Bayesian A* Tree Search with Expected O(N) Convergence Rates for Road Tracking

James M. Coughlan and A. L. Yuille
Smith-Kettlewell Eye Research Institute,
2232 Webster Street,
San Francisco, CA 94115, USA.
Tel. (415) 561-0536. Fax. (415) 561-1610.
Email yuille@attila.ski.org, coughlan@attila.ski.org

November 28, 1998

Abstract *This paper develops a theory for the convergence rates of A* algorithms for Bayesian inference. We are specifically concerned with real-world vision problems, such as road tracking [11], which can be formulated in terms of maximizing a reward function derived using Bayesian probability theory. Such problems are well suited to A* tree search and it can be shown [25] that many algorithms proposed to solve them are special cases, or variants, of A*. Moreover, the Bayesian formulation naturally defines a probability distribution on the ensemble of problem instances (see Pearl Chp 5, [20]), which we call the* **Bayesian Ensemble**. *We analyze the Bayesian ensemble, using techniques from information theory, and mathematically prove expected time convergence rates of algorithms. These rates depend on an "order parameter" which characterizes the difficulty of the problem. In particular, we study: (i) an admissible A* algorithm which uses pruning and (ii) an inadmissible A* algorithm. In both cases we prove expected convergence rates with O(N) node expansions (where N is the problem size) and also expected constant time sorting per node expansion. We also characterize the expected errors as functions of the order parameter. Our proofs break down at critical values of the order parameters but, in related work [27], we prove that the search task becomes impossible by any algorithm at critical values of closely related order parameters. We conclude that A* is a very effective way of solving such problems in the regimes in which they can be solved.*

Keywords: (I) Heuristic Search, (II) A*, (III) Order Parameters and Phase Transitions, (IV) NP-complexity versus Typical Complexity, (V) Bayesian Computer Vision.

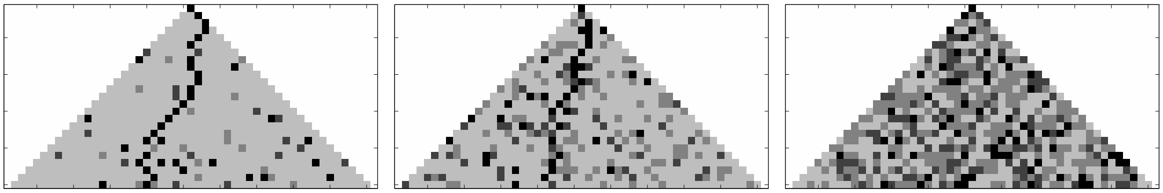

Figure 1: The difficulty of detecting the target path in clutter depends, by our theory [27], on the order parameter K. The larger K the less computation required. Left, an easy detection task with $K = 0.8647$. Middle, a harder detection task with $K = 0.2105$. Right, an impossible task with $K = -0.7272$.

1 Introduction

Heuristic search is a fundamental problem of Artificial Intelligence [20],[24],[22]. The A* algorithm is a particularly important approach to heuristic search which has been elegantly expounded and analyzed by Pearl [20].

More recently, it has become apparent [25] that a class of real world vision problems, formulated as Bayesian inference [17], can be solved using A* algorithms. This class includes tasks such as the detection and tracking of paths in noise/clutter, see figure (1). In particular, it was shown [25] that many of the algorithms used to solve these tasks (see, for example, [19],[2],[9],[11],[10]) could be interpreted as special cases, or variants, of A* algorithms. Incidently, a consequence of applying A* to Bayesian problems is that the prior probabilities, an essential component of the Bayesian approach, can be used to make stronger heuristic predictions than in standard A*, see [11],[28], which can result in improved performance.

An advantage of expressing algorithms in a uniform framework, such as A*, is that it enables us to do theoretical and experimental comparisons between different algorithms to determine which ones are most effective. Moreover, one may hope to identify characteristics of the problem domain which determine the difficulty of the search tasks independently of the algorithm used. If so, then it may be possible to design optimally effective algorithms to solve the problems. These are the issues that we investigate in this paper. (See also, our related work [27]).

Broadly speaking, there are two strategies for evaluating the effectiveness of algorithms. The first is the worst case analysis used in much of computer science [8]. The second involves determining the convergence rates on typical problem situations (i.e. those which typically occur). This form of analysis requires having a probability distribution on the ensemble of problem instances. Karp and Pearl provided a fascinating analysis of binary tree A* search using this approach (see Chp 5 [20],[14]). We argue that this second approach is of more relevance to the problems we are concerned with and so we will study it in our paper. Interestingly, however, there are some recent studies showing that order parameters exist for NP-complete problems and that these problems can be easy to solve for certain values of the order parameters [4],[23]. The connection between this approach and our own is a topic for further research.

We emphasize that the Bayesian formulation of our problems *naturally gives rise to a probability distribution on the ensemble of problem instances*, which we call the *Bayesian*

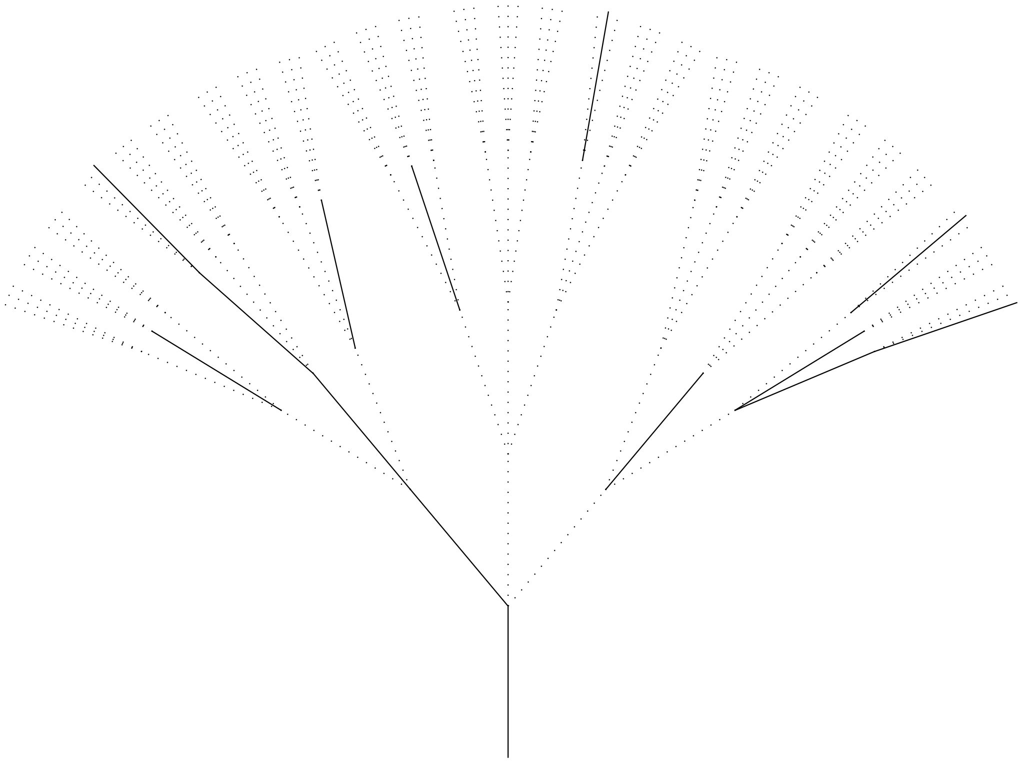

Figure 2: A simulated road tracking problem where dark lines indicate strong edge responses and dashed lines specify weak responses. The branching factor is three. The data was generated by stochastic sampling using a simplified version of the models analyzed in this paper. In this sample there is only one strong candidate for the best path (the continuous dark line) but chance fluctuations have created subpaths in the noise with strong edge responses. The A* algorithm must search this tree starting at the bottom node.

Ensemble. This allows us to build on the foundations established by Karp and Pearl [20] to obtain expected convergence rates. Technically, our proofs involve adapting techniques from information theory, such as Sanov's theorem, which were developed to bound the probability of rare events occurring [6]. (For the road tracking problem, a rare event would be when a subpath in the background noise/clutter has greater reward than a subpath of the true road – i.e. looks more like a road). Our proofs rely on three basic elements: (i) Sanov's theorem which shows that the probability of rare events decreases *exponentially* with the length of the subpath, (ii) An *onion peeling* strategy which allows us to recursively analyze the search tree, and (iii) the use of standard techniques for summing, and bounding, exponential series generated by (i). See figure (2) for an illustration of this search task.

In particular, we formulate the problem of detecting object curves in images to be one of Bayesian inference [17]. This assumes that statistical knowledge is determined for the images and the curves, see sections (2). Such statistical knowledge has often been used in computer vision for determining optimization criteria to be minimized and techniques have been developed to learn it form real data [29]. We want to go one step further and use this statistical knowledge to determine good search strategies. In particular, we can prove that for certain curve and boundary detection algorithms we will, with high probability, obtain expected A* convergence rates by examining a number of nodes which is linear in the size N of the problem. In addition, the expected sort time per node is constant (note that this does *not* necessarily imply that the expected time for the problem is linear in N). Moreover, our analysis helps determine important characteristics of the problem, similar to order parameters in statistical physics, which quantify the difficulty of the problem. These order parameters determine the constants in the convergence rate formulae and also determine the expected errors.

As we will show, our convergence bounds become infinite at certain values of these order parameters. Is this an artifact of our proofs? Or is it a limitation of the A* search strategy? In related work [27], we prove instead that it corresponds to a fundamental difficulty with the problem. As proven in [27] similar order parameters characterize the difficulty of solving the problem *independently* of the algorithm employed. Moreover, at critical values of these order parameters there is a phase transition and the problem becomes insolvable. These fundamental bounds show that our proofs in this paper only break down as we enter the regime where the problem is unsolvable by any algorithm. The A* algorithm remains effective as we approach the critical value of the order parameters although, not surprisingly, the convergence rates get very slow.

The first section (2) of this paper describes the probabilistic formulation of road tracking that we use to prove our results. Section (3) introduces Sanov's theorem and illustrates how it can be applied to bound the probabilities of rare events. (We give a proof of Sanov's Theorem in Appendix 1). In section (4) we prove convergence rates of an admissible A* algorithm which uses pruning. In section (5), we extend our results to the more interesting, and challenging, case of inadmissible heuristics. We conclude by placing this work in a larger context and summarizing recent extensions.

2 Mathematical Formulation of Road Tracking

Tracking curved objects in real images is an important practical problem in computer vision. We consider a specific formulation of the problem of road tracking from aerial images by Geman (D.) and Jedynak [11]. Their approach used a novel active search algorithm to track a road in an aerial photograph with empirical convergence rates of $O(N)$ for roads of length N. Their algorithm is highly effective for this application and is arguably the best currently available. In previous work [25], we showed that Geman and Jedynak's algorithm was a close approximation to A*. Other search algorithms such as Dijktra and Dynamic Programming used in related visual search problems [19], [2],[9], [16]. [10],[5] can be shown to be special cases of A* [25].

Our approach assumes that both the intensity properties and the geometrical shapes of the target path (i.e. the edge contour) can be determined statistically. This path can be considered to be a set of elementary path segments joined together. We first consider the intensity properties along the edge and then the geometric properties.

The image properties of segments lying on the path are assumed to differ, in a statistical sense, from those off the path. More precisely, we can design a filter $\phi(.)$ with output

$\{y_x = \phi(I(x))\}$ for a segment at point x so that:

$$\begin{aligned} P(y_x) = P_{on}(y_x), &\quad if \text{ ``}x\text{''} \; lies \; on \; the \; true \; path \\ P(y_x) = P_{off}(y_x), &\quad if \text{ ``}x\text{''} \; lies \; off \; the \; true \; path. \end{aligned} \tag{1}$$

For example, we can think of the $\{y_x\}$ as being values of the edge strength at point x and P_{on}, P_{off} being the probability distributions of the response of $\phi(.)$ on and off an edge. The set of possible values of the random variable y_x is the *alphabet* with *alphabet size* M. See [11],[5] examples of distributions for P_{on}, P_{off} used in computer vision applications.

We now consider the geometry of the target contour. We require the path to be made up of connected segments $x_1, x_2, \ldots, x_N$. There will be a Markov probability distribution $P_g(x_{i+1}|x_i)$ which specifies prior probabilistic knowledge of the target. It is convenient, in terms of the graph search algorithms we will use, to consider that each point x has a set of Q neighbours. Following terminology from graph theory, we refer to Q as the *branching factor*. We will assume that the distribution P_g depends only on the relative positions of x_{i+1} and x_i. In other words, $P_g(x_{i+1}|x_i) = P_{\Delta g}(x_{i+1} - x_i)$. An important special case is when the probability distribution is uniform for all branches (i.e. $P_{\Delta g}(\Delta x) = U(\Delta x) = 1/Q \; \forall \Delta x$).

By standard Bayesian analysis, the optimal path $X^* = \{x_1^*, \ldots, x_N^*\}$ maximizes the sum of the log posterior:

$$E(X) = \sum_i \log \frac{P_{on}(y_{(x_i)})}{P_{off}(y_{(x_i)})} + \sum_i \log \frac{P_{\Delta g}(x_{i+1} - x_i)}{U(x_{i+1} - x_i)}, \tag{2}$$

where the sum i is taken over all points on the target (which may, or may not, be a fixed number). $U(x_{i+1} - x_i)$ is the uniform distribution and its presence merely changes the log posterior $E(X)$ by a constant value. It is included to make the form of the intensity and geometric terms similar, which simplifies our later analysis.

We will refer to $E(X)$ as the *reward* of the path X which is the sum of the *intensity rewards* $\log \frac{P_{on}(y_{(x_i)})}{P_{off}(y_{(x_i)})}$ and the *geometric rewards* $\log \frac{P_{\Delta g}(x_{i+1}-x_i)}{U(x_{i+1}-x_i)}$.

It is important to emphasize that our results can be extended to higher-order Markov chain models (provided they are shift-invariant). We can, for example, define the x variable to represent spatial orientation *and* position of a small edge segment. This will allow our theory to apply to models, such as snakes, used in recent successful vision applications [2], [11]. (It is straightforward to transform the standard energy function formulation of snakes into a Markov chain by discretizing and replacing the derivatives by differences. The smoothness constraints, such as membranes and thin plate terms, will transform into first and second order Markov chain connections respectively). Recent work by Zhu [33] shows that Markov chain models of this type can be learnt using Minimax Entropy Learning theory from a representative set of examples. Indeed Zhu goes further by demonstrating that other Gestalt grouping laws can be expressed in this framework and learnt from representative data.

Reward functions, such as equation (2), are ideally suited to A* graph/tree search algorithms [20],[22] and we will therefore analyze A* algorithms later in this paper, see section (5). As we will describe, A* searches the nodes – possible branches of the road/snake – which are most promising. The "goodness" $f(n)$ of a node n is $g(n) + h(n)$ where $g(n)$ is the reward to get to the node and $h(n)$ is a heuristic reward to get to the finish from n. The A* algorithm starts at the top of the tree and evaluates the child nodes (i.e. those connected to the top node by a single arc). These child nodes are placed in the *queue*. As the algorithm proceeds it selects the member of the queue with best evaluation, removes it from the queue, expands its children and enters them in the queue.

The evaluation of the nodes is based on the reward to reach it from the top node (i.e. the sum of the log posteriors) and on a heuristic reward based on anticipated future performance. More precisely, a path segment ending at x has a total reward $f(x) = g(x) + h(x)$ (note that the nonoverlapping path requirement implies that x determines a unique path to the initialization point). The choice of heuristic reward $h(x)$ is very important to the algorithm [20]. It can be proven that if $h(x)$ is an upper bound on the reward to get to the end of the path then A* is guaranteed to find the global maximum eventually. An A* algorithm whose heuristic satisfies this bound is called *admissible*. One that does not is called *inadmissible*. The problem is that admissible A* algorithms are guaranteed to find the best result but may do so slowly. By contrast, inadmissible $A*$ algorithms are often faster but may fail in certain cases.

Karp and Pearl [20] provided a theoretical analysis of convergence rates of A* search. They studied a binary tree where the rewards for each arc were 0 or 1 and were specified by a probability p. They then studied the task of finding the minimum cost path. This is an interesting task but it differs from ours in many respects. From our perspective, it resembles the task of finding the best path in the noise/clutter rather than detecting a true target in the presence of noise/clutter.

There are three elements to our proofs. The first is the use of Sanov's theorem to put exponential bounds on the probabilities of rare events – this theorem is described in section (3) and a proof is given in Appendix 1. The second is the onion peeling strategy to recursively explore the search tree, this is described in section (4). The third is the summation of exponential series, generated by Sanov's theorem, which is described in Appendix 2.

3 Sanov's Theorem

This section introduces results from the theory of types [6] which we will use to prove our results. We will be particularly concerned with Sanov's theorem, which we give a proof of in Appendix 1. To motivate this material we will apply it to the problem of determining whether a given set of measurements are more likely to come from a road or non-road but *without* making any geometrical assumptions about the likely shape of the road. The theorem assumes that we have an underlying distribution Q which generates a set of N independent identically distributed (i.i.d.) samples. From each sample set we can determine an empirical histogram, or *type*, see figure (3,4). The law of large numbers states that these empirical histograms (when normalized) must become similar to the distribution Q as $N \mapsto \infty$. Sanov's theorem puts bounds on *how fast* the empirical histograms converge (in probability) to the underlying distribution. Thereby it puts bounds on the probability of rare events.

Recall, see Appendix 1, that Sanov Theorem states:

Sanov's Theorem. *Let $y_1, y_2, ..., y_N$ be i.i.d. from a distribution $Q(y)$ with alphabet size J and E be any closed set of probability distributions. Let $Pr(\vec{\phi} \in E)$ be the probability that*

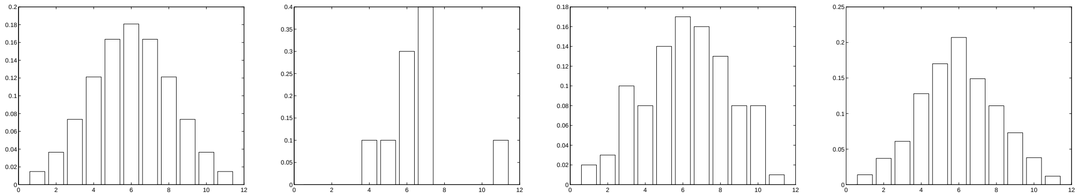

Figure 3: Samples from an underlying distribution. Left to right, the original distribution, followed by histograms, or types, from 10, 100, and 1000 samples from the original. Observe that for small numbers of samples the types tend to differ greatly from the true distribution. But for large N the law of large numbers says that they must converge (with high probability).

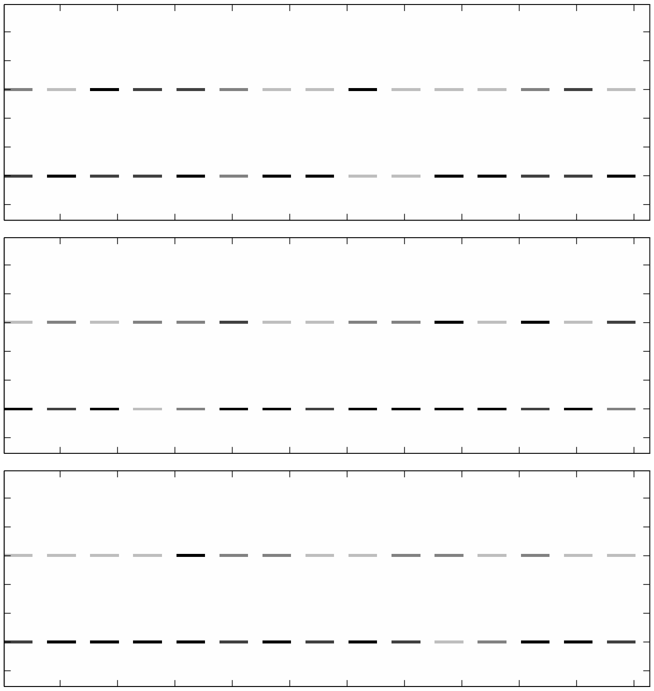

Figure 4: Sequences of edge values rendered using four gray levels ranging from light gray to black. In each pair, one sequence is drawn i.i.d. from $P_{on} = (0.1, 0.1, 0.3, 0.5)$ and the other from $P_{off} = (0.5, 0.3, 0.1, 0.1)$. Although individual edge values are unreliable, taken as a whole it is clear that the top sequences are from P_{off} and the bottom sequences from P_{on}. The Chernoff distance between P_{on} and $P_{off}$is 0.2311 nats.

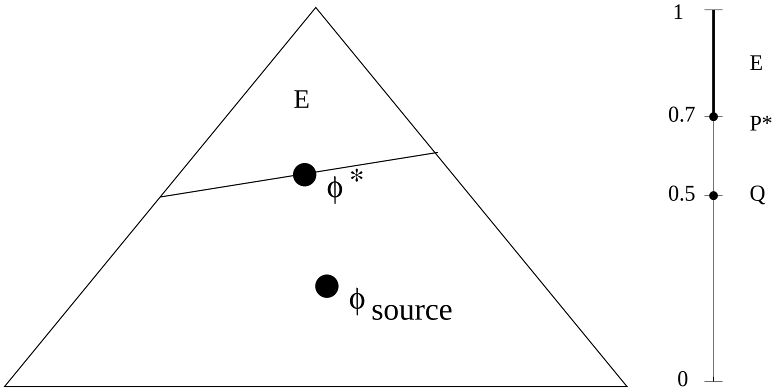

Figure 5: Left, Sanov's theorem. The triangle represents the set of probability distributions. ϕ_{source} is the distribution which generates the samples. Sanov's theorem states that the probability that a type, or empirical distribution, lies within the subset E is chiefly determined by the distribution ϕ^* in E which is closest to ϕ_{source}. Right, Sanov's theorem for the coin tossing experiment. The set of probabilities is one-dimensional and is labelled by the probability $p(head)$ of tossing a head. The unbiased distribution Q is at the centre, with $P(head) = 1/2$, and the closest element of the set E is P^* such that $P^*(head) = 0.7$.

the type of a sample sequence lies in the set E. Then:

$$\frac{2^{-ND(\vec{\phi}^*||Q)}}{(N+1)^J} \leq Pr(\vec{\phi} \in E) \leq (N+1)^J 2^{-ND(\vec{\phi}^*||Q)}, \tag{3}$$

where $\phi^* = \arg\min_{\phi \in E} D(\phi||Q)$ is the distribution in E that is closest to Q in terms of Kullback-Leibler divergence, given by $D(\phi||Q) = \sum_{y=1}^{J} \phi(y) \log(\phi(y)/Q(y))$.

This is illustrated by figure (5). Intuitively, it shows that, when considering the chance of a set of rare events happening, we essentially only have to worry about the "most likely" of the rare events (in the sense of Kullback-Leibler divergence). Most importantly, it tells us that the probability of rare events falls off *exponentially* with the Kullback-Leibler divergence between the rare event (its type) and the true distribution. This exponential fall-off is critical for proving the results in this paper. Note that Sanov' theorem involves an *alphabet factor* $(N+1)^J$. This alphabet factor becomes irrelevant at large N (compared to the exponential term). It does, however, require that the distribution Q is defined on a finite space, or can be well approximated by a quantized distribution on a finite space.

Sanov's theorem can be illustrated by a simple coin tossing example, see figure (5). Suppose we have a fair coin and want to estimate the probability of observing more than 700 heads in 1000 tosses. Then set E is the set of probability distributions for which $P(head) \geq 0.7$ $(P(head) + P(tails) = 1)$. The distribution generating the samples is $Q(head) = Q(tails) = 1/2$ because the coin is fair. The distribution in E closest to Q is $P^*(head) = 0.7, P^*(tails) = 0.3$. We calculate $D(P^*||Q) = 0.119$. Substituting into Sanov's theorem, setting the alphabet size $J = 2$, we calculate that the probability of more than 700 heads in 1000 tosses is less than $2^{-119} \times (1001)^2 \leq 2^{-99}$.

In this paper, we will only be concerned with sets E which involve the rewards of types.

These sets will therefore be defined by linear constraints on the types – in particular, constraints such as $\vec{\phi} \cdot \vec{\alpha} \geq T$, where $\alpha(y) = \log(P_{on}(y)/P_{off}(y))$, $y = 1, ..., J$. (We define $\vec{\phi} \cdot \vec{\alpha} = \sum_{y=1}^{J} \phi(y)\alpha(y))$. This will enable us to derive results which will not be true for arbitrary sets E. We will often, however, be concerned with the probabilities that the rewards of samples from one distribution are greater than those from a second. It is straightforward to generalize Sanov's theorem to deal with such cases.

Theorem 1. *The probabilities that the loglikelihoods of sequence of samples with N elements from off-road or on-road are above, or below, the threshold T are bounded above and below as follows:*

$$(N+1)^{-J} 2^{-ND(\phi_T||P_{on})} \leq Pr\{\vec{\phi}^{on} \cdot \vec{\alpha} < T\} \leq (N+1)^{J} 2^{-ND(\phi_T||P_{on})}, \tag{4}$$

$$(N+1)^{-J} 2^{-ND(\phi_T||P_{off})} \leq Pr\{\vec{\phi}^{off} \cdot \vec{\alpha} > T\} \leq (N+1)^{J} 2^{-ND(\phi_T||P_{off})}, \tag{5}$$

where $\phi_T(y) = P_{on}(y)^{1-\lambda(T)} P_{off}(y)^{\lambda(T)}/Z(T)$, and $\lambda(T) \in [0,1]$ is a scalar which depends on the threshold T, and $Z(T)$ is a normalization factor. The value of $\lambda(T)$ is determined by the constraint $\vec{\phi}_T \cdot \vec{\alpha} = T$.

Proof. *We apply Sanov's theorem setting $E_{on} = \{\vec{\phi}^{on} : \vec{\phi}^{on} \cdot \vec{\alpha} \leq T\}$ and $E_{off} = \{\vec{\phi}^{off} : \vec{\phi}^{off} \cdot \vec{\alpha} \geq T\}$. Determining the closest distribution $\phi_T \in E_{on}$ to P_{on} reduces to constrained minimization using Lagrange multipliers (ν and λ):*

$$\sum_{y=1}^{J} \phi_T(y) \log \frac{\phi_T(y)}{P_{on}(y)} + \nu\{\sum_{y=1}^{J} \phi_T(y) - 1\} + \lambda\{\vec{\phi}_T \cdot \vec{\alpha} - T\}. \tag{6}$$

This can be solved to give $\phi_T(y) = P_{on}^{1-\lambda(T)}(y) P_{off}^{\lambda(T)}(y)/Z(T)$ with $\lambda(T)$ being determined by the constraint $\vec{\phi}_T \cdot \vec{\alpha} = T$ (recalling $\alpha(y) = \log\{P_{on}(y)/P_{off}(y)\}$). A similar argument applies to P_{off} and the same constraint, $\vec{\phi}_T \cdot \vec{\alpha} = T$, applies to both cases. Hence results.

We have not yet specified the threshold T. There are two important natural choices. The first is based on minimizing the *asymptotic error rate* of the classification – the rate of *falsely classifying a sequence of on-road samples as coming from off-road and vice versa* (i.e. we give equal weight to the false positives and false negatives),

Corollary 1. *The asymptotic error rate is minimized by setting $T = 0$. The error rate in this case is determined by the Chernoff information $C(P_{on}, P_{off})$, where the Chernoff information is defined by the Kullback-Leibler divergence to the distribution ϕ_T^c halfway between P_{on} and P_{off}. More precisely, $C(P_{on}, P_{off}) = D(\phi_T^c||P_{on}) = D(\phi_T^c||P_{off})$ for the unique distribution ϕ_T^c, of form $\phi_T(y) = P_{on}(y)^{1-\lambda(T)} P_{off}(y)^{\lambda(T)}/Z(T)$, which satisfies this constraint.*

Proof. *The error rates fall off as $2^{-ND(\phi_T||P_{on})}$ and $2^{-ND(\phi_T||P_{off})}$. $\phi_T(y)$ is of form $P_{on}^{1-\lambda(T)}(y) P_{off}^{\lambda(T)}/Z(T)$ and has only one degree of freedom. As $\lambda(T)$ increases $D(\phi_T||P_{on})$ decreases and $D(\phi_T||P_{off})$ increases. Therefore there is a unique minimum error rate for T^c such that $D(\phi_T^c||P_{on}) = D(\phi_T^c||P_{off})$, which defines the Chernoff information. Observe that $\sum_{y=1}^{J} \phi_T^c(y) \log P_{on}(y)/P_{off}(y) = 0$.*

The second natural choice of T corresponds to estimating the probability that the rewards of sequence of samples from on-road are less than the *expected* rewards for sequence of samples from off-road (or vice versa). This gives:

Corollary 2. *The probability that sequence of samples from on-road have lower rewards than the average reward for off-road sequence of samples is less than* $(N+1)^J 2^{-ND(P_{off}||P_{on})}$ *and greater than* $(N+1)^{-J} 2^{-ND(P_{off}||P_{on})}$.

Proof. *We set the threshold* T *to be the average reward,* $-D(P_{off}||P_{on})$*, of a sequence of samples generated by off-road. The result of Theorem 1 shows that we must set* $\vec{\phi}_T = P_{off}$ *to satisfy the optimization constraint.*

The second case has two sequences of samples as input (one each from on-road and off-road) and the task is to classify them correctly. The best decision rule to to classify the sequence of sample with higher reward to be on-road and the other to be off-road. Therefore for this task we only care about the chances that a sequence of samples from on-road will have lower reward than a sequence of sample from off-road. Our main result is:

Theorem 2. *The probability that a sequence of samples from on-road has lower reward than a sequence of samples from off-road is bounded below by* $(N+1)^{-2J} 2^{-2NB(P_{on},P_{off})}$ *and above by* $(N+1)^{2J} 2^{-2NB(P_{on},P_{off})}$*, where* $B(P_{on},P_{off}) = -\log\{\sum_{y=1}^{J} P_{off}^{1/2}(y) P_{on}^{1/2}(y)\}$*. (*$N$ *is the number of elements in each sequence of sample.)*

Proof. *This is a generalization of Sanov's theorem to the case where we have two probability distributions and two types. We define* $E = \{(\vec{\phi}^{on}, \vec{\phi}^{off}) : \vec{\phi}^{off} \cdot \vec{\alpha} \geq \vec{\phi}^{on} \cdot \vec{\alpha}\}$*. We then apply the same strategy as for the Sanov proof but applied to the product space of the two distributions* P_{on}, P_{off}*. This requires us to minimize:*

$$f(\vec{\phi}^{off}, \vec{\phi}^{on}) = ND(\vec{\phi}^{off}||P_{off}) + ND(\vec{\phi}^{on}||P_{on})$$
$$+\tau_1\{\sum_{y=1}^{J} \phi^{off}(y) - 1\} + \tau_2\{\sum_{y=1}^{J} \phi^{on}(y) - 1\} + \gamma\{\vec{\phi}^{on}\cdot\vec{\alpha} - \vec{\phi}^{off}\cdot\vec{\alpha}\}, \tag{7}$$

where the τ*'s and* γ *are Lagrange multipliers. The function* $f(.,.)$ *is convex in the* $\vec{\phi}$ *and the Lagrange constraints are linear. Therefore there is a unique minimum which occurs at:*

$$\phi^{off*}(y) = \frac{P_{on}^{\gamma}(y) P_{off}^{1-\gamma}(y)}{Z[1-\gamma]}, \quad \phi^{on*}(y) = \frac{P_{on}^{1-\gamma}(y) P_{off}^{\gamma}(y)}{Z[\gamma]}, \tag{8}$$

subject to the constraint $\vec{\phi}^{on}\cdot\vec{\alpha} = \vec{\phi}^{off}\cdot\vec{\alpha}$*. The unique solution occurs when* $\gamma = 1/2$ *(because this implies* $\vec{\phi}^{off*} = \vec{\phi}^{on*}$ *and so the constraints are satisfied.) We define* $\vec{\phi}_{Bh} = \vec{\phi}_{\lambda^{-1}(1/2)} = P_{on}^{1/2} P_{off}^{1/2}/Z[1/2]$ *("Bh" is short for Bhattacharyya). We therefore obtain:*

$$(N+1)^{-2J} 2^{-N\{D(\vec{\phi}_{Bh}||P_{off}) + D(\vec{\phi}_{Bh}||P_{on})\}} \leq Pr\{(\vec{\phi}^{off}, \vec{\phi}^{on}) \in E\}$$
$$\leq (N+1)^{2J} 2^{-N\{D(\vec{\phi}_{Bh}||P_{off}) + D(\vec{\phi}_{Bh}||P_{on})\}}. \tag{9}$$

We define $B(P_{on},P_{off}) = (1/2)\{D(\vec{\phi}_{Bh}||P_{off}) + D(\vec{\phi}_{Bh}||P_{on})\}$*. Substituting in for* $\vec{\phi}_{Bh}$ *from above yields* $B(P_{on},P_{off}) = -\log\{\sum_{y=1}^{J} P_{off}^{1/2}(y) P_{on}^{1/2}(y)\}$*. Hence result.*

This result tells us that the order parameter for the second task is $2B(P_{on},P_{off})$. This is just another measure of the distance between P_{on} and P_{off} and we will refer to it as the

Bhattacharyya distance (because it is identical to the Bhattacharyya bound for Bayes error, see [21]). Once again the problem becomes increasingly hard as the distributions become more similar but there is no critical point and no phase transition.

4 Tree Search: A* and pruning

In this section we consider an algorithm which uses an admissible A* heuristic and a pruning mechanism. (In the subsequent section, we will show that better results can be achieved using an inadmissible heuristic. But the results in this section are easier to prove and more intuitive). The idea is to examine the paths chosen by the A* heuristic. As the length of candidate path reaches an integer multiple of N_0 we prune it based on its intensity reward and its geometric reward evaluated on the previous N_0 segments, which we call a *segment block*. The reasoning is that few false paths will survive this pruning for long but the target path will survive with high probability.

We prune on the intensity by eliminating all paths whose intensity reward, averaged over the last N_0 segments, is below a threshold T (recall that $-D(P_{off}||P_{on}) < T < D(P_{on}||P_{off})$ and we will usually select T to take values close to $D(P_{on}||P_{off})$). In addition, we prune on the geometry by eliminating all paths whose geometric rewards, averaged over the last N_0 segments, are below $\hat{T}$ (where $-D(U||P_{\Delta g}) < \hat{T} < D(P_{\Delta g}||U)$ with $\hat{T}$ typically being close to $D(P_{\Delta g}||U)$). More precisely, we discard a path provided (for any integer i):

$$\frac{1}{N_0}\sum_{i=zN_0}^{(z+1)N_0-1}\log\frac{P_{on}(y_i)}{P_{off}(y_i)} < T, \ \textit{or} \ \frac{1}{N_0}\sum_{i=zN_0}^{(z+1)N_0-1}\log\frac{P_{\Delta g}(\Delta x_i)}{U(\Delta x_i)} < \hat{T}. \tag{10}$$

There are two important issues to address: (i) With what probability will the algorithm converge?, (ii) How long will we expect it take to converge? The next two subsections put bounds on these issues.

4.1 Probability of Convergence

When will the algorithm converge to the target? The admissible heuristic means that the A* algorithm will converge to the path with greatest reward that survives pruning. There are therefore two types of error to consider : (i) a false path has better reward than the true path, and (ii) the true path gets eliminated by the pruning.

We analyzed the first kind of errors in our related paper [27] where we put bounds on these errors in terms of the order parameter $K = 2B(P_{on}||P_{off}) + 2B(P_{\Delta g}||U) - \log Q$. Essentially the expected size of the error (measured by the number of false segments on the path of highest reward) decreases exponentially with $K > 0$. As $K \mapsto 0$ the error bounds we obtain become infinite and at $K = 0$ there is a phase transition to a regime ($K < 0$) where the target is essentially undetectable (because, with high probability) there are many completely false paths which have higher rewards than the true path).

To quantify the second type of error, we calculate the probability that the target (true) path survives the pruning. This gives a lower bound on the probability of convergence[1].

[1] An upper bound on the probability of failure is a lower bound on the probability of success.

We choose T large and write the fall-off factors as $D(P_T||P_{on}) = \epsilon_1(T)$, $D(P_T||P_{off}) = D(P_{on}||P_{off}) - \epsilon_2(T)$ where $\epsilon_1(T), \epsilon_2(T)$ are positive and $(\epsilon_1(T), \epsilon_2(T)) \mapsto (0,0)$ as $T \mapsto D(P_{on}||P_{off})$. Similarly, we choose $\hat{T}$ to be large and obtain fall-off factors $D(P_{\hat{T}}||P_{\Delta g}) = \hat{\epsilon}_1(\hat{T})$, $D(P_{\hat{T}}||U) = D(P_{\Delta g}||U) - \hat{\epsilon}_2(\hat{T})$.

The pruning rules removes path segments for which the intensity reward r_I or the geometric reward r_g fails the pruning test. The probability of failure by removing a block segment of the true path, with rewards r^t_I, r^t_g, is $Pr(r^t_I < T \;\; or \;\; r^t_g < \hat{T}) \leq Pr(r^t_I < T) + Pr(r^t_g < \hat{T}) \leq (N_0+1)^M 2^{-N_0 \epsilon_1(T)} + (N_0+1)^Q 2^{-N_0 \hat{\epsilon}_1(\hat{T})}$, where we have used Theorem 1 to put bounds on the probabilities. The probability of pruning out any N_0 segments of the true path can therefore be made arbitrarily small by choosing $T, \hat{T}$ so as to make $N_0 \epsilon_1$ and $N_0 \hat{\epsilon}_1$ large.

It should be emphasized that the algorithm will not necessarily converge to the exact target path. The admissible nature of the heuristic means that the algorithm will converge to the path with highest reward which has survived the pruning. It is highly probable that this path is close to the target path and results reported in [27] enable us to quantify this claim.

4.2 Bounding the Number of False Paths

Suppose we face a Q-nary tree. We can order the false paths by the stage at which they diverge from the target (true) path, see figure (6). For example, at the first branch point the target path lies on only one of the Q branches and there are $Q-1$ false branches which generate the first set of false paths F_1. Now consider all the $Q-1$ false branches at the second target branch, these generate set F_2. As we follow along the true path we keep generating these false sets F_i. The set of all paths is therefore the target path plus the union of the F_i $(i = 1, \ldots, N)$. To determine convergence rates we must bound the amount of time we spend searching the F_i. If the expected time to search each F_i is constant then searching for the target path will at most take $constant \cdot N$ steps.

A key concept here is the onion-like structure of the tree representation, see figure (6). This structure allows us to classify all paths in terms of sets $F_1, F_2, F_3, ...$ which depend on where they branch off from the true path. Paths which are always bad (i.e. completely false) correspond to F_1. Paths which are good for one segment, and then go bad, form F_2 and so on. Our previous results have compared the properties of paths in F_1 to those of the true path. To understand the probabilities of paths in F_2 relative to the true path, we simply have to peel off the first layer of the onion (i.e. remove the first arc of the true path) and the comparison of the rest of the true path to F_2 reduces to our previous result for F_1. Thus our results for F_1 can be readily adapted to $F_2, F_3, ...$ Observe that paths in F_i share the first $(i-1)$ arcs with the true path, by definition, and hence have the same partial rewards for these arcs. Therefore we often only need to compare the rewards for the remaining arcs. (Variants of this argument will be used throughout the paper.)

Consider the set F_i of false paths which leave the true path at stage i. We will apply our analysis to block segments of F_i which are completely off the true path. If $(i-1)$ is an integer multiple of N_0 then all block segments of F_i will satisfy this condition. Otherwise, we will start our analysis at the next block and make the worse case assumption that all path segments until this next block will be searched. Since the distance to the next block is

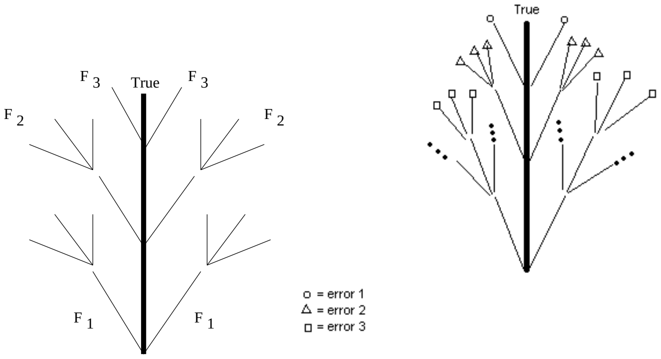

Figure 6: Left: We can divide the set of paths up into N subsets $F_1, ..., F_N$ as shown here. Paths in F_1 are completely off-road. Paths in F_2 have one on-road segment and so on. Intuitively, we can think of this as an onion where we peel off paths stage by stage. Right: When paths leave the true path they make errors which we characterize by the number of false arcs. For example, a path in F_1 has error N, a path in F_i has error $N+1-i$.

at most $N_0 - 1$, this gives a maximum number of Q^{N_0-1} starting blocks for any branch of F_i. Each F_i also has $Q-1$ branches and so this gives a generous upper bound of $(Q-1)Q^{N_0-1}$ starting blocks for each F_i.

For each starting block, we wish to compute (or bound) the expected number of blocks that are explored thereafter. This requires computing the *fertility* of a block, the average number of paths in the block that survive pruning. Provided the fertility is smaller than one, we can then apply results from the theory of branching processes to determine the expected number of blocks searched in F_i.

The fertility q is the number of paths that survive the geometric pruning times the probability that each survives the intensity pruning. This can be bounded (using Theorem 1) by $q \leq \hat{q}$ where:

$$\begin{aligned}\hat{q} &= Q^{N_0}(N_0+1)^Q 2^{-N_0\{D(P_{\Delta g}||U)-\hat{\epsilon}_2(\hat{T})\}}(N_0+1)^M 2^{-N_0\{D(P_{on}||P_{off})-\epsilon_2(T)\}} \\ &= (N_0+1)^{Q+M} 2^{-N_0\{D(P_{on}||P_{off})-H(P_{\Delta g})-\epsilon_2(T)-\hat{\epsilon}_2(\hat{T})\}}, \qquad (11)\end{aligned}$$

where we used the fact that $D(P_{\Delta g}||U) = \log Q - H(P_{\Delta g})$.

Observe that the condition $\hat{q} < 1$ can be satisfied provided $D(P_{on}||P_{off}) - H(P_{\Delta g}) > 0$. This condition is intuitive, it requires that the edge detector information, quantified by $D(P_{on}||P_{off})$, must be greater than the uncertainty in the geometry measured by $H(P_{\Delta g})$. In other words, the better the edge detector and the more predictable the path geometry then the smaller $\hat{q}$ will be.

We now apply the theory of branching processes to determine the expected number of blocks explored from a starting block in F_i , $\sum_{z=0}^{\infty} \hat{q}^z = 1/(1-\hat{q})$. The number of branches of F_i is $(Q-1)$, the total number of segments explored per block is at most Q^{N_0}, and we explore at most Q^{N_0-1} segments before reaching the first block. The total number of F_i is N. Therefore the expected total number of segments wastefully explored is at most $N(Q-1)\frac{1}{1-\hat{q}}Q^{2N_0-1}$. We summarize this result in a theorem:

Theorem 3. *Provided $\hat{q} = (N_0 + 1)^{Q+M} 2^{-N_0 K} < 1$, where the order parameter $K = D(P_{on}||P_{off}) - H(P_{\Delta g}) - \epsilon_2(T) - \hat{\epsilon}_2(\hat{T})$, then the expected number of false segments explored is at most $N(Q-1)\frac{1}{1-\hat{q}} Q^{2N_0 - 1}$.*

Comment The requirement that $\hat{q} < 1$ is chiefly determined by the *order parameter* $K = D(P_{on}||P_{off}) - H(P_{\Delta g}) - \epsilon_2(T) - \hat{\epsilon}_2(\hat{T})$. Our convergence proof requires that $K > 0$ and will break down if $K < 0$. Is this a limitation of our proof? Or does it correspond to a fundamental difficulty in solving this tracking problem?

In more recent work [27] we extend the concept of order parameters and show that they characterize the difficulty of visual search problem *independently of the algorithm.* In other words, as $K \mapsto 0$ the problem becomes impossible to solve by any algorithm. There will be too many false paths which have better rewards than the target path. As $K \mapsto 0$ there is a phase transition in the ease of solving the problem (see Karp and Pearl [20],[14] for an earlier example of a phase transition of this type).

5 Tree Search: A* and Inadmissible Heuristics.

We now consider the more important case of inadmissible heuristics. The convergence results for these cases are harder to prove than those in the previous section. But the convergence rates are better (e.g. smaller convergence factors).

Our main result of this section is to prove convergence of A* algorithms with inadmissible heuristics. We prove that convergence is achieved with O(N) expected nodes opened and we put bounds on the expected errors of the solutions. We also prove that the expected sorting costs per node explored are constant (i.e. independent of N).

5.1 A* Convergence for the Bhattacharyya Heuristic

We now want to consider a traditional A* search strategy using a heuristic function but no pruning. In this section, we will formulate the problem for any heuristic and then obtain bounds for a special case, which we call the *Bhattacharyya heuristic* (again because it is directly related to the Bhattacharyya bound). In the following section, we will generalize our results of other inadmissible heuristics.

For a node W_M, at distance M from the start, we let $g(W_M)$ be the measured reward and $h(W_M)$ is the heuristic function. The A* algorithm proceeds by searching the node in the queue for which the combined reward $f(W_M) = g(W_M) + h(W_M)$ is greatest. How many nodes (or arcs) do we expect to search by this strategy? And what are the expected errors in our solutions?

The reward to reach W_M is just the reward of the log-likelihood data and prior terms along the path from the start to W_M. We define the *heuristic reward* $h(W_M) = (N - M)(H_L + H_P)$ where H_L and H_P are constants (H_L and H_P are heuristics for the likelihood and the prior respectively). As we will show, there are optimal values for H_L and H_P to take and convergence will break down for H_L and H_P outside a specific regime.

Observe that a path segment will be visited only if the reward to get to it (including its heuristic reward) is sufficiently high. More precisely, *if a segment n of a false path is searched then this implies that its reward is better than the reward of at least one point on the target* *path.* This is because the A* algorithm always maintains a queue of nodes to explore and searches the node segment with highest reward. The algorithm is initialized at the start of the target path and so an element of the target path will always lie in the queue of nodes that A* considers searching. Hence a node will never be explored if its reward is lower than all the rewards on the target path segments.

Since the length of all possible paths is constant we can ignore the constant factor $N(H_L + H_P)$ and the heuristic will then merely penalize path segments which have been tested. Then a false path of length n and a true path of length m will have effective rewards denoted by the random variables $S_{off}(n)$ and $S_{on}(m)$:

$$\begin{aligned} S_{off}(n) &= \sum_{i=1}^{n} \{\log \frac{P_{on}(y_{x_i})}{P_{off}(y_{x_i})} - H_L\}_{off} + \sum_{i=1}^{n} \{\log \frac{P_{\Delta G}(x_{i+1} - x_i)}{U(x_{i+1} - x_i)} - H_P\}_{off}, \\ S_{on}(m) &= \sum_{i=1}^{m} \{\log \frac{P_{on}(y_{x_i})}{P_{off}(y_{x_i})} - H_L\}_{on} + \sum_{i=1}^{m} \{\log \frac{P_{\Delta G}(x_{i+1} - x_i)}{U(x_{i+1} - x_i)} - H_P\}_{on}, \end{aligned} \tag{12}$$

where the subscripts off and on are used to denote false and true paths respectively (paths with a mixture of true and false segments will be dealt with later).

We now define types $\vec{\phi}^{off}, \vec{\psi}^{off}, \vec{\phi}^{on}, \vec{\psi}^{on}$ for false and true road samples with $\vec{\phi}$ corresponding to the data and $\vec{\psi}$ to the prior. These types are normalized so that their components sum to 1, i.e. $\sum_{\mu=1}^{M} \phi_\mu = 1$, $\sum_{\nu=1}^{Q} \psi_\nu = 1$. The types will be computed for samples of variables lengths n, m. These lengths will be clear from the context so we will not label them explicitly (i.e. we will not use notation like $\vec{\phi}_n$ to denote types taken from n samples).

Therefore we express the rewards of two sequences $S_{off}(n)$ and $S_{on}(m)$ by:

$$\begin{aligned} S_{off}(n) &= n\{\vec{\phi}^{off} \cdot \vec{\alpha} - H_L\} + n\{\vec{\psi}^{off} \cdot \vec{\beta} - H_P\}, \\ S_{on}(m) &= m\{\vec{\phi}^{on} \cdot \vec{\alpha} - H_L\} + n\{\vec{\psi}^{on} \cdot \vec{\beta} - H_P\}, \end{aligned} \tag{13}$$

where $\alpha(y) = \log(P_{on}(y)/P_{off}(y))$ and $\beta(\delta x) = \log(P_{\Delta g}(\delta x)/U(\delta x))$.

Recall that if a segment n of a false path is searched then its reward must be better than at least one point on the target path. This means that we should consider $Pr\{\exists m : S_{off}(n) \geq S_{on}(m)\}$. This, however, is hard to compute so we bound it above by $\sum_{m=0}^{\infty} Pr\{S_{off}(n) \geq S_{on}(m)\}$ (using Boole's inequality).

Our first result is Theorem 4, which is proven using Sanov's theorem (including the use of constrained optimization to find the fall-off coefficients) and results for the sums of exponential series. The main point of this result is to show that the chance of an off-road path having greater reward than *any* true road path falls off exponentially with the length of the off-road path.

We first define two *sub-order parameters* $\Psi_1 = D(\vec{\phi}_{Bh}||P_{off}) + D(\vec{\psi}_{Bh}||U)$ and $\Psi_2 = D(\vec{\phi}_{Bh}||P_{on}) + D(\vec{\psi}_{Bh}||P_{\Delta G})$. These parameters will determine the convergence and error rates of the algorithm by means of the two functions:

$$C_1(\Psi) = \{\frac{1}{1 - 2^{-\{\Psi - \epsilon\}}} + \Xi(\epsilon, \Psi)\}\ ,\ C_2(\Psi) = \{\frac{e^{-\{\Psi - \epsilon\}}}{(1 - e^{-\{\Psi - \epsilon\}})^2} + \hat{\Xi}(\epsilon, \Psi)\}, \tag{14}$$

where $\Xi, \hat{\Xi}$ are defined in Appendix 2. The order parameter for the problem is $K = \Psi_1 + \Psi_2 - \log Q$ which, as was shown in [27], is the quantity which depends whether a solution to the problem can be found *by any algorithm.*

Theorem 4. *The A^* algorithm, using the Bhattacharyya heuristic $H^*_L = \vec{\phi}_{Bh} \cdot \vec{\alpha}$ and $H^*_P = \vec{\psi}_{Bh} \cdot \vec{\beta}$, gives:*

$$Pr\{S_{off}(n) \geq S_{on}(m)\} \leq \{(n+1)(m+1)\}^{2J+2Q} 2^{-(n\Psi_1 - m\Psi_2)}. \tag{15}$$

Moreover, the probability of a particular false path segment being searched falls off, to first order in n, as $C_1(\Psi_2) 2^{-n\Psi_1}$ where n is the number of segments by which this path segment diverges from the target path.

Proof. *This first part of the proof is again a generalization of Sanov applied to product distributions, see Theorem 2. The new twist is that we have different length factors n and m and the heuristics. But for the Bhattacharyya heuristics this will make no difference. (We deal with the more general heuristics later in subsection (5.2). Define:*

$$E = \{(\vec{\phi}^{off}, \vec{\psi}^{off}, \vec{\phi}^{on}, \vec{\psi}^{on}) : n\{\vec{\phi}^{off} \cdot \vec{\alpha} - H^*_L + \vec{\psi}^{off} \cdot \vec{\beta} - H^*_p\} \geq m\{\vec{\phi}^{on} \cdot \vec{\alpha} - H^*_L + \vec{\psi}^{on} \cdot \vec{\beta} - H^*_p\}\}. \tag{16}$$

Applying the strategy from Theorem 2, we must minimize:

$$\begin{aligned} f(\vec{\phi}^{off}, \vec{\psi}^{off}, \vec{\phi}^{on}, \vec{\psi}^{on}) &= nD(\vec{\phi}^{off}||P_{off}) + nD(\vec{\psi}^{off}||U) + mD(\vec{\phi}^{on}||P_{on}) + mD(\vec{\psi}^{on}||P_{\Delta G}) \\ &+\tau_1\{\sum \vec{\phi}^{off} - 1\} + \tau_2\{\sum \vec{\psi}^{off} - 1\} + \tau_3\{\sum \vec{\phi}^{on} - 1\} + \tau_4\{\sum \vec{\psi}^{on} - 1\} \\ &+\gamma\{m\{\vec{\phi}^{on} \cdot \vec{\alpha} - H^*_L + \vec{\psi}^{on} \cdot \vec{\beta} - H^*_p\} - n\{\vec{\phi}^{off} \cdot \vec{\alpha} - H^*_L + \vec{\psi}^{off} \cdot \vec{\beta} - H^*_p\}\}, \end{aligned} \tag{17}$$

where the τ's and γ are Lagrange multipliers. As before, we know that this function $f(.,.,.,.)$ is convex so there is a unique minimum. Observe that $f(....)$ consists of four terms of form $nD(\vec{\phi}^{off}||P_{off}) + \tau_1\{\sum \vec{\phi}^{off}\} - n\gamma\vec{\phi}^{off} \cdot \vec{\alpha}$ which are coupled by shared constants. These terms can be minimized separately to give:

$$\vec{\phi}^{off*} = \frac{P^{\gamma}_{on} P^{1-\gamma}_{off}}{Z[1-\gamma]}, \quad \vec{\phi}^{on*} = \frac{P^{1-\gamma}_{on} P^{\gamma}_{off}}{Z[\gamma]}, \quad \vec{\psi}^{off*} = \frac{P^{\gamma}_{\Delta G} U^{1-\gamma}}{Z_2[1-\gamma]}, \quad \vec{\psi}^{on*} = \frac{P^{1-\gamma}_{\Delta G} U^{\gamma}}{Z_2[\gamma]}, \tag{18}$$

subject to the constraint given by equation (16).

As before, we see that the unique solution occurs when $\gamma = 1/2$. In this case:

$$\vec{\phi}^{off*} \cdot \vec{\alpha} = H^*_L = \vec{\phi}^{on*} \cdot \vec{\alpha}, \quad \vec{\psi}^{off*} \cdot \vec{\beta} = H^*_P = \vec{\psi}^{on*} \cdot \vec{\beta}. \tag{19}$$

The solution occurs at $\vec{\phi}_{Bh}, \vec{\psi}_{Bh}$ ($\vec{\phi}_{\lambda^{-1}(1/2)}$ and $\vec{\psi}_{\mu^{-1}(1/2)}$). Hence the first result.

We must now sum over m to obtain the bound that $P\{\exists m : S_{off}(n) \geq S_{on}(m)\}$. For large m, the alphabet terms are unimportant and we just need to sum the geometric series. However, we must add extra terms $\Xi(\epsilon, \Psi_2)$ to correct for the alphabet factors for small m, see Appendix 2 for details. Hence

$$Pr\{\exists m \ : \ S_{off}(n) \geq S_{on}(m)\} \leq (n+1)^{2J+2Q} C_1(\Psi_2) 2^{-n\Psi_1}. \tag{20}$$

We can now state our main result about the convergence of A* using the Bhattacharyya heuristic. Our result, Theorem 5, builds on Theorem 4 by adding the onion peeling argument combined with the summation of exponential series.

Theorem 5. *Provided $\Psi_1 > \log Q$, the expected number of searches is $O(N)$ in the size of the problem and is bounded above by $C_1(\Psi_2)C_1(\Psi_1 - \log Q)N$. Moreover, the expected error in convergence is bounded above by $C_1(\Psi_2)C_2(\Psi_1 - \log Q)$, which is small, independent of the size N of the problem, and decays exponentially with $\Psi_1 - \log N$. The order parameter $K = \Psi_1 + \Psi_2 - \log Q$.*

Proof. *We use the onion peeling strategy to express the expectation in terms of the expected number of nodes searched in $F_1, F_2, F_3 ..., F_N$. By the structure of our problem the expectations will be bounded by the same number for all F_i. Therefore the bound is linear provided the expectation for F_1 is finite. More precisely, we get $\sum_{i=1}^{N}\{1 + |F_i|\}$, where $|F_i|$ is the cardinality of F_i.*

Theorem 4 gives us a bound that a specific path of length n in F_1 will have higher reward than any subpath of the true path (a subpath must start at the beginning of the target path). We determine that the expected number of paths of length n, with rewards higher than any subpath of the target path, is bounded above by $C_1(\Psi_2)(n+1)^{2J+2Q}Q^n 2^{-n\Psi_1}$, see equation (20), where $C_1(\Psi)$ is specified by equation (14). This can be summed over n again taking care with the alphabet factors, see Appendix 2 to obtain $C_1(\Psi_2)\{\frac{1}{1-2^{-(\Psi_1-\log Q)+\epsilon}} + \Xi(\epsilon, (\Psi_1 - \log Q))\} = C_1(\Psi_1 - \log Q)C_1(\Psi_2)$. This can always be summed provided $\Psi_1 > \log Q$. Our first result follows.

To put bounds on the expected errors of the algorithm we measure the error in terms of the expected number of off-road arcs. We use the onion peeling strategy again and consider the probability $Pr(n)$ that A will explore a path in F_{N+1-n} to the end, for any n, instead of proceeding along the true path. If this happens we will get an error of size n. The expected error can then be bounded above by $\sum_{n=0}^{\infty} Pr(n)n$.*

We want to put an upper bound on $Pr(n)$. Observe that a path in F_{N+1-n} will be followed to the end only if its reward is greater than the heuristic reward along the true path, or the reward of one arc of the true path plus the heuristic reward for the remainder, or the reward for two true arcs plus the heuristic reward for the rest, and so on. We can apply Sanov to get probability bounds for these by using the constraints $n\{\vec{\phi}^{off}\cdot\vec{\alpha}+\vec{\psi}^{off}\cdot\vec{\beta}\} \geq m\{\vec{\phi}^{on}\cdot\vec{\alpha}+\vec{\psi}^{on}\cdot\vec{\beta}\} + (n-m)\{H_L^* + H_P^*\}$, where $m = 0, ..., n$ is the number of arcs of the true path that are explored. These constraints, of course, are the same constraints $n\{\vec{\phi}^{off}\cdot\vec{\alpha} + \vec{\psi}^{off}\cdot\vec{\beta} - H_L^* - H_P^*\} \geq m\{\vec{\phi}^{on}\cdot\vec{\alpha} + \vec{\psi}^{on}\cdot\vec{\beta} - H_L^* - H_P^*\}$ which we used in Theorem 4 above. Therefore, by Boole's inequality,

$$Pr(n) \leq Q^n \sum_{m=0}^{\infty}\{(n+1)(m+1)\}^{2J+2Q} \times 2^{-\{n\Psi_1 + m\Psi_2\}}. \tag{21}$$

As before, we can sum the series with respect to m, see Appendix 2, to obtain:

$$Pr(n) \leq C_1(\Psi_2)(n+1)^{2J+2Q}2^{-n\{\Psi_1 - \log Q\}}. \tag{22}$$

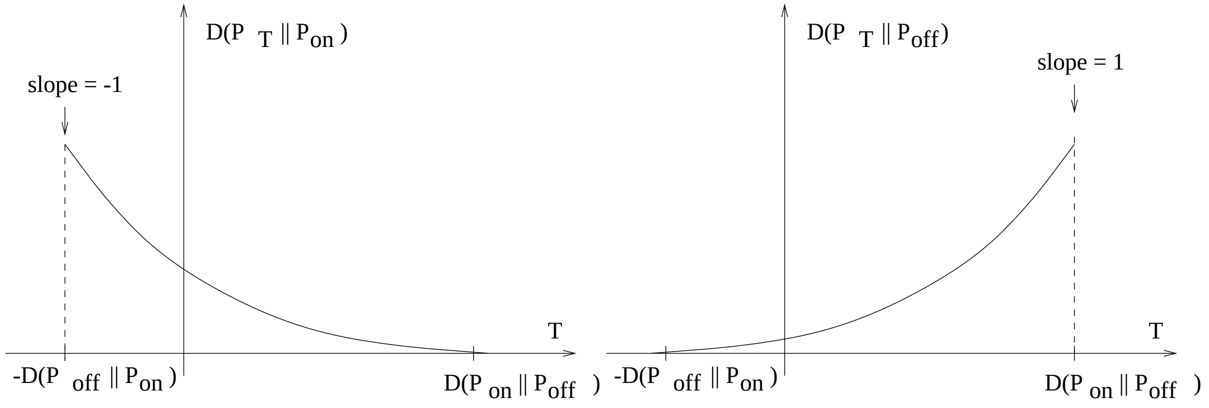

Figure 7: Left, $D(\vec{\phi}_T||P_{on})$ is a convex function of T with its minimum at $T = D(P_{on}||P_{off})$. Right, similarly $D(P_T||P_{off})$ in convex with minimum at $T = -D(P_{off}||P_{on})$.

The expected error is then bounded above by $\sum_{n=1}^{\infty} nPr(n)$. *The dominant, exponential terms, can be summed as before (see Appendix 2) yielding:*

$$< Error > \leq C_2(\Psi_1 - \log Q)C_1(\Psi_2). \tag{23}$$

5.2 Alternative Heuristics

Our results in the previous section used the Bhattacharyya heuristics H_L^*, H_P^*. These choices of heuristic were special in that they enabled us to put bounds on the probability of $Pr\{S_{off}(n) \geq S_{on}(m)\}$ with fall-off factors which depend only on $D(\vec{\phi}_{Bh}||P_{on})$ and $D(\vec{\psi}_{Bh}||P_{\Delta G})$. But these results leave several unanswered questions. Are these specific heuristics optimal in some sense? Are our results stable to small changes in the heuristic values? This section answers these questions by obtaining convergence results for other values of the heuristics. These new results show that the Bhattacharyya heuristics lead to faster convergence rates.

These proofs are complicated because the sets E corresponding to the rare events, and the exponential fall off rates, depend nontrivially on the on- and off- path lengths m and n. They requires us to first prove a convexity result concerning how the fall-off factors, such as $D(\vec{\phi}_T||P_{on})$, vary with the threshold T. This result, Theorem 6, is illustrated by figure (7).

Theorem 6. *Let* $\vec{\phi}_T(y) = P_{on}^{1-\lambda(T)}(y)P_{off}^{\lambda(T)}(y)/Z(T)$, *then* $D(\vec{\phi}_T||P_{on})$ *and* $D(P_T||P_{off})$ *are convex functions of* T *which attain minima of zero at* $T = D(P_{on}||P_{off})$ *and* $T = -D(P_{off}||P_{on})$ *respectively, see figure (7). Moreover,* $D(\vec{\phi}_T||P_{on}) = D(\vec{\phi}_T||P_{off}) - T$.

Proof. *The statement* $D(\vec{\phi}_T||P_{on}) = D(\vec{\phi}_T||P_{off}) - T$ *follows from the identity:*

$$\sum_y \vec{\phi}_T(y) \log\{\frac{\vec{\phi}_T(y)}{P_{on}(y)}\} = \sum_y \vec{\phi}_T(y) \log\{\frac{\vec{\phi}_T(y)}{P_{off}(y)} \frac{P_{off}(y)}{P_{on}(y)}\}. \tag{24}$$

By differentiating equation (24) we observe that the equations (28) are consistent. It therefore is sufficient to prove the first equation.

Differentiating $D(\vec{\phi}_T||P_{on}) = \sum_y \vec{\phi}_T(y) \log \frac{\vec{\phi}_T(y)}{P_{on}(y)}$ yields:

$$\frac{d}{dT}D(\vec{\phi}_T||P_{on}) = \sum_y \frac{d\vec{\phi}_T(y)}{dT}\log\frac{\vec{\phi}_T(y)}{P_{on}(y)} + \sum_y \frac{d\vec{\phi}_T(y)}{dT} = \sum_y \frac{d\vec{\phi}_T(y)}{dT}\log\frac{\vec{\phi}_T(y)}{P_{on}(y)}, \tag{25}$$

because $\sum_y \frac{d\vec{\phi}_T(y)}{dT} = d/dT \sum_y \vec{\phi}_T(y) = 0$. Using $\vec{\phi}_T(y) = P_{on}^{1-\lambda(T)}(y)P_{off}^{\lambda(T)}(y)/Z(T)$ we re-express this as:

$$\frac{d}{dT}D(\vec{\phi}_T||P_{on}) = \sum_y \frac{d\vec{\phi}_T(y)}{dT}\log\frac{P_{off}^{\lambda(T)}(y)}{P_{on}^{\lambda(T)}(y)Z(T)} = -\lambda(T)\sum_y \frac{d\vec{\phi}_T(y)}{dT}\log\frac{P_{on}(y)}{P_{off}(y)}. \tag{26}$$

The Lagrange term in equation (6) implies $\sum_y \vec{\phi}_T(y)\log\frac{P_{on}(y)}{P_{off}(y)} = T$ and differentiating yields:

$$\sum_y \frac{d\vec{\phi}_T(y)}{dT}\log\frac{P_{on}(y)}{P_{off}(y)} = 1. \tag{27}$$

Combining equations (26) and (27) gives the result:

$$\frac{d}{dT}D(\vec{\phi}_T||P_{on}) = -\lambda(T), \quad \frac{d}{dT}D(\vec{\phi}_T||P_{off}) = 1 - \lambda(T), \tag{28}$$

We can solve explicitly for $\lambda(D(P_{on}||P_{off})) = 0$ and $\lambda(-D(P_{off}||P_{on})) = 1$. It is clear that as the threshold T decreases then $\lambda(T)$ decreases because $\vec{\phi}_T$ becomes closer to P_{off}. Hence $d\lambda/dT < 0$. The result follows.

Armed with this theorem, we now proceed to prove results about convergence rates. We first define a function $\mu(T)$ which is the analogue of $\lambda(T)$ for $P_{\Delta G}$ and U. Observe that there is an ambiguity in H_L and H_P because only their sum, $H_L + H_P$, appears in the rewards. To remove this ambiguity we impose the constraint that $\lambda(H_L) = \mu(H_P)$. We then define $\hat{H}_L, \hat{H}_P$ by the conditions $\lambda(\hat{H}_L) + \lambda(H_L) = 1$ and $\mu(\hat{H}_P) + \mu(H_P) = 1$.

We start by proving an analogue of the first part of Theorem 4. This shows that the chance of an off-road path of length n having greater reward than a true road path of length m falls off exponentially with a factor $g(m;n)$. Unfortunately this factor is no longer linear in m and n as it was for the Bhattacharyya heuristic (this linearity enables us to sum the resulting series easily). Instead we need to bound $g(m;n)$ below by a function of form $c_1 m + c_2 n$ (for some constants c_1, c_2). This requires the use of Theorem 6 and an analysis of how $g(m;n)$ varies with n, m, see figure (8).

Theorem 7.

$$Pr\{S_{off}(n) \geq S_{on}(m)\} \leq \{(n+1)(m+1)\}^{2J+2Q}2^{-g(m;n)}, \tag{29}$$

where:

$$g(m;n) \geq n\{D(\phi_{\hat{H}_L}||P_{off}) + D(\psi_{\hat{H}_P}||U)\} + m\{D(\phi_{H_L}||P_{on}) + D(\psi_{H_P}||P_{\Delta G})\}, \ \textit{if} \ H_L > \hat{H}_L,$$
$$g(m;n) \geq n\{D(\phi_{H_L}||P_{off}) + D(\psi_{H_P}||U)\} + m\{D(\phi_{\hat{H}_L}||P_{on}) + D(\psi_{\hat{H}_P}||P_{\Delta G})\}, \ \textit{if} \ H_L < \hat{H}.(30)$$

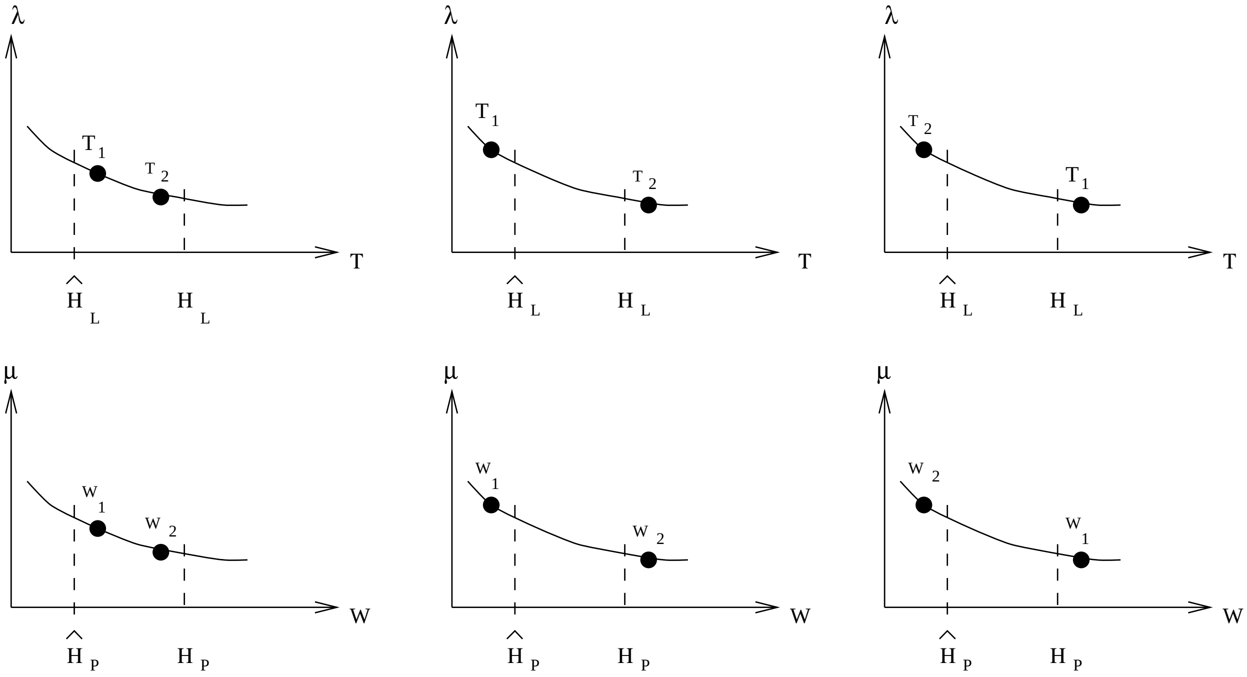

Figure 8: This figure illustrates the three cases of the argument in Theorem 7. The variables T_1, T_2, W_1, W_2 are functions of m, n and their values determine $g(m;n)$. If we use the Bhattacharyya heuristic then $T_1 = T_2 = H_L^*$ and $W_1 = W_2 = H_P^*$ for all values of m, n. They can therefore be thought of as "effective heuristics" and it is necessary to understand their "dynamics" as m, n vary. In Theorem 7, we show that they are restricted to lie in the ranges illustrated by the left-most column (the configurations in the centre and rightmost column are inconsistent with the three constraints (32,33) which enables us to put bounds on $g(m;n)$.

Proof. *We start by following the proof of Theorem 4 but with the definition of set E changed to allow for different heuristics H_L, H_P. We minimize $f(.,.,.,.)$ and obtain similar expressions for $\phi_{T_1} = \phi^{off*}, \psi_{W_1} = \psi^{off*}, \phi_{T_2} = \phi^{on*}, \psi_{W_2} = \psi^{on*}$ except that the minimization no longer occurs at $\gamma = 1/2$. The fall-off rate is determined by:*

$$g(m;n) = n\{D(\phi_{T_1}||P_{off}) + D(\psi_{W_1}||U)\} + m\{D(\phi_{T_2}||P_{on}) + D(\psi_{W_2}||P_{\Delta G})\}, \tag{31}$$

where

$$m\{T_2 + W_2 - H_L - H_P\} = n\{T_1 + W_1 - H_L - H_P\}, \tag{32}$$

and

$$\begin{aligned} \lambda(T_1) + \lambda(T_2) = 1 = \mu(W_1) + \mu(W_2), \\ \lambda(T_1) = \mu(W_1), \quad \lambda(T_2) = \mu(W_2). \end{aligned} \tag{33}$$

Recall, that to remove the ambiguity in H_L and H_P we imposed the constraint that $\lambda(H_L) = \mu(H_P)$. We also defined $\hat{H}_L, \hat{H}_P$ by the conditions $\lambda(\hat{H}_L) + \lambda(H_L) = 1$ and $\mu(\hat{H}_P) + \mu(H_P) = 1$ which implies that $\lambda(\hat{H}_L) = \mu(\hat{H}_P)$.

There are two situations to consider: (i) $H_L \geq \hat{H}_L$, which implies $H_P \geq \hat{H}_P$ (this follows from the equations at the end of the previous paragraph plus the fact that $\lambda(.)$ and $\mu(.)$ are monotonically decreasing functions), and (ii) $H_L \leq \hat{H}_L$, which implies $H_P \leq \hat{H}_P$.

We claim, in case (i), that $T_1, T_2 \in [\hat{H}_L, H_L]$ and $W_1, W_2 \in [\hat{H}_P, H_P]$, see figure (8). Moreover,

$$g(m;n) \geq n\{D(\phi_{\hat{H}_L}||P_{off}) + D(\psi_{\hat{H}_P}||U)\} + m\{D(\phi_{H_L}||P_{on}) + D(\psi_{H_P}||P_{\Delta G})\}. \tag{34}$$

Moreover, in situation (ii) we claim that $T_1, T_2 \in [H_L, \hat{H}_L]$ and $W_1, W_2 \in [H_P, \hat{H}_P]$, and

$$g(m;n) \geq n\{D(\phi_{H_L}||P_{off}) + D(\psi_{H_P}||U)\} + m\{D(\phi_{\hat{H}_L}||P_{on}) + D(\psi_{\hat{H}_P}||P_{\Delta G})\}. \tag{35}$$

We prove the results only for situation (i) because the proofs for situation (ii) are exactly analogous. The condition $\lambda(T_1) + \lambda(T_2) = 1$ implies that there are only three possible cases: either both $T_1, T_2 \in [\hat{H}_L, H_L]$ or, using the monotonicity of $\lambda(T)$, that $T_1 > H_L$ and $T_2 < \hat{H}_L$, or $T_1 < \hat{H}_L$ and $T_2 > H_L$. The first case will ensure that $W_1, W_2 \in [\hat{H}_P, H_P]$ which solves the problem. The second requires that $W_1 > H_P$ and $W_2 < \hat{H}_P$ but this is inconsistent with the requirement that $m\{T_2 + W_2 - H_L - H_P\} = n\{T_1 + W_1 - H_L - H_P\}$ (because the left hand side is negative and the right hand side is positive). Similarly, the third case implies that $W_1 < \hat{H}_P$ and $W_2 > H_P$ which again contradicts the equality. Thus the only possible situation is the first case.

Moreover, as $n \mapsto \infty$, we have $T_1 \mapsto H_L, W_1 \mapsto H_P, T_2 \mapsto \hat{H}_L, W_2 \mapsto \hat{H}_P$. (This is because $T_1 \leq H_L$ and $W_1 \leq H_P$ so as $n \mapsto \infty$ we have $T_1 + W_1 - H_L - H_P \mapsto 0$).

Given this result, it is now straightforward to prove the analogues of the second half of Theorem 4 and of Theorem 5 (using almost exactly the same proofs). More precisely, we first prove that the chance of an off-road path having greater reward than *any* part of the true road path falls off exponentially with the length of the off-road path. Then we obtain the rate of convergence and an upper bound on the expected error.

Theorem 8. *Let* $\hat{\Psi}_1 = D(\vec{\phi}_{\hat{H}_L}||P_{off}) + D(\vec{\psi}_{H_P}||U)$, $\hat{\Psi}_2 = D(\vec{\phi}_{H_L}||P_{on}) + D(\vec{\psi}_{H_P}||P_{\Delta G})$, *then the probability that an off-road path has greater reward than any on-road path is bounded above by:*

$$Pr\{\exists m \ : \ S_{off}(n) \geq S_{on}(m)\} \leq C_1(\hat{\Psi}_2)(n+1)^{2J+2Q}2^{-n\Psi_1}, \tag{36}$$

and, provided $\hat{\Psi}_1 > \log Q$, *the expected number of searches is* $O(N)$ *in the problem size* N *and is less than* $C_1(\hat{\Psi}_2)C_1(\hat{\Psi}_1 - \log Q)N$. *The expected error is bounded above by* $C_1(\hat{\Psi}_2)C_2(\hat{\Psi}_1 - \log Q)$, *which is independent of* N *and decays exponentially with* $\hat{\Psi}_1 - \log N$. *The order parameter* $K = \hat{\Psi}_1 + \hat{\Psi}_2 - \log Q$.

Proof. *We adapt the proofs of Theorems 4,5 but replacing* Ψ_1, Ψ_2 *with* $\hat{\Psi}_1, \hat{\Psi}_2$.

5.3 How to Sort the Queue

We have shown that the expected number of nodes searched is linear in N. But the convergence rate of the algorithm will depend on sorting the queue of nodes that we want to expand. After all, if we have order N nodes in the queue then we may have to spend $O(\log N)$ time searching the queue to determine which node to expand.

We now show that this may not be necessary and the expected search time for each step is constant. To see this, let us use a simple linked list data structure where we order the nodes in the queue according to their rewards (instead of a more sophisticated data structure, like a heap – see, for example, [10], [5]). By our previous theorems, the queue will contain, on average, order N elements. A* proceeds by expanding the top node and must adjust the queue to accommodate its children. Provided we can place the children in their correct position in the queue by only looking, at most, at a constant set of queue elements then the expected search time is constant.

How bad can the children of the best node be? The worst incremental reward they can get will be a negative number. It is convenient to represent this as $-\Lambda$, where Λ is positive and where $-\Lambda = \min_y \log P_{on}(y)/P_{off}(y) + \min_x \log P_{\Delta G}(x)/U(x) - H_L - H_P$.

We wish to put bounds on the expected number of nodes in the queue with rewards which are smaller by at most Λ than that of the best node. We do not know the reward of the best node, but we do know that there is always a true path segment (i.e. it consists entirely of on-road arcs) in the queue, whose length we can call n. It therefore suffices to put bounds on the expected number of paths in the queue with rewards greater than the reward of the on-path of length n minus Λ.

This can be done by a slightly more complicated variant of the proofs of Theorems 7 and 8. We consider the case when $H_L > \hat{H}_L$ and $H_P > \hat{H}_P$ (the alternative case can be solved by adapting the following argument). Suppose the longest true partial path in the queue is of length n and has reward r^t. We must consider the probabilities that paths in $F_1, ..., F_n$ have rewards higher than $r^t - \Lambda$. (We do not need to consider paths in $F_i, \ \ i > n$ because they involve children of nodes in the queue and so cannot be in the queue.) Applying the onion argument, for each $m \leq n$, we must bound the probability that any off-path of any length has reward higher than the true reward for n arcs minus Λ. Following the standard application of Sanov's theorem, we define the set:

$$E = \{(\vec{\phi}^{off}, \vec{\psi}^{off}, \vec{\phi}^{on}, \vec{\psi}^{on}) : \ m(\vec{\phi}^{off} \cdot \vec{\alpha} + \vec{\psi}^{off} \cdot \vec{\beta} - H_L - H_P) + \Lambda \geq n(\vec{\phi}^{on} \cdot \vec{\alpha} + \vec{\psi}^{on} \cdot \vec{\beta} - H_L - H_P)\}. \tag{37}$$

Following the proof of Theorem 7 this gives thresholds: T_1, T_2, W_1, W_2 (as before, these thresholds are functions of n and m), where:

$$m(T_1 + W_1 - H_L - H_P) + \Lambda = n(T_2 + W_2 - H_L - H_P), \tag{38}$$

$$\lambda(T_1) + \lambda(T_2) = 1, \ \mu(W_1) + \mu(W_2) = 1, \ \lambda(T_1) = \mu(W_1), \ \lambda(T_2) = \mu(W_2). \tag{39}$$

The fall-off depends on

$$g(n : m) = m\{D(\vec{\phi}_{T_1}||P_{off}) + D(\vec{\psi}_{W_1}||U)\} + n\{D(\vec{\phi}_{T_2}||P_{on}) + D(\vec{\psi}_{W_2}||P_{\Delta G})\}. \tag{40}$$

Now, again following Theorem 7, we would like to put lower bounds on $g(n : m)$. For Theorem 7 we were able to prove that $T_1, T_2 \in [\hat{H}_L, H_L]$ and $W_1, W_2 \in [\hat{H}_P, H_P]$ (for the situation where $H_L > \hat{H}_L$ and analogous results hold for the situation with $H_L < \hat{H}_L$). The Λ term prevents these results from being true. However, for large enough n or m the Λ term becomes negligible and we will prove that $T_1, T_2 \in [\hat{H}_L - \epsilon, H_L + \epsilon]$ and $W_1, W_2 \in [\hat{H}_P - \epsilon, H_P + \epsilon]$. These contain the most important terms and, as we will show, make only a constant contribution to the expected sorting cost. The contributions for small m and n are, of course, also constant.

We first show, that for any fixed n, the thresholds T_1, W_1 increase monotonically with m and tend to H_L, H_P as $m \mapsto \infty$ and, similarly, T_2, W_2 decrease monotonically with m and tend to $\hat{H}_L, \hat{H}_P$. From $\lambda(T_1) = \mu(W_1)$, see equation (39), and the monotonicity of the functions $\lambda(.)$ and $\mu(.)$, we see that the coupling between T_1 and W_1 means that they have to increase, or decrease, together. Similarly, T_2 and W_2 must either decrease, or increase, together. By equation (38), we see that at $m = 0$ we have $T_2(0) + W_2(0) = H_L + H_P + \Lambda/n$ which implies, by equation (39), that $T_1(0) + W_1(0) < \hat{H}_L + \hat{H}_P$. Equation (38) enforces that $T_1 \mapsto H_L, \ W_1 \mapsto H_P$ as $m \mapsto \infty$ which implies that $T_2 \mapsto \hat{H}_L$ and $W_2 \mapsto \hat{H}_P$. Therefore, we see that $T_1 + W_1$ increases overall from $m = 0$ as $m \mapsto \infty$ and conversely $T_2 + W_2$ decreases. But are these changes monotonic? From equation (38), we see that provided $T_1 + W_1 < H_L + H_P$ then it is inconsistent for T_1 and W_1 to decrease and T_2 and W_2 to increase. However, it is impossible for $T_1 + W_1 > H_L + H_P$ because, by equation (38), this would imply that $T_2 + W_2 > H_L + H_P$ (recall that $\Lambda > 0$) which is inconsistent with equations (39). So we conclude that the only possibility is for T_1 and W_1 to increase monotonically and T_2 and W_2 to decrease monotonically.

Now select a number N_0, chosen so that $N_0(\epsilon) \geq \Lambda/\epsilon$, and let $n \geq N_0$. Then for $m = 0$, we see that $T_2 < H_L + \epsilon$ and $W_2 < H_P + \epsilon$ (this follows from equations (38,39)). Moreover, $T_1 > \hat{H}_L - \hat{\epsilon}$ and $W_1 > \hat{H}_P - \hat{\epsilon}$ (where $\hat{\epsilon}$ is defined by equation (39)). As m increases T_1, W_1 increase monotonically to H_L, H_P and T_2, W_2 decrease monotonically to $\hat{H}_L, \hat{H}_P$. Therefore we have:

$$g(m : n) \geq m\{D(\vec{\phi}_{\hat{H}_L - \hat{\epsilon}}||P_{off}) + D(\vec{\psi}_{\hat{H}_P - \hat{\epsilon}}||U)\} + n\{D(\vec{\phi}_{H_L + \epsilon}||P_{on}) + D(\vec{\phi}_{H_P + \epsilon}||P_{\Delta G})\} \ \ \forall \ n > N_0(\epsilon), \tag{41}$$

which ensures that the fall-off factors are bounded below for large n.

We now deal with the case of small n (i.e. $n < N_0(\epsilon)$) and large m. We claim that there is a specific value M_0 such that for $m > M_0$ we have $T_1, T_2 \in [\hat{H}_L, H_L]$ and $W_1, W_2 \in [\hat{H}_P, H_P]$, in which case we can use the same bounds for $g(m;n)$ as above (see equation (41)). This claim is proven by setting $M_0 = \Lambda/(H_L + H_P - \hat{H}_L - \hat{H}_P)$ and substituting into equation (38) to obtain $\Lambda(T_1 + W_1 - \hat{H}_L - \hat{H}_P) = n(T_2 + W_2 - H_L - H_P)(H_L + H_P - \hat{H}_L - \hat{H}_P)$. The consistency conditions, imposed by equation (39), mean that this equation's only solution is $T_1 = \hat{H}_L$, $W_1 = \hat{H}_P$, $T_2 = H_L$, and $W_2 = H_P$ (all other possibilities can be shown to be inconsistent using equation (39). The monotonicity increase of T_1, W_1, and decrease of T_2, W_2, ensure that, for $m > M_0$, the $T_1, T_2 \in [\hat{H}_L, H_L]$ and $W_1, W_2 \in [\hat{H}_P, H_P]$.[2]

The final situation is when $n < N_0(\epsilon)$ *and* $m < M_0$. This is a finite case so we do not need to obtain bounds. We can simply exhaustively count the number of arc segments. (This is extremely conservative).

We now put all these results together. Let $\hat{n}$ be the length of the true path segment in the queue (by the nature of A* there can only be one such true path segment in the queue at any time). The expected number of queue members with rewards higher than the true segment minus Λ is obtained by summing over the possible segments in $F_1, F_2, ..., F_{\hat{n}}$. We can deal with the cases $m < M_0$ *and* $n < N_0(\epsilon)$ by exhaustive counting which yields a finite number. For each $n \leq \hat{n}$ we can use the bounds given by equation (41) and apply the arguments from Theorem 7 to sum over m for fixed n obtaining a term which decays exponentially with n. Finally, we can apply the arguments from Theorem 8 to sum over n. The exponential decay factor means that this sum will converge for any value of $\hat{n}$ (even as $\hat{n} \mapsto \infty$). Hence we get constant expected sorting costs.

We summarize this result as a theorem:

Theorem 9 *The expected sorting rate per node is constant and independent of the size N of the problem.*

6 Conclusion

Our analysis shows it is possible to track certain classes of image contours with linear expected node expansions (and linear expected sorting time per node). We have shown how the convergence rates, and the choice of A* heuristics, depend on order parameters which characterize the problem domain. In particular, the entropy of the geometric prior and the Kullback-Leibler distance between P_{on} and P_{off} allow us to quantify intuitions about the power of geometrical assumptions and edge detectors to solve these tasks. Not surprisingly, the easiest target curves to detect are those for which the edge detector is most informative and the prior geometric knowledge most constraining. Our analysis allows us to quantify these intuitions. See [18] for analysis of the forms of P_{on}, P_{off} arising in typical images.

Our more recent work [27] has extended this work by showing that similar order parameters can be used to specify intrinsic (algorithm independent) difficulty of the search

[2] Observe that M becomes infinite if we use the Bhattacharyya heuristic (i.e. when $H_L = \hat{H}_L$ and $H_P = \hat{H}_P$). This is because the regions $[\hat{H}_L, H_L]$ and $[\hat{H}_P, H_P]$ shrink to points H_L^* and H_P^* and the T's and W's only reach them asymptotically. This requires a modification of the proof to obtain bounds on M for which $\max\{|T_1 - H_L^*|, |W_1 - H_P^*|, |T_2 - H_L^*|, |W_2 - H_L^*|\} < \epsilon$.

problem and that phase transitions occur when these order parameters take critical values. Fortunately, the proofs in this paper break down at closely related critical points. Therefore A* algorithms are an effective way to solve this problem in the regime for which it can be solved.

As shown in [25] many of the search algorithms proposed to solve vision search problems [19],[2], [11] are special cases of A* (or close approximations). We therefore hope that the results of this paper will throw light on the success of the algorithms and may suggest practical improvements and speed ups, see [5] for promising preliminary results.

Crucial to our analysis has been the use of Bayesian probability theory both to determine an optimization criterion for the problem we wish to solve *and* to define the *Bayesian ensemble* of problem instances. Analysis of the Bayesian ensemble led to the definition of order parameters which characterized the difficulty of the problem. It will be interesting to compare our results with those obtained by [4],[23] for completely different classes of problems and using different techniques. This is a topic for further research.

Acknowledgements

We want to acknowledge funding from NSF with award number IRI-9700446, from the Center for Imaging Sciences funded by ARO DAAH049510494, and from the Smith-Kettlewell core grant, and the AFOSR grant F49620-98-1-0197 to A.L.Y. Lei Xu drew our attention to Pearl's book on heuristics and we thank Abracadabra books for obtaining a second hand copy for us. We would also like to thank Dan Snow and Scott Konishi for helpful discussions as the work was progressing and Davi Geiger for providing useful stimulation. Also influential was Bob Westervelt's joking request that he hoped James Coughlan's PhD thesis would be technical enough to satisfy the Harvard Physics Department. David Forsyth, Jitendra Malik, Preeti Verghese, Dan Kersten, Suzanne McKee and Song Chun Zhu gave very useful feedback and encouragement. Finally, we wish to thank Tom Ngo for drawing our attention to the work of Cheeseman and Selman.

7 Appendix 1: The Theory of Types

This appendix derives the basic concepts and mathematical machinery that we will need to prove our results.

For concreteness, we will assume that we dealing with the likelihood function terms only. In other words, we are only concerned with the measurements of the local road detectors and we *ignore* any knowledge about the likely geometrical configurations of the road.

We have a sequence of samples $\vec{y} = y_1, y_2, ..., y_N$ of the responses of the road detector. The optimal tests for determining whether the samples come from P_{on} or P_{off} will depend on the *log-likelihood ratio*[3] (see the Neyman-Pearson lemma [6]):

[3] This can be thought of as the maximum likelihood test between two hypotheses which are equally likely a priori.

$$\log\{\frac{P_{on}(y_1,....y_N)}{P_{off}(y_1,....y_N)}\} = \log\{\prod_{i=1}^{N}\frac{P_{on}(y_i)}{P_{off}(y_i)}\} = \sum_{i=1}^{N}\log\{\frac{P_{on}(y_i)}{P_{off}(y_i)}\}. \tag{42}$$

The larger the log-likelihood ratio then the more probable that the measurement sample $\vec{y} = (y_1, y_2, ..., y_N)$ came from the on-road rather than off-road (if the log-likelihood ratio is zero then both on-road and off-road are equally probable). But we need to consider the probabilities that a random sample from off-road has higher log-likelihood ratio than a sample from on-road. This requires us to put probabilistic bounds on the probabilities of unlikely events. This can be done by adapting the theory of types, see [6].

Any sample $\vec{y} = (y_1, y_2, ..., y_N)$ determines an empirical histogram, or *type*, $\vec{\phi}(\vec{y})$ which is an J-dimensional vector whose components $\phi_1, ..., \phi_J$ are the proportions of responses ϕ_i which take values $1, ..., J$. (i.e. $\phi(y) = (1/N)\sum_{i=1}^{N}\delta_{y_i,y}$). The key point is that *all the relevant properties of the sample will depend only on its type* (in view of the i.i.d. assumption). This includes the result of the log-likelihood test, see equation (42), which we can re-express as:

$$\log\{\frac{P_{on}(y_1,....y_N)}{P_{off}(y_1,....y_N)}\} = \sum_{y=1}^{J}(N\phi(y))\log\{P_{on}(y)/P_{off}(y)\}. \tag{43}$$

It is important to observe that this is simply the dot-product, $N\vec{\phi}\cdot\vec{\alpha}$, of the type $\vec{\phi}$ with a weight vector $\vec{\alpha}$ (for the equation above, $\vec{\alpha}$ has components $\alpha(y) = \log\{P_{on}(y)/P_{off}(y)\}$). Most of the quantities that we are concerned with, such as the rewards of paths and the convergence rates of algorithms, will depend on dot products of this form. The theory of types proceeds by putting probabilistic bounds on types which can then be used to put probability bounds on the dot products. For the results which follow it is convenient to divide out by the size factor N. We therefore consider the average of the log-likelihood with respect to the sequence of samples – i.e. $(1/N)\sum_{i=1}^{N}\log P_{on}(y_i)/P_{off}(y_i)$.

There are five key lemmas that we will use about types [6]:

Lemma 1. The total number of types $\leq (N+1)^J$. (This is a very generous upper bound which occurs because each component of the type vector $\vec{\phi}$ can take at most $N+1$ possible values).

Lemma 2. The probability $Q^N(\vec{y})$ for any sequence of samples $\vec{y}$ drawn i.i.d. from $Q(y)$ depends only on the *entropy* $H(\vec{\phi}(\vec{y}))$ of the type of the sequence and the Kullback-Leibler distance $D(\vec{\phi}(\vec{y})||Q)$ between the type and the distribution Q, and is given by:

$$Q^N(\vec{y}) = F(\vec{\phi}(\vec{y})) = 2^{-N\{H(\vec{\phi}(\vec{y}))+D(\vec{\phi}(\vec{y})||Q)\}}. \tag{44}$$

(The probability of the sequence can be expressed as $\prod_{y=1}^{J}Q(y)^{N\phi(y)} = 2^{N\sum_{y=1}^{J}\phi(y)\log Q(y)}$ and we use $H(\vec{\phi}) + D(\vec{\phi}||Q) = -\sum_{y=1}^{J}\phi(y)\log Q(y)$ to obtain the result.)

Lemma 3. The probability $P(\vec{\phi})$ that a sequence has type $\vec{\phi}$ is given by:

$$P(\vec{\phi}) = F(\vec{\phi})\left|T(\vec{\phi})\right|, \tag{45}$$

where $\left|T(\vec{\phi})\right| = \sum_{\vec{y}:\vec{\phi}(\vec{y})=\vec{\phi}} 1$ is the number of distinct sequences with type $\vec{\phi}$. (This follows from $P(\vec{\phi}) = \sum_{\vec{y}} \delta_{\vec{\phi},\vec{\phi}(\vec{y})} Q^N(\vec{y})$ and substituting equation (44)).

Lemma 4. We can bound the size of each type class by [6]:

$$\frac{2^{NH(\vec{\phi})}}{(N+1)^J} \leq \left|T(\vec{\phi})\right| \leq 2^{NH(\vec{\phi})}. \tag{46}$$

(Not surprisingly, the larger the entropy $H(\vec{\phi})$ the bigger the type class.)

Lemma 5. We can put a bound on $P(\vec{\phi})$ by combining Lemmas 2, 3, and 4. This gives:

$$\frac{2^{-ND(\vec{\phi}||Q)}}{(N+1)^J} \leq P(\vec{\phi}) \leq 2^{-ND(\vec{\phi}||Q)}. \tag{47}$$

From these basic lemmas we can derive the main result we need. We are particularly interested in putting bounds of the probability that a type $\vec{\phi}$ lies within a certain set of types E. For example, for our road tracking task we define the *reward* of a type $\vec{\phi}$ to be $\vec{\phi} \cdot \vec{\alpha}$. It will then be important to bound the probability that sequences of samples from off-road have rewards above a specific threshold T. To do this, we define $E_T = \{\vec{\phi} : \vec{\phi} \cdot \vec{\alpha} \geq T\}$ and ask for the probability, $Pr(\vec{\phi} \epsilon E_T)$, that the type of a sequence of samples from off-road will lie within E_T.

The main result is called Sanov's theorem:

Sanov's Theorem. *Let $y_1, y_2, ..., y_N$ are i.i.d. from a distribution $Q(y)$ with alphabet size J and E be any closed set of probability distributions. Let $Pr(\vec{\phi} \in E)$ be the probability that the type of a sample sequence lies in the set E. Then:*

$$\frac{2^{-ND(\vec{\phi}^*||Q)}}{(N+1)^J} \leq Pr(\vec{\phi} \in E) \leq (N+1)^J 2^{-ND(\vec{\phi}^*||Q)}, \tag{48}$$

where $\phi^* = \arg\min_{\phi \in E} D(\phi||Q)$ is the distribution in E that is closest to Q in terms of Kullback-Leibler divergence.

Proof. *It is straightforward to see that $\max_{\vec{\phi}\epsilon E} P(\vec{\phi}) \leq Pr(\vec{\phi}\epsilon E) \leq |E| \max_{\vec{\phi}\epsilon E} P(\vec{\phi})$. From Lemma 5, we can put upper and lower bounds on $\max_{\vec{\phi}\epsilon E} P(\vec{\phi})$ in terms of $\vec{\phi}^* = \arg\min_{\vec{\phi}\epsilon E} D(\vec{\phi}||Q)$. This gives the result using Lemma 1 to put $1 \leq |E| \leq (N+1)^J$.*

Appendix 2: Bounding the Sums of Exponential Series

We often need to sum series which contain geometric decay terms and alphabet factors. The geometric terms dominate the series for large m but for small m the alphabet terms become important. Our approach is to sum the geometric series and add a correction factor for the alphabet terms.

Bounding series like $\sum_{m=0}^{\infty}(m+1)^A 2^{-Bm}$.

We describe two methods for summing, or bounding, series which contain exponential decay terms and alphabet factors. The alphabet factors are usually bounded by polynomial terms (see section (5)). It should be emphasized, however, that the polynomial bounds on the alphabet factors are not tight and, in particular, will be misleading for small m. We see two strategies.

The first strategy is to sum the series directly using the polynomial bounds for the alphabet factors. To do this, we define $G_1(B,A) = \sum_{m=0}^{\infty}(m+1)^A 2^{-Bm}$, where A is a positive integer corresponding to the alphabet factors and B is the exponential decay factor. Observe that $G_1(B,0) = \sum_{m=0}^{\infty} 2^{-Bm} = \frac{1}{1-2^{-B}}$. Differentiating $G_1(B,0)$ with respect to B introduces polynomial terms inside the summation. It is then straightforward to verify that:

$$G_1(B,A) = \frac{e^B}{(\log_e 2)^A}(-1)^A \frac{d^A}{dB^A}\frac{1}{1-2^{-B}}. \tag{49}$$

The second strategy takes into account the inaccuracies of the alphabet factor terms. For small m, the alphabet factors become important and so they should be modelled accurately. We will not do this here. Instead we observe that given any number ϵ we can pick a number $M(\epsilon, A)$ such that $(m+1)^A \leq 2^{m\epsilon}\ \forall\ m \geq M_0(\epsilon, A)$. We can sum the series to obtain:

$$G_2(B,A) = \frac{1}{1-2^{-(B-\epsilon)}} + \Xi(A,B,\epsilon), \tag{50}$$

where $\Xi(A,B,\epsilon)$ is a (positive) correction caused by the terms for $m < M_0(\epsilon, A)$ (the sum underestimates these terms because $(m+1)^A \geq 2^{(M\epsilon)}$, $\forall\ m < M_0(\epsilon, A)$.)

Bounding series like $\sum_{m=0}^{\infty} m(m+1)^A 2^{-Bm}$.

In addition, we will often need to bound sums such as:

$$\sum_{m=0}^{\infty} m 2^{-Bm}(m+1)^A. \tag{51}$$

As above, we pick a number ϵ and $M_0(\epsilon, A)$ such that $(m+1)^A < e^{m\epsilon}$, $\forall\ m > M_0(\epsilon, A)$. We can divide the sum into two parts:

$$\sum_{m=0}^{\infty} m 2^{-(B-\epsilon)m} + \hat{\Xi}(\epsilon, A, B), \tag{52}$$

where $\hat{\Xi}(\epsilon, A, B)$ is a correction factor used to correct for the alphabet factors for small $m < M_0(\epsilon, A)$.

Let $f(x) = \sum_{m=0}^{\infty} 2^{xm} = 1/(1-2^x)$. Then it is straightforward to differentiate both sides with respect to x to obtain $\sum_{m=0}^{\infty} m 2^{xm} = \frac{2^x}{(1-2^x)^2}$. We can therefore express:

$$\sum_{m=0}^{\infty} m2^{-Bm}(m+1)^A = \frac{2^{-B}}{(1-2^{-B})^2} + \hat{\Xi}(\epsilon, A, B). \tag{53}$$

References

[1] R. Balboa. PhD Thesis. Department of Computer Science. University of Alicante. Spain. 1997.

[2] M. Barzohar and D. B. Cooper, "Automatic Finding of Main Roads in Aerial Images by Using Geometric-Stochastic Models and Estimation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 459-464, 1993.

[3] R. E. Bellman, *Applied Dynamic Programming.* Princeton University Press, 1962.

[4] P. Cheeseman, B. Kanefsky, and W. Taylor. "Where the Really Hard Problems Are". In *Proc. 12th International Joint Conference on A.I.*. Vol. 1., pp 331-337. Morgan-Kaufmann. 1991.

[5] J. Coughlan, D. Snow, C. English, and A.L. Yuille. "Efficient Optimization of a Deformable Template Using Dynamic Programming". In *Proceedings Computer Vision and Pattern Recognition. CVPR'98.* Santa Barbara. California. 1998.

[6] T.M. Cover and J.A. Thomas. **Elements of Information Theory**. Wiley Interscience Press. New York. 1991.

[7] M.A. Fischler and R.A. Erschlager. "The Representation and Matching of Pictorial Structures". *IEEE. Trans. Computers.* C-22. 1973.

[8] M.R. Garey and D.S. Johnson. **Computers and Intractability: A Guide to he Theory of NP-Completeness**. W.H. Freeman and Co. New York. 1979.

[9] D. Geiger, A. Gupta, L.A. Costa, and J. Vlontzos. "Dynamic programming for detecting, tracking and matching elastic contours." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-17, March 1995.

[10] D. Geiger and T-L Liu. "Top-Down Recognition and Bottom-Up Integration for Recognizing Articulated Objects". In *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition.* Ed. M. Pellilo and E. Hancock. Venice, Italy. Springer-Verlag. May. 1997.

[11] D. Geman. and B. Jedynak. "An active testing model for tracking roads in satellite images". *IEEE Trans. Patt. Anal. and Machine Intel.* Vol. 18. No. 1, pp 1-14. January. 1996.

[12] U. Grenander, Y. Chow and D. M. Keenan, *Hands: a Pattern Theoretic Study of Biological Shapes,* Springer-Verlag, 1991.

[13] D.W. Jacobs. “Robust and Efficient Detection of Salient Convex Groups”. *IEEE Trans. Patt. Anal. and Machine Intel.* Vol. 18. No. 1, pp 23-37. January. 1996.

[14] R.M. Karp and J. Pearl. “Searching for an Optimal Path in a Tree with Random Costs”. *Artificial Intelligence.* 21. (1,2), pp 99-116. 1983.

[15] M. Kass, A. Witkin, and D. Terzopoulos. ”Snakes: Active Contour models”. In *Proc. 1st Int. Conf. on Computer Vision.* 259-268. 1987.

[16] N. Khaneja, M.I. Miller, and U. Grenander. “Dynamic Programming Generation of Geodesics and Sulci on Brain Surfaces”. Submitted to *PAMI.* 1997.

[17] D.C. Knill and W. Richards. (Eds). **Perception as Bayesian Inference**. Cambridge University Press. 1996.

[18] S. Konishi, A.L. Yuille, J.M. Coughlan, and S.C. Zhu. In preparation 1998.

[19] U. Montanari. “On the optimal detection of curves in noisy pictures.” *Communications of the ACM*, pages 335–345, 1971.

[20] J. Pearl. **Heuristics**. Addison-Wesley. 1984.

[21] B.D. Ripley. **Pattern Recognition and Neural Networks**. Cambridge University Press. 1995.

[22] S. Russell and P. Norvig. “Artificial Intelligence: A Modern Approach. Prentice-Hall. 1995.

[23] B. Selman and S. Kirkpatrick. “Critical Behaviour in the Computational Cost of satisfiability Testing”. Artificial Intelligence. 81(1-2); 273-295. 1996.

[24] P.H. Winston. **Artificial Intelligence**. Addison-Wesley Publishing Company. Reading, Massachusetts. 1984.

[25] A.L. Yuille and J. Coughlan. ”Twenty Questions, Focus of Attention, and A*: A theoretical comparison of optimization strategies.” In *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition.* Ed. M. Pellilo and E. Hancock. Venice, Italy. Springer-Verlag. May. 1997.

[26] A.L. Yuille and J.M. Coughlan. “Convergence Rates of Algorithms for Visual Search: Detecting Visual Contours”. In *Proceedings NIPS’98.* 1998.’

[27] A.L. Yuille and J.M. Coughlan. “Visual Search: Fundamental Bounds, Order Parameters, Phase Transitions, and Convergence Rates”. Submitted to *Transactions on Pattern Analysis and Machine Intelligence.* 1998.

[28] A.L. Yuille and J. Coughlan. ”An A* perspective on deterministic optimization for deformable templates”. To appear in *Pattern Recognition Letters.* 1998.

[29] S.C. Zhu, Y. Wu, and D. Mumford. "Minimax Entropy Principle and Its Application to Texture Modeling". Neural Computation. Vol. 9. no. 8. Nov. 1997.

[30] S.C. Zhu and D. Mumford. "Prior Learning and Gibbs Reaction-Diffusion". IEEE Trans. on PAMI vol. 19, no. 11. Nov. 1997.

[31] S.C. Zhu and D. Mumford. "GRADE: A framework for pattern synthesis, denoising, image enhancement, and clutter removal." In Proceedings of International Conference on Computer Vision. Bombay. India. 1998.

[32] S-C Zhu, Y-N Wu and D. Mumford. FRAME: Filters, Random field And Maximum Entropy: — Towards a Unified Theory for Texture Modeling. Int'l Journal of Computer Vision 27(2) 1-20, March/April. 1998.

[33] S.C. Zhu. "Embedding Gestalt Laws in Markov Random Fields". Submitted to *IEEE Computer Society Workshop on Perceptual Organization in Computer Vision.*