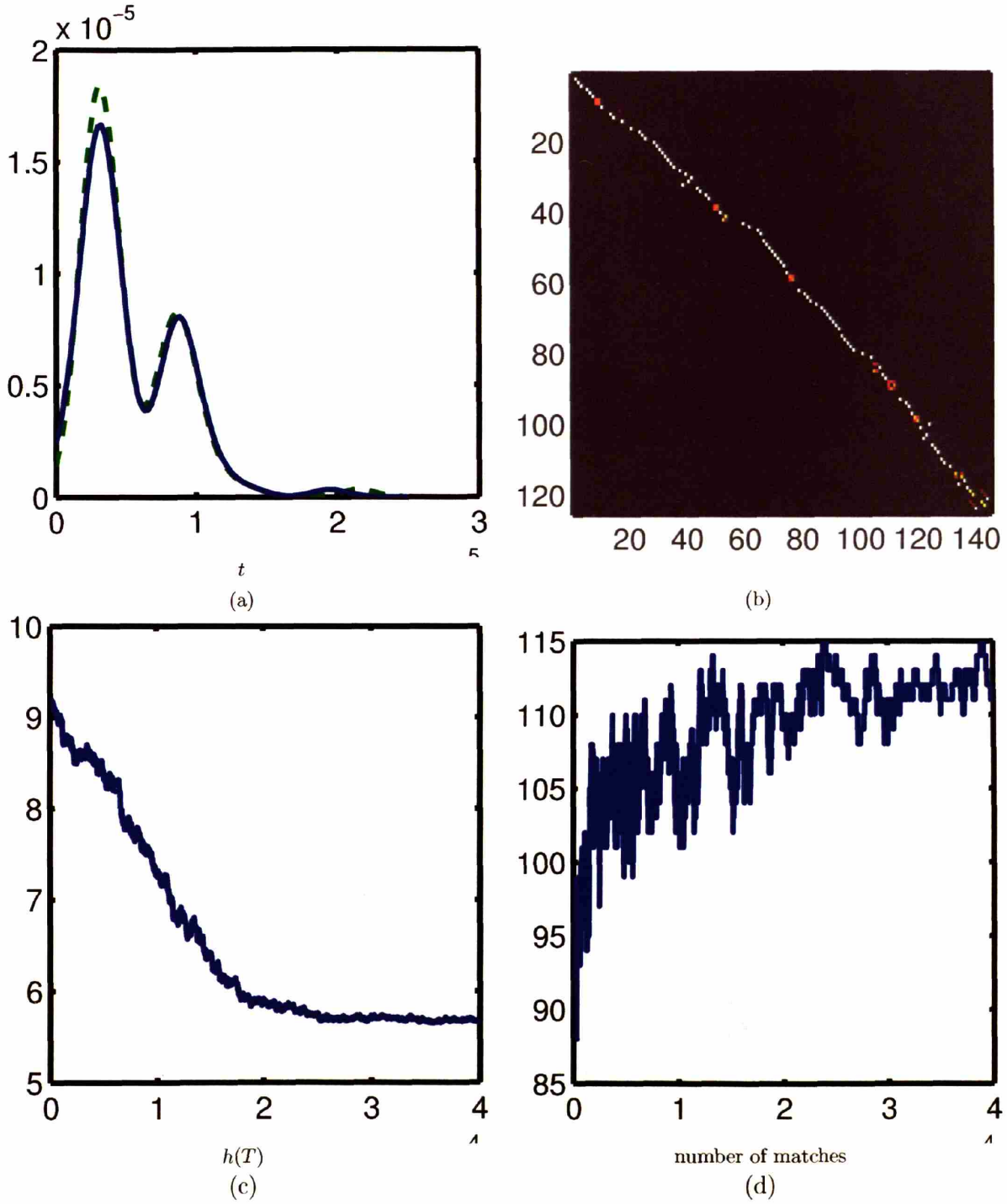


Figure 4-6: Our method on the road data. (a) Estimated transition distribution. (b) Samples from the posterior distribution of correspondences $p(\pi|O)$ (true matching along the diagonal). (c) Entropy of the transition distribution vs. MCMC iteration. (d) Number of correspondences vs. MCMC iteration.

Figure 4-7: Examples of objects matched by our method. The second from the right has been considered an outlier by the algorithm.

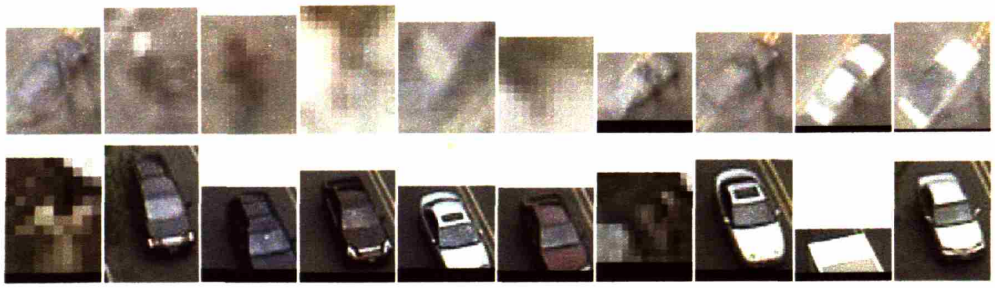


Figure 4-8: Examples of objects matched by naive raw pixel appearance.

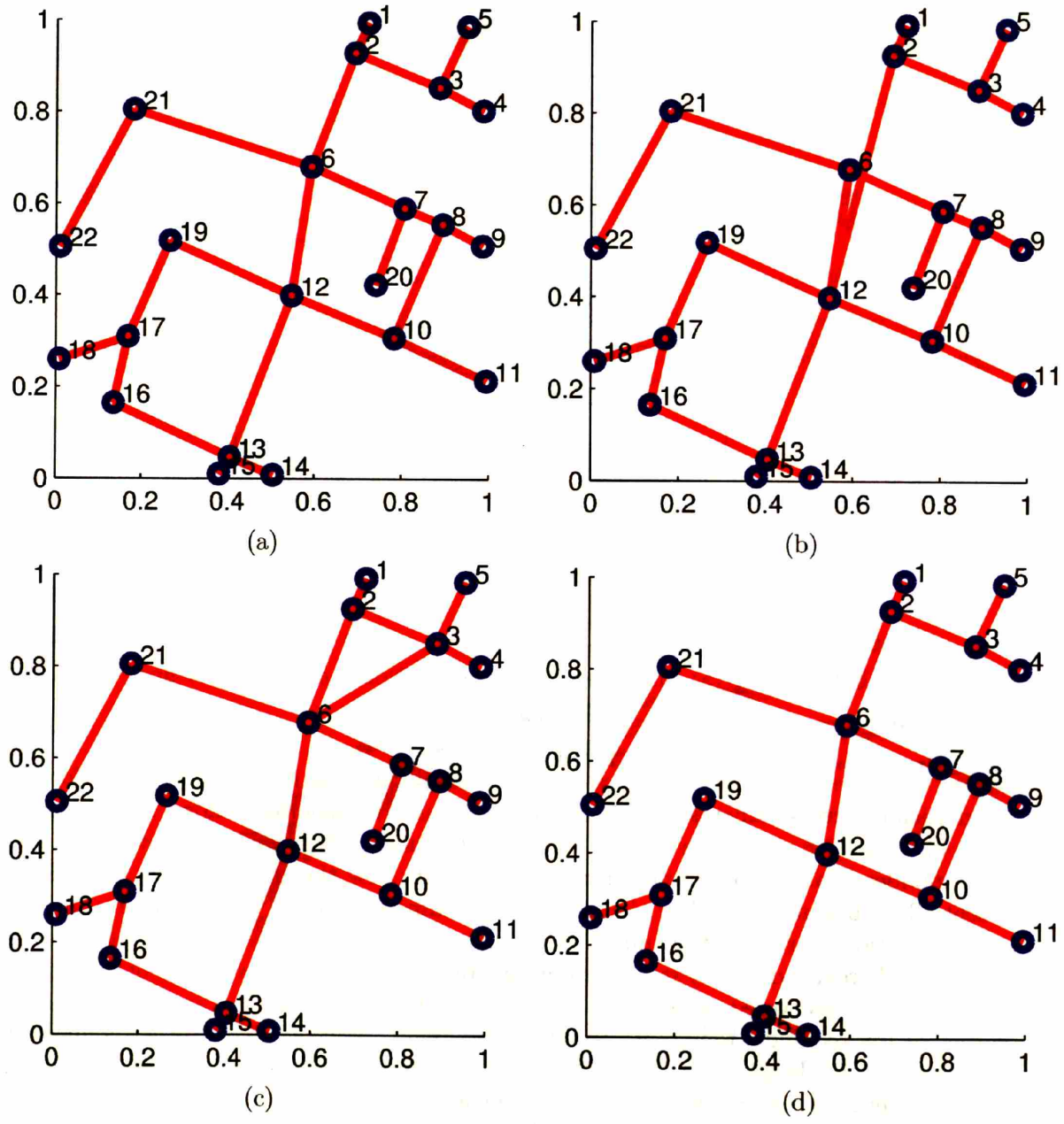


Figure 4-9: (a) The true simulated network of cameras. (b)-(d) Examples of recovered graphs of the simulated traffic network for different MI thresholds. Here the camera locations are assumed known for visualization purposes, but our algorithm is agnostic to this information.

Figure 4-10: Examples of tracked vehicles in the real traffic network.

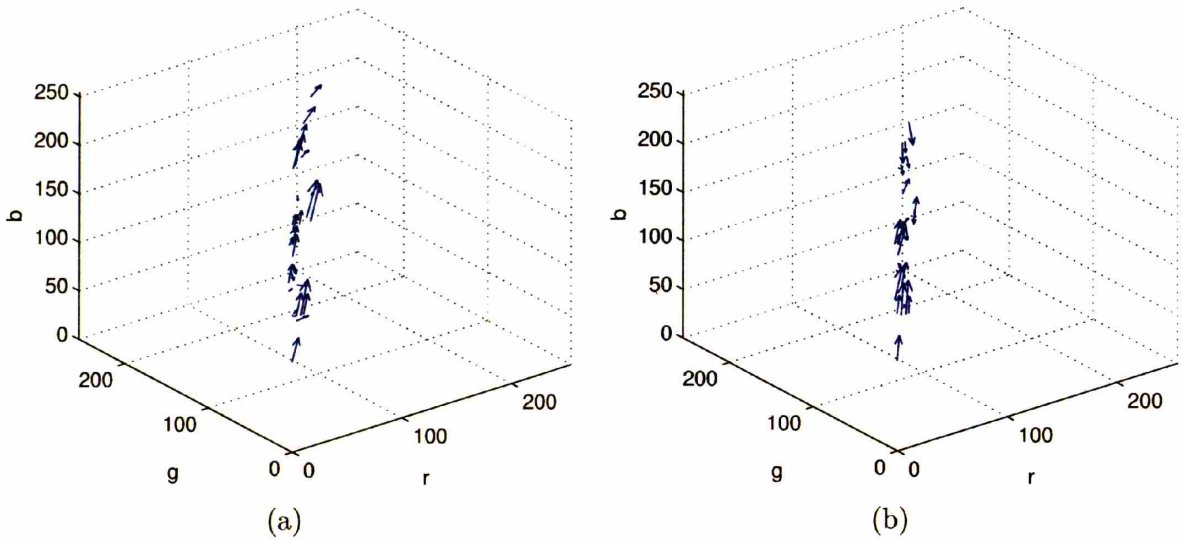
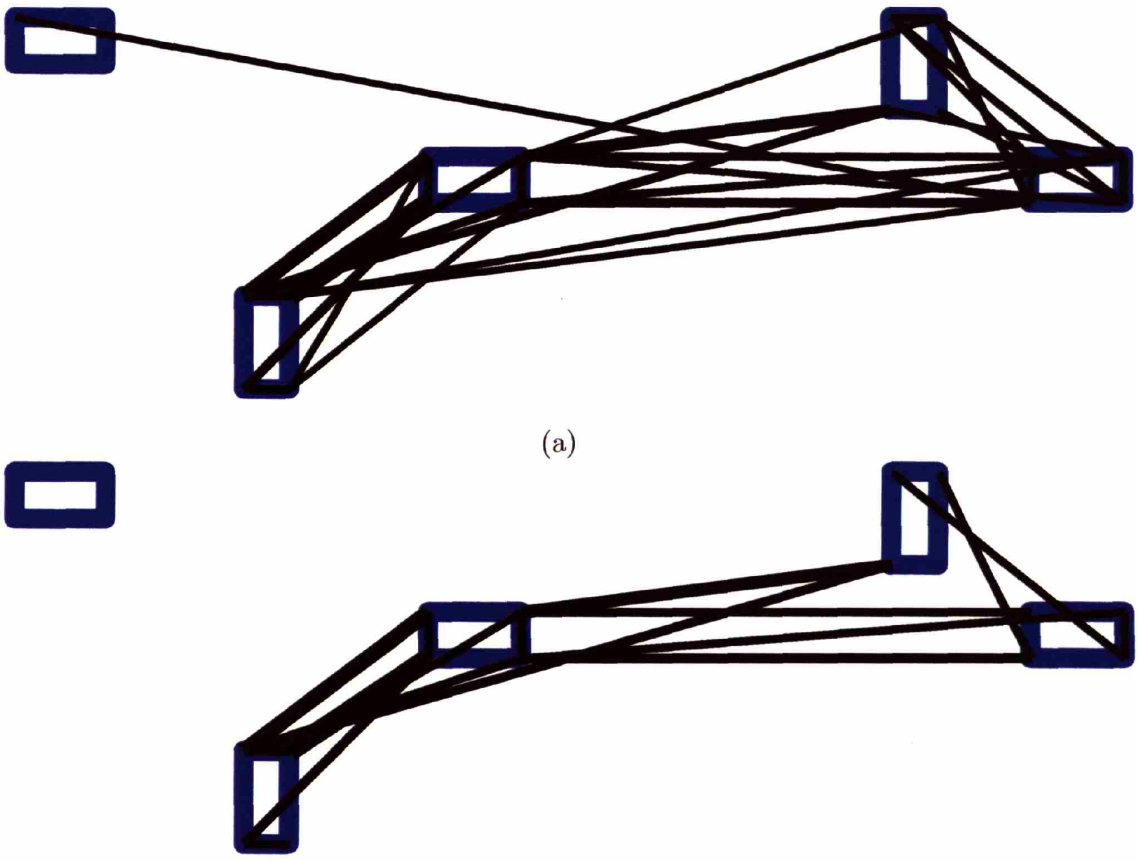


Figure 4-11: (a) Matching color flow (b) Non-matching color flow

sensor responses. In addition, for wide area surveillance the lighting conditions for vary dramatically across cameras. Color transformations are modeled as flows in RGB space, $c_y = c_x + \Delta c$, for an RGB vector c. Figure 4-11 shows estimated color flows for a good correspondence and an essentially random one. Note how the corresponding color flow is essentially a brightening, while the non-corresponding one is less unstructured. The total transformation entropy is the sum of the temporal and color transition entropies. In this case we had greater difficulty inferring the camera transition topology. Many of the primary transitions are recovered as shown in Figure 4-12. Each rectangle is a camera where each corner of the rectangle represents an entry/exit point. Weaker second order connections also show up. We believe these difficulties are primarily caused by the lack of data. Many of the links between cameras had only about 30 correspondences. In addition, accurate times of departure and arrivals were only available at frame resolution.

We have described how to formulate the inference of camera network topology in terms of maximally dependent matching. This method generalizes previous work by removing restrictive assumptions and enables more complicated transition distributions between cameras. We have shown results on both simulated and real data using the algorithm described in the previous chapter.



(a)

(b)

Figure 4-12: Links inferred for the real traffic network. Line thickness is proportional to strength of statistical dependence. (a) low MI threshold (b) high MI threshold.

4.6 Summary

We have analyzed the problem of estimating statistical dependence when the correspondence or matching between observations is unknown. In particular we considered the case of minimal restrictive assumptions where the distributions are unknown and non-parametric. We proposed to find the matching that maximized the statistical dependency the resulting data. In other words, we have a distribution-free matching problem. The corresponding maximally dependent matching decision problem was shown to be NP-complete. Nevertheless, in the following chapter, we show how an approximate solution can be obtained via a randomized algorithm.

Chapter 5

Conclusion

Estimation of statistical dependence is the key underlying task for problems which involve learning the structure of probabilistic models. We have seen two demonstrations of this for the problems of object interaction and matching. In object interaction, statistical dependence arises as a natural, quantitative measure of interaction. In matching, statistical dependence serves as an optimization criterion by which to judge the quality of a matching. We have shown theoretically how this approach generalizes previous work and empirically how it outperforms previous methods.

An advantage of formulating our problems in terms of statistical dependence estimation is that we can leverage the large body of existing work on information theory and model selection. The ideas from these fields allow us to view interaction and matching in terms of statistical dependency structure and uncertainty in prediction.

Our primary goal has always been to simplify the motivating problems as much as possible so that we could precisely formulate and analyze the underlying task. Although abstracting has given us much insight into the problem, the practical application of these ideas requires a move in the opposite direction. That is, studying more constrained models will lead to more practical algorithms.

To recap, our primary contributions are two-fold:

1. Formulate object interaction in terms of dependency structure model selection,
   (a) Analyze the relationship between Bayesian, information theoretic/geometric, and classical methods for statistical dependence estimation,
   (b) Empirical validation on simulated and real interaction data,
2. Formulate matching problem in terms of maximizing statistical dependence,
   (a) Recast previous matching methods in our formulation,
   (b) Prove intractability of exact maximally dependent matching,
   (c) Generalize previous non-overlapping camera matching, and show improved results on simulated and real data.

It is important to remember that our motivating applications are problems in discovery and description, as opposed to simply recognition. We are given observed data and must infer the model structure, which involves deciding the statistical dependency relationships between RVs. Description is more difficult than recognition because the answer consists of more information than simply a class label. In our case, the description is the dependency structure of data.

5.1 Future Work

Estimating statistical dependency structure is not a new idea. We have applied it to different problems and drawn explicit connections to model selection and information theory. We believe that statistical dependence may also be useful in many other problems. Below we discuss some of these avenues for future work.

In the object interaction problem, we have assumed that motion trajectories are given, and remarked how accurate measurements of interacting trajectories are actually difficult to acquire with current tracking technology. This is because interacting objects are often close to each other and may occlude one another. One way to tackle these issues is to retain our approach but start with the actual video data and infer both the trajectories and interactions jointly. We expect that knowing that two objects are interacting should help any tracking system reduce the uncertainty in predicting their states. Perhaps the simplest way to expand our model is to consider the nodes currently in the causal dependency graphs as hidden, and to add observation nodes for the video data. Although the corresponding computations may be more involved with this type of all-encompassing model, the inference should be more accurate.

Another extension of our object interaction model is to allow the interaction state to vary over time. We expect that this occurs naturally, such as when two pedestrians start off moving independently, then walk together, and finally move independently again. One way to capture this type of process is with a hidden Markov model on top of the dependency estimation [79].

For matching, one line of future work involves exploring the relationship between transformations between matched objects and object similarity. These issues have been partially explored in non-matching contexts [87, 62, 81, 61]. The matching problem can be viewed as a search for most similar pairs if we regard the similarity between two observations as a function of the probability of the transformation between them. For example, two observations of the same object under different lighting conditions leads to a color transformation between them. We expect the true matching between observations to correspond to a distribution of transformations that assigns high probability to the matched pairs. Thus, optimizing for the matching may enable us to learn the natural similarity between corresponding observations in two cameras or more generally two different settings.

Bibliography

[1] S. Amari. *Differential-geometrical methods in statistics.* Number 28 in Lecture Notes in Statistics. Springer-Verlag, 1985.

[2] D. Ayers and M. Shah. Monitoring human behavior from video taken in an office environment. *Image and Vision Computing*, 19(1):833–846, 2001.

[3] J. Beirlant, E. J. Dudewicz, L. Gyoerfi, and E. C. Van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–40, 1997.

[4] James O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Number 14 in Statistics. Springer-Verlag, second edition, 1985.

[5] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, 1995.

[6] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, March 2001.

[7] Matthew Brand. Learning concise models of human activity from ambient video. Technical Report 97-25, Mitsubishi Electric Research Labs, 1997.

[8] Hilary Buxton. Generative models for learning and understanding dynamic scene activity. In *1st International Workshop on Generative-Model-Based Vision*, 2002.

[9] C. Castel, L. Chaudron, and C. Tessier. What is going on? a high level interpretation of sequences of images. In *European Conference on Computer Vision*, pages 13–27, 1996.

[10] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

[11] Robert Collins, Alan Lipton, and Takeo Kanade. A system for video surveillance and monitoring. In *American Nuclear Society Eight International Topical Meeting on Robotic and Remote Systems*, 1999.

[12] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[13] Ingemar J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[14] Harald Cramér. *Mathematical Methods of Statistics.* Princeton Univeristy Press, 1946.

[15] S. Dance and T.M. Caelli. On the symbolic interpretation of traffic scenes. In *Asian Conference on Computer Vision*, pages 798–801, 1993.

[16] Frank Dellaert. Addressing the correspondence problem: A Markov Chain Monte Carlo approach. Technical report, Carnegie Mellon University School of Computer Science, 2000.

[17] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[18] W. Feller. *An Introduction to Probability Theory and Its Applications.* John Wiley & Sons, 1968.

[19] J. Fernyhough, A. Cohn, and David Hogg. Constructing qualitative event models automatically from video input. *Image and Vision Computing*, 18:81–103, 2000.

[20] R. B. Fisher. Self-organization of randomly placed sensors. In *European Conference on Computer Vision*, 2002.

[21] David A. Forsyth, John A. Haddon, and Sergey Ioffe. The joy of sampling. *International Journal of Computer Vision*, 41(1/2), 2001.

[22] Michael R. Garey and David S. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness.* W.H. Freeman and Company, 1979.

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

[24] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo In Practice.* Chapman & Hall/CRC, 1996.

[25] S. Gold, C. P. Lu, A. Rangarajan, S. Pappu, and E. Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. In *Neural Information Processing Systems*, 1995.

[26] S.G. Gong, J. Ng, and J. Sherrah. On the semantics of visual behaviour, structured events and trajectories of human action. *Image and Vision Computing*, 20(12):873–888, October 2002.

[27] W. Eric L. Grimson, Lily Lee, Raquel Romano, and Chris Stauffer. Using adaptive tracking to classify and monitor activities in a site. In *Computer Vision and Pattern Recognition*, 1998.

[28] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, August 2000.

[29] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57, 1970.

[30] David Heckerman. A tutorial on learning with Bayesian networks. *Learning in Graphical Models*, pages 301–354, 1998.

[31] F. Heider and M. Simmel. An experimental study of apparent behaviour. *America Journal of Psychology*, 57:243–259, 1944.

[32] Berthold Klaus Paul Horn. *Robot Vision.* MIT Press, 1986.

[33] T. Huang and S. Russell. Object identification in a Bayesian context. In *International Joint Conference on Artificial Intelligence*, 1997.

[34] Alexander T. Ihler, John W. Fisher, and Alan S. WIllsky. Nonparametric hypothesis tests for statistical dependency. *IEEE Transactions on Signal Processing*, 52(8):2234–2249, August 2004.

[35] Stephen S. Intille and Aaron F. Bobick. A framework for recognizing multi-agent action from visual evidence. In *AAAI*, pages 518–525, 1999.

[36] Yuri A. Ivanov, Aaron F. Bobick, Christopher Stauffer, and W. Eric L. Grimson. Visual surveillance of interactions. In *International Workshop on Visual Surveillance*, pages 82–89, 1999.

[37] Omar Javed, Sohaib Khan, Zeeshan Rasheed, and Mubarak Shah. Camera handoff: Tracking in multiple uncalibrated stationary cameras. In *IEEE Workshop on Human Motion*, 2000.

[38] Omar Javed, Zeeshan Rasheed, Khurram Shafique, and Mubarak Shah. Tracking across multiple cameras with disjoint views. In *International Conference on Computer Vision*, 2003.

[39] Harold Jeffreys. *Theory of Probability.* Oxford University Press, 3 edition, 1961.

[40] G. Johansson. Visual perception of biological motion. *Perception and Psychophysics*, 14(2):201–211, 1973.

[41] Neil Johnson, Aphrodite Galata, and David Hogg. The acquisition and use of interaction behaviour models, 1998.

[42] Michael I. Jordan, editor. *Learning in Graphical Models.* MIT Press, 1998.

[43] Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

[44] Vera Kettnaker and Ramin Zabih. Bayesian multi-camera surveillance. In *Computer Vision and Pattern Recognition*, 1999.

[45] Ross Kindermann and J. Laurie Snell. *Markov Random Fields and Their Applications.* American Mathematical Society, 1980.

[46] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[47] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[48] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[49] Solomon Kullback. *Information Theory and Statistics.* Dover Publications, Inc., 1968.

[50] S. Lauritzen. *Lectures on Contingency Tables.* University of Aalborg Press, 1982.

[51] Erik G. Learned-Miller and John W. Fisher III. ICA using spacings estimates of entropy. *Journal of Machine Learning Research*, 4, 2003.

[52] Lily Lee, Raquel Romano, and Gideon Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

[53] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[54] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[55] Dimitrios Makris, Tim Ellis, and James Black. Bridging the gaps between cameras. In *Computer Vision and Pattern Recognition*, 2004.

[56] Richard Mann, Allan Jepson, and Jeffrey Mark Siskind. The computational perception of scene dynamics. *Computer Vision and Image Understanding: CVIU*, 65(2):113–128, 1997.

[57] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, October 2000.

[58] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[59] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44:335–341, 1949.

[60] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. In *Computer Vision and Pattern Recognition*, 2000.

[61] Erik G. Miller. *Learning from One Example in Machine Vision by Sharing Probability Densities.* PhD thesis, Massachusetts Institute of Technology, 2002.

[62] Erik G. Miller, Kinh Tieu, and Chris P. Stauffer. Learning object-independent modes of variation with feature flow fields. Technical Report AIM-2001-021, Massachusetts Institute of Technology Artificial Intelligence Laboratory, 2001.

[63] R. J. Morris and David C. Hogg. Statistical models of object interaction. *International Journal of Computer Vision*, 37(2):209–215, 2000.

[64] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[65] H-H Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing*, 6(2):59–74, 1988.

[66] Nuria M. Oliver, Barbara Rosario, and Alex Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[67] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization.* Dover, 1988.

[68] A. Papoulis. *Probability, Random Variables and Stochastic Processes.* McGraw-Hill, third edition, 1991.

[69] V. Parameswaran and R. Chellappa. Human action-recognition using mutual invariants. *Computer Vision and Image Understanding*, 98(2):294–324, May 2005.

[70] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33, 1962.

[71] Hanna Pasula, Stuart Russell, Michael Ostland, and Ya'acov Ritov. Tracking many objects with many sensors. In *International Joint Conference on Artificial Intelligence*, 1999.

[72] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988.

[73] Judea Pearl. *Causality.* Cambridge University Press, 2000.

[74] Ali Rahimi, Brian Dunagan, and Trevor Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Computer Vision and Pattern Recognition*, 2004.

[75] Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publishing, 1989.

[76] Y. Bar Shalom and T. E. Fortmann. *Tracking and Data Association.* Academic-Press, 1988.

[77] Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27, 1948.

[78] J. A. Simpson and Edmund S. Weiner. *The Oxford English Dictionary.* Oxford University Press, 1989.

[79] Michael Siracusa and John Fisher. Modeling and estimating dynamic dependency structure : Applications to audio-visual speaker labeling. In *Snowbird Learning Workshop*, 2006.

[80] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

[81] Chris Stauffer, Erik G. Miller, and Kinh Tieu. Transform-invariant image decomposition with similarity templates. In *Neural Information Processing Systems*, 2001.

[82] Chris Stauffer and Kinh Tieu. Automated multi-camera planar tracking correspondence modeling. In *Computer Vision and Pattern Recognition*, 2003.

[83] Gilbert Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, 1986.

[84] D'arcy Wentworth Thompson. *On Growth and Form: The Complete Revised Edition.* Dover Publications, Inc., 1992.

[85] Kinh Tieu. Categories of relative motion. In *MIT AI Lab Student Oxygen Workshop*, 2002.

[86] Kinh Tieu, Gerald Dalley, and W. Eric. L. Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *International Conference on Computer Vision*, 2005.

[87] Kinh Tieu and Erik G. Miller. Unsupervised color constancy. In *Neural Information Processing Systems*, 2003.

[88] Shimon Ullman. Analysis of visual motion by biological and computer systems. *IEEE Computer*, 14(8):57–69, 1981.

[89] O. Vasicek. A test for normality based on sample entropy. *Journal of the Royal Statistical Society, Series B*, 38(1), 1976.

[90] Paul Viola and William M. Wells, IIII. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2), 1997.

[91] Xiaogang Wang, Kinh Tieu, and W. Eric L. Grimson. Learning semantic scene models by trajectory analysis. In *European Conference on Computer Vision*, 2006.

[92] David Wolf. Mutual information as a Bayesian measure of independence. Technical Report 94, Los Alamos Unlimited Release, 1994.