Unsupervised Semantic Perception, Summarization, and Autonomous Exploration for Robots in Unstructured Environments

Yogesh Girdhar

School of Computer Science
McGill University, Montréal
July 2014

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Ph.D.

List of Tables

List of Figures

List of Algorithms

Acknowledgements

The work presented in this thesis would have been absolutely impossible to produce without the continuous encouragement, support, and inspiration from my thesis supervisor Greg Dudek. Not only has Greg taken a personal interest in promoting me and my work, but he has also provided key support every time I felt I was stuck in an intellectual local minima.

I would like to thank my PhD qualifying exam committee members Doina Precup, Joelle Pineau, Mike Langer, and Luc Devroye for giving me sound advice early in my work.

Doina has been a great teach and mentor. Her class on machine learning provided me with a strong foundation upon which I felt comfortable in exploring the machine learning literature. Outside the classroom, Doina has given me continuous feedback on my research and provided much needed direction.

I would like to thank Luc for introducing me to the beautiful world of randomness and Markov chains. His class on probabilistic analysis has been my favorite class in grad school, while also being the most painful (in a good way). Analysis of the secretaries hiring problem that we present in this thesis was done in close collaboration with Luc.

I would like to thank the members of the Mobile Robotics Lab (MRL) at McGill, all of whom have tolerated me for such a long time, and kept this journey playful.

Philippe Giguere has been a good friend and colleague. Without his intellectual support, knowledge of Aqua related black magic, and general comradery, this PhD would have been a lot less fun and a lot more challenging.

Ioannis Rekleitis is a great friend, great diver, and a great underwater videographer. His support, especially while being underwater, has been critical for completion of this work. On terra firma, Ioannis has been a constant source of interesting conversation from the widest variety of topics.

Florian Shkurti and Anqi Xu have been excellent lab mates, and we have on several projects. I could not have asked for more skillful and fun colleagues.

I am thankful to Chris Prahacs and Bir Bikram Dey for keeping Aqua alive after so many years of constant physical and electrical abuse. Every time I have talked to them, I have learned something new.

It has been a pleasure working with David Whitney and Arnold Kalmbach, both of whom

came to the lab as undergrads, and taught me many things. Collaborating with them was a lot of fun.

I would like to thank my friends outside the lab. Ivan Savov for interesting conversations about life, the universe, and the nature of reality. Misha Rabinovich, whose creative misuse of my code has produced several interesting works of art. Aditya Bhatia, on old friend, with whom I have been able to talk about everything else.

I would like to thank my family for supporting me through this long journey; my wife Sofia Ibarraran for giving me continuous joy; my parents Pushpa Girdhar and Kishore Girdhar for opening the doors for this wonderful journey; and my sisters Karuna Shree and Ritu Shree, for always being on my side.

Abstract

This thesis explores the challenges involved in building autonomous exploration and monitoring systems, and makes contributions on four fronts: describing the semantic context of the collected data, summarizing this information, deciding where to collect this data, and making optimal online irrevocable decisions for physical sample collection.

Making high level decisions based on the environmental context of a robot's location requires that we first describe what is being observed in a semantic space with higher level of abstraction than the low level sensor reading. ROST, a realtime online spatiotemporal topic modeling technique that we develop in this thesis solves the problem of obtaining such high level descriptors. Topics in this case represent the latent causes (such as objects and terrains), which produce these observations. ROST extends previous work on topic modeling by efficiently taking into account the spatiotemporal context of an observation, and using a novel Gibbs sampling technique to refine the topic label assignment in realtime, making it suitable for processing streaming sensor data such as video and audio observed by a robot. Our experiments suggest that taking into account the spatiotemporal context of observations results in better topic labels that have higher mutual information with ground truth labels, compared to topic modeling without taking into account the spatiotemporal context. Moreover we show that the perplexity of the online topic model using the proposed Gibbs sampler is competitive with batch Gibbs sampler.

Given a scene descriptor such as bag-of-words, location, or topic distribution, the thesis then proposes a novel online summarization algorithm, which unlike previous techniques focuses on building a navigation summary containing all the surprising scenes observed by the robot. We argue that the summaries produced by the algorithm (called extremum summaries) are ideal for monitoring and inspections tasks, where the goal is to maintain a small set of images that is representative of the diversity of what has been observed. Although computation of an optimal summary, even in the batch case is NP-hard, we empirically show that the approximate online algorithm presented in the thesis produces summaries with cost that is statistically indistinguishable from batch summaries, while running on natural datasets. Cost was measured as the distance of the farthest sample from a sample in the summary.

Collecting data from an environment to build a topic model or a summary requires a robot to traverse this environment. If the geographic size of this region of interest is small then we can simply use any space filling curve to plan this path. However, for larger areas this might not be possible, and hence we propose an information theoretic exploration technique which biases the path towards locations with high information gain in topic space. The resulting topic models were empirically shown to perform better than topic models learned with other competing exploration algorithms, such as free space exploration. Performance was measured in terms of mutual information with ground truth labels, and mutual information with topic labels computed in batch mode with complete knowledge of the environment.

Many exploration robots are required to collect samples and perform some chemical or physical analysis. Often such a task requires making irrevocable decisions on whether to select the current sample or not. This thesis presents a novel formulation of this task as an instance of the secretaries hiring problem. We examine several existing variants of this problem, and present an optimal solution to a new variant of the secretaries hiring problem, where the goal is to maximize the probability of identifying the top K samples online and irrevocably.

Together, the contributions of this thesis are a step towards developing fully autonomous robotic agents that can be used in collaboration with humans to explore dangerous unknown environments.

Résumé

Cette thèse explore les défis rencontrés afin de réaliser des systèmes autonomes d'exploration et de surveillance. Elle apporte des contributions sur quatre fronts : description du contexte sémantique d'un jeu de donnée, résumé automatique de cette information, décision sur les endroits où collecter ces données et finalement la prise de décision irrévocable liée à la saisie ou le traitement d'échantillons physiques.

À fin de prendre des décisions de haut niveau basées sur le contexte du milieu environnant d'un robot, nous devons d'abord décrire les observations dans un espace sémantique d'un niveau d'abstraction plus élevé que celui représenté par l'espace des mesures des capteurs. ROST, un modèle de sujet (topic modeling) d'abstraction spatio-temporel en ligne que nous avons développé dans cette thèse, résout le problème de l'obtention de telle description de haut niveau. Dans le cas qui nous intéresse, les sujets (topics) représentent les causes latentes (telles que les objets ou terrains présents) produisant les observations. ROST étend des travaux antérieurs sur la modélisation de sujet, en tenant compte de manière efficace du contexte spatio-temporel d'une observation, et en utilisant un nouvel échantillonneur Gibbs pour raffiner l'assignation d'étiquette de sujet. Cette approche rend possible le traitement de flux de données en temps-réel, tels que la vidéo et l'audio capturés par un robot. Nos expérimentations suggèrent que tenir compte de ce contexte spatio-temporel donne un étiquetage plus proche de la vérité-terrain (du point de vue de l'information mutuelle), comparativement à ne pas en tenir compte. De plus, nous montrons que la mesure de perplexité de cette modélisation en ligne utilisant note échantillonneur Gibbs est compétitive avec un échantillonneur Gibbs par lot.

À partir d'un descripteur de scène tel que sac de mots (bag-of-words), location ou distribution de sujets, nous proposons un nouvel algorithme de résumé en ligne qui, contrairement aux approches précédentes, se concentre sur la construction d'un résumé de navigation contenant les scènes dites surprenantes observées par le robot. Nous soutenons que les résumés produits par cet algorithme (appelés résumés extremums) sont idéaux pour les tâches de surveillance et d'inspection. Pour ces tâches, l'objectif est de conserver un petit ensemble d'images représentatif de la diversité des observations. Bien que la complexité en temps de

calcul d'un résumé optimal (même en traitement par lot) soit NP-dur, nous avons démontré de manière empirique que notre algorithme approximatif de résumé en ligne produit des résumés avec des coûts qui sont statistiquement impossibles à distinguer des résumés obtenus par lot, pour des jeux de données naturelles. Ce coût était établi en fonction de la distance de l'échantillon le plus éloigné d'un des échantillons conservés dans le résumé.

Afin de construire un modèle de sujet ou un résumé pour un environnement donné, un robot doit le traverser pour effectuer une collecte de données. Si l'étendue géographique de la région d'intérêt est petite, alors il est possible de simplement la couvrir avec une trajectoire basée sur n'importe quelle courbe remplissant l'espace. Cependant, pour des régions plus étendues, cette approche devient peu pratique. Par conséquent, nous proposons une technique d'exploration basée sur la théorie de l'information, qui dirige la trajectoire vers les endroits contenant plus d'information dans l'espace des sujets (topic space). Nous démontrons empiriquement que les modèles de sujets découlant de cette approche performent mieux que les modèles de sujets appris par l'entremise d'autres algorithmes, tel que l'algorithme d'exploration par les espaces libres (free space exploration). Cette performance a été mesurée en terme d'information mutuelle avec la vérité-terrain, et avec les étiquettes de sujets estimées à partir d'un algorithme par lot possédant une connaissance complète de l'environnement.

De nombreux robots explorateurs doivent faire la collecte d'échantillons physiques ainsi que certaines analyses chimique ou physique in-situ. Très souvent, ces tâches impliquent la prise de décisions irrévocables à savoir si un tel échantillon doit être sélectionné ou non. Nous présentons une nouvelle formulation de cette tâche comme une du problème de l'embauche de secrétaires (secretaries hiring problem). Nous examinons plusieurs variantes de ce problème, et présentons une solution à la nouvelle variante proposée. Pour cette dernière, l'objectif est de maximiser la probabilité d'identifier les K meilleurs échantillons, de manière irrévocable.

Ensemble, les contributions de cette thèse constituent une avance vers le développement d'agents robotisés entièrement automatisés, permettant l'exploration de l'environnements dangereux en collaboration avec les humains.

Table of Contents

Chapter 1

Introduction

Imagine sending a robot to explore an unknown environment, such as to the bottom of an ocean or to another planet. Communicating with this robot will be slow and unreliable and hence direct control of the vehicle is not possible. Moreover, we would like the robot to identify a few samples to be collected and returned. This thesis deals with the challenges of implementing such a system and presents solutions to each of these problems.

Exploration robots will be useful in many different scenarios where we would like to monitor vast geographic regions for any abnormalities. We want to monitor our forests for wildlife, fires, diseases and illegal logging; our borders and coastlines for any intrusions; and our oceans for archeological discoveries and animal biodiversity. These environment monitoring and exploration tasks, when done by a humans, can be dangerous and expensive. Use of automated intelligent robots is thus desirable and useful. This thesis explores some of the challenges involved in building such exploration and monitoring systems. In particular we address four subproblems in building such robots in this thesis: realtime online semantic perception to extract the context of the observations being made by the robot, summarizing the robot's experience concisely using a few observations, the online decision theoretic process of identifying samples for further inspection or return, and planning information driven exploration paths that results in better semantic perception models.

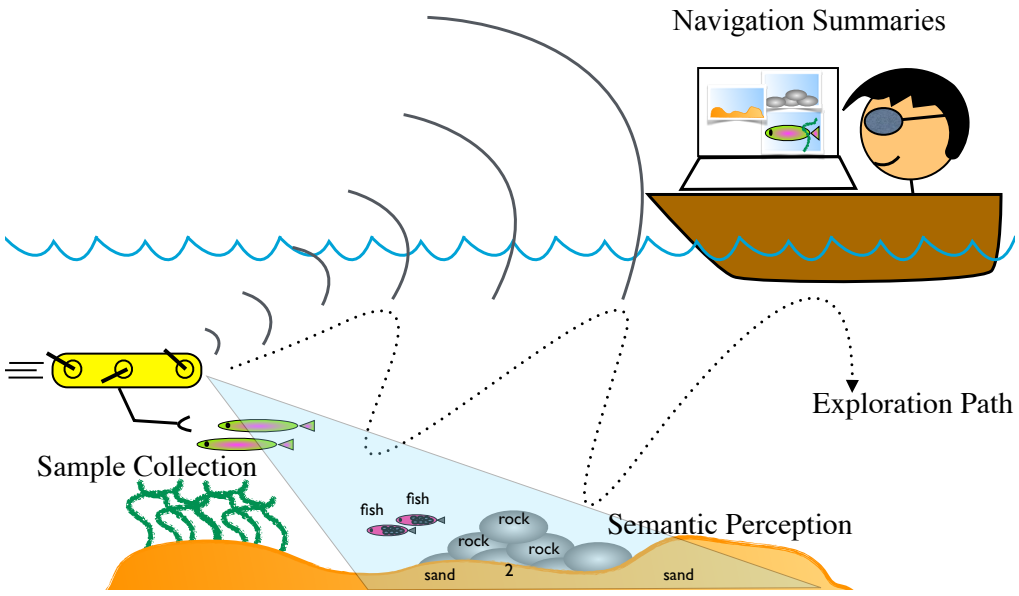


Figure 1.1: Various subproblems that needs to be solved while building exploration robots, which we examine in this thesis.

1.1 Seeing the Bigger Picture

One of the big challenges in building exploration robots is to detect when what is being observed is new and interesting, from the stream of low level sensor data that is being collected by the robot. For example when we look at images of grass or bush, every image will be different if we compare them in terms of low level local features such as pixels, edges, even though conceptually they might have the same content. We would like our robot to be intelligent enough so that it can see the world at a higher level of abstraction, because without it every image will appear different even while having similar thematic content or identical objects within it. One way to address this problem is through the use of *topic models*.

Topic modeling techniques have been used to analyze large collections of text documents. Using topic modeling, it is possible to automatically extract semantically relevant underlying themes behind the text documents, and describe each document in terms of these high level themes. Intuitively, each topic is represented by a sparse set of words which are likely to co-occur in a document. For example, words such as 'protein', 'DNA', 'amino-acid' are likely to co-occur and hence their presence indicates a common topic (which we might label as 'biochemistry', whereas words such as 'algorithms', 'complexity', 'NP-hard' indicate the

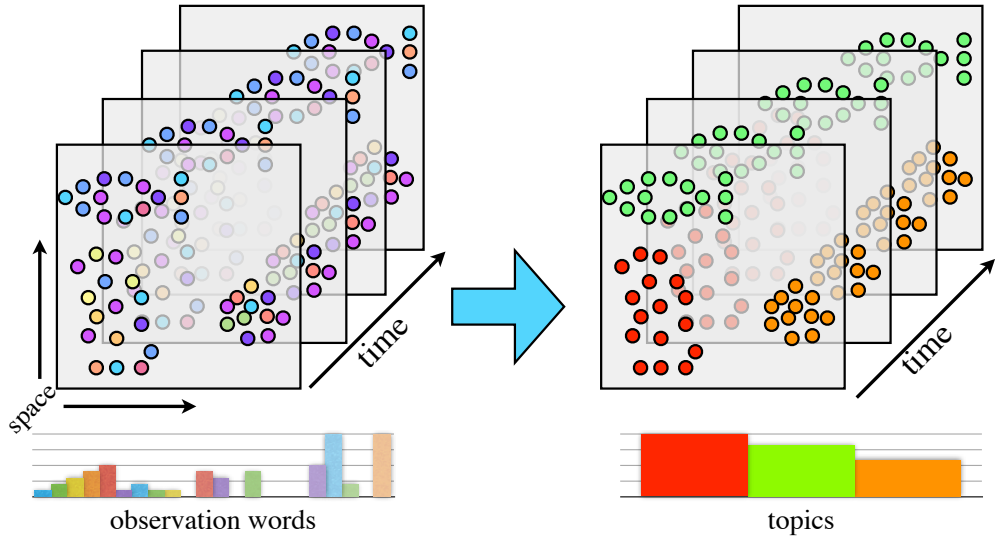


Figure 1.2: Given low level observations (words), we would like to compute topic labels that are representative of high level spatiotemporal phenomena. The distribution of these topic labels can then be used as a semantic scene descriptor.

presence of a different topic (which we might label as 'computer science'). A document which has all of these words together might indicate that the document has content from both of these topics, such as a paper on bioinformatics. What is notable is that it is possible to do topic modeling unsupervised, without any prior knowledge about possible different topics.

In the context of visual data, topic modeling can be used to extract high level visual scene constructs given the low level visual features, and then use these high level visual constructs (topics) to describe the observed scene. The analogue of a text word in visual data is a visual word, which could represent a low level visual pattern, a color, or a sharp change in color. For example, consistent co-occurrence of scales like texture, gray color, and triangular tail shape in observed data indicates a common underlying topic – a fish. Figure 1.2 illustrates this idea visually.

One fundamental difference between topic modeling of text data and visual data as observed by a robot is that visual words have spatiotemporal context. Unlike text documents which are typically assumed to be independent of other text documents, consecutive images observed by a robot are likely to contain similar topics. Moreover, within an image, visual words which are spatially close to each other are likely to have the same underlying topic label.

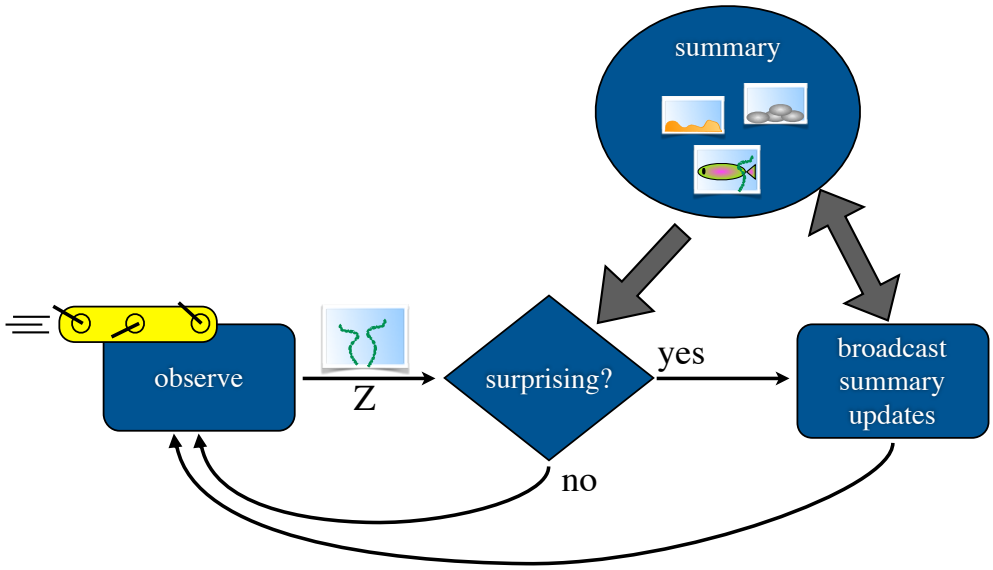


Figure 1.3: Online navigation summaries are useful for providing mission updates, especially when communication bandwidth is limited.

Taking advantage of this correlation is important for making sure we can extract consistent topic labels for the observed visual data. Another important consideration is that observation data is streaming, and for topic modeling to be useful in a robotic context we must be able to compute topic labels for incoming observations in realtime, while still ensuring the topic labels are globally consistent. In Section 3 we present ROST - a realtime online spatiotemporal topic modeling framework which extends the previous work on topic modeling to enable realtime operation, while taking advantage of spatiotemporal correlations in the data.

Work presented in Chapter 3 has appeared in [41, 42] and borrows much of its text and results.

1.2 A Summary of Surprises

The result of any exploratory voyage is a large amount of data, which at the end is likely to be analyzed by a human operator. Summarization of this data is thus essential for it to be useful, however this must be done in a manner which includes interesting observations. Imagining our data as a collection of points in some high dimensional space, the summarization task is

then identifying a small subset of k points which are representative of the collected data. It is common to use a clustering algorithm such as k-means to summarize data, however this is not always useful for producing navigation summaries. A k-means summary would consist of points which are representative of the mean appearance of what was observed. For example, a k-means summary of an ocean floor exploration mission would probably only contain images of lifeless sand or rock. Instead, a summary containing outliers that are surprising or interesting is more useful for the purpose of monitoring or surveillance. In Section 4.3 we propose such a summarization strategy, which works by choosing k points such that the maximum cost of a data point, defined as the distance of that point to the closest summary point, is minimized. Such an extremum summary would ensure that every point in the dataset is close to at least one of the summary points. If we measure the distance between observation points in information theoretic topic space, our summaries will be thematically meaningful.

Extremum summaries are representative of the diversity of what has been observed, and hence are a useful tool to send back to the operator as exploration mission updates. However, due to on-board computational constraints, and communication constraints because of the distance (planetary missions) or the medium (underwater missions), it is most often impossible to recompute and send the summary at each time step. In such a scenario it is desirable to have an online summarization algorithm that can communicate to the operator only the changes to the summary by sending back only the surprising observations that exceed a certain surprise threshold score. Figure 1.3 shows a general overview of such an online summarization algorithm. If there is no prior knowledge about the environment, then this threshold score must by chosen dynamically. In Section 4.4 we present such an online summarization algorithm, and analyze its behavior.

Work presented in Chapter 4 has appeared in [36, 37, 39], and borrows much of its text and results.

1.3 Sample Collection

Collecting samples is an important part of many exploration missions. For a planetary exploration mission, we might want to identify locations to drill, or select rocks for sample return. For an underwater exploration mission we might want to collect water samples to bring to

the surface. In all such cases in which the act of collecting a sample is irrevocable, and there is no prior knowledge about the environment, we can pose the sample collection problem as an instance of the Secretaries Hiring Problem.

Secretaries Hiring Problem refers to the hypothetical task of hiring the best candidate from a given set of interviewing candidates. There are n applicants who have signed up for the interview. After interviewing each candidate we can rank them relative to all other candidates seen so far, and we must either hire or reject the candidate immediately after the interview. We are not allowed to go back and hire a previously rejected candidate. In Chapter 5 we introduce an extension to the secretaries hiring problem in which the goal is to hire the k top candidates, and present an optimal solution. We demonstrate how such an approach can be used to select information rich observations.

Work presented Chapter 5 has appeared in the following publications: [34, 35] and borrows much of its text and results.

1.4 Modeling Curiosity for Learning Better Perception Models

A simple approach to collecting environment data is to use some kind of a space filling path through this environment. This is however not ideal because the amount of information collected about different spatial phenomena is proportional to their area. Underwater, this might mean that most of the data collected only contains observations of lifeless sand or rocks, and very occasionally we might have a few samples corresponding to something interesting such as thermal vents, marine life, or archeological sites. A better strategy for collecting data might be to have the robot behave like an explorer or a vacationing tourist; moving swiftly over regions with familiar sights, while paying much more attention, i.e., collecting much more data when something novel or interesting is in view. In Chapter 6 we show that such a strategy collects higher quality data, which when used to train a topic model can distinguish between terrains much more effectively than other techniques such as simple space coverage.

Work presented in Chapter 6 has appeared in the following publication: [42, 40] and borrows much of its text and results.

1.5 Thesis Contributions

The research contribution of this thesis can be summarized as below:

- *Realtime Online Spatiotemporal Topics:* We developed a topic modeling technique that extends the previous work on online LDA, by taking advantage of the spatiotemporal context of the observed data while using a novel Gibbs sampler to enable realtime processing of the sensor data stream. We showed that our approach results in better topic labels that have higher mutual information with ground truth labels, compared to topic models that do not take into account the spatiotemporal context. Moreover, we showed that the perplexity of the online topic model using the proposed Gibbs sampling is competitive with batch Gibbs sampling.

- *Navigation Summaries:* We formulated the navigation summary problem as an instance of the k-centers problem, and proposed an online algorithm which approximates the optimal solution. Experiments with several different real world datasets showed that the performance of the online summarization algorithm is statistically indistinguishable from the offline solution.

- *k-Secretaries Hiring Problem:* We proposed a new formulation of the sample collection problem as an instance of the Secretaries Hiring problem. We derived an optimal solution to the generalized k secretaries hiring problem, in which the goal is to select all top k candidates.

- *Curiosity Based Exploration:* We developed a formulation of exploration problem that aims to maximize information gain in topic space, which we showed that it results in learning better terrain models of a landscape. We validated the effectiveness of the proposed exploration technique over candidate techniques by computing mutual information between the terrain maps generated through the use of the learned terrain model, and hand labeled ground truth, on three different datasets.

- *Implementation & Demonstration:* We implemented the proposed topic modeling and exploration technique on an underwater robot. We showed that the emergent behavior of the robot has a striking similarity to that of biological organisms, and that with no prior training, we can observe behaviors such as staying within the confines of a coral reef while collecting data, or following a diver.

Chapter 2

Background

This thesis builds upon several existing ideas. In this chapter we will discuss some of these ideas which are most relevant to understanding this thesis.

In Section 2.1 we begin by reviewing Dirichlet distributions and their properties. We use these distributions to model how various objects and terrain types are distributed in a scene. In Section 2.2 we discuss the idea of topic modeling which has been used to extract high level descriptors for text and image data. In Section 2.3 we discuss various low level representations of an image that can be used for topic modeling. In Section 2.4 we discuss a Bayesian approach to modeling surprise, which is a useful idea for building curious robots. In Section 2.6 we discuss various formulations of the exploration problem, and how they differ from each other. In Section 2.7 we discuss various robotic platforms which have been used for the exploration task in the literature, and the three robots which we have used to conduct experiments for this thesis.

2.1 Dirichlet Distributions

Dirichlet distributions are most commonly used as priors for mutinomial or categorical distributions. A categorical distribution is used to describe the potential outcome of an experiment with K discrete outcomes, whose probabilities are given individually. A multinomial distribution generalizes the categorical distribution by allowing multiple runs of the experiment.

For example, we can use a categorical distribution to model a single roll of a die, and a multinomial for $n > 1$ rolls. To simplify notation, we will refer to both these distributions as discrete distributions.

We can use a K-tuple $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ to model the probability distribution for each of the K faces of a die. If however we would like to model the generative process which produces these dice, we can use a Dirichlet distribution, which is a distribution over discrete distributions; i.e., a sample from a Dirichlet distribution is the multinomial parameter vector $\boldsymbol{\theta}$.

$$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K) \tag{2.1}$$

$$\boldsymbol{\theta}|\boldsymbol{\alpha} = (\theta_1, \ldots, \theta_K) \sim \text{Dirichlet}(\boldsymbol{\alpha}) \tag{2.2}$$

$$\boldsymbol{x} = (x_1, \ldots, x_K) \sim \text{Discrete}(\boldsymbol{\theta}) \tag{2.3}$$

Here $x \sim Y$ means x is a sample from distribution Y, and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ is the Dirichlet parameter vector, sometimes also known as the concentration parameter.

A very common special case of Dirichlet distribution is the symmetric Dirichlet distribution, in which all of elements in the parameter vector $\boldsymbol{\alpha}$ have the same value. Symmetric Dirichlet distributions are often used for placing priors on discrete distributions, when we do not have prior knowledge regarding the k components of the distribution.

The Dirichlet distribution is given by the equation:

$$\text{Dirichlet}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{B}\prod_{k=1}^{K}\theta_k^{\alpha_k - 1}, \tag{2.4}$$

where $B = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)}$ is a normalizing constant. For a symmetric K dimensional Dirichlet distribution, this simplifies to:

$$\text{Dirichlet}(\boldsymbol{\theta}|K, \alpha) = \frac{1}{B}\prod_{k=1}^{K}\theta_k^{\alpha - 1}, \tag{2.5}$$

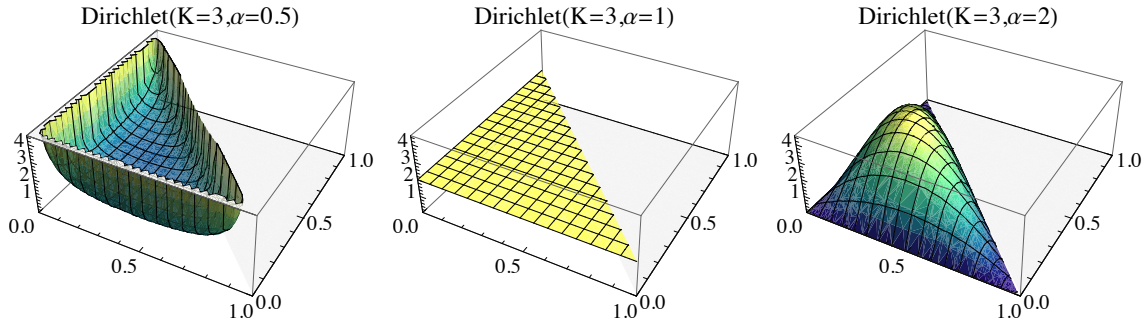


Figure 2.1: Symmetric Dirichlet probability density function for different values of parameter α. With $\alpha < 1$, most of the mass is on the boundary of the probability simplex, implying that a discrete distribution sample is likely to have low entropy; with $\alpha > 1$, most of the mass is concentrated in the center of the probability simplex, which implies that the sampled distribution is likely to have high entropy, with mass distributed uniformly over the K categories; with $\alpha = 1$, all discrete distributions are equally likely.

Figure 2.1 shows a plot of a symmetric Dirichlet probability density function (PDF) for $K = 3$ and three different values of α. The probability simplices shown in the plots demonstrate how changing α can be used to control the sparseness of the samples from the Dirichlet. The most interesting case is when $\alpha < 1$, which implies that the discrete distributions sampled from the Dirichlet distribution are likely to be sparse with high entropy. This is the most of common reason for using Dirichlet priors while modeling discrete phenomenon. Many natural phenomenon are discrete and sparse. For example, if we are trying to model what objects a robot might see when it is out exploring the world, then we know that although the world consist of many different objects, at any given time we might only see a small subset of possible objects.

Figure 2.2 shows sample draws from a Dirichlet distribution over six categories. For $\alpha > 1$ we see that draws from the Dirichlet distribution are essentially uniform discrete distributions over the six categories. The more interesting case of $\alpha < 1$ demonstrates the sparseness prior enforced by the Dirichlet. We see that most of the sampled multinomials have most of the mass on only a few categories.

2.1.1 Expected Entropy of Samples

The symmetric Dirichlet parameter α can be used to control the expected entropy of the samples from the discrete distribution. A high α value corresponds to a discrete distribution with high entropy and vice versa. In fact the entropy of the discrete distribution has a sharp peak around the expected entropy, with low variance, even for high values of K [82].

The expected entropy of the symmetric Dirichlet draws $(\theta_1, \ldots, \theta_K) \sim \text{Dirichlet}(\alpha)$ as computed by [111], is given by:

$$\begin{aligned} \mathbb{E}\left[H(\alpha, K)\right] &= -\mathbb{E}[\log \theta_i] && (2.6) \\ &= \Psi(K\alpha + 1) - \Psi(\alpha + 1), && (2.7) \end{aligned}$$

where $\Psi(x) = \frac{d}{dx} \log \Gamma(x)$ is the digamma function. The plots in Figure 2.3 show how the expected entropy of the Dirichlet changes as a function of α and K.

2.1.2 Dirichlet as a Conjugate Prior to a Discrete Distribution

The Dirichlet probability density function (pdf) is the conjugate prior of a discrete distribution in Bayesian inference. This means that if the prior hypothesis is modeled by a Dirichlet distribution, and the evidence is in the form of a discrete distribution, then the posterior is guaranteed to be a Dirichlet distribution. A discrete distribution is defined by a K-tuple of counts corresponding to each of the K categories:

$$\text{Discrete}(\boldsymbol{x}|\boldsymbol{\theta}) \propto \prod_{k=1}^{K} \theta_k^{x_k}, \tag{2.8}$$

where x_k is the number of occurrences observed for each of the K categories, and θ_k is the probability of a sample falling into category k. If θ has a Dirichlet prior:

$$\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha}) \tag{2.9}$$

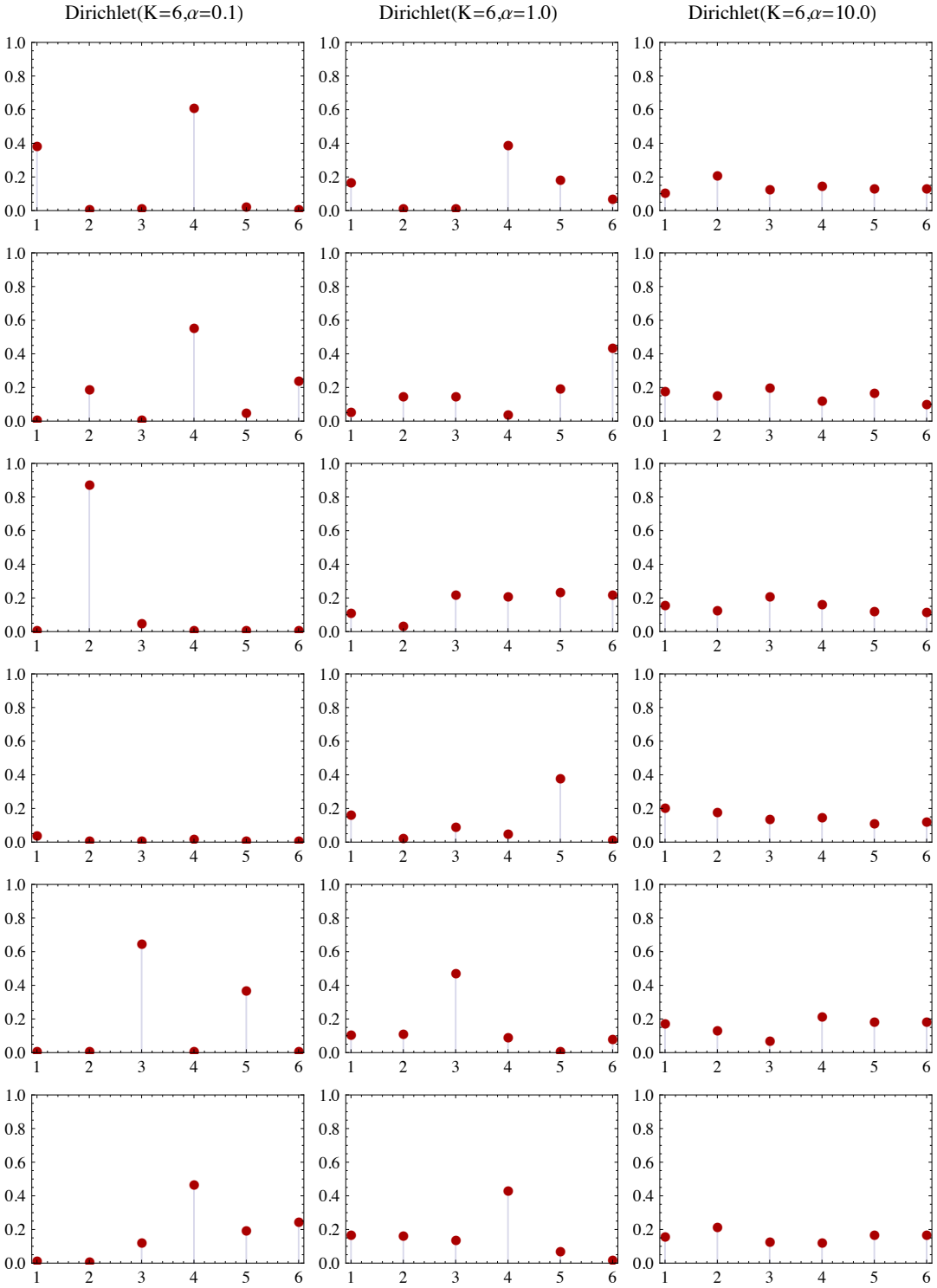


Figure 2.2: Example of draws from a symmetric Dirichlet distribution for different values of parameter α. High α gives multinomials with close to uniform probability distribution across the six categories. Values of $\alpha < 1$ give sparse multinomial samples with most of the mass on only a few categories.

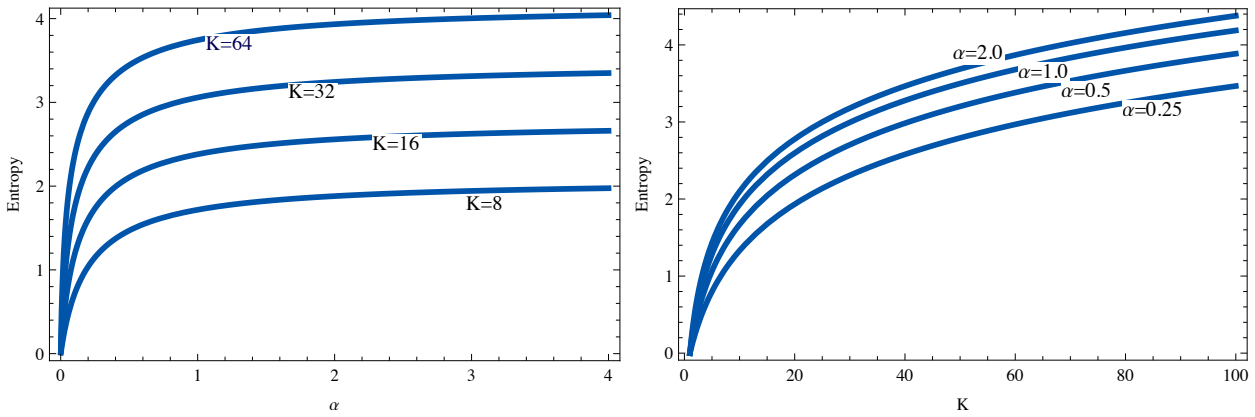


Figure 2.3: The Expected entropy(in base e) of symmetric Dirichlet draws as a function of the parameter α, and the number of categories K.

then the posterior $\boldsymbol{\theta}$ is drawn from:

$$\begin{aligned}
\boldsymbol{\theta}|\boldsymbol{x},\boldsymbol{\alpha} &\sim \text{Discrete}(\boldsymbol{x}|\boldsymbol{\theta})\ \text{Dirichlet}(\boldsymbol{\theta}|\boldsymbol{\alpha}) && (2.10)\\
&= \frac{1}{B}\prod_{k=1}^{K}\theta_k^{x_k}\prod_{k=1}^{K}\theta_k^{\alpha_k-1} && (2.11)\\
&= \frac{1}{B}\prod_{k=1}^{K}\theta_k^{x_k+\alpha_k-1} && (2.12)\\
&= \text{Dirichlet}(\boldsymbol{\alpha}+\boldsymbol{x}), && (2.13)
\end{aligned}$$

where B is the normalizing constant. It is interesting to note that intuitively, we can interpret the prior $\boldsymbol{\alpha}$ as virtual counts that can just be added the observation counts to compute the posterior.

2.2 Semantic Perception via Topic Modeling

From Plato's theory of Forms [88] to Hume's bundle of perceptions [56], philosophers have long been debating how we perceive our world semantically. It is only recently, however, that we have been able to build computational models of semantic perception, where we can extract high level latent information out of low level observations.

Observations can be of many different types. In text documents, the low level observations

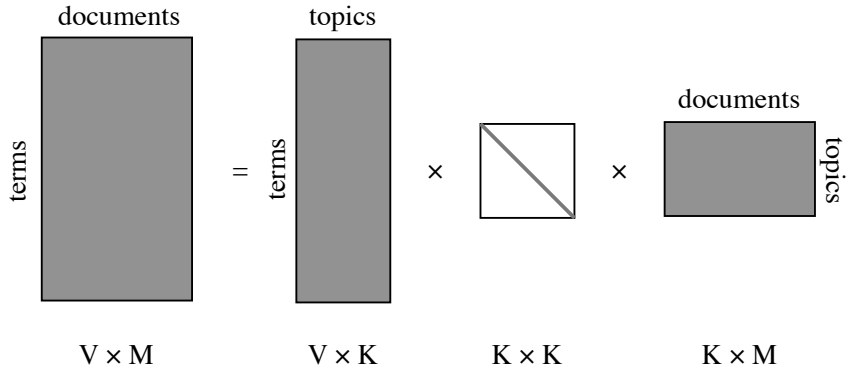


Figure 2.4: Latent Semantic Analysis(LSA) uses Singular Value Decomposition to decomposes the term-document matrix into three smaller matrices. Here V is the number of terms or the vocabulary size, M is the number of documents, and $K = \min(V, M)$.

are the words, which when taken collectively might describe a high level idea. In image data, low level observations might be the color, texture, edges and shapes in the image, which together might represent a higher level idea such as an object (see Figure 1.2). Topic modeling techniques aim to find these high level ideas (topics) from low level observations (words). Topic models are useful in many search and retrieval scenarios where we would like to find a document based on thematic similarity rather than exact similarity. For example, given a picture of a car, we would like to find other pictures with cars in them rather then just finding other picture with the same car in the same pose. For text documents, we might want to search for documents that are about similar topics, rather than doing exact word searches.

Many different topic models have been developed thus far. We will discuss a them in the following sections.

2.2.1 Latent Semantic Analysis (LSA)

Deerwester et al. [23] first proposed a singular value decomposition (SVD) based solution to topic modeling. Given a large term-document matrix of observed word occurrences, LSI decomposes this data into a three smaller matrices:

$$X = \Phi S \Theta, \tag{2.14}$$

where X is the word occurrence matrix for M documents and V possible terms (vocabulay size), Φ is a $V \times K$ matrix with orthonormal columns that describe each of the K topics using a V dimensional term vector, Θ is a $K \times M$ matrix with orthonormal rows that describes all M documents using K dimensional topic vectors, and S is a $K \times K$ diagonal matrix (see Figure 2.4). The diagonal matrix S has has at most $K = \min(V, M)$ singular values, which are all positive and (by convention) in decreasing order.

The beauty of this approach is that if we keep only the top $k < K$ values in S, and remove the corresponding columns and rows in Φ and Θ, this gives us a representation of the documents in a lower dimensional space. This works because these k dimensions correspond to the directions along which there is most variance in the data. Hence, describing documents using just these k dimensions would be sufficient to explain most of the variance in the data.

Let $\widetilde{S}$ be the diagonal matrix of size $k < K$, and $\widetilde{\Phi}$, $\widetilde{\Theta}$ be the corresponding smaller matrices, then we have:

$$X \approx \widetilde{X} = \widetilde{\Phi}\widetilde{S}\widetilde{\Theta}, \tag{2.15}$$

where $\widetilde{X}$ approximatex X in the least square sense. Moreover, we can treat the columns of $\widetilde{S}\widetilde{\Theta}$ as points in a k dimensional space describing each document. Taking dot products of these k dimensional vectors gives us a semantic distance between the documents.

The columns of the $\widetilde{\Phi}\widetilde{S}$ matrix on the other hand can be treated as the description of the k topics.

Papadimitriou et al. [84] have shown that LSA does indeed recover the latent structure in the corpora and that it handles synonyms well.

2.2.2 Probabilistic Latent Semantic Analysis (PLSA)

Despite the success of LSA, it has some significant shortcomings. First, the L_2 norm approximation used by LSA is not appropriate for word count data, and is hard to justify. Second, because LSA assumes that topic descriptors are orthonormal vectors, it is unable to handle polysemy. Hence, modeling a word such as 'crane' to mean either the bird or the construction equipment is not possible. In the vision context this would mean that it might not be possible

to use the same low level features in the description of multiple high level topics representing different objects.

Hoffman [53] introduced the idea of probabilistic Latent Semantic Analysis(PLSA) for text documents, which models the probability of observing a word w_i in a given document M as:

$$\mathbf{P}(w_i = v|d_i = m) = \sum_{k=1}^{K} \mathbf{P}(w_i = v|z_i = k)\mathbf{P}(z_i = k|d_i = m), \tag{2.16}$$

where w_i takes a value between $1 \dots V$, where V is the vocabulary size; z_i is the hidden variable or topic label for w_i that takes a value between $1 \dots K$, where K is the number of topics, and is much smaller than V; and d_i is the document number, which can take a values between $1 \dots M$, where M is the total number of documents.

The central idea is the introduction of a latent variable z, which models the underlying topic, or the context responsible for generating the word. Each document M in the given corpora is modeled using a distribution $\theta_m(k) = \mathbf{P}(z_i = k|d_i = m)$ over topics, and each topic is modeled using a distribution $\phi_k(v) = \mathbf{P}(w_i = v|z_i = k)$ over the set of vocabulary words. During the training phase, these distributions are learned directly using an EM algorithm.

The distribution of topics in a document gives us a low dimensional semantic description of the document, which can be used to compare it semantically with other documents. The problem with this approach is that since the dimensionality of the model is very large, a lot of training data is required. Moreover, it is easy to overtrain for a given data set.

2.2.3 Latent Dirichlet Allocations (LDA)

Latent Dirichlet Allocation (LDA), proposed by Blei et al. [11] mitigates the training problem by placing a Dirichlet prior on θ. Subsequently Griffiths et al. [49] proposed placing a Dirichlet prior on ϕ in addition to θ. Placing Dirichlet priors, as discussed in Section 2.1, can encourage the distributions to be sparse, which has been shown to give semantically more relevant topics. The generative process for words in documents according to LDA is as following:

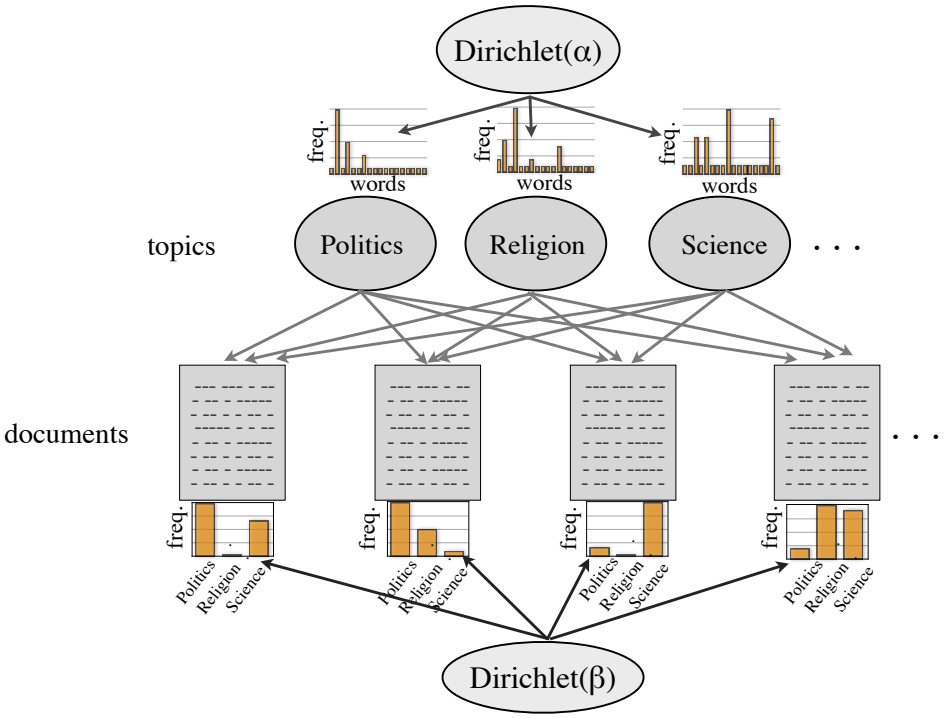


Figure 2.5: Latent Dirichlet Allocation (LDA) is a popular technique for describing a set of documents as a mixture of topics, which are themselves described as distribution over words. Both the topic-word and document-topic distributions have Dirichlet priors, which biases these distributions towards being sparse.

1. Word distribution for each topic k:

$$\phi_k \sim \text{Dirichlet}(\beta).$$

2. Topic distribution for words in document m :

$$\theta_m \sim \text{Dirichlet}(\alpha).$$

3. Topic label for a word in document m:

$$z \sim \text{Discrete}(\theta_m).$$

4. Word label:

$$w \sim \text{Discrete}(\phi_z).$$

Here $y \sim Y$ implies that random variable y is sampled from the distribution Y, and z is the topic label for the word observation w. Each topic k is modeled by the distribution ϕ_k over V possible words in the observation vocabulary. Figure 2.5 visualizes this generative model.

Griffiths et al. [49] proposed a collapsed Gibbs sampler for LDA, where the state is the topic assignments for all the words in all the documents, which is different from the original variational approximation based approach proposed by Blei et al. [11].

Given the ith word observation w_i in document d_i, the posterior topic distribution is given by:

$$\mathbf{P}(z_i = k | w_i = v, d_i = m) \propto \frac{n^v_{k,-i} + \beta}{\sum_{v=1}^{V}(n^v_{k,-i} + \beta)} \cdot \frac{n^k_{m,-i} + \alpha}{\sum_{k=1}^{K}(n^k_{m,-i} + \alpha)}, \tag{2.17}$$

where $n^w_{k,-i}$ counts the number of words of type w in topic k, excluding the ith word, and $n^k_{m,-i}$ is the number of words with topic label k in document m, excluding the topic label for the ith word, and α, β are the Dirichlet hyper-parameters, K is the total number of topics, and V is the vocabulary size.

2.2.4 Topic Modeling of Visual Data

Given images of scenes with multiple objects, topic modeling has been used to discover objects in these images in an unsupervised manner. Bosch et al. [12] used PLSA and a SIFT based [76] visual vocabulary to model the content of images, and used a nearest neighbor classifier to classify the images.

Fei-Fei et al.[29] have demonstrated the use of LDA to provide an intermediate representation of images, which was then used to learn an image classifier over multiple categories.

Instead of modeling the entire image as a document, Spatial LDA (SLDA) [110] models a subset of words, close to each other in an image as a document, resulting in a better encoding of the spatial structure. The assignment of words to documents is not done *a priori*, but is instead modeled as an additional hidden variable in the generative process.

Geometric LDA (gLDA) [86] models the LDA topics using words that are augmented with spatial position. Each topic in gLDA can be visualized as a pin-board where the visual words are pinned at their relatively correct positions. A document is assumed to be generated by first sampling a distribution over topics, and then for each word, sampling a topic label from this distribution, along with the transformation from the latent spatial model to the document (image). These transformations are all assumed to be affine, to model the change in viewpoints.

LDA has been extended to learn a hierarchical representation of image content. Sivic et al.[96] used hierarchical LDA (hLDA) [10] for automatic generation of meaningful object hierarchies. Like LDA, hLDA also models documents as a mixture of topics; however, instead of the flat topics used in LDA, topics in hLDA correspond to a path in a tree. These topics become more specialized as they travel farther down from the root of the tree.

2.3 Visual Words

2.3.1 Pixels

Probably the simplest visual words are the pixels in an image captured by the robot. For color perception, typically the cameras capture the images as intensity values in the red, green and blue (RGB) color channels. The RGB color space is the most common representation for color images; however, many other color spaces exist.

The Hue-Saturation-Value (HSV) color space is useful in modeling the color of an object due to an independent hue channel. HSV color space is modeled as a cylinder, with the hue channel represented by the angles along the central vertical axis, "saturation" corresponds to radial distance from the central vertical axis, and "value" or brightness corresponds to the distance along the vertical axis.

The Lab color space [57] is an opponent color space designed to approximate human vision. Lab colors are perceptually more uniform, i.e., same amount of changes in the color value should have similar perceptual difference. In a computer vision context, it is useful to use the Lab color space when the goal is to approximate human behavior.

2.3.2 Gabor Filter Based Textons

A 2D Gabor filter is characterized by a preferred spacial frequency and an orientation. It is essentially the product of a Gaussian and a Sine function of a given frequency, with a given orientation. An example is shown in Figure 2.6. Gabor filters have been used extensively for texture classification [22, 108, 59, 50].

Given a bank of Gabor filters of different scales and orientation, such as ones shown in Figure 2.6, we can construct a vector containing filter responses for each pixel. Clustering these response vectors (typically using k-means) would give us a small set of prototype response vectors that are called *textons* [60, 78]. Given such a vocabulary of textons, we can then map filter response vectors from an image to the closest texton in the vocabulary, giving us a texton word description of that image.

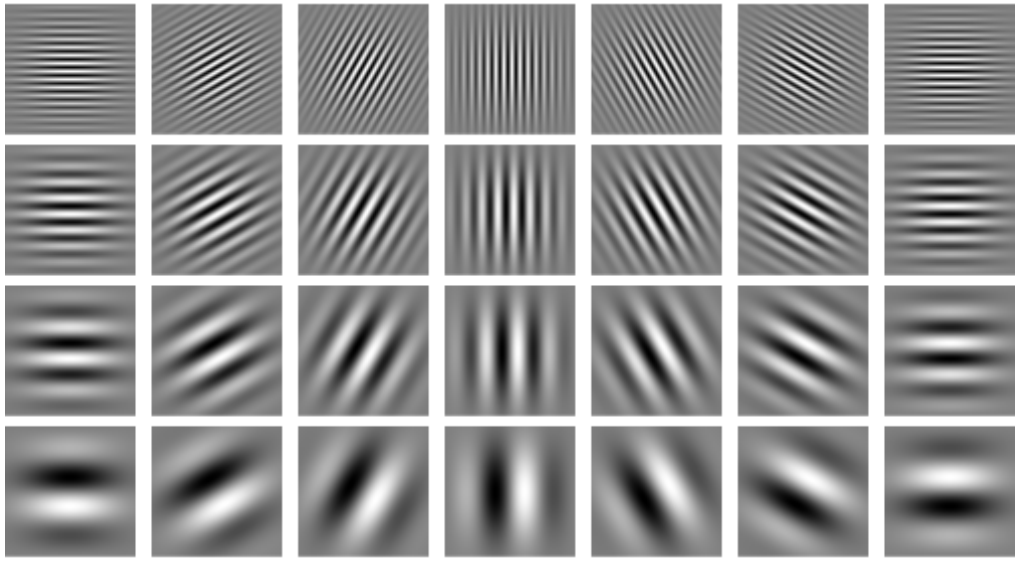


Figure 2.6: Sine Gabor filters with orientation $0^o, 30^o, \cdots, 180^o$ and wavelength 4,8,16,32 pixels.

2.3.3 Local Visual Features

Both SIFT [76] and SURF [6] features are typically stored using 128 or 64 dimensional floating points vectors, and doing nearest neighbor queries require computing L1 distances between these two vectors, which can be computationally expensive for a large number of features. More recently, binary feature descriptors such as Oriented BRIEF (ORB) [90], have been shown to perform well. The distance between two binary feature vectors can be computed by taking the Hamming distance between the bit strings, which can be implemented very efficiently using XOR operations that are available on most CPUs. Hence, such binary feature descriptors are much more suitable for applications requiring realtime operation.

2.3.4 Building Visual Vocabularies

Topic modeling of images requires that the general idea of a textual word be replaced by visual words. One approach to generate these visual words from visual features is that described by Sivic et al. [98]. To generate a visual vocabulary, we first extract visual features such as SIFT [76] or SURF [6] from an unrelated dataset, with high visual diversity. These features are then clustered using the k-means algorithm, with V clusters corresponding to the desired vocabulary size. The cluster centers of these V clusters represent the visual words in the

vocabulary. Now, to extract visual words from a given image, first we extract its visual features, and then map each feature to the index of the closest visual word in the vocabulary.

2.4 Bayesian Surprise

Itti and Baldi [58] formally defined Bayesian surprise in terms of the difference between posterior and prior beliefs about the world. They showed that observations which lead to high Kullback-Leibler(KL) divergence [72] between posterior and prior visual appearance hypotheses, are very likely to attract human attention.

The relative entropy or KL divergence between two probability mass functions $p(x)$ and $q(x)$ is defined as:

$$d_{\mathrm{KL}}(p\|q) = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}. \tag{2.18}$$

The KL divergence can be interpreted as the inefficiency in coding a random variable from distribution p, when assuming its distribution to be q.

In this thesis we represent surprise with the symbol ξ:

$$\xi = d_{\mathrm{KL}}(\text{posterior}\|\text{prior}). \tag{2.19}$$

Suppose we have a set of summary images $\boldsymbol{S} = \{S_i\}$, which visually summarizes all observations so far. Let F be a random variable representing the presence of some visual feature. For example, F could represent the presence of a given color, or a visual word [98, 97]. Let π^- be the prior probability distribution over all such features.

$$\pi^- = \mathbf{P}(F|\boldsymbol{S}) \tag{2.20}$$

Similarly, we can define the posterior probability distribution π^+, after observing a new image Z:

$$\pi^+ = \mathbf{P}(F|Z,\boldsymbol{S}). \tag{2.21}$$

Using Itti and Baldi's definition of surprise, we can then define *surprise* ξ in observing an image Z, given a summary $\boldsymbol{S}$ as:

$$\xi(Z|\boldsymbol{S}) \;=\; d_{\mathrm{KL}}\left(\pi^{+}\middle\|\pi^{-}\right). \tag{2.22}$$

The surprise $\xi(Z|\boldsymbol{S})$ can be interpreted as the amount of information gained by observing Z. Ideally, we would like to choose a summary set such that the information gained after observing any random image from the terrain is small. In such a case, this would imply that the summary images already contain most of the information about the world.

2.5 Navigation Summaries

Given the vast amount of data that might be collected by a robot on a mission, there is a need for algorithms that can summarize this data is a meaningful manner. A navigation summary is a synopsis of observations made by a robot on a trajectory. The term *vacation snapshot problem* [14] was coined to refer to the algorithm task analogous to what many tourists face: summarize their vacation using a small set of images. The task of generating a navigation summary can be seen as a generalization of the vacation snapshot problem, as it incorporates not just image data, but any other sensor data, including location.

We discuss below a few summarization techniques which have relevance in the robotics context.

2.5.1 PCA Residual Error Based Summaries

Dudek and Lobos [26] used Principal Component Analysis (PCA) [107] to compute a low dimensional representation of the images in the current summary, and then used the residual reconstruction error of a new image to decide whether to include it into the summary or not. With each inclusion of a summary image, a new PCA basis set was computed. The proposed inclusion criterion was a linear function of visual novelty, measured using PCA reconstruction residual error, as well as geographic novelty, measured as the Hausdorff distance [52] of the image from the location of other images in the summary.

2.5.2 Star Clustering

Given a distance threshold, star clustering [1] first builds a neighborhood graph of the input samples, in which two nodes are connected if their distance is less than the given threshold. The Star clustering algorithm then repeatedly finds the nodes with the highest degree, and removes them (along with their neighbors) from the graph. These high degree nodes are then the cluster centers which form the summary.

Paul et al. [85] have used Star clustering to build navigation summaries. Each image was described using a topic distribution, computed using Latent Dirichlet Allocation (discussed in Section 2.2.3).

2.5.3 Gaussian Mixture Models

Summarizing benthic (sea floor) images is an especially difficult problem due to the general lack of order, symmetry, and orientation of the visual data. Steinberg et al. [102] used a Gaussian mixture model to cluster benthic stereo images, while using a Variation Dirichlet Process [74] to automatically infer the number of clusters. Although this work did not use location information in the clustering process, the resulting cluster labels were shown to be spatially contiguous, indicating correctness. The computed labels were shown to outperform those obtained with spectral clustering and EM Gaussian mixture models, when compared with hand labeled ground truth data.

2.5.4 Landmark Detection

Generating summaries from observed image data is related to the problem of identifying landmark views in a view based mapping system. Ranganathan and Dellaert [89], have worked on the problem of identifying such a set of landmark locations, and then building a topological map from them. They modeled each observed image using a SIFT bag-of-words histogram, which they use to learn a Multivariate Polya posterior and prior model. The KL divergence between these two distributions gives us the "Bayesian Surprise" [58]. If this surprise score is a local maximum above the mean, then the images were selected.

Another example is the work on view-based maps by Konolige et al. [71]. In this work, the goal was to identify a set of representative views and the spatial constraints among them. These views are then used to localize the robot. One could consider these selected views as a summary.

FAB-MAP, proposed by Cummins et al. [20] perform loop closure by identifying common geographic origins of images, which might not be completely similar. This produces a clustering of images where the clusters correspond to geographic locations.

2.5.5 Video Summarization

Video summarization is a related problem, which mainly focuses on identifying cut-scenes. Truong and Venkatesh [106] provide a good review of many summarization strategies in the context of video data, and Valdes and Martinez [109] present a taxonomy for classifying these different techniques. A notable early example of video summarization is the work by Gong and Liu [45], who propose video summaries produced by exploiting principal components representation of the color space of the video frames. They used a set of local color histograms and computed a singular value decomposition (SVD) of these local histograms to capture the primary statistical properties (with respect to a linear model) of how the color distribution varied. This allowed them to detect frames whose color content deviated substantially from the typical frame.

2.6 Exploration

Gaining knowledge about our environment is a never-ending quest for humanity. Direct exploration by humans, although tempting, puts strong limitations on what can be explored. Fortunately, through the use of robotics, we can continue this tradition of exploration without putting human lives in danger. The use of autonomous robots for exploration is a necessity for space and ocean exploration [7], where there are strong communication bottlenecks.

In the following sections we briefly look at some common variants of the exploration problem.

2.6.1 Coverage of Known Environments

If we have prior knowledge about the world then perhaps the simplest form of exploration is coverage, where the goal is to make the robot pass through every point in the given spatial region of interest. If the space is free of obstacles, then we can simply use a zig-zag path, sometimes known as a boustrophedon path to cover the world. In the case of known obstacles, Choset et al. [17] proposed boustrophedon cell decomposition of the world such that each cell can be covered by a simple boustrophedon path; then, given this decomposition, a path can be planned through all the cells. This would result in complete coverage.

Mannadiar and Rekleitis [79] later proposed splitting some boustrophedon cells so that the robot does not need to move over previously covered cells, resulting in paths guaranteeing optimal coverage. These paths have been extended for use with the general class of non-holonomic robots, such as aerial vehicles [112].

2.6.2 Exploration for Improving Navigation

Navigating a robot through free space is a fundamental problem in robotics. Yamauchi [113] defined exploration as the "act of moving through an unknown environment while building a map that can be used for subsequent navigation". Yamauchi's proposed solution involved moving the robot towards the frontier regions in the map, which were described as the boundary between known free space and the uncharted territories.

If we have an inverse sensor model of the range sensor, it is possible to compute locations in the world which would maximize the utility of the sensor reading in resolving the obstacle position and shape. Grabowski [47] proposed such an exploration strategy in which the goal is to maximize the understanding of obstacles rather than the exposure to free space. In this approach, the robot identifies the location with the next best view, where a sonar sensor reading would have the greatest utility in improving the quality of the representation of an obstacle.

If there is no external localizer available to the robot, then it is desirable that the robot explores, maps, and localizes in the environment at the same time. Sim, Dudek and Roy [94] take the approach of finding trajectories at each step that explore new regions while

minimizing the localization uncertainty of the robot as it re-enters a previously mapped region.

Bourgault et al. [13] and Stachniss et al. [101] have proposed an exploration strategy which uses gradient ascent to move the robot towards areas of high entropy which would maximize map information gain, while still keeping the robot localized.

Kollar et al. [70, 69] formulated the exploration problem as a constrained optimization problem, where the goal is to find a path that maximizes map accuracy with the constraint of complete map coverage. To do this, the algorithm first identifies the locations on the map that are essential for coverage, and then uses these locations to constrain the trajectory that maximizes map accuracy.

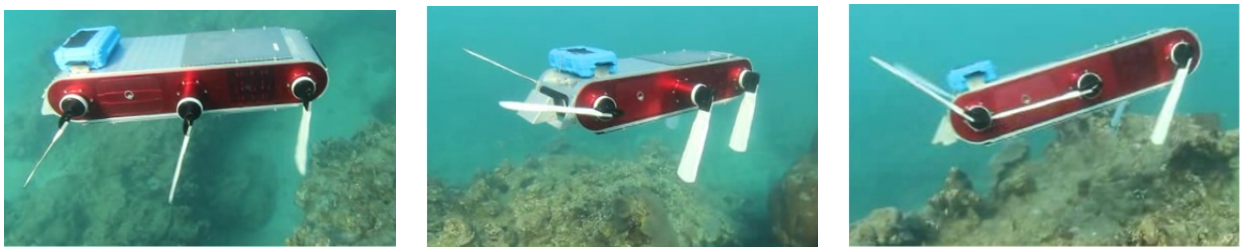
2.6.3 Exploration for Monitoring Spatiotemporal Phenomena

In underwater and aerial environments, obstacle avoidance and map building tasks are typically not of primary concern.

Binney et al. [9] have described an exploration technique to optimize monitoring spatiotemporal phenomena by taking advantage of the submodularity of the objective function. Bender et al. [8] has proposed a Gaussian process based exploration technique for benthic environments, which uses an experiment specific utility function. Das et al. [21] have presented techniques to autonomously observe oceanographic features in the open ocean. Hollinger et al. [54] have studied the problem of autonomously studying underwater ship hulls by maximizing the accuracy of the sonar data stream. Smith et al. [100] have looked at computing robot trajectories which maximize the information gained, while minimizing the deviation from the planned path.

2.7 Robotic Platforms

In this work we have used several different robotic platforms for collecting data and conducting experiments. The following sections gives a brief overview of these robots.



(a) (b) (c)

Figure 2.7: Pictures showing the flippers' angle due to the action of the autopilot system, during one of the sea trials. (a) the robot is performing a heave-up maneuver to maintain depth and attitude at zero forward speed. (b) the robot is executing a combined heave up, pitch up and slow forward speed maneuver. (c) the robot is performing a pitch-up maneuver at high speed.

2.7.1 Aqua Amphibious robot

Aqua [25, 91] is an amphibious six legged robot capable of autonomous operation. Although vehicles like Dorado [93] and Hugin [80] are extremely capable for long distance and deep water exploration, Aqua's propulsion is based on six flippers that can provide motion in five degrees of freedom, which is more suitable for tasks requiring high maneuverability, such as coral reef exploration. By using a novel combination of gaits, Aqua can move at various speeds while maintaining its orientation, despite external disturbances [33, 87]. Figure 2.7 shows various flipper poses for different desired swimming speeds.

Aqua is equipped with four cameras: a fish-eye camera in the front for environment awareness, a pair of stereo cameras in the front for depth perception, and a back camera. It is possible to take downward-looking images using a mirror mount. Aqua can be programmed underwater visually using these cameras [27] by showing it a sequence of fiducial markers such as ARTags [63].

Aqua is also capable of walking on land, and through the use of recently designed flippers [24] can perform both swimming and walking using the same set of flippers.

2.7.2 MARE Surface vehicles

MARE (Marine Autonomous Robotic Explorer) [43] is a sea-worthy and collapsible airboat robot capable of autonomous visual exploration of a given region over an ocean or a lake, and

Figure 2.8: MARE is an autonomous airboat made from off-the-shelf components, and is capable of exploring turbulent open water environments. Its discreet air-propelled design and hydrodynamically stable catamaran hull structure make MARE suitable for long-term deployment in all types of bodies of water. Since it has no moving parts in contact with the water, MARE can explore marine ecosystems while introducing minimal disturbance.

it has been used to collect data for several experiments in this thesis. It is a differential-drive, air-propelled vehicle with a downward-pointing camera, as shown in Figure 2.8. Its chassis follows a catamaran hull design, which makes it hydro-dynamically stable and also allows it to accommodate heavy and voluminous payloads. Its air propellers enable MARE to explore marine environments with minimal disturbance of the water surface and the aquatic life. These factors together make MARE a favorable choice as a sea-worthy surface exploration platform, with numerous application possibilities.

To manage the execution of these exploration strategies while automatically taking into account constraints such as range of operation, wireless connectivity to homebase, and battery levels, MARE uses a powerful software framework and accompanying programming tool called Graphical State Space Programming (GSSP) [75]. GSSP allows the robot operator to program execution plans for experiment sessions, where each plan can comprise a set of location-specific activities, different reactive behaviors, as well as various fail safe mechanisms.

Figure 2.9: Unicorn UAV

2.7.3 Unicorn Aerial Vehicle

The Unicorn UAV is a kite-sized fixed-wing aerial vehicle with a 1 m wingspan. It operates at an average ground speed of 14 m/s for up to 30 minutes of flight time. This vehicle is equipped with multiple sensors, including IMU, GPS, and pressure-based altitude and speed sensors. These devices are integrated with an on-board autopilot micro-processor, which uses them to navigate autonomously based on waypoint directives issued from the home base. Communication between the autopilot and the home base is established using a high-power radio modem, which allows the UAV to be controlled at multi-kilometer ranges. The Unicorn is also equipped with a CCD camera mounted on a pan-tilt gimbal, which allows the home base to receive a live aerial feed through an analog radio frequency channel.

Chapter 3

Realtime Online Spatiotemporal Topic Modeling

3.1 Introduction

Making decisions based on the environmental context of a robot's locations requires that we first model the context of the robot observations, which in turn might correspond to various semantic or conceptually higher level entities that compose the world. If we are given an observation model of these entities that compose the world then it is easy to describe a given scene in terms of these entities using this model; likewise, if we are given a labeling of the world in terms of these entities, then it is easy to compute the observation model for each individual entity. The challenge comes from doing these two tasks together, unsupervised, and with no prior information. This chapter presents ROST, a realtime online spatiotemporal topic modeling framework that attempt to solve this problem of assigning high level labels to low level streaming observations.

Topic modeling techniques, as discussed in Section 2.2, were originally developed for unsupervised semantic modeling of text documents. These algorithms automatically discover the main themes (topics) that underly these documents, which can then be used to compare these documents based on their semantic content.

Topic modeling of observation data captured by a mobile robot faces additional challenges

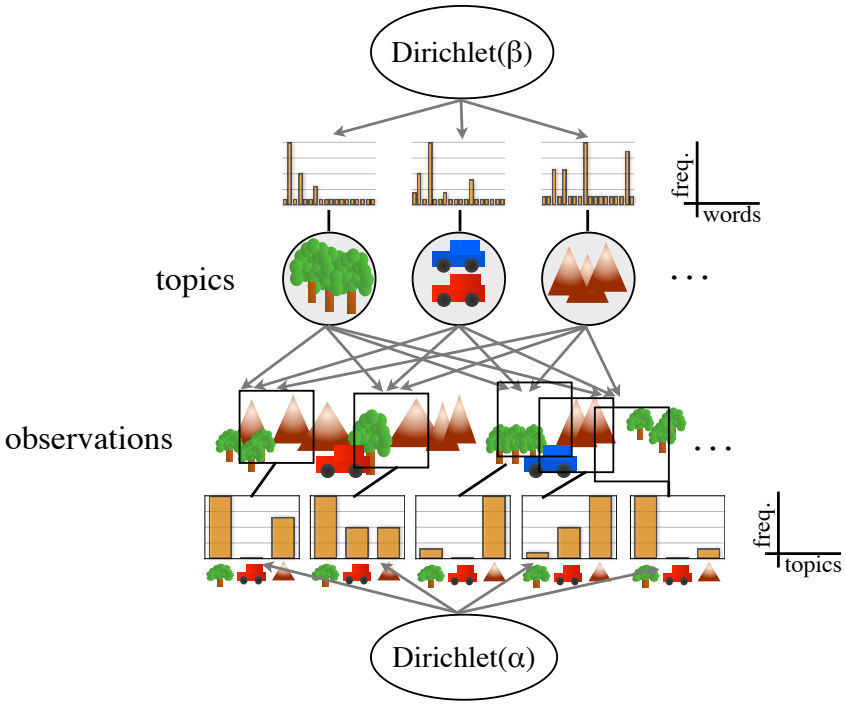


Figure 3.1: Spatiotemporal Topics: As a robot observes the world, we would like its observations to be expressed as a mixture of topics with perceptual meaning. We model the topic distribution of all possible overlapping spatiotemporal regions or neighborhoods in the environment, and place a Dirichlet prior on their topic distribution. The topic distribution of the current observation can then be inferred given the topic labels for the neighborhoods in the view. Modeling neighborhoods allows us to use the context in which the current observation is being made to learn its topic labels. To guarantee realtime performance, we only refine a constant number of neighborhoods in each time step, giving higher priority to recently observed neighborhoods.

compared to topic modeling of a collection of text documents, or images that are mutually independent.

- Robot observations are generally dependent on its location in space and time, and hence the corresponding semantic descriptor must take into account the location of the observed visual words during the refinement, and use it to compute topic priors that are sensitive to changes in time and the location of the robot.
- The topic model must be updated online and in realtime, since the observations are generally made continuously at regular intervals. When computing topic labels for a new observation, we must also update topic labels for previous observations in the light on new incoming data.

ROST extends previous work on text and image topic modeling to make it suitable for processing streaming sensor data such as video and audio observed by a robot, and presents approximations for posterior inferencing that work in realtime. Topics in this case model the latent causes that produce these observations.

3.2 Generative Process for Observations

An observation word is a discrete observation made by a robot. Given the observation words and their location, we would like to compute the posterior distribution of topics at this location. Let w be the observed word at location x. We assume the following generative process for the observation words:

1. word distribution for each topic k:

$$\phi_k \sim \text{Dirichlet}(\beta),$$

2. topic distribution for words at location x :

$$\theta_x \sim \text{Dirichlet}(\alpha),$$

3. topic label for w:

$$z \sim \text{Discrete}(\theta_x),$$

4. word label:

$$w \sim \text{Discrete}(\phi_z),$$

where $y \sim Y$ implies that random variable y is sampled from distribution Y, and z is the topic label for the word observation w. Each topic is modeled by distribution ϕ_k over V possible word in the observation vocabulary.

$$\phi_k(v) = \mathbf{P}(w = v|z = k) =\propto n_k^v + \beta, \tag{3.1}$$

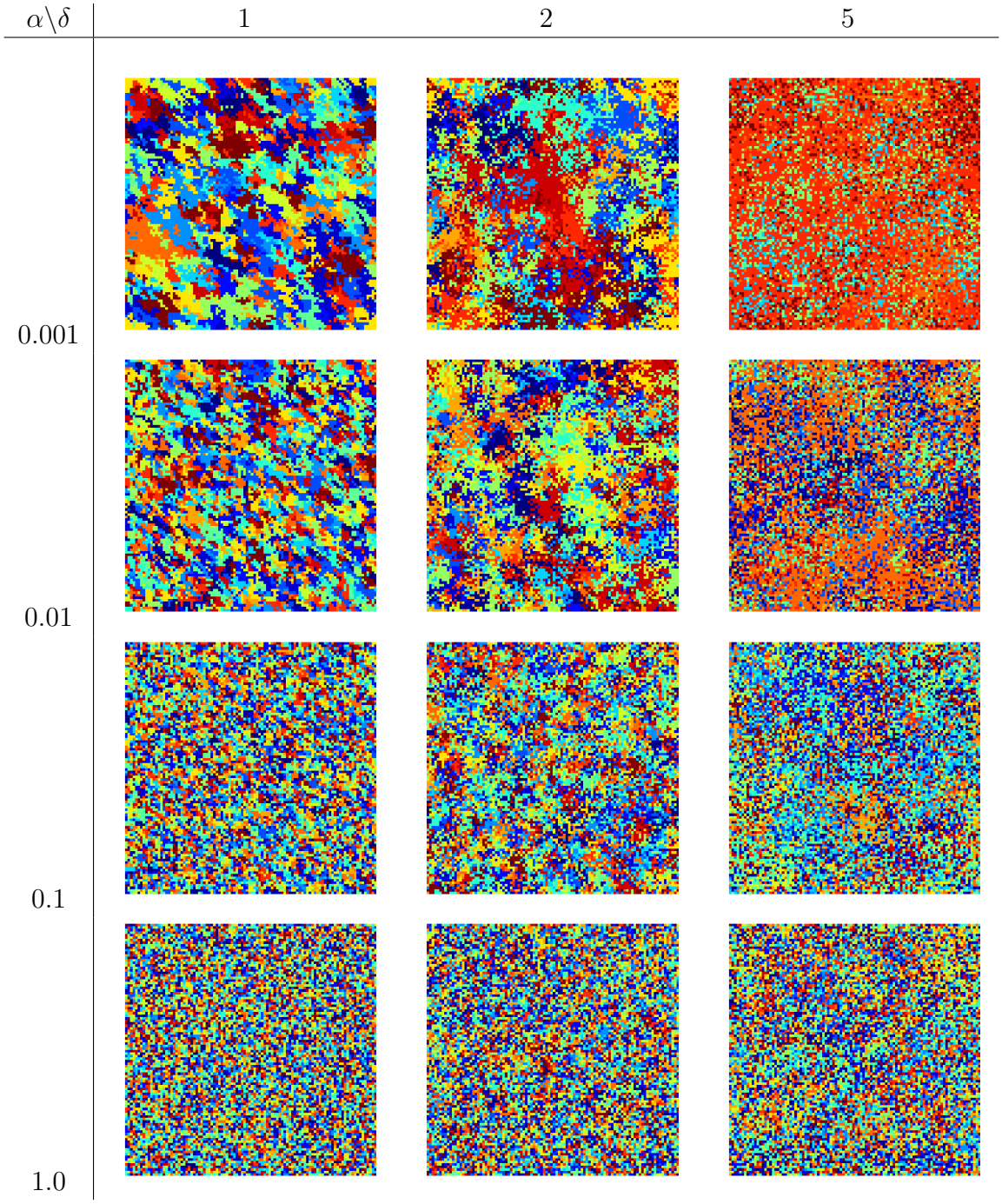


Figure 3.2: Samples from ROST's topic generative process in 2D. The table shows random maps sampled from the generative process used to characterize spatial terrain information in 2D worlds. Columns show variation in neighborhood size δ, and rows show variation in Dirichlet concentration parameter α. We see that changes in α control cluster sizes, whereas changes in δ control mixing of adjacent clusters. These samples were generated using a Gibbs sampler with burn-in period of 1000 iterations

where n_k^v is the number of times we have observed word v taking topic label k, and β is the Dirichlet prior hyperparameter. Topic model $\Phi = \{\phi_k\}$ is a $K \times V$ matrix that encodes the global topic description information shared by all locations.

The main difference between this generative process and the generative process of words in a text document as proposed by LDA (see Section 2.2.3) is in step 2. The context of words in LDA is modeled by the topic distribution of the document, which is independent of other documents in the corpora. We relax this assumption and instead propose the context of an observation word to be defined by the topic distribution of its spatiotemporal neighborhood. This is achieved via the use of a kernel. The topic distribution at location x is thus defined as:

$$\theta_x(k) = \mathbf{P}(z = k|x) \propto \left(\sum_y K(x - y) n_y^k \right) + \alpha, \tag{3.2}$$

where $K(\cdot)$ is the kernel, α is the Dirichlet prior hyperameter and, n_y^k is the number of times we observed topic k at location y.

Figure 3.2 shows random samples from the topic generative process, with $x \in \mathbb{R}^2$, number of topics $K = 16$, and a simple box kernel of radius $\delta = \{1, 2, 5\}$. As we saw in Section 2.1, samples from Dirichlet with $\alpha < 1$ are sparse distributions, which are responsible for the clustering effect seen in the figure. A kernel with larger width results in smoother boundaries between these clusters. This clustering effect of Dirichlet priors is our main motivation for their use in modeling the latent entities responsible for producing the observation. Our assumption is that in natural scenes, the observations emitted by the same topic will cluster together in space and time. These samples were generated using a Gibbs sampler with burn-in period of 1000 iterations.

3.3 Approximating Neighborhoods using Cells

The generative process defined above models the clustering behavior of observations from a natural scene well, but is difficult to implement because it requires keeping track of the topic distribution at every location in the world. This is computationally infeasible for any large dataset. For the special case when the kernel is a uniform distribution over a finite region, we

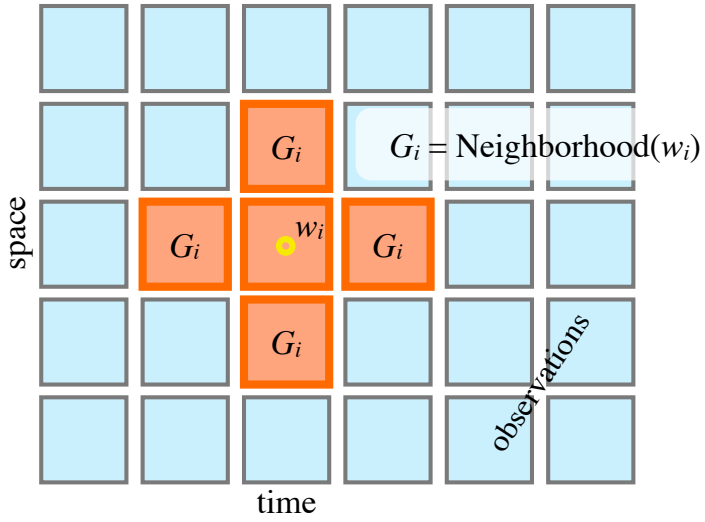


Figure 3.3: Each cell shown corresponds to a spatiotemporal bucket containing all the observation from that region. We refine the topic label for a word w_i in an observation by taking into account the spatiotemporal context G_i of the observation.

can assume a cell decomposition of the world, and approximate the topic distribution around a location by summing over topic distribution of cells in and around the location.

Let the world be decomposed into C cells, in which each cell $c \in C$ is connected to its neighboring cells $G(c) \subseteq C$. Let $c(x)$ be the cell that contains points x. In this thesis we only experiment with a grid decomposition of the world in which each cell is connected to its four nearest neighbors. However, the general ideas presented here are applicable to any other topological decomposition of spacetime.

The topic distribution around x can then be approximated using cells as:

$$\theta_x(k) \propto \left(\sum_{c' \in G(c(x))} n^k_{c'} \right) + \alpha \tag{3.3}$$

Due to this approximation, the following properties emerge:

1. $\theta_x = \theta_y$ if $c(x) = c(y)$, i.e., all the points in a cell share the same neighborhood topic distribution.
2. The topic distribution of the neighborhood is computed by summing over the topic

Initialize $\forall i, z_i \sim$ Uniform($\{1, \ldots, K\}$)
while *true* **do**
  foreach *cell* $c \in C$ **do**
    foreach *word* $w_i \in c$ **do**
      $z_i \sim \mathbf{P}(z_i = k | w_i = v, x_i)$
      Update Θ, Φ given the new z_i by updating n_k^v and n_G^k
    end
  end
end

Algorithm 1: Batch Gibbs sampling

distribution of the neighboring cells rather than individual points.

We take advantage of these properties while doing inference in realtime.

3.4 Realtime Inference using Gibbs Sampling

Given a word observation w_i, its location x_i, and its neighborhood $G_i = G(c(x_i))$, we use a Gibbs sampler similar to one described in Section 2.2.3 to assign a new topic label to the word, by sampling from the posterior topic distribution:

$$\mathbf{P}(z_i = k | w_i = v, x_i) \propto \frac{n_{k,-i}^v + \beta}{\sum_{v=1}^{V}(n_{k,-i}^v + \beta)} \cdot \frac{n_{G_i,-i}^k + \alpha}{\sum_{k=1}^{K}(n_{G_i,-i}^k + \alpha)}, \tag{3.4}$$

where $n_{k,-i}^w$ counts the number of words of type w in topic k, excluding the current word w_i, $n_{G_i,-i}^k$ is the number of words with topic label k in neighborhood G_i, excluding the current word w_i, and α, β are the Dirichlet hyper-parameters. Note that for a neighborhood size of 0, the above Gibbs sampler is equivalent to the LDA Gibbs sampler proposed by Griffiths et al.[49], where each cell corresponds to a document. Algorithm 1 shows a simple iterative technique to compute the topic labels for the observed words in batch mode.

In the context of robotics we are interested in the online refinement of observation data. After each new observation, we only have a constant amount of time to do topic label refinement. Hence, any online refinement algorithm that has computational complexity which increases with new data, is not useful. Moreover, if we are to use the topic labels of an

```
while true do
    Add new observed words to their corresponding cells.
    Initialize ∀i ∈ M_T, z_i ∼ Uniform({1, ..., K})
    while no new observation do
        t ∼ P(t|T)
        foreach cell c ∈ M_t do
            foreach word w_i ∈ c do
                z_i ∼ P(z_i = k|w_i = v, x_i)
                Update Θ, Φ given the new z_i by updating n_k^v and n_G^k
            end
        end
    end
    T ← T + 1
end
```

Algorithm 2: ROST Gibbs sampler

incoming observation for making realtime decisions, then it is essential that the topic labels for the last observation converge before the next observation arrives.

Since the total amount of data collected grows linearly with time, we must use a refinement strategy that efficiently handles global (previously observed) data and local (recently observed) data.

Our general strategy is described by Algorithm 2. At each time step we add the new observations to the model, and then randomly pick observation times $t \sim \mathbf{P}(t|T)$, where T is the current time, for which we resample the topic labels and update the topic model.

We discuss the choice of $\mathbf{P}(t|T)$ in the following sections.

3.4.1 Now Gibbs Sampling

The simplest way of processing streaming observation data to ensure that the topic labels from the last observation have converged is to only refine topics from the last observation till

the next observation has arrived.

$$\mathbf{P}(t|T) = \begin{cases} 1, & \text{if } t = T \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

We call this the Now Gibbs sampler. This is analogous to o-LDA approach by Banerjee and Basu [5].

If R is our computation budget, defined as the expected number of observation time-steps our system can refine between the arrival times of two consecutive observations, and $r(t)$ be the number of times observations in M_t have been refined after time T, then this approach gives each observation R amount of resources.

$$\mathbb{E}\{r(t)\} \quad = \quad R \tag{3.6}$$

Although this sounds fair, the problem is that no information from the future is used to improve the understanding of the past data.

3.4.2 Uniform Gibbs Sampling

A conceptually opposite strategy is to uniform randomly pick an observation from all the observations thus far, and refine the topic labels for all the words in this observation.

$$\mathbf{P}(t|T) = 1/T \tag{3.7}$$

This is analogous to the incremental Gibbs sampler for LDA proposed by Canini et al.[16].

Let M_t be the set of cell containing observations at time t, R be the number of observations our system can refine between two observations, and $r(t)$ be the number of times observations in M_t have been refined after time T. The expected value of $r(t)$ is then:

$$\mathbb{E}\{r(t)\} \quad = \quad R\left(\frac{1}{t} + \frac{1}{t+1} + \cdots + \frac{1}{T}\right) \tag{3.8}$$

$$\approx \quad R(\log T - \log t). \tag{3.9}$$

We see that older observations are sampled disproportionally higher than newer observations, and topic labels of new observations might take a long time to converge. In fact, if τR is the expected number of iterations it takes for topic labels of an observation to converge, where $\tau < 1$ is a constant, then all observations after time $t' = 1/\tau$ would never be able to converge in the time before the next observation arrives. This is a big problem for a real-time system such as the one being proposed in this chapter, where we need the topic labels of the last observations to actuate the robot.

3.4.3 Age Proportional Gibbs Sampling

A seemingly good in-between approach might be to bias the random sampling of observations to be refined in favor of picking recent observations, with probability proportional to its timestamp.

$$\mathbf{P}(t|T) = \frac{t}{\sum_{i=1}^{T} i} \tag{3.10}$$

Then, the expected number of times this observation is refined is given by:

$$\begin{aligned} \mathbb{E}\{r(t)\} &= R\left(\frac{t}{\sum_{i=1}^{t} i} + \frac{t}{\sum_{i=1}^{t+1} i} + \cdots + \frac{t}{\sum_{i=1}^{T} i}\right) && (3.11) \\ &\approx 2R\frac{(T-t)}{T}. && (3.12) \end{aligned}$$

When a new observation is made, the expected number of refinements it will gets before the next observation arrives is $Rt/\sum t \approx 2R/t$, which implies that if t' is the time after which it will not have sufficient number of refinements, then:

$$\begin{aligned} \frac{2R}{t'} &= \tau R && (3.13) \\ \implies t' &= \frac{2}{\tau} && (3.14) \end{aligned}$$

Hence, we see that this strategy, although better than uniform random sampling (for which we computed $t' = 1/\tau$), is still not useful for long term operating of the robot.

3.4.4 Exponential Gibbs Sampling

Using a geometric distribution we can define the probability of refinement of timestep t, at current time T

$$\mathbf{P}(t|T) = q(1-q)^{T-t}, \tag{3.15}$$

where $0 < q < 1$ is a parameter. Using this distribution for picking refinement samples ensures that on average qR number of refinements are spent on refining the most recent observations, and the remaining $(q-1)R$ refinement iterations are spent on refining other recent observations. In the limit $T \to \infty$, observations in each time-step are refined $\mathbb{E}\{r(t)\} = R$ number of times, similar to Now Gibbs Sampler. This approach, however, allows new information to influence some of the recent past observations, resulting in lower global perplexity of the learned model.

3.4.5 Mixed Gibbs Sampling

We expect both Now and Exponential Gibbs samplers to be good at ensuring the topic labels for the last observation converges quickly (to a locally optimal solution), before the next observation arrives, whereas Uniform and Age-proportional Gibbs samplers are better at finding globally optimal results.

One way to balance both these performance goals is to combine these global and a local strategies. We consider four such approaches in this thesis:

Uniform+Now:

$$\mathbf{P}(t|T) = \begin{cases} \eta, & \text{if } t = T \\ (1-\eta)/(T-1), & \text{otherwise} \end{cases} \tag{3.16}$$

AgeProportional+Now:

$$\mathbf{P}(t|T) = \begin{cases} \eta, & \text{if } t = T \\ (1-\eta)\frac{t}{\sum_{i=1}^{T-1} i}, & \text{otherwise} \end{cases} \tag{3.17}$$

Uniform+Exp:

$$\mathbf{P}(t|T) = \eta q(1-q)^{T-t} + (1-\eta)/T \tag{3.18}$$

AgeProportional+Exp:

$$\mathbf{P}(t|T) = \eta q(1-q)^{T-t} + (1-\eta)\frac{t}{\sum_{i=1}^{T} i} \tag{3.19}$$

Here $0 \leq \eta \leq 1$ is the mixing proportion between the local and the global strategies.

3.5 Computational Complexity of ROST

Assuming that cells in ROST are stored in a hash table, indexed by the cell's spatiotemporal coordinates, computing the distribution of topics in a neighborhood of the a given word requires $O(K)$ time, where K is the number of topics. This can be done by maintaining the frequency count of topics for each cell, apart from the topic labels for each word in the cell. Every time the topic label of a word in a cell changes, we also update the topic frequency count. Now, to compute the distribution of topics in the neighborhood of an observed word, we simply query all the neighboring cells, and accumulate the topic frequency counts. Since the maximum number of cell neighbors is constant, and it requires $O(1)$ time to query a cell, and $O(K)$ time to accumulate the topic counts, the total time needed to compute $\phi_x(k) = \mathbf{P}(z|x)$ is $O(K)$. Moreover, since $\phi_x(k)$ is the same for all the words in a cell, we can memoize this distribution, giving us an even lower amortized cost of $O(K/\bar{W_C})$, where $\bar{W_C}$ is the average number of words in a cell.

In Comparison, Spatial LDA (SLDA) [110] models the document variable representing the spatiotemporal context of an observed word as another random variable. A word is likely to be assigned to a document if its location is close to other words in the document, and has the same topic label as other words in the document. Computing the topic distribution of the spatial context (document) of the word requires $O(KW_M)$ time, where W_M is the number of words in an image.

3.6 Visualizing Spatiotemporal Topics

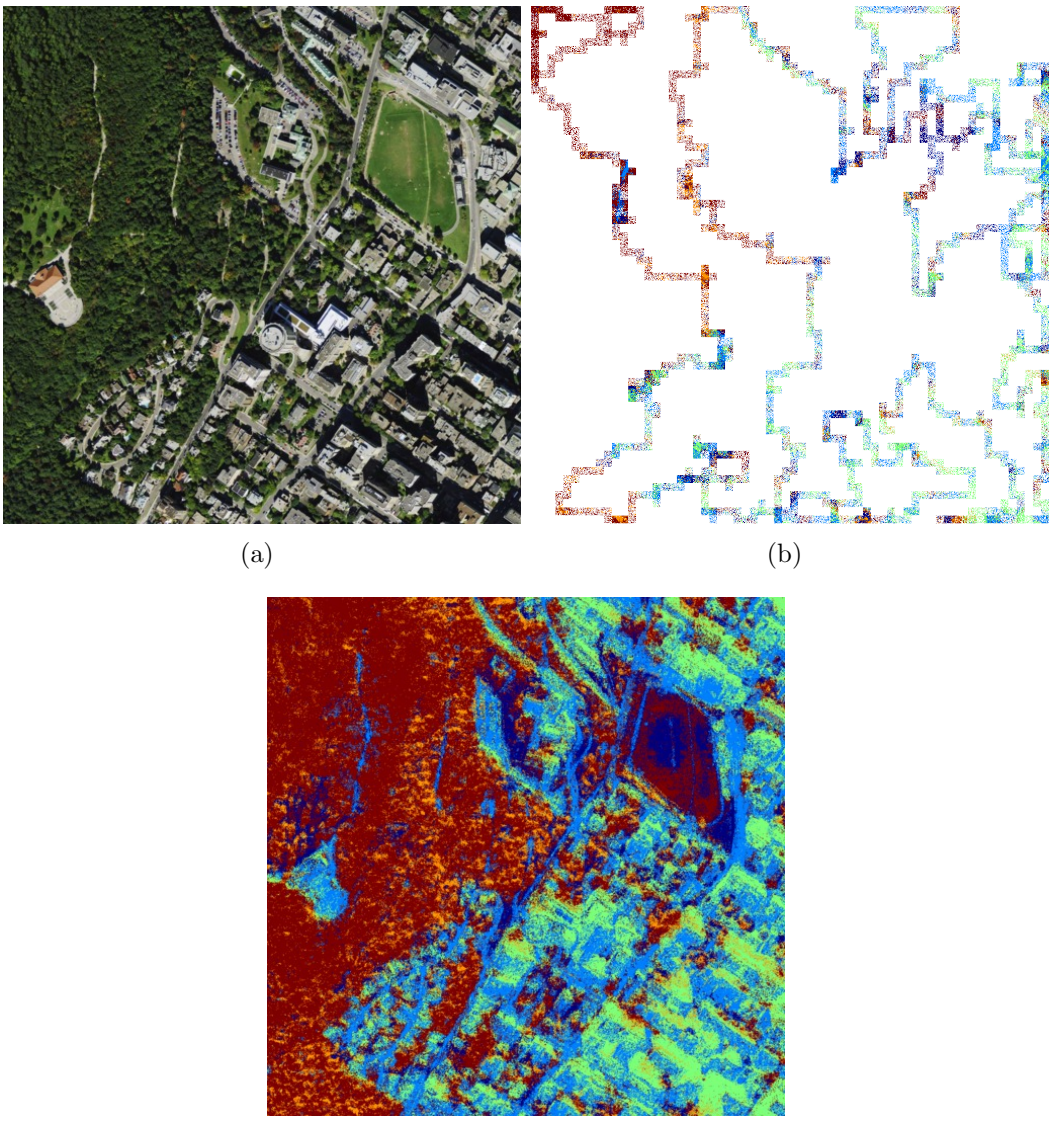
Topic Modeling of Online Terrain Observation Data

Consider a scenario in which a robot traverses a path on a 2D plane, and we would like to topic model the observations made by it to discover and recognize different terrains in the scene. We simulate this via a path on a 2D satellite image, as shown in Figure 3.4. Each cell in this example is a square with width of 16 pixels, and we extract Texton and ORB visual words as described in Section 2.3. Color of each pixel corresponds to the topic label. The path in (b) shows the final topic labels at the end of the path, computed using the proposed Uniform+Now Gibbs sampling. We see that topic labels computed online correspond well with the underlying terrain. In (c) we show the topic labels computed in batch mode for reference.

Evolution of Topics over Time

Streaming video is essentially streaming spatiotemporal data, where visual features have temporal coordinates corresponding to the frame number, and spatial coordinates corresponding to the pixel coordinates of the feature. Using ROST we can take advantage of the fact that the topic label for a visual word in a video frame is likely to be similar to neighboring topic labels within the same image, and also in the adjacent frames.

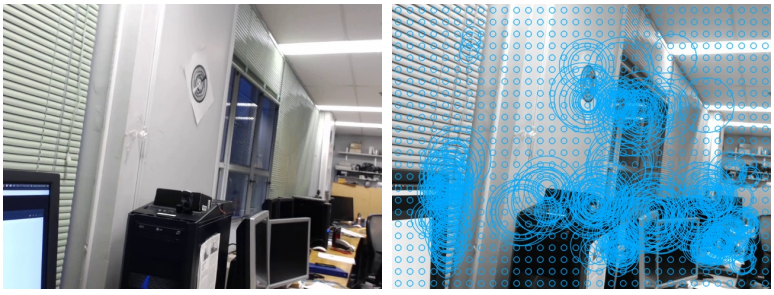
Topics learned by ROST evolve over time as more data is observed. Several different video demonstrations of ROST are available at: `http://cim.mcgill.ca/mrl/girdhar/`. Figure 3.5 shows an example of this topic evolution in an indoor environment. The images (a)-(d) are in temporal sequential order, and show the growth of the topic model as time elapses. The circles correspond to extracted visual words, and the color corresponds to the different topic labels. Initially, we see that the entire scene is labeled with the same topic label, as is shown in Figure 3.5(a). After a few time steps, we see that ROST is successfully able to segment the scene using two topics corresponding to the upper and lower half of the room (Figure 3.5(b)). Eventually, we see that many more topic labels are being used to represent different parts of the room (Figure 3.5(c,d)).



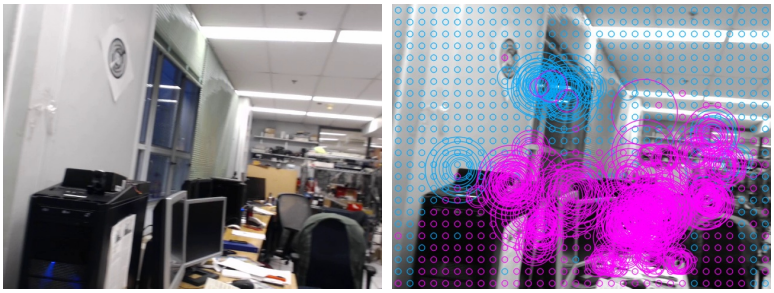
(a) (b)

(c)

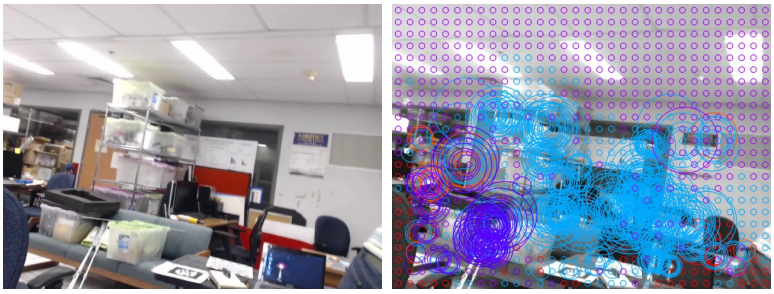
Figure 3.4: Spatial topic modeling of terrain data. (a) Given map (b) Simulated path followed by a robot. Each block corresponds to a step in the robot's path and also a cell in our topic model. Each pixel corresponds to an observation word, and the color of the pixel corresponds to the topic label. (c) Topic labels for the entire map computed in batch mode.



(a)



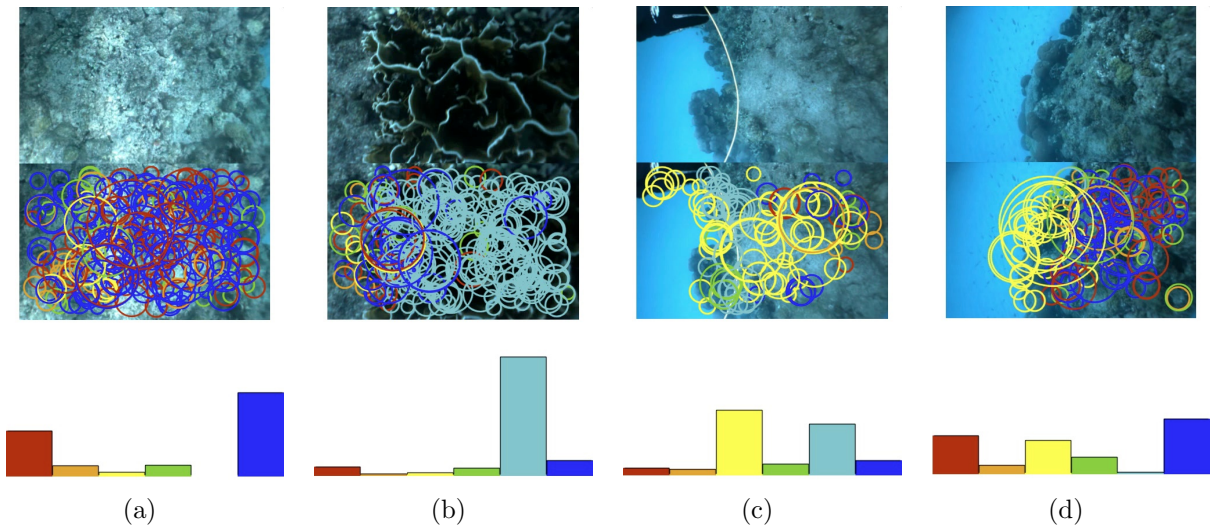
(b)



(c)

(d)

Figure 3.5: Topics learned by ROST evolve over time to capture the growing complexity of the scene. The figures are temporally sequenced, and show more topics being used to label different parts of the scene as more time elapses.



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

Figure 3.6: Example of topics learned from ocean floor images taken by the Aqua robot, for a single trajectory. Each visual word is marked by a circle, the size of which corresponds to the size of the visual feature. Histograms depicting the content of each color-coded topic are shown below.

Topic Modeling of Underwater Data

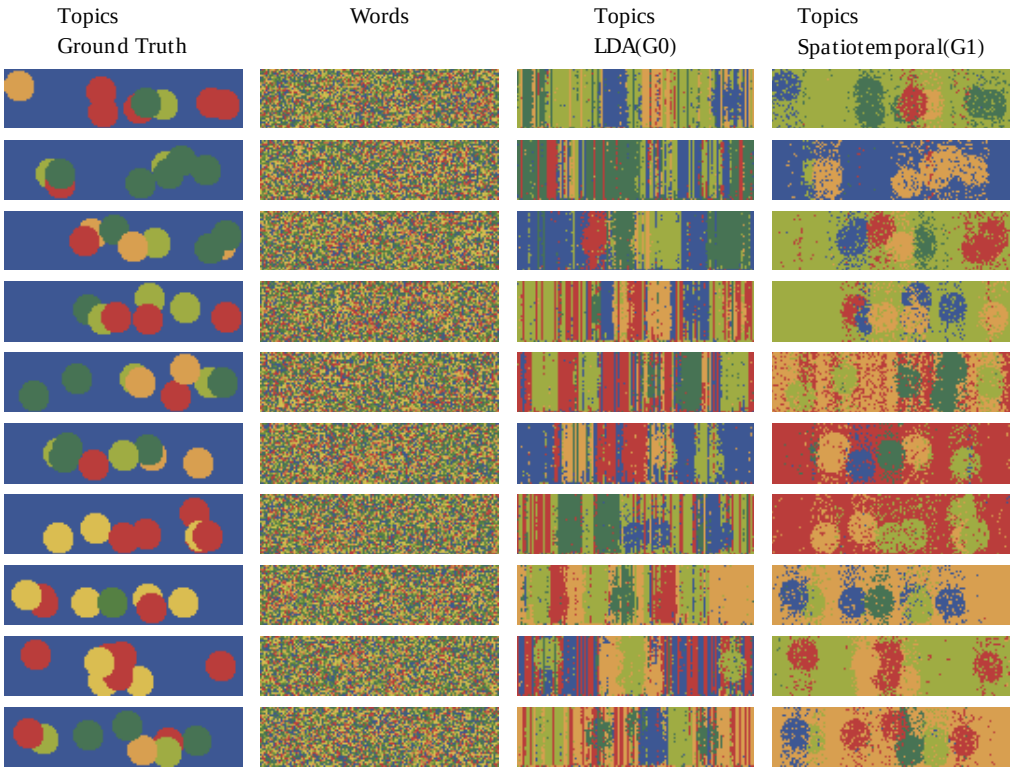
Some examples of the topics learned from the underwater dataset are shown in Fig 3.6. We see that the topics are representative of underlying physical phenomenon, and do well in describing scenes where a mixture of these exist. Red and blue topics are being used to represent rocks in the dataset, yellow for the sand-rock boundary, and cyan for the fire coral and the white rope. We set both summary size and topic size to 6 for our experiments. The hyper-parameters for the LDA were determined empirically.

3.7 Experiments

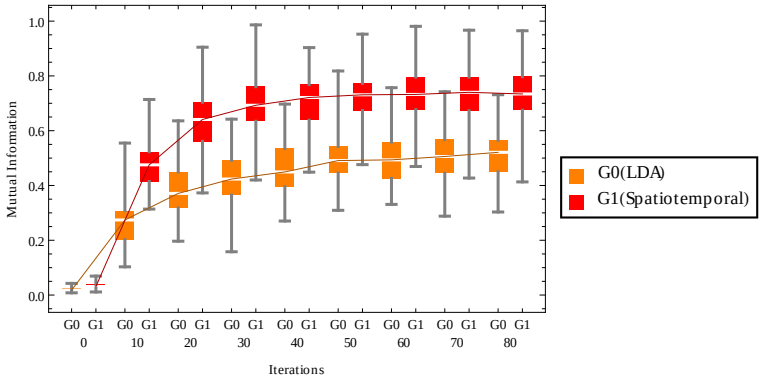
3.7.1 Evaluating ROST on Artificial Data

Generative Process

To test the effectiveness of the proposed spatiotemporal topic modeling in recovering the latent structure in the data we used artificially generated datasets, for which the ground truth is



(a) Artificially generated datasets along with the topics computed by the topic models.



(b) Mutual Information between the computed topic labels and the ground truth.

Figure 3.7: We generated 100 artificial datasets out of which we show the first 10 in (a). Each column corresponds to a timestep, and each pixel corresponds to a word observation. The color of the pixel corresponds to the topic/word label. In (b) we show the Mutual Information between the results and the ground truth. We see that topic model which takes into account its two adjacent neighbors (G1), consistently outperforms the topic model that does not use the neighborhood information (G0), which is equivalent of the standard LDA generative model.

known. Each dataset was generated in the following manner. As shown in Figure 3.7(a) column 1, we generated ground truth topic labels such that each column corresponds to an observation, and each pixel corresponds to a word. Each dataset has 8 objects(disks), spanning multiple observations, and are placed randomly. Each object has a topic label $z \in [0, K)$ that is chosen randomly from one of the $K = 5$ topics. Topic labels are shown by the pixel color. Given the topic label of a word $z = k$, we generated word labels by randomly picking $v \in [0, V)$ from distribution $\mathbf{P}(v|z)$. We used vocabulary size $V = 100$. The vocabulary was equally divided between the K topics such that V/K words were preferred exclusively by each topic. To add noise, we used a non-zero probability of emitting a word not related to the given topic.

$$\mathbf{P}(v|z) \propto \begin{cases} 1+\eta & \text{if v is related to } z \\ \eta & \text{otherwise} \end{cases} \tag{3.20}$$

We used $\eta = 0.1$ for all our experiments. Hence, the probability that a we observe a word related to the underlying topic is: $\frac{(1+\eta)V/K}{(1+\eta)V/K+\eta(V-V/K)} = 0.73$

Examples of the resulting words are shown in Figure 3.7(a) column 2. We can see that original pattern is invisible to human eyes. Finally, we used these generated words as input to the proposed spatiotemporally smoothed topic model, and refined the topics in batch mode for 80 iterations. A neighborhood size of 1 implies that while refining each observation, represented as a column in the dataset image, we take into account the two adjacent columns. A neighborhood size of 0 corresponds to the traditional LDA, where each column is refined independently. Figure 3.7(a) columns 3,4 show the resulting topic labels. We used hyper-parameter values $\alpha = 0.1$ and $\beta = 1.0$ for all our experiments. We experimented with 100 randomly generated datasets, out of which the first 10 are shown in Figure 3.7(a). We see that in most cases, use of the proposed technique results in much more accurate topic labels.

Evaluation of Neighborhood Modeling

To quantitatively evaluate these results, we computed the Mutual Information between the ground truth topic labels and the proposed methods. Mutual Information $I(X, Y)$ essentially measures the reduction in Entropy of a random variable X, after observing random variable

Y.

$$I(X,Y) = H(X) - H(X|Y) \tag{3.21}$$

$$= \sum_{x,y} \mathbf{P}(x,y) \log \frac{\mathbf{P}(x,y)}{\mathbf{P}(x)\mathbf{P}(y)} \tag{3.22}$$

To compute Mutual Information between the ground truth labels, and the topic model generated labels, we set x to topic label from the ground truth, y to the topic labels produced by the topic models. Figure 3.7(b) shows the box-whisker plot of the mutual information between the ground truth, and the two topic models, for 100 datasets. The box corresponds to 75% and 25% quantiles, and the line in the middle is the median mutual information across all datasets. The whiskers show the minimum and maximum range of the Mutual Information score. We see that the proposed spatiotemporal topic model clearly outperforms traditional LDA, as it has higher Mutual Information score.

Evaluating Gibbs Sampling Strategies - Perplexity

Computing perplexity is a useful way of evaluating generative models. Per word perplexity of a document could be intuitively interpreted as the uncertainty in recreating the word labels from the topic label distribution being used by the model to describe a document.

Per word perplexity of the set of observed words M_t, at time t is defined as:

$$\text{Perplexity}(M_t) = exp\left(-\frac{\sum_i^W \log p(w_i|x_i, M_t)}{W_t}\right), \tag{3.23}$$

where W_t is the number of words in observation M_t, and w_i is the ith word in the observation with spatiotemporal location x_i. The log likelihood of a word in an observation can be computed using an expression similar to Eq. 3.4, with the difference that instead of considering $\mathbf{P}(z|G)$, the probability of a topic given its neighborhood, we use $\mathbf{P}(z|M)$, the probability of a topic given the observation to which it belongs.

We generated 100 random datasets, and then computed their topic labels in realtime using various Gibbs sampling strategies. The dataset has 256 time steps, and at each time step we observed 32 words from a vocabulary of size 100. There are 5 topics in the ground

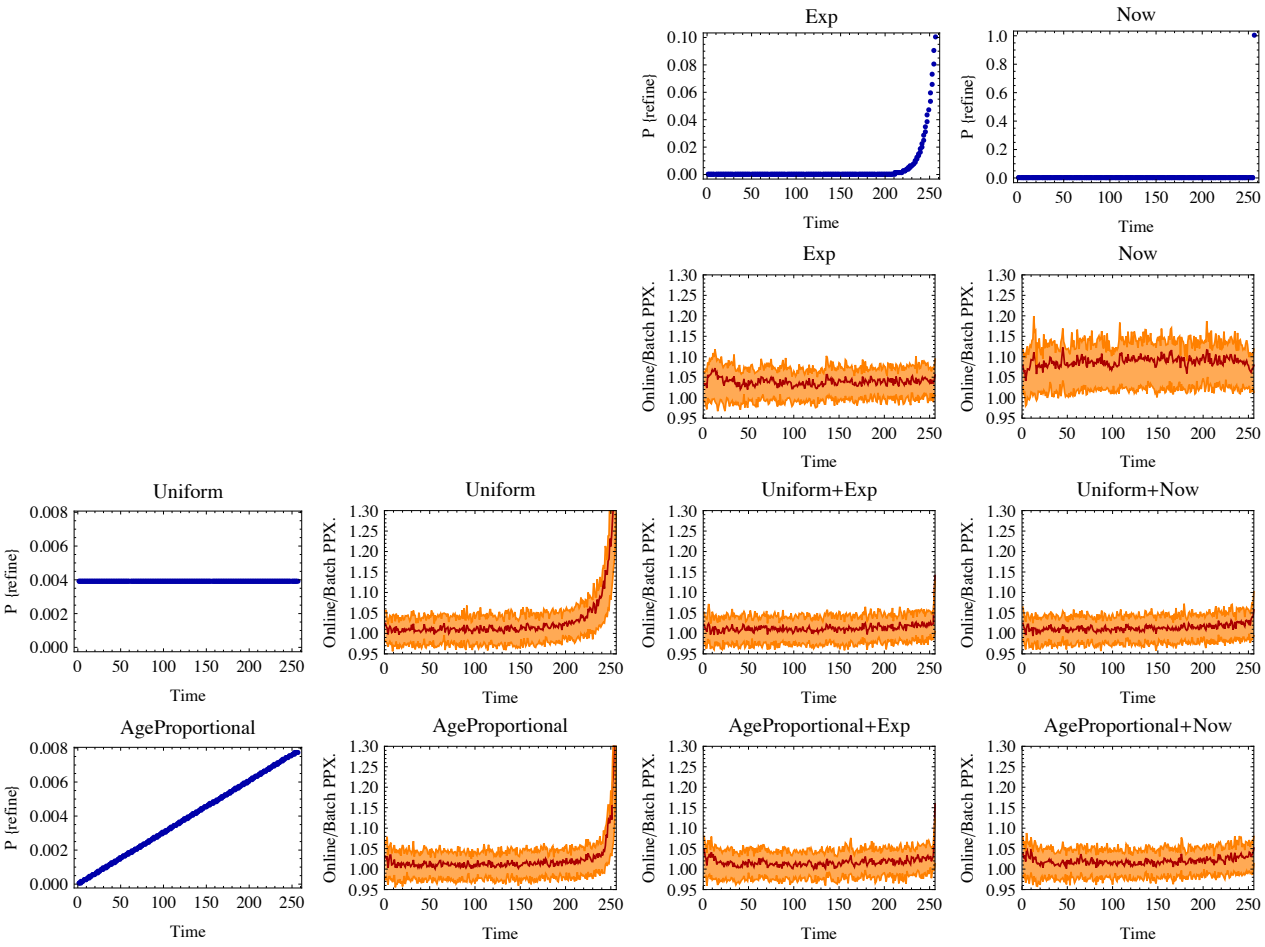


Figure 3.8: Evaluation of Gibbs sampling strategies using perplexity competitive ratio. Plots show the competitive ratio of the proposed online Gibbs samplers, measured in terms of the mean perplexity scores of the observations, immediately after the last observation. A score of 1.0 implies that the topic labels computed online have the same perplexity score as topic labels computed in batch, till convergence. The results were computed using 100 randomly generated datasets, each with 256 time-steps, and 32 words per time-step. We see that the use of local sampling techniques (Exp, Now) result in higher perplexity across the timeline, whereas the use of global sampling techniques (Uniform, AgeProportional) result in high perplexity of recent observations. We observe consistently better results when we process the data using mixed techniques.

truth, however while computing the topics, we set the number of maximum topics to $K = 10$, mitigating the bias due to knowledge of the actual number of topics. Each time step was fixed to 1 millisecond. For the mixed Gibbs samplers, we fixed $\eta = 0.5$.

Figure 3.8 shows the competitive ratio of different Gibbs sampling techniques discussed in this chapter. The left column and the top row show the distribution representing the probability of refinement of a given time-step, at current time $T = 257$. The two global refinement distributions - Uniform and AgeProportional are on the left column, and the two local refinement distributions - Exp and Now are on the top row. The inner plots in red show the ratio of perplexity of each time-step computed online to perplexity of the time-step computed using batch technique. The online perplexity was computed after observing the last time-step, at $T = 257$. The orange lines correspond to the 1/4 and 3/4 percentile scores.

The Now Gibbs sampling strategy focuses on refining only the last observation at each iteration. We see that the resulting perplexity is consistently high for the entire timeline (row 2, col 4). The exponential Gibbs sampling (with parameter $q = 0.1$) improves on the Now Gibbs sampler, resulting in lower overall perplexity (row 2, col 3).

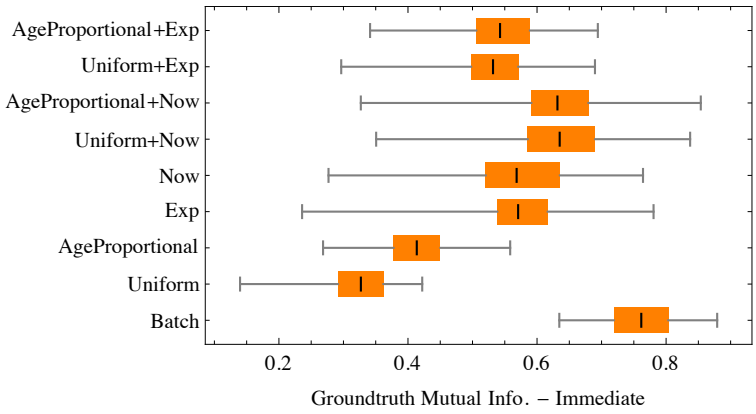
The global samplers - Uniform and AgeProportional have a better competitive ratio than the local samplers for most of the timeline except the most recent observations (row 3-4, col 2). We see that Age proportional performs better than uniform for recent observations.

The mixed Gibbs samplers (row 3-4, col 3-4) perform better than all other techniques. These samplers use 50% of their time using a global sampler, and the remaining time using a local sampler. The resulting numerical scores are summarized in Table 3.1. We see that Uniform+Exp has the best competitive ratio over the entire timeline, however AgeProportional+Now performs the best in terms of the perplexity of the last observation.

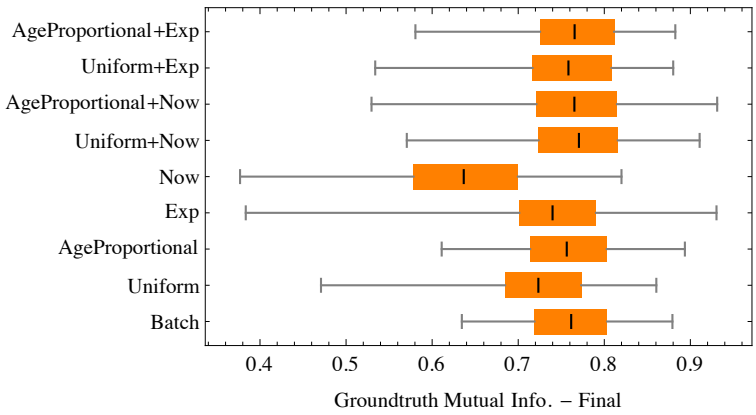
Evaluating Gibbs Sampling Strategies - Ground Truth Mutual Information

Figure 3.9 shows the mutual information of the online topic labels and the ground-truth labels. The box whiskers plot on the top shows the mutual information score of the topic labels, one time step after they were observed. The bottom plot shows the mutual information score of the topic labels after making the last observation at $T = 257$.

We see that the mixed Gibbs samplers perform well in general, and Uniform+Now per-



(a)



(b)

Figure 3.9: Evaluation of Gibbs sampling strategies using ground truth mutual information. (a) Mutual information score using computed topic labels, one time step after the observation time. (b) Mutual information score computed using topic labels after the last observation has been processed.

Gibbs Sampler	PPXRatio Mean	PPXRatio LastObs.	MeanMI Immediate	MeanMI Final
AgeProp.+Exp	1.01697	1.15927	0.542681	0.765524
Uniform+Exp	**1.01317**	1.14120	0.531800	0.758349
AgeProp.+Now	1.01975	**1.05130**	0.631617	0.765263
Uniform+Now	1.01379	1.05747	**0.635104**	**0.770538**
Now	1.08815	1.07921	0.568375	0.636721
Exp	1.03903	1.05523	0.570709	0.739895
AgeProp.	1.02443	1.62573	0.413637	0.756399
Uniform	1.03070	1.48012	0.327006	0.723406
Batch	1.0	1.0		0.761544

Table 3.1: Summary of Results. PPXRatio is the ratio of perplexity of the data given the topic labels computed online, to the topic labels computed in batch. PPXRatio-Mean is the mean score over all 100 datasets and all 256 time-steps. PPXRatio-LastObs. is the mean score of the observations collected at time-step 256, measured at time-step 257. MeanMI is the Mean mutual information of the computed labels and the ground truth labels, computed immediately after observation, and after the last observation at time-step 257.

forms the best in both the scenarios (immediate and final topic labels). The surprising result is that the proposed online Gibbs sampler outperforms even the batch computation of topic labels. We hypothesize that this is due to the fact that in batch inference we initialize the topic labels for the entire dataset randomly, all at once. As a result the posterior topic distribution for an observation word is completely random. In the online setting, the posterior topic distribution is biased by the converged topic distribution of the previous time steps, resulting in a posterior distribution which might be more relevant for natural data.

The resulting numerical scores are summarized in Table 3.1.

3.7.2 Evaluating ROST on Streaming Video Data

Setup

We evaluated the performance on ROST in analyzing videos using three different datasets with millions of visual words. We used a mixed vocabulary to describe each frame, with 5000 ORB words, 256 intensity words (pixel intensity), and 180 hue words (pixel hue), for a total vocabulary size of 5436. Although it is difficult to substantiate the optimality of the

Name	size	N(timesteps)	N(words)	$\frac{\text{N(words)}}{\text{N(timesteps)}}$	Vocab. size
2objects	720x480	1158	1741135	1503	5436
aerial	640x480	3600	8190231	2275	5436
underwater	1024x638	2569	4809869	1872	5436

Table 3.2: Video datasets for evaluating ROST

vocabulary, our experiments have suggested that once the vocabulary size is sufficiently large, there is limited sensitivity to its precise value [38].

Some key statistics for these datasets is shown in Table 3.2.

The *2objects* dataset show a simple scenario in which two different objects appear on a textured (wood) background randomly, first individually and finally together.

The *aerial* dataset was collected using Unicorn UAV (Sec. 2.7.3) over a coastal region. The UAV performs a zig-zag coverage pattern over buildings, forested areas and ocean.

The *underwater* dataset was collected using Aqua (Sec 2.7.1) as it swims over a coral reef. The dataset contains a variety of complex underwater terrain such as different coral species, rocks, sand, and divers.

The video files corresponding to these datasets are available at [1].

To focus on analyzing the effects of spatiotemporal neighborhoods, and various Gibbs samplers, we fixed all other parameters of the system. We used cells of size 64x64 pixels with temporal width of 1 time step, Dirichlet parameters $\alpha = 0.1, \beta = 0.5$, number of topics $K = 16$.

Spatiotemporal Neighborhoods

We hypothesize that using spatiotemporal neighborhood will result in better topic models, that will have lower perplexity. We ran ROST multiple times for each dataset, with using spatiotemporal neighborhoods ($G_R = 1$) and without ($G_R = 0$), and computed the mean perplexity for each time step. Neighborhood radius $G_R = 1$ implies that each cell is connected to its 2 closest cells along each dimension, which in this case corresponds to 4 spatial neighbors

[1] `http://cim.mcgill.ca/mrl/girdhar/`

Dataset →	2objects		aerial		underwater	
$G_R \downarrow$, T_R (in ms.)→	40	160	40	160	40	160
1 (proposed)	70.1645	55.9029	887.417	575.283	517.411	348.581
0 (LDA)	78.4882	64.7831	970.409	639.424	554.629	386.074

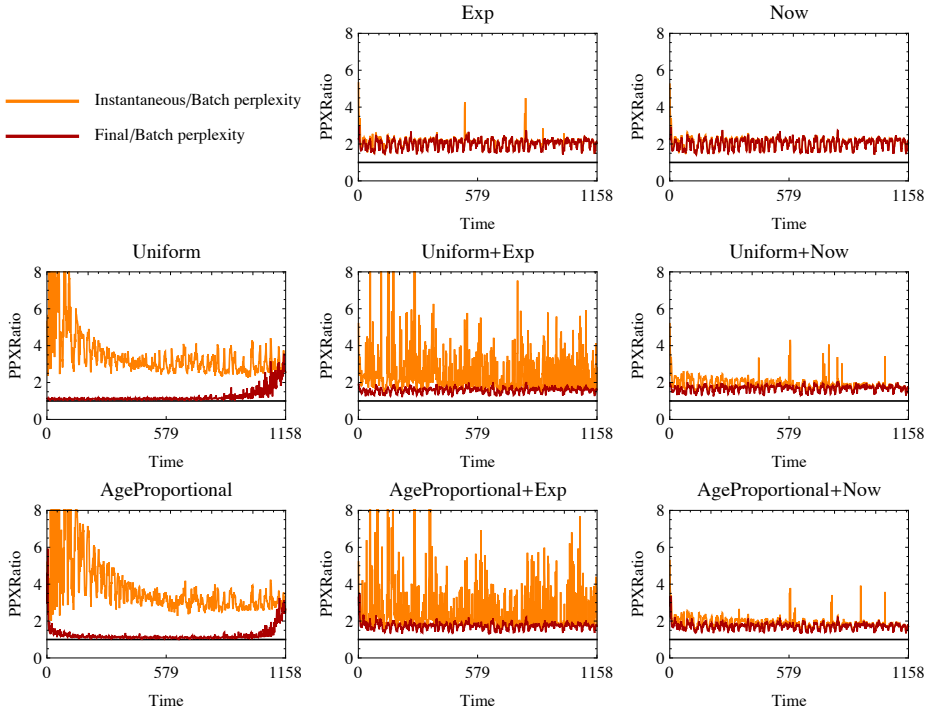
Table 3.3: Mean batch perplexity of a word in the dataset for different refinement times, and neighborhoods (lower is better). G_R is the neighborhood radius, where $G_R = 0$ is similar to using plain LDA, and $G_R = 1$ corresponds to refining each cell in the context of its 6 spatiotemporal neighbors. T_R is the average refinement time (in milliseconds) per time step. The proposed algorithm is superior with a perplexity of at most 0.93 times the LDA perplexity, and on average 0.90 times the LDA perplexity.

and 2 temporal neighbors, and we refine the topic labels in a cell in the context of other words in its neighboring cells. Neighborhood radius $G_R = 0$ implies that topic labels in each cell are refined in the context of just the words in its own cell, which is similar to plain LDA, where each document corresponds to a cell. The results, computed by averaging over 10 random restarts, are presented in Table 3.3.

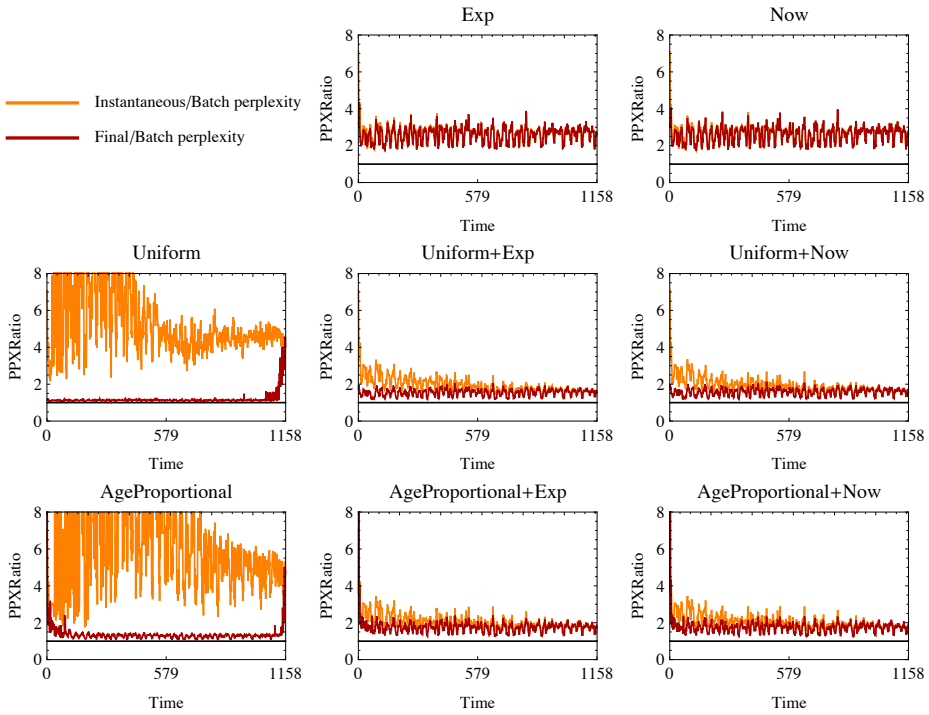
We see that using spatiotemporal neighborhood consistently results in lower perplexity (with statistical significance), irrespective of the datasets or refinement time. The proposed algorithm is superior with a perplexity of at most 0.93 times the LDA perplexity, and on average 0.90 times the LDA perplexity. This is likely due to the better priors on topic distribution for a word in a cell, computed using using topic distribution of the neighboring cells.

Realtime Gibbs Samplers

To evaluate the proposed realtime Gibbs samplers on real data, we performed the following experiment. For each video dataset, and for each Gibbs sampler, we computed the topic labels and perplexity online, with 10 random restarts. We then compared the mean perplexity of words, one time step after their arrival (instantaneous), and after all observations have been made (final), with the perplexity of topic labels computed in batch. For a fair comparison, we used the same refinement time per time step (T_R) for both batch and online cases. The resulting perplexity plots are shown in Figures 3.10, 3.11, and 3.12. The mean perplexity scores for the entire datasets are shown in Tables 3.5 (instantaneous perplexity), and 3.4 (final perplexity).

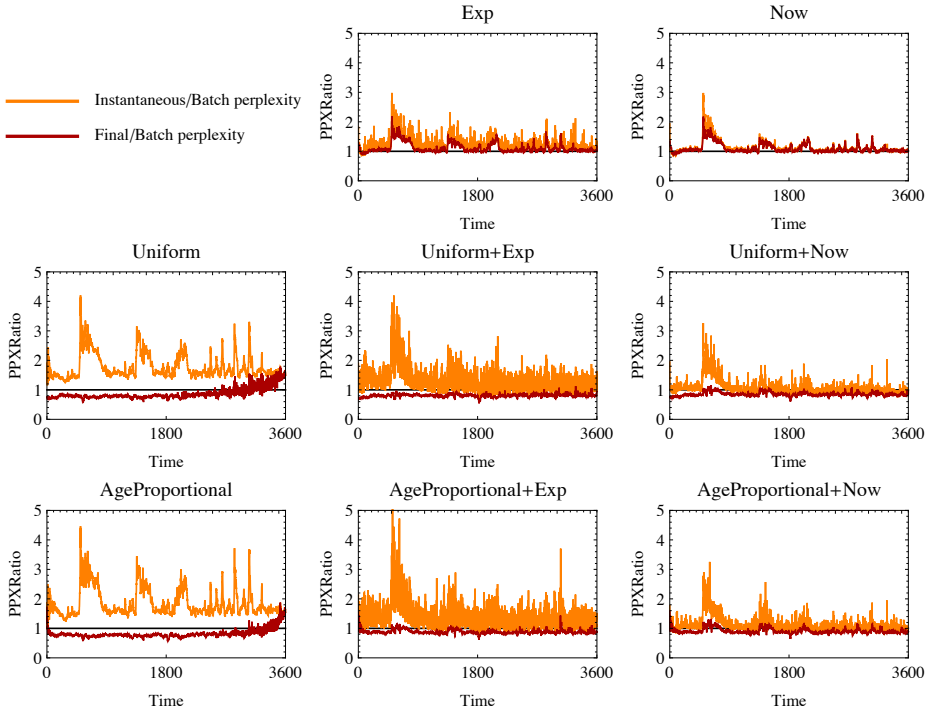


(a) Refinement time $T_R = 40$ ms.

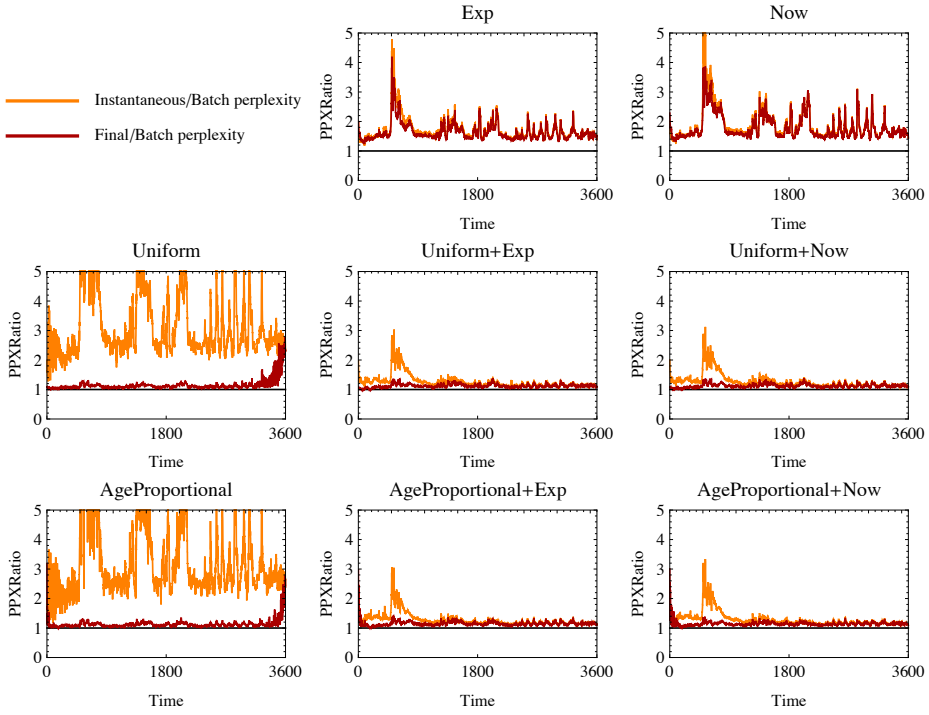


(b) Refinement time $T_R = 160$ ms.

Figure 3.10: 2Objects dataset – ratio of instantaneous and final perplexity to batch perplexity, for each time step

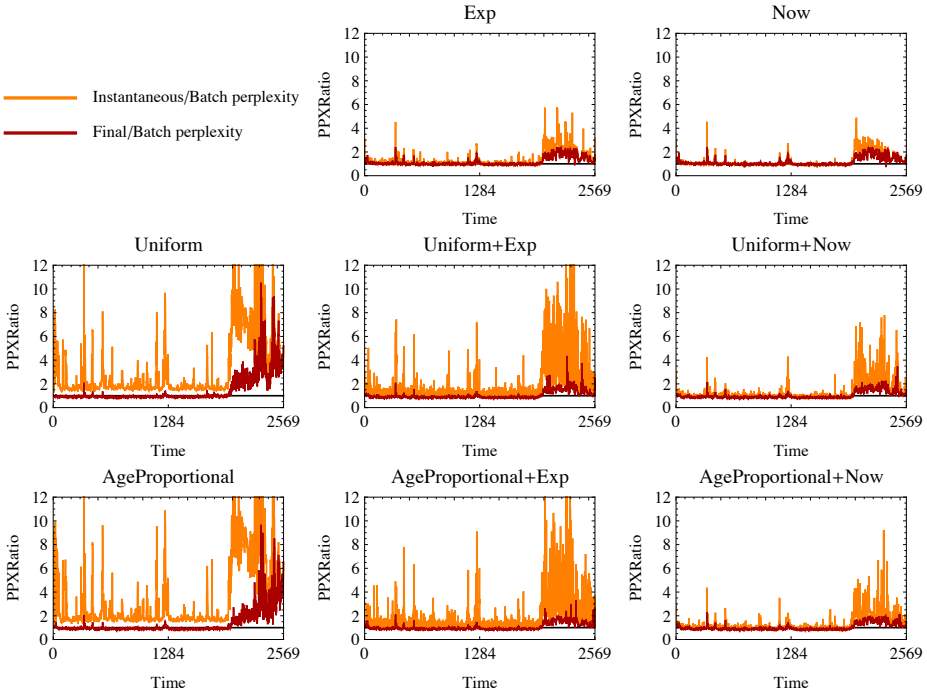


(a) Refinement time $T_R = 40$ ms.

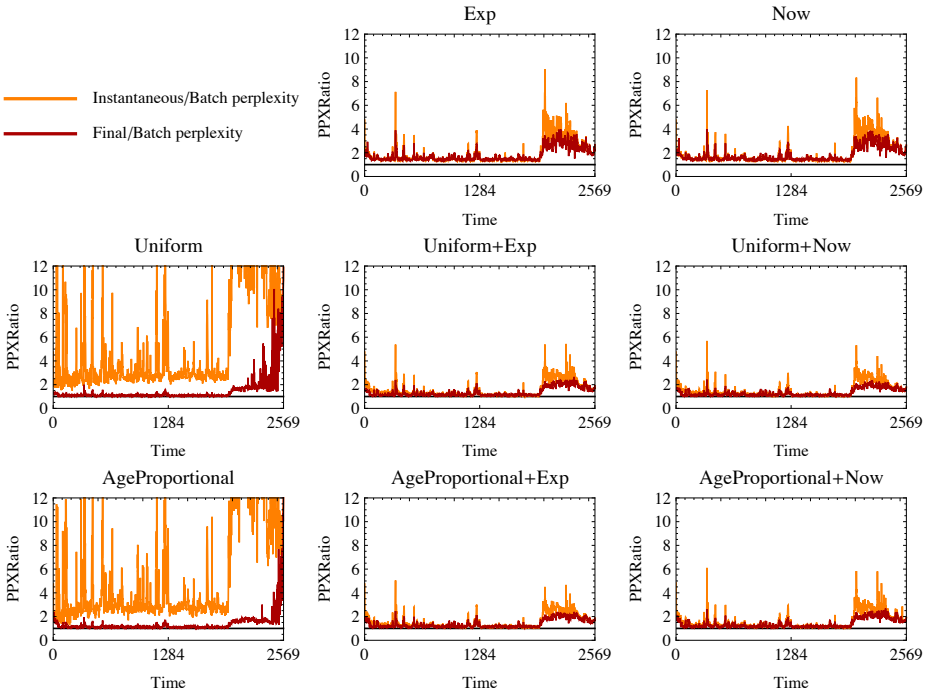


(b) Refinement time $T_R = 160$ ms.

Figure 3.11: Aerial dataset – ratio of instantaneous and final perplexity to batch perplexity, for each time step



(a) Refinement time $T_R = 40$ ms.



(b) Refinement time $T_R = 160$ ms.

Figure 3.12: Underwater dataset – ratio of instantaneous and final perplexity to batch perplexity, for each time step

Dataset →	2objects		aerial		underwater	
Alg.↓, T_R →	40	160	40	160	40	160
Uniform	1.21513	**1.15597**	0.8816	1.14133	0.993211	**1.13398**
AgeP.	**1.20734**	1.26958	0.832377	1.12045	0.969214	1.14734
Exp.	1.7994	2.27297	1.0656	1.54894	1.00544	1.48878
Now	1.79106	2.27042	1.05654	1.6628	0.998835	1.49153
Uni+Now	1.55508	1.42057	0.842953	1.10998	**0.88532**	1.15279
AgeP+Now	1.59996	1.57503	0.893272	1.13423	0.894387	1.193
Uni+Exp	1.48055	1.39821	**0.808759**	**1.10826**	0.887202	1.16382
AgeP+Exp	1.60926	1.57668	0.874776	1.13783	0.906826	1.1859

Table 3.4: Mean final perplexity of a time step, computed online, normalizes by batch perplexity (given in Tab. 3.3) . T_R is the mean refinement time in milliseconds for a time step.

Dataset →	2objects		aerial		underwater	
Alg.↓, T_R →	40	160	40	160	40	160
Uniform	3.50014	4.82605	1.66513	2.79854	1.90751	3.14073
AgeP.	3.71459	5.09174	1.71593	2.85599	1.99735	3.13597
Exp.	1.85907	2.31035	1.17068	1.59899	1.07371	1.49085
Now	1.82919	2.31286	1.08926	1.71185	1.00659	1.50235
Uni+Now	1.75666	**1.77041**	**0.99453**	1.22674	**0.988827**	**1.24204**
AgeP+Now	**1.73974**	1.88394	1.02863	1.24738	0.996776	1.27098
Uni+Exp	2.26779	1.78492	1.2795	**1.21763**	1.35703	1.24965
AgeP+Exp	2.32011	1.87336	1.33975	1.23843	1.37546	1.25631

Table 3.5: Mean instantaneous perplexity perplexity of a time step, computed online, normalizes by batch perplexity (given in Tab. 3.3). T_R is the mean refinement time in milliseconds for a time step.

From our experiments we find that although Uniform and Age Proportional Gibbs samplers perform well when it comes to final perplexity of the dataset, they however perform poorly when measuring instantaneous perplexity. Low instantaneous perplexity, which is measured one time step after an observation is made, is essential for use of topic modeling in robotic applications. We would like to make decisions based on current observations, and hence low instantaneous perplexity is crucial. We find that the mixed Gibbs samplers such as Uniform+Now perform consistently well. Note that all experiments with the mixed Gibbs samplers were performed with a fixed mixing ratio $\eta = 0.5$, giving equal weight to local and global refinement. We are confident that better tuning of this variable will result in even better performance of ROST.

3.8 Summary

In this chapter we proposed ROST, a realtime online spatiotemporal topic modeling technique that models the latent context of the streaming spatiotemporal observation data collected by a robot. We evaluated the performance of ROST in two ways.

First, using several artificial and natural datasets we showed that the proposed technique results in significantly better topic labels, when compared with Latent Dirichlet Allocation based topic model that does not take into account spatiotemporal context of observed data. We measured the performance in terms of mutual information with the ground truth data, and perplexity of the observations given the computed topic labels.

Second, we measured the performance of several Gibbs samplers, including those proposed by o-LDA and incremental LDA, in terms of how well the topic labels converge, one time step after the observation time. This measurement criterion is essential for evaluating the performance of the proposed technique in the context of robotics, where we need to make instantaneous decisions. We showed that the proposed mixed Gibbs samplers such as Uniform+Now perform consistently better than other samplers which just focus on recent observation, or which refine all observation with equal probability.

Chapter 4

Navigation Summaries

4.1 Introduction

A navigation summary is a synopsis of observations made by a robot on a trajectory [26]. This problem is related to the *vacation snapshot problem*[14], a term that was coined to refer to the algorithmic task analogous to what many tourists face: to summarize their vacation using a small set of images.

The summarization problem can be defined in different ways, however, for applications such as monitoring and exploration, we are interested in summaries which are representative of the full range observed measurements, and not just focus on the mean. Popular clustering-based summarization techniques such as kMeans or other clustering based techniques are not useful in this context as they ignore the outliers, which often are the most interesting observations. This work poses the summarization problem as a sampling problem, where we would like to identify samples in a summary such that given this summary, none of the remaining observed data would be surprising. We define the surprise score of an observation as its information gain over the observations in the summary. A good summary then minimizes the maximum surprise score of the observed data. Figure 4.1 shows an illustrated example of a navigation summary.

Navigation summaries can either be computed offline, once the path traversal is over, or online as the observations are being made. In this chapter we will discuss both versions of

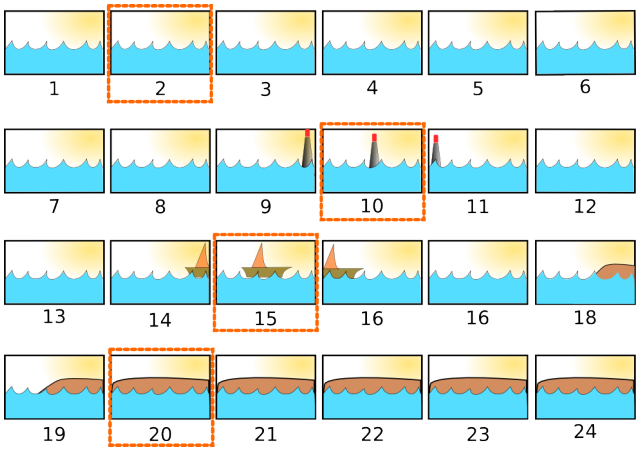


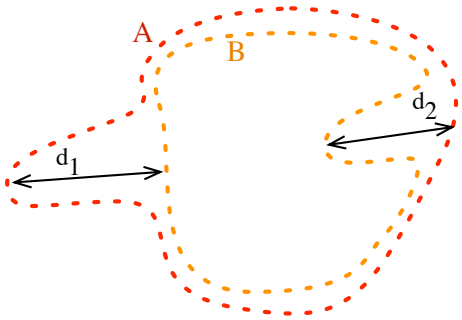
Figure 4.1: An illustrated example of a navigation summary. The sequence of images represent the observations made by a robot as it is traversing a terrain. The boxes indicates one possible choice of the summary images. These images capture both the surprise, and the mean appearance of the terrain.

the problem.

In Section 4.3, we explore several offline strategies for picking the summary images. First strategy uses the k-medoids clustering to produce summaries. The second strategy models the problem as an instance of the classical k-centers problem, which is NP-Hard, but has a simple approximate solution that involves greedily picks the most surprising images. This is similar to picking the images at the corners of the high dimensional manifold formed by images in the information space. The third strategy models the problem as an instance of the classical *set cover* problem, which again is NP-hard, but has a polynomial time approximate solution.

In Section 4.4, we propose an online strategy for picking extremum summary images. By online summarization we mean that the decision to include a sample in the summary set is made and immediately after it is acquired. Such summaries are useful in proving mission updates over a narrow communication channel [61].

This chapter focuses on observation data in the form of images and the corresponding locations, however, the techniques discussed can be applied to observations coming from any other type of sensors.



$$d_1 = \sup_{a \in A} \inf_{b \in B} d(a, b)$$
$$d_2 = \sup_{b \in B} \inf_{a \in A} d(a, b)$$
$$d_H(A, B) = \max(d_1, d_2)$$

Figure 4.2: Hausdorff Metric $d_H(A, B)$ measures the distance between two sets A and B.

4.2 Set Theoretic Surprise

In Section 2.4 we discussed the idea of Bayesian Surprise, which has been shown to correlate well with human attention. In this section, we will extend the idea of Bayesian surprise, to make it suitable for application to the summarization problem. Instead of modeling the properties of the world with a single distribution, we propose to maintain a set of local hypotheses, each corresponding to an observation sample in the summary set. This set of distributions can then be interpreted as the prior hypothesis of the properties of the world. Using a set of distributions to model the properties allows us to model worlds with arbitrary complexity, by simply increasing or decreasing the size of the summary set.

Similarly, to measure the surprise of a given observation, we model the posterior using the set containing all the observations in the summary and the given observation. Now, analogous to Bayesian surprise, we would like to measure the distance between these two distribution sets. The Hausdorff metric provides a natural way to compute distance between two such sets. For two sets A, B, the Hausdorff distance between the sets is defined as

$$d_H(A, B) = \max \left\{ \sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b) \right\}. \tag{4.1}$$

Figure 4.2 illustrates this graphically.

However, since the prior and posterior hypothesis sets only differs by one element (the

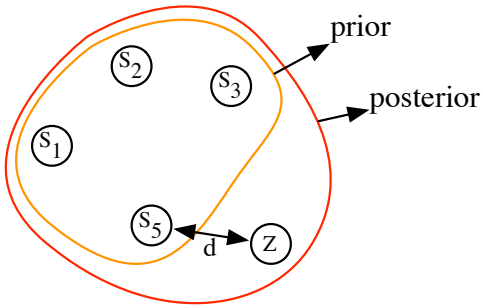


Figure 4.3: *Set Theoretic Surprise.* We model our prior using a set containing the summary images $\{S_i\}$, and the posterior using a set containing summary images and the observed image Z. Each image is represented as a bag-of-words histogram, normalized to form a probability distribution. The surprise is then defined as the Hausdorff distance between these sets of probability distribution. We use KL divergence as the distance metric. Since the two sets only differ by one element, the Hausdorff distance can be simplified to only finding the closest element in the summary set.

given observation), the surprise of a given observation Z given the summary set $\boldsymbol{S} = \{S_i\}$ is:

$$\xi(Z|\boldsymbol{S}) \quad = \quad \inf_i \xi(Z|S_i) \tag{4.2}$$

We call this measure of surprise "Set Theoretic Surprise". This is visualized graphically in Figure4.3.

Each observation can be represented using a variety of descriptors depending on the sensors used. If $d_r(Z_i, Z_j)$ is the normalized distance between two observations Z_i, Z_j using descriptor r, then we define surprise:

$$\xi(Z_i|Z_j) = \max_r d_r(Z_i, Z_j). \tag{4.3}$$

For visual bag-of-words descriptors, we use symmetric KL divergence distance function, and for location data we use Euclidean distance to compute the surprise.

Surprise $\xi(Z|\boldsymbol{S})$ can be interpreted as the amount of information gained in observing Z. Given the set of all the observations $\boldsymbol{Z} = \{Z_i\}$, ideally we would like to choose a summary

set $\boldsymbol{S} = \{S_i\}$ such that the mean surprise

$$\bar{\xi}(\boldsymbol{Z}|\boldsymbol{S}) = \frac{1}{|\boldsymbol{Z}|}\sum_i \xi(Z_i|\boldsymbol{S}), \tag{4.4}$$

and max surprise

$$\hat{\xi}(\boldsymbol{Z}|\boldsymbol{S}) = \max_i \xi(Z_i|\boldsymbol{S}), \tag{4.5}$$

are both low.

A summary with low $\bar{\xi}$ implies that it captures the mean properties of the world well. A summary with low $\hat{\xi}$ implies that it captures the outliers well.

In the following sections, we will explore different techniques to choose samples in the summary set, offline and online.

4.3 Batch Summarization

We are given a set of observations $\boldsymbol{Z}$ and number of desired samples k in the summary set $\boldsymbol{S}$. The goal is to come up with a set of observations called summary set $\boldsymbol{S} \subset \boldsymbol{Z}, |\boldsymbol{S}| = k$, such that the mean surprise $\bar{\xi}$ and max surprise $\hat{\xi}$ are low.

The k-medoids summaries (see Section 4.3.1) minimizes the mean distance between these cluster centers and other points within the cluster. Hence, if we use the surprise function described in (4.3) as the distance function for k-medoids, the cluster centers then correspond to a summary which minimizes the mean surprise $\bar{\xi}$.

The k-centers summaries (see Section 4.3.2) minimizes the max distance between these cluster centers and other points within the cluster. This correspond to a summary which minimizes the max surprise $\hat{\xi}$.

The k-cover summaries (see Section 4.3.3) maximize the number of samples covered by the summary, given a threshold distance. We will present variants of this techniques, which will allow us to tune the behavior of the summaries to lie in between kMedoid and kCenter summaries.

4.3.1 k-Medoid Summaries

Initialize medoids $\boldsymbol{S} = \{S_1, \ldots, S_k\}$ by randomly picking k points from $\boldsymbol{Z}$
while *true* **do**
  Assign each point in $\boldsymbol{Z}$ that is not in $\boldsymbol{S}$ to the cluster with the closest medoid point in $\boldsymbol{S}$
  Recompute new medoids $\boldsymbol{S}$ for each cluster
  if $\boldsymbol{S}$ *does not change* **then**
    return $\boldsymbol{S}$
  end
end

Algorithm 3: k-MEDOIDS $(\boldsymbol{Z}, k)$

The most common popular clustering algorithm for summarization is perhaps the k medoids clustering algorithm [64].

Given a set of observation $\boldsymbol{Z} = \{Z_1, \ldots, Z_n\}$ and number of desired samples k, k-medoids clustering defines the cost of a given set of cluster center points $\boldsymbol{S} = \{S_1, \ldots, S_k\}$ as

$$\text{C}ost(\boldsymbol{S}) = \frac{1}{|\boldsymbol{Z}|} \sum_i \min_j d(Z_i, S_j) = \bar{\xi}(\boldsymbol{Z}|\boldsymbol{S}), \tag{4.6}$$

where the center points $\boldsymbol{S} \subseteq \boldsymbol{Z}$, and $d(.)$ is a distance metric on the input samples. The goal now is the find $\boldsymbol{S}$ that minimizes this cost. These cluster centers can be considered as our summary because they minimize the distance between these cluster centers and other points within the cluster.

The k-medoids clustering problem is similar to k-means [77], with difference that k-medoids uses one of the input samples to designate the cluster centers, whereas k-means can use any point in space as cluster centers.

The k-medoids clustering, like k-means clustering is NP-hard, however good heuristic solutions exist. The standard algorithm to solve the k-medoids clustering problem is called Partitioning Around Medoids [65], which is described in Algorithm 3.

4.3.2 k-Center Summaries

Given the set of all observations $\boldsymbol{Z} = \{Z_i\}$, and the number of desired samples k in the summary set, we would like to find the set $\boldsymbol{S} \subseteq \boldsymbol{Z}, |\boldsymbol{S}| = k$, such that the

$$Cost(\boldsymbol{S}|\boldsymbol{Z}) = \max_i \min_j d(Z_i, S_j) = \hat{\xi}(\boldsymbol{Z}|\boldsymbol{S}) \tag{4.7}$$

is minimized. This is like finding centers of k balls of smallest (but equal) sizes, which cover all the points in $\boldsymbol{Z}$. This is essentially the k-centers problem.

Approximate Solution

If the distance function obeys the triangle inequality, then not only is the k-center problem NP-hard, but Huse and Nemhauser [55] showed that an α-approximation of this problem is also NP-hard for $\alpha < 2$ (i.e., for any approximate that guarantees an approximate this good).

Consider the greedy strategy presented in Algorithm 4, which we refer to as the *Extremum Summary* algorithm. We initialize the summary with an arbitrary observation, then in each iteration, we choose an observation which is farthest away from the observations in the current summary, and add it to the summary set. This algorithm has an approximation ratio of 2, and hence is likely the best we can do unless P=NP [46].

$\boldsymbol{S} \leftarrow \{Z_{\text{random}}\}$
$\boldsymbol{Z} \leftarrow \boldsymbol{Z} \setminus Z_{\text{random}}$
repeat
  $m \leftarrow \text{argmax}_i \ \min_j d(Z_i, S_j)$
  $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{Z_m\}$
  $\boldsymbol{Z} \leftarrow \boldsymbol{Z} \setminus Z_m$
until $|\boldsymbol{S}| \geq k$
return $\boldsymbol{S}$

Algorithm 4: ExtremumSummary $(\boldsymbol{Z}, k)$. Compute a summary as a subset of input samples $\boldsymbol{Z}$, by greedily picking the samples farthest away from samples in the current summary.

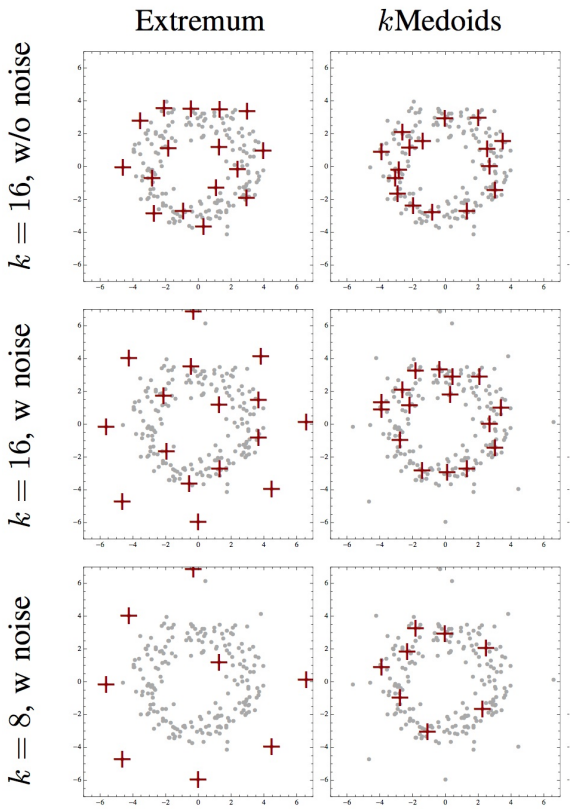


Figure 4.4: Extremum vs k-Medoids Summaries. The dataset consists of 200 points generated randomly around a circle in $\mathbb{R}^2$. The summaries generated by the two algorithms are shown in the first row. Since there are no outliers in the dataset, the summaries seem similar. In the second row, we add 8 extra samples from a different distribution, which are all outliers in the context of the other points. Adding these outliers highlights the differences between the two strategies. We see that extremum summary favors picking the outliers, whereas the k-medoids summary ignores these outliers completely. In the last row, we reduce the summary size and see the differences exaggerated even more. The extremum summary is almost entirely made up of the outliers, whereas the k-medoid summary is only representative of the mean.

k-Center Summaries Favor Outliers

Figure 4.4 highlights the characteristic differences in summaries generated by the extremum summary algorithm, and the k-Medoids algorithm. The dataset consists of 200 points generated randomly around a circle in $\mathbb{R}^2$. The summaries generated by the two algorithms are shown in the first row. Since there are no outliers in the dataset, the summaries seem similar.

In the second row of Figure 4.4, we add 8 extra samples from a different distribution, which are all outliers in the context of the other points. Adding these outliers highlights the differences between the two strategies. We see that extremum summary favors picking the outliers, whereas the k-medoids summary ignores these outliers completely.

In the last row of Figure 4.4, we reduce the summary size and see the differences exaggerated even more. The extremum summary is almost entirely made up of the outliers, whereas the k-medoid summary is still only representative of the mean.

Although a k-medoids summary might be useful when we want to model the mean properties of an environment, if however we are interested in identifying the range of what was observed, then an extremum summary is more useful since its objective function ensures that each observations is close to at least one of the summary samples.

4.3.3 Summaries Using Set Cover Methods

The task of selecting samples in the summary set can also be modeled as an instance of the classical *Set Cover Problem*.

We define the cover of an observation Z to be the set of all observations that are only surprising up-to a surprise threshold ξ_T:

$$C(Z|\xi_T) = \{\{Z_j\} : \xi(Z_j|Z) < \xi_T\}\,. \tag{4.8}$$

Similarly we define the cover of a summary set $\boldsymbol{S}$:

$$C(\boldsymbol{S}|\xi_T) = \bigcup_{S_i \in \boldsymbol{S}} C(S_i|\xi_T). \tag{4.9}$$

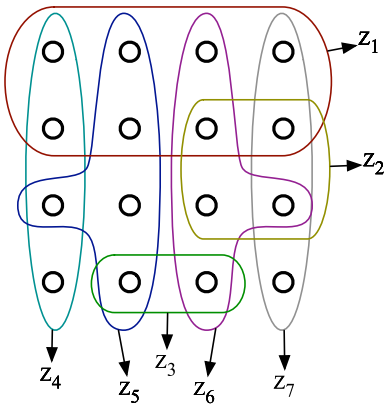


Figure 4.5: An instance of the *Set Cover* problem. The goal is to find the smallest number of sets, which span all the elements. In this example, set $\{Z_4, Z_5, Z_6, Z_7\}$ is the smallest set of sets that cover all the elements in the universe.

Our goal now is to find the minimal set of samples $\boldsymbol{S}$, which cover the entire terrain. This is essentially an instance of the classical *Set Cover* problem with $|\boldsymbol{Z}|$ elements in the universe and $|\boldsymbol{S}|$ sets which span the universe. Figure 4.5 shows an example of an instance of the *Set Cover* problem.

Set cover, like k-centers, is known to be NP-hard [62]. Hence we use a greedy strategy to pick our samples. There are several variants of the set-cover problem that are relevant for producing summaries. These are discussed below.

Fixing Coverage Ratio

$\boldsymbol{S} \leftarrow \{S_{\text{init}}\}$
repeat
  $Z_{\max} \leftarrow \underset{Z_i \in \boldsymbol{Z}}{\operatorname{argmax}} \left| C(\{Z_i\} \cup \boldsymbol{S}|\xi_T) \setminus C(\boldsymbol{S}|\xi_T) \right|$
  $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{Z_{\max}\}$
  $\boldsymbol{Z} \leftarrow \boldsymbol{Z} \setminus Z_{\max}$
until $\frac{\left|C(\boldsymbol{S}|\xi_T)\right|}{|\boldsymbol{Z}|} < \gamma$
return $\boldsymbol{S}$

Algorithm 5: γ-CoverSummary $(\boldsymbol{Z}|\xi_T, \gamma)$. Computes a summary as a subset of all observations $\boldsymbol{Z}$, given the surprise threshold ξ_T, by greedily picking the observations with maximum cover. We stop when the coverage ratio is more than γ.

Algorithm 5 greedily picks $Z_{\max} \in \boldsymbol{Z}$ which provides maximum additional cover, and then adds it to the summary set $\boldsymbol{S}$. The algorithm stops when the coverage ratio exceeds the parameter γ. Setting coverage parameter $\gamma = 1$ implies that the algorithm stops when the summary covers all the observations. Lower values of γ can be used for higher noise tolerance, at the cost of increased likelihood of missing a truly surprising sample.

To initialize the summary set, S_{init} is chosen to be the sample with smallest mean distance to other samples. The number of samples in the summary obtained using this greedy strategy, is guaranteed to be no more than $OPT \log |\boldsymbol{Z}|$, where OPT is the number of sets in the optimal summary set [18].

Max k-Cover Given a Surprise Threshold

Algorithm 6 greedily picks k summary samples which provide maximum combined cover for the summary set. This problem of finding the optimal set of samples with maximum cover is known as the *Max k-Cover* problem. *Max k-Cover*, like *Set Cover* is also known to be NP-Hard, and the greedy approach to this problem approximates the optimal solution to within a ratio of $1 - 1/e$ [99].

$\boldsymbol{S} \leftarrow \{S_{\text{init}}\}$
repeat
  $Z_{\max} \leftarrow \underset{Z_i \in \boldsymbol{Z}}{\operatorname{argmax}} \left| C(\{Z_i\} \cup \boldsymbol{S} | \xi_T) \setminus C(\boldsymbol{S} | \xi_T) \right|$
  $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{Z_{\max}\}$
  $\boldsymbol{Z} \leftarrow \boldsymbol{Z} \setminus Z_{\max}$
  $k \leftarrow k - 1$
until $k > 0$
return $\boldsymbol{S}$

Algorithm 6: K-CoverSummary $(\boldsymbol{Z}|k, \xi_T)$. Computes a summary of size k, given the surprise threshold ξ_T by greedily picking samples with maximum cover.

Max k-Cover Given a Coverage Ratio

Often times we do not have a way of estimating a good value for the surprise threshold ξ_T. Instead, we can fix the size of the summary k, and then find the smallest value of ξ_T that

Algorithm →	Extremum		k-Medoid		k-Cover(1)		k-Cover(0.9)	
Dataset ↓	$\bar{\xi}$	$\hat{\xi}$	$\bar{\xi}$	$\hat{\xi}$	$\bar{\xi}$	$\hat{\xi}$	$\bar{\xi}$	$\hat{\xi}$
Ring(100) no outliers	0.086	0.175	0.067	0.207	0.073	0.160	0.0661	0.205
Ring(100) with outliers	0.077	0.168	0.055	0.276	0.071	0.160	0.055	0.273
Street View	0.110	0.205	0.078	0.253	0.082	0.208	0.073	0.364
Aerial View	0.154	0.263	0.095	0.547	0.101	0.220	0.086	0.522

Table 4.1: Evaluation of batch summaries. We see that kMedoid summaries have low mean mean surprise $\bar{\xi}$, whereas kCenter summaries (extremum summaries) have low max suprise $\hat{\xi}$. k-Cover summaries can behave either like extremum summaries or kMedoid summaries, by tuning the coverage ratio parameter.

gives us the desired coverage ratio.

Using Algorithm 6, we can hence define ξ_T as:

$$\xi_T(k) = \min_{\xi_{T'}} \left\{ \boldsymbol{S} = \textsc{k-CoverSummary}(\boldsymbol{Z}|k, \xi_{T'}), \frac{\left|C(\boldsymbol{S}|\xi_{T'})\right|}{|\boldsymbol{Z}|} \geq \gamma \right\}. \tag{4.10}$$

We can use a binary search procedure to find optimal ξ_T, and either terminate after a fixed number of iterations or when interval size is smaller than the smallest inter-sample distance.

4.3.4 Evaluation of Batch Summaries

We tested the summarization strategies presented in this chapter on datasets collected by different classes of vehicle moving outdoors, and on artificially generated datasets. We evaluate the summaries based on their mean and max surprise scores. Table 4.1 summarizes the mean and max surprise results for each dataset.

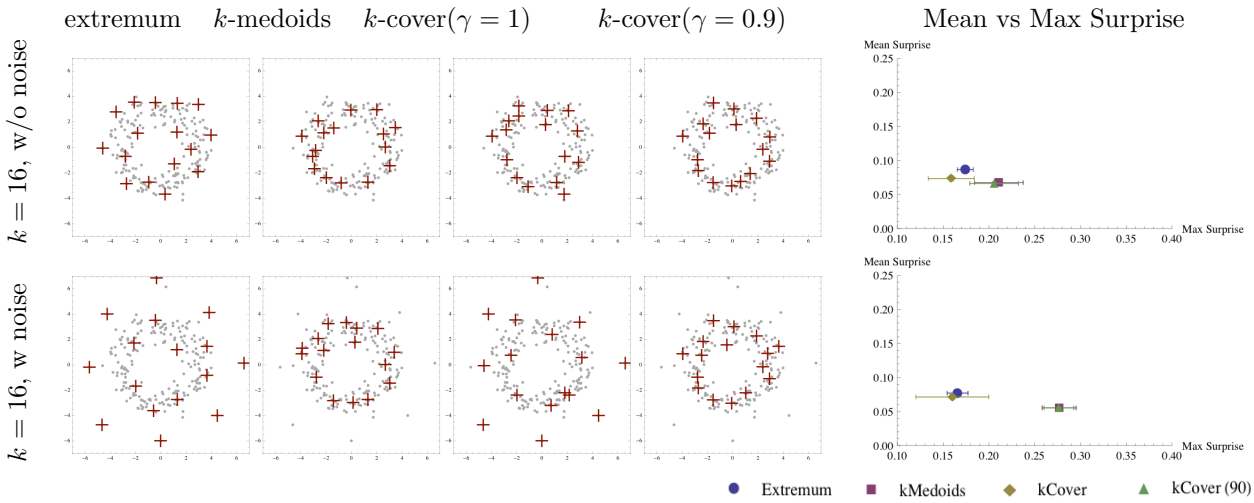


Figure 4.6: Summaries for Noisy Datasets. When the data has little noise (row 1), we see that all algorithms perform similarly. When some noisy samples are added to the same data(row 2), both extremum summary algorithm and k-cover($\gamma = 1.0$) favour picking these outliers. However, by lowering γ, we can make k-cover immune to noise and perform similar to k-medoids. Mean vs Max surprise plots shown are averages for 100 randomly generated datasets from the same distribution. The dataset shown above is one specific example.

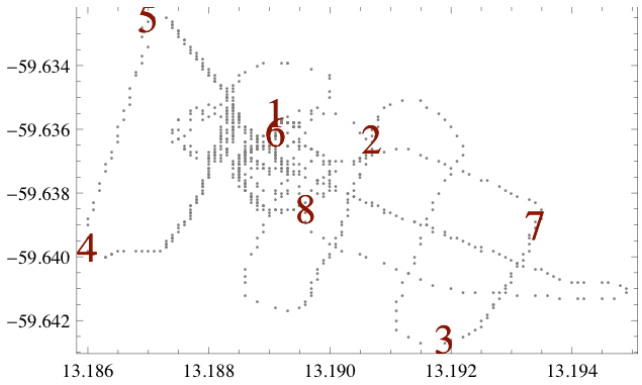
$\mathbb{R}^2$ Ring Dataset

In Figure 4.6, we show 200 samples generated randomly around a circle in $\mathbb{R}^2$. We then add 8 extra samples from a different distribution, representing the outliers. This dataset allows us to visualize the difference between the three proposed strategies clearly.

In Figure 4.6 (row 1), we see that when there is little noise, and the size of the summary is large ($k = 16$), all algorithms perform similarly, with k-cover(γ=1) having the lowest max surprise, and both k-cover(γ=0.9) and k-medoids having lowest mean surprise.

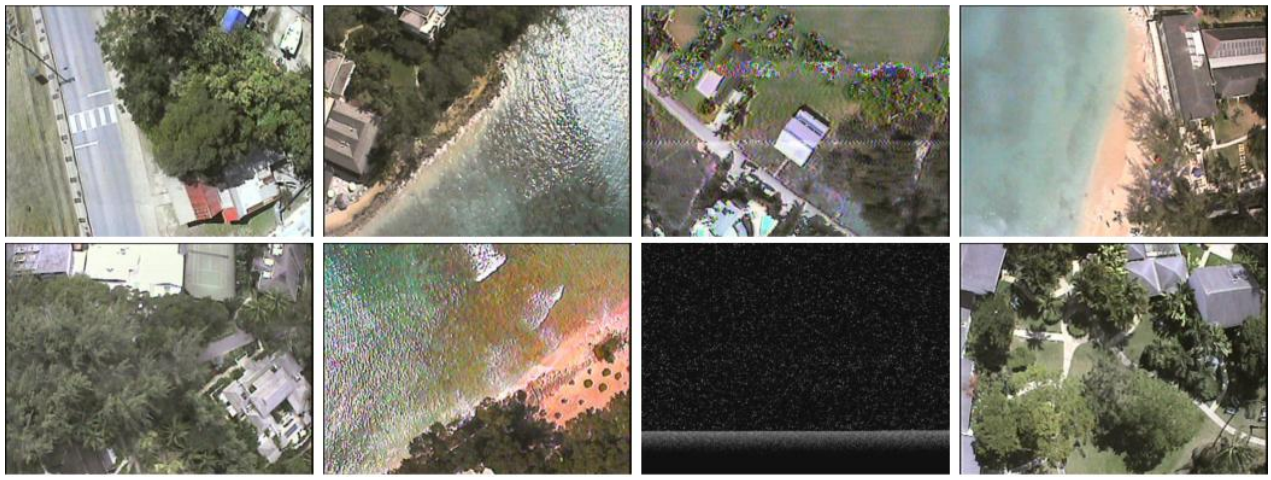
Figure 4.6 (row 2) shows the result of adding outliers to this dataset. We see that the k-cover(γ=1) and the extremum summaries look similar and both sample the outliers and the mean samples well. The k-cover(γ=1) and the k-medoids summary on the other hand completely ignore the outliers. As a result they have high max surprise and low mean surprise.

(a) Summary Images

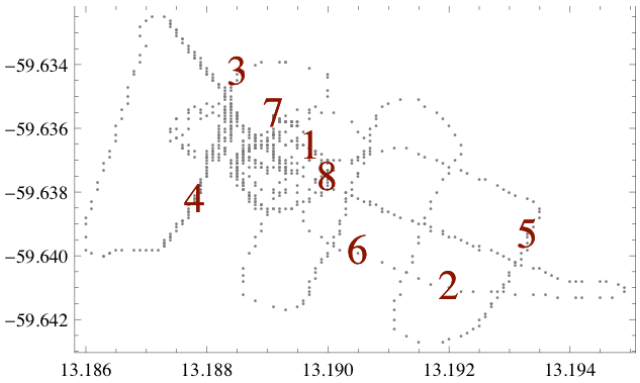


(b) Summary Image Locations

Figure 4.7: Aerial View Dataset: Extremum Summary. The algorithm prefers picking outliers and hence prefers picking noisy images (#1,3,4), or images at the geographic corners (#3,4,5,7). Image 6 is of grass, taken just before the landing.



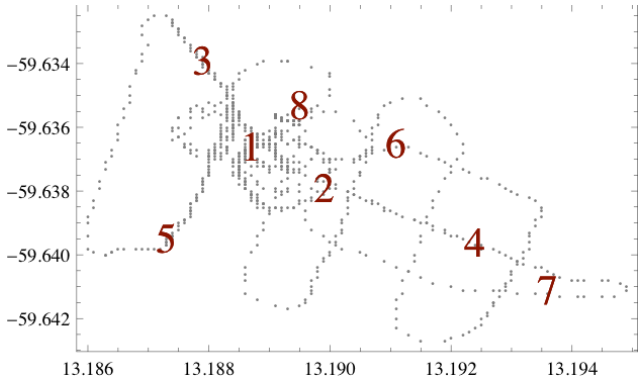
(a) Summary Images



(b) Summary Image Locations

Figure 4.8: Aerial View Dataset: k-Cover Summary with $\gamma = 1$. The algorithm is forced to choose summary picks which cover all samples including the noisy samples, but not necessarily pick them. Hence, it mitigates the effect of noise, while not completely eliminating it. We see the inclusion of some obvious outliers like image #7

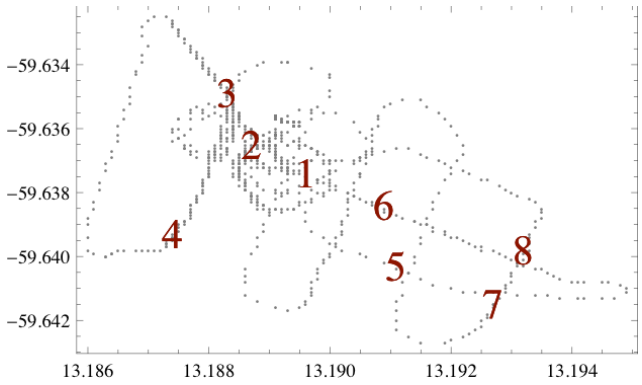
(a) Summary Images



(b) Summary Image Locations

Figure 4.9: Aerial View Dataset : k-Cover Summary with $\gamma = 0.9$. The k-cover algorithm with $\gamma = 0.9$ does a good job of ignoring outliers completely, and presents a clearer picture of what was observed during flight: ocean(#5), beaches(#2,7), buildings(#4,6), fields(#1,8).

(a) Summary Images



(b) Summary Image Locations

Figure 4.10: Aerial View Dataset : k-Medoids Sumary. The algorithm produces a summary that is similar to k-cover summary with $\gamma = 0.9$

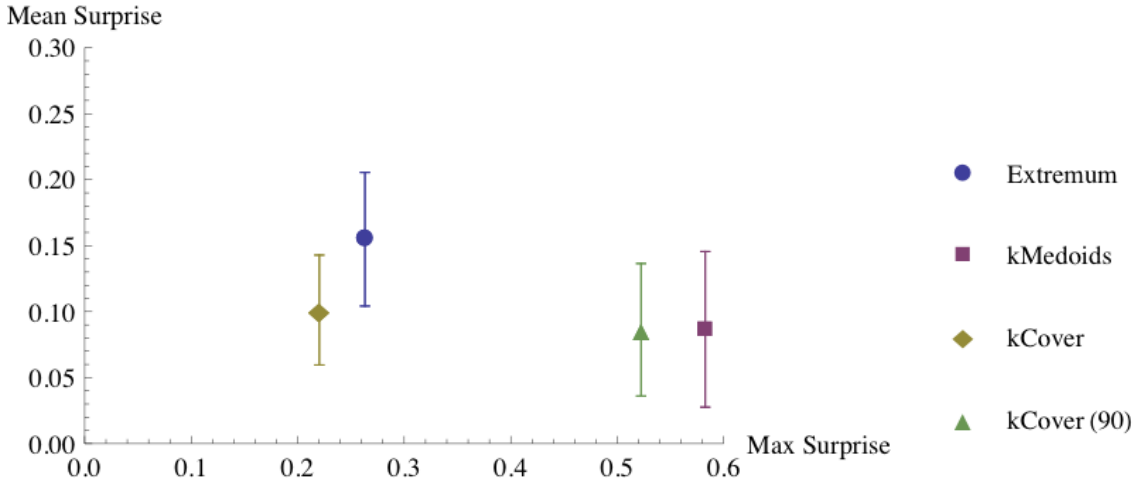


Figure 4.11: Aerial View Dataset - Mean vs Max Surprise. Images were transmitted by a UAV over an analogue transmission channel, and hence contains noise due to interference. The plot shows mean and max surprise score for the four different types of summaries that we consider.

Aerial View Dataset

The aerial view (AV) dataset contains 847 images taken by a GPS equipped unmanned aerial vehicle (UAV) (Section 2.7.3). The images in this dataset were transmitted over an analog channel before being captured digitally. As a result, some of the images in the AV dataset are noisy due to temporary signal losses, and interference with on board electronics and radios. This results in the dataset having many undesirable outlier images.

Visual distance between images was computed using symmetric KL divergence between SURF [6] bag-of-words descriptors, with a vocabulary of 1000 words. Geographic distance was computed by taking the Euclidean distance between the corresponding GPS coordinates. These distances were normalized to be between [0-1]. For each pair or images we set their distance as the maximum of the visual or geographic distance.

Figure 4.7 highlights the problem of using a noise sensitive algorithm such as the extremum summary algorithm on noisy data. Since the extremum summary algorithm favors picking outliers, the summary generated contains several images with visible noise related artifacts, and at least one image with complete loss of signal. However due to the constraints imposed by geographic data, it still manages to get good geographic coverage as shown in the map in Fig 4.7. The map shows location of each image in the summary. We see that images 3,4,5 and 7 correspond to the four corners of the map, which is the behavior expected from

the extremum summary algorithm.

Figure 4.8 shows images form the k-cover summary with coverage ration γ=1. Qualitatively we see that it provides a much better representation of the environment than the extremum summary. However, coverage ratio γ=1 implies that every sample in the observation set must be covered by a summary sample. The algorithm is therefore forced to pick image 7, which is the result of a temporary signal loss, and is not representative of the environment.

Figure 4.9 show the k-cover summary with $\gamma = 0.9$. We see that the algorithm is able to ignore the noisy samples, and produces a navigation summary with good geographical and visual coverage. We see that all major features of the environment: ocean, beach, buildings, fields, and trees, are represented in the summary.

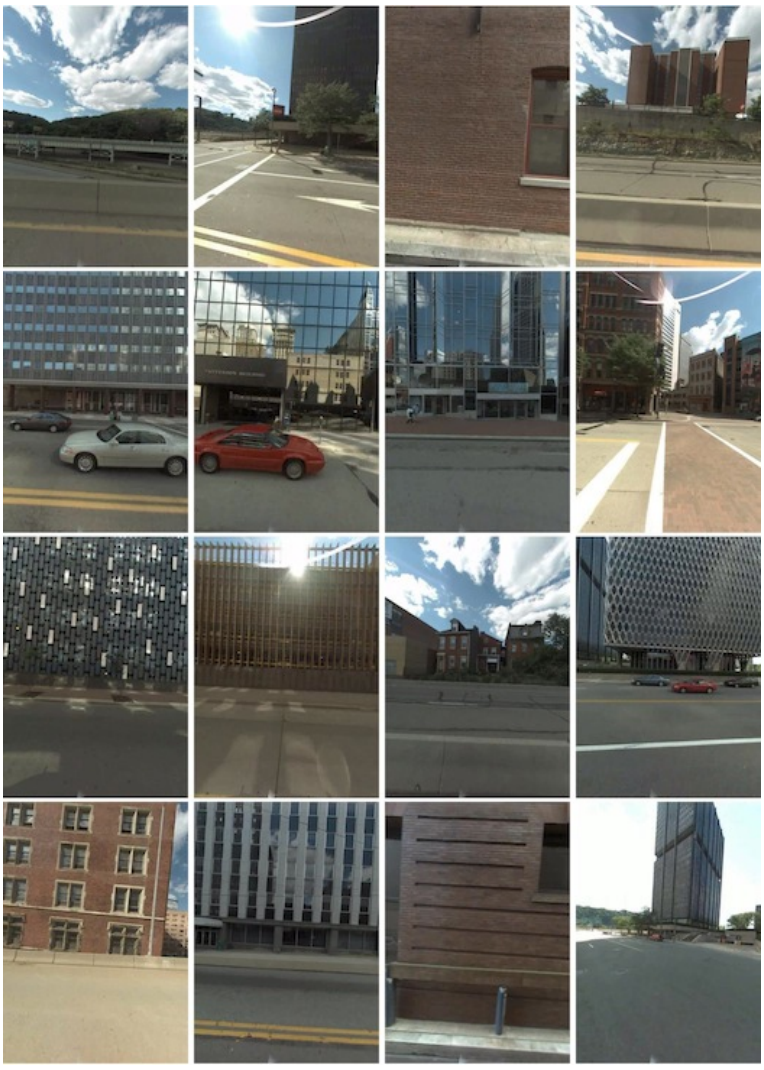
Figure 4.10 show the k-medoids algorithm produces a summary very similar to the k-cover(γ=1) summary, as both algorithms ignore noisy samples.

Street View Dataset

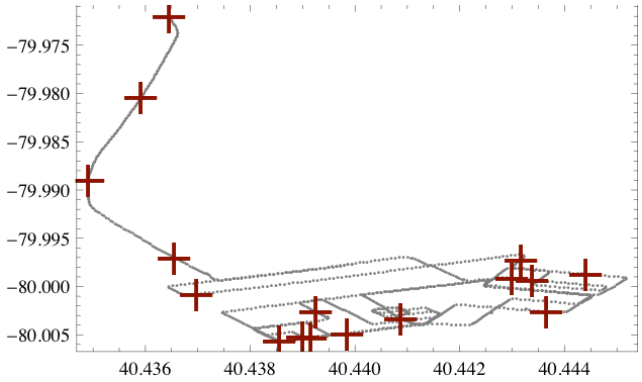
The street view (SV) dataset contains 1255 geo-tagged images of a city centre, subsampled from the data originally collected by Google Inc., for their street view application. Unlike the AV dataset, the SV dataset has negligible visual noise. We computed inter sample distance in the same way as the aerial view dataset; combining visual and geographic distances.

Figure 4.12 shows the summary generated by the extremum summary algorithm which favors picking outliers or the corner samples. Upon closer inspection, it seems that many of the selected images correspond to images of buildings with different repetitive patterns, completely occupying the camera's field of view. Such images would be represented by histograms with a sharp peak at different locations, when using the bag-of-words representation of an image. As a result, the KL divergence between these images is very high, and hence, they are favored by the extremum summary algorithm.

Figure 4.15 shows the summary produced by the k-medoids algorithm. Compared to the images in the extremum summary, the images in the k-medoids summary will not have a peaky bag-of-words histogram. The images in the summary do not contain repetitive patterns and represented by many different visual words. As a result, the selected images share visual

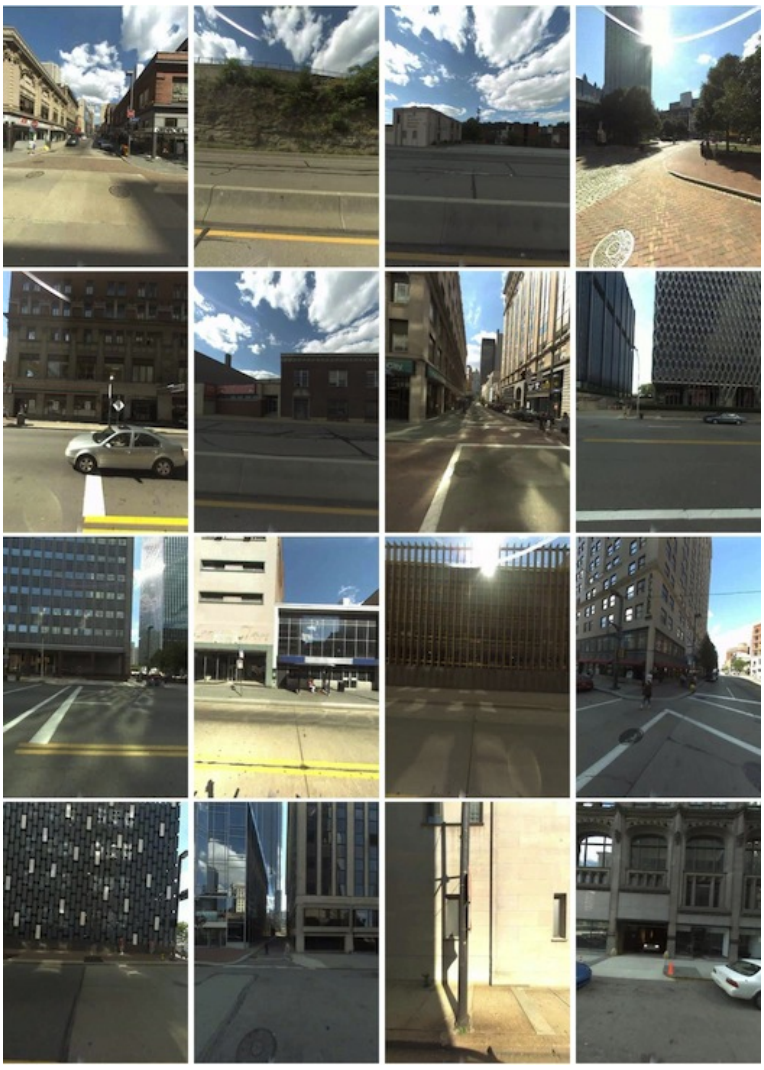


(a) Summary Images

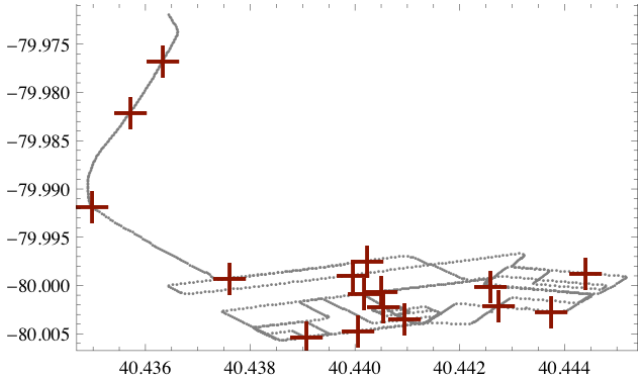


(b) Summary Image Locations

Figure 4.12: Street View Dataset: Extremum Summary

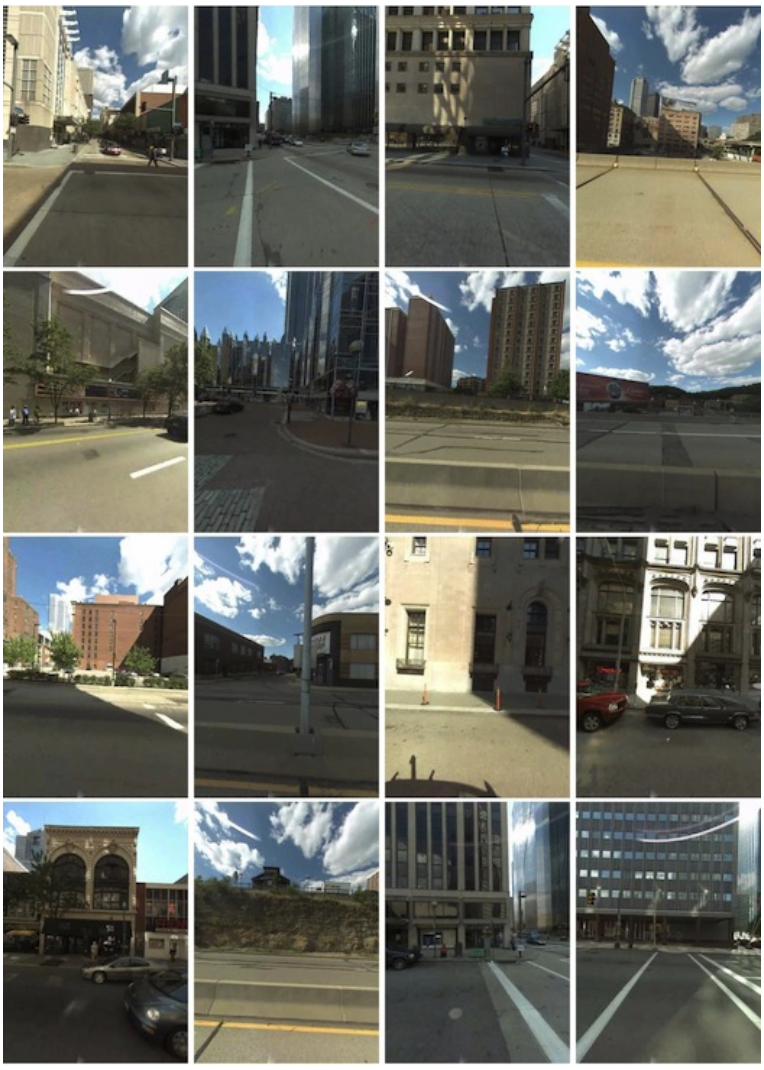


(a) Summary Images

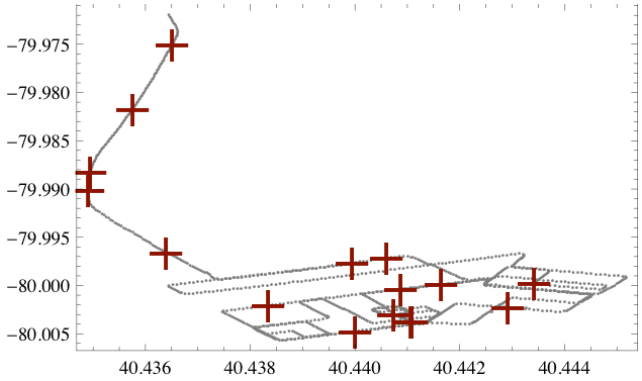


(b) Summary Image Locations

Figure 4.13: Street View Dataset: k-Cover Summary with $\gamma = 1.0$.

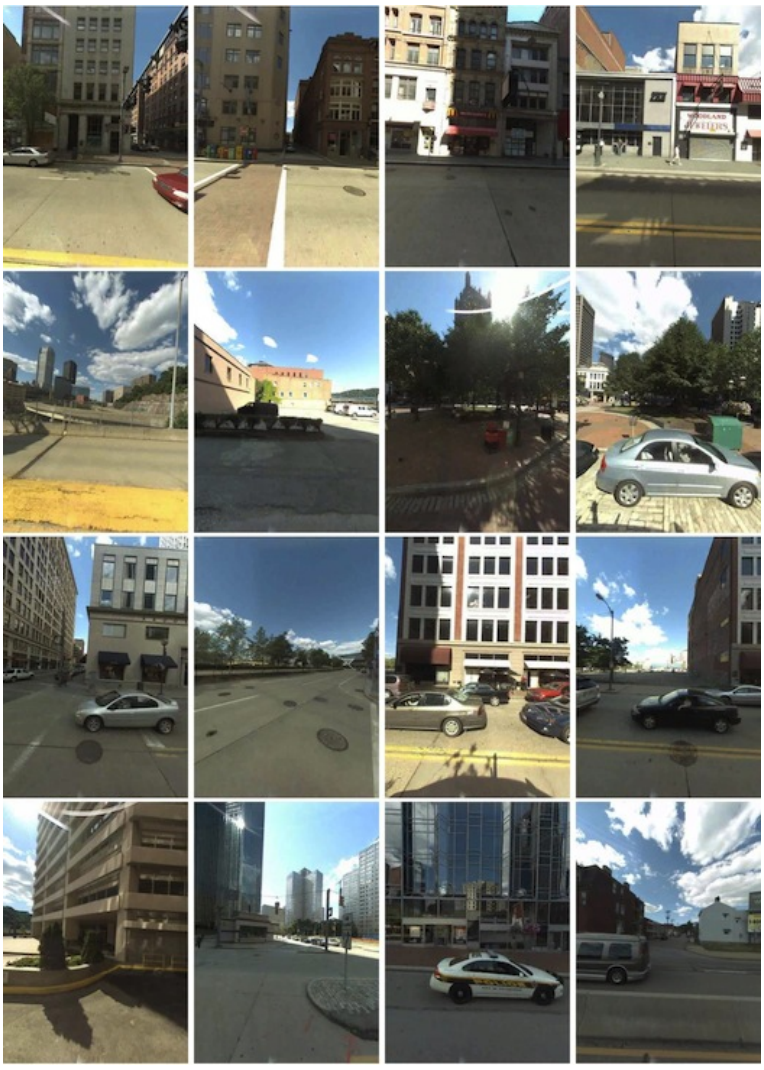


(a) Summary Images

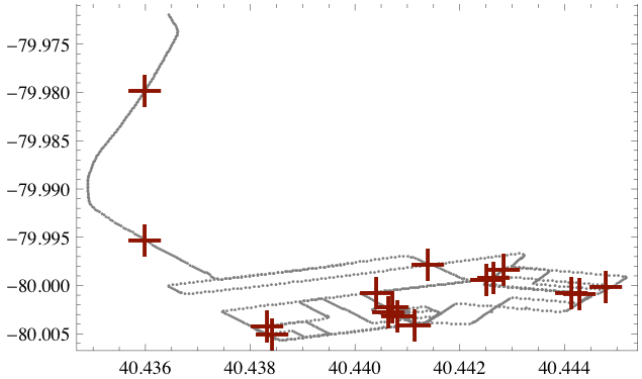


(b) Summary Image Locations

Figure 4.14: Street View Dataset: k-Cover Summary with $\gamma = 0.9$.



(a) Summary Images



(b) Summary Image Locations

Figure 4.15: Street View Dataset: k-Medoids Summary

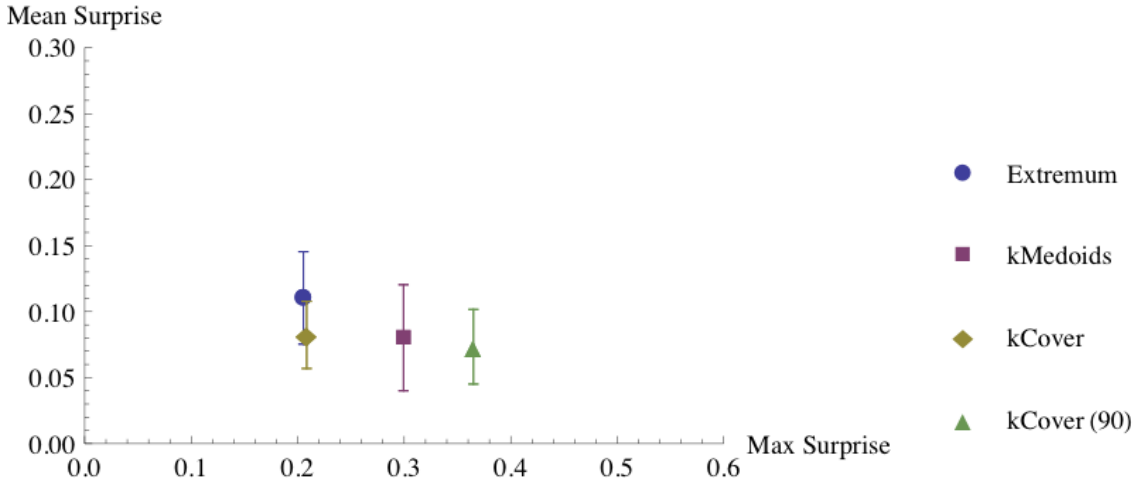


Figure 4.16: Street View Dataset: Mean vs Max Surprise. Summaries were generated using surprise score calculated over visual and geographic distance. Extremum summary algorithm focuses on finding the outliers and results in low max surprise. Most images are of unique buildings which are rare in the dataset, but are extremely different in appearance from everything else. The k-medoids algorithm produces a summary with no outliers and low mean surprise. The k-cover algorithm has in-between characteristics and produces a summary with low mean and max surprise. It contains some of outliers which exist in the extremum summary , while not ignoring the samples representing the mean appearance.

properties with many other images in the dataset, and are good candidates for characterizing the mean appearance of the environment.

Figure 4.13 show the summary generated using the k-cover algorithm. Using k-cover with $\gamma = 1.0$ is a good choice for this dataset because there is negligible noise in the dataset, and hence the outliers correspond to truly surprising observations. The summary produced by k-cover does not completely ignore the outliers, and nor does it only pick samples representing the mean. It finds a good balance between lowering the mean and max surprise. Using k-cover with $\gamma = 0.9$ produces summaries (shown in Figure 4.14) with high max surprise and low mean surprise.

This is shown by the plot in Figure **??**. The k-cover summary has some of the outliers from extremum summary, while still having some images which are more representative of the environment.

4.4 Online Extremum Summaries

Consider the task of deciding whether to include an incoming observation into the summary or not, immediately after its arrival. If we want a robot to collect a diverse set of samples such as rocks, then the currently collected samples correspond to a navigation summary of rocks observed thus far. For such robotic applications, the decision to include a sample to the summary must be made online.

Broder et. al. [15] named the strategy of picking samples above the mean or median score of the previous picks as "Lake Wobegon" hiring strategies [1]. Variation of such a strategy have supposedly been used by companies like Google and GE to hire a continuous stream of employees. We take inspiration from this idea to pick our summary samples.

4.4.1 Picking Above the Mean

Given the current summary $\boldsymbol{S} = \{S_i\}$, we define the score of an observation Z_t, observed at time t as:

$$\text{Score}(Z_t) = \min_i d(Z_t, S_i). \tag{4.11}$$

Similarly, we can define the picking threshold γ as the mean score of the samples currently in the summary:

$$\gamma = \frac{1}{|\boldsymbol{S}|} \sum_i \min_{j, j \neq i} d(S_i, S_j) \tag{4.12}$$

Now for each incoming observation, we compute its score given the current summary, and then if the score is above the current picking threshold γ, we add it to the current summary. Algorithm 7 summarizes one iteration of this algorithm.

[1] Named after the fictional town "Lake Wobegon", where according to the Garrison Keillor "all the women are strong, the men are good looking, and all the children are above average."[15]

threshold $\leftarrow \frac{1}{|\boldsymbol{S}|} \sum_i \min_{j,j \neq i} d(S_i, S_j)$
if $\min_j d(Z_t, S_j) >$ *threshold* **then**
  $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{Z_t\}$
end
return $\boldsymbol{S}$

Algorithm 7: OnlineSummaryUpdate $(\boldsymbol{S}, Z_t)$. Updates the summary $\boldsymbol{S}$ by picking the incoming observation Z_t if its score is above the mean score of observations already in the summary.

4.4.2 Analysis of Picking Rate

To simplify our exposition and analysis, we assume that all observations lie on a high dimensional grid, and two consecutive observations only differ by one step on this grid, which corresponds to a distance of 1 unit.

Let the threshold after k picks be γ_k. Then, the score of the observation which leads to the next pick must be $\gamma_k + 1$. Using this, we can define the threshold after k picks recursively as:

$$\gamma_k = \frac{(k-1) * \gamma_{k-1} + \gamma_{k-1} + 1}{k} \tag{4.13}$$

$$= \gamma_{k-1} + \frac{1}{k} \tag{4.14}$$

$$\approx \log k. \tag{4.15}$$

Hence, the threshold grows as $\Theta(\log k)$ with the number of picks, and as a result the picking rate should slow down with time.

Given a threshold γ_k, it will take us at least γ_k time to pick a new sample, since in each time step the observation changes only by 1 unit. Hence, the expected time for the next pick is greater than γ_k.

Let T_k be the expected time for k picks. We then know that

$$\begin{aligned} T_k &\geq T_{k-1} + \gamma_k && (4.16) \\ &= \sum_{i=1}^{k} \gamma_i && (4.17) \\ &\approx \sum_{i=1}^{k} \log k = \log k!. && (4.18) \end{aligned}$$

Using Sterling's approximation, we get,

$$T_k \geq k \log k - k \tag{4.19}$$

If we solve the above equation for equality, we can get an upper bound on the number of picks in a given time:

$$k(T) \leq \frac{T}{\text{ProductLog}\left(\frac{T}{e}\right)}. \tag{4.20}$$

This upper bound is useful in determining the memory requirements of a system that would guarantee a given duration of exploration by a robot. Figure 4.17 shows a plot of the upper bound along with plots of number of picks over time for the street view dataset described in Section 4.4.4, with 10 random starting times.

4.4.3 Trimming Strategies

Given infinite time, Algorithm 7 will give us a summary of infinite size. To make the summary size tractable, we can trim the summary when its size exceeds a desired size by removing a sample. We would like to do this in a way which ensure that the cost of the summary stays low.

Two simple strategies for identifying the sample to be discarded are the following:

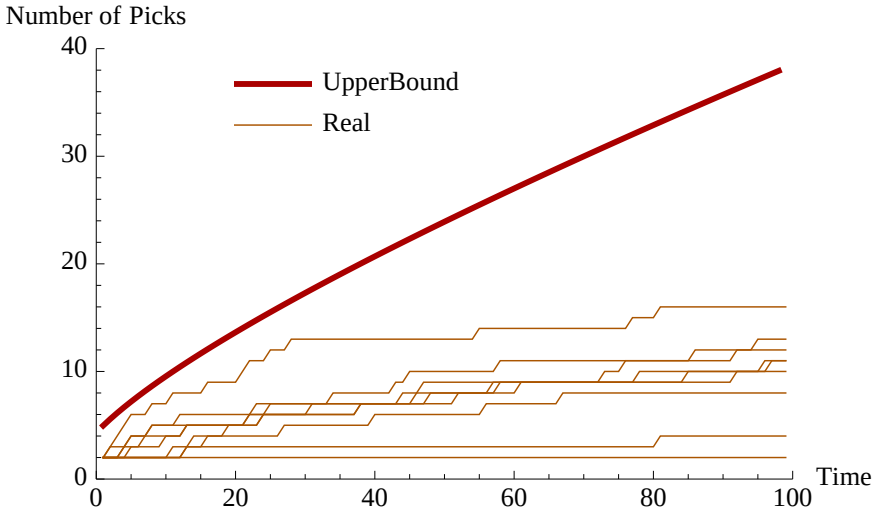
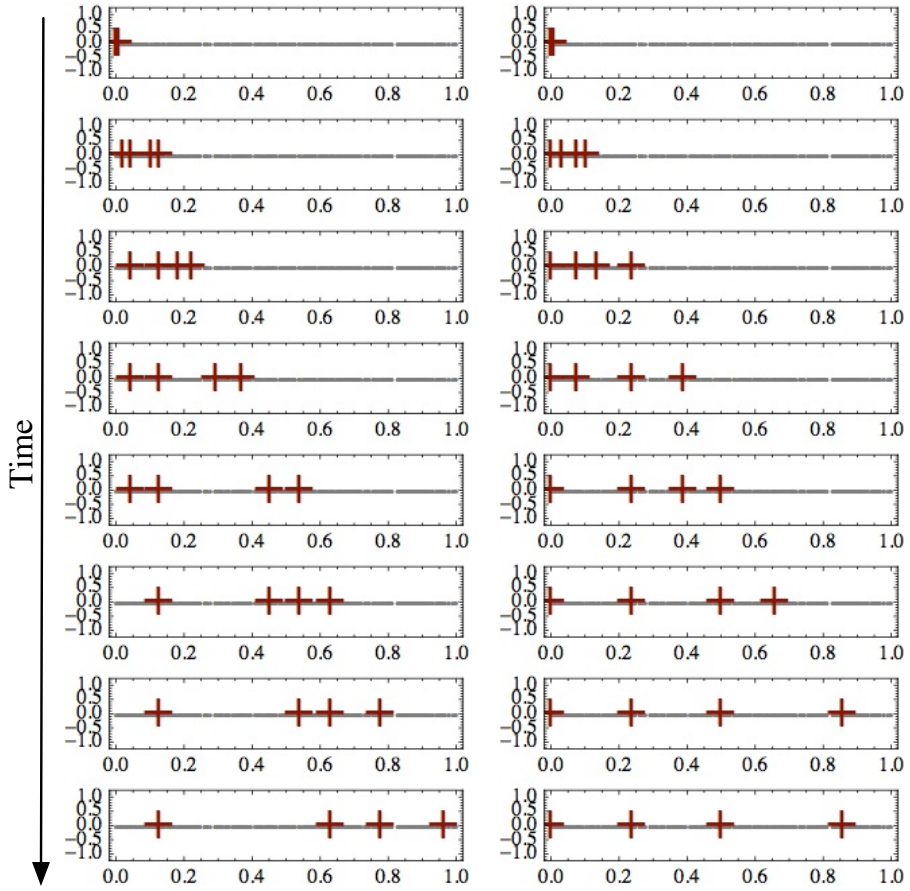


Figure 4.17: Upper Bound on Picking Rate. The plot shows number of picks by Algorithm 7 over time for 10 random starting locations in the street view dataset. The upper bound defined in Equation 4.20 is shown in thick red.

$threshold \leftarrow \frac{1}{|\boldsymbol{S}|} \sum_i \min_{j,j \neq i} d(S_i, S_j)$
if $\min_j d(Z_t, S_j) > threshold$ **then**
  $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{Z_t\}$
end
if $|\boldsymbol{S}| > k$ **then**
  TrimSummary($\boldsymbol{S}$)
end
return $\boldsymbol{S}$

Algorithm 8: k-OnlineSummaryUpdate $(\boldsymbol{S}, Z_t, k)$. Updates the summary $\boldsymbol{S}$ of size k by picking the incoming observation Z_t if its score is above the mean score of observations already in the summary. If the summary size exceeds k, then we trim the summary.



(a) Strategy 1 (b) Strategy 2

Figure 4.18: Discarding Strategies. There are 400 points in the dataset, distributed randomly over (0,1), and presented sequentially to the algorithm in monotonically increasing order of x axis values. The red '+' sign corresponds to the location of samples in the current summary. The successive state of the summary as time progresses is shown by successive rows. With strategy 1, we see that after a few time steps, the summary splits into two groups, and stays split for rest of the duration. This is because the minimum score sample to be discarded is always the 1st sample in the right group. Strategy 2 does not have this problem, and clearly has lower cost as defined by Equation 4.7.

Strategy 1: We can discard sample S_i if it has the lowest score defined as:

$$\text{Score}(S_i) = \min_{j, j \neq i} d(S_i, S_j). \tag{4.21}$$

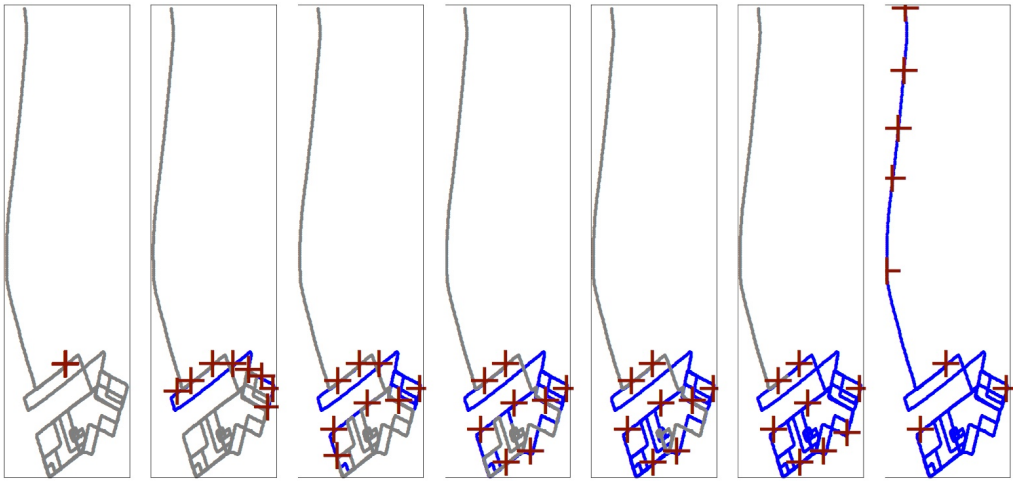
This identifies sample with the smallest nearest neighbour distance. Although this strategy seems reasonable, it fails in the simplest of cases as demonstrated in Figure 4.18(a). There are 400 points in the dataset, distributed randomly over (0,1), and presented sequentially to the algorithm in monotonically increasing order of x axis values. The red '+' sign corresponds to the location of samples in the current summary. The successive state of the summary as time progresses is shown by successive rows. We see that after a few time steps, the summary splits into two groups, and stays split for rest of the duration. This is because the minimum score sample to be discarded is always the 1st sample in the right group. Hence, this is not a good strategy.

Strategy 2: Summary trimming can also be modeled as summarization problem. The current summary is of size $k + 1$, from which we would like to select k representative samples. We can hence run the extremum summary algorithm (shown in Algorithm 4), with $\boldsymbol{S}$ initialized using the last selected sample. This technique is immune to problem faced by strategy 1 described above. Figure 4.18(b) shows progression of the summary on the same dataset, using strategy 2. Using this strategy clearly produces lower cost summaries as defined by Equation 4.7, and hence is the recommended strategy. The computational complexity of a summary update using this strategy is $O(k^2)$.

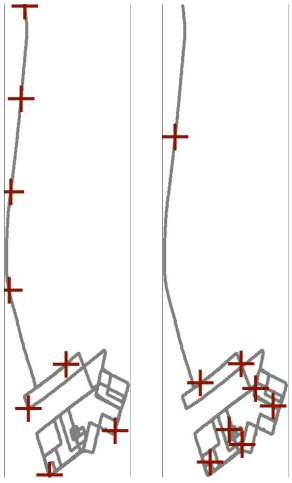
4.4.4 Evaluating Online Extremum Summaries

Location Based Summaries

Consider the task of collecting k samples such that no visited location is very far from one of the locations from which a sample was taken. We would like to do this even when the path taken by the robot is not known in advance. These samples for example could be rocks collected by a planetary rover as it is exploring a cave like environment, and we would like to have a good representation of the kind of rocks present in that area. Also due to energy, physical or logistics constraints, we might not able to come back to a previously visited location.

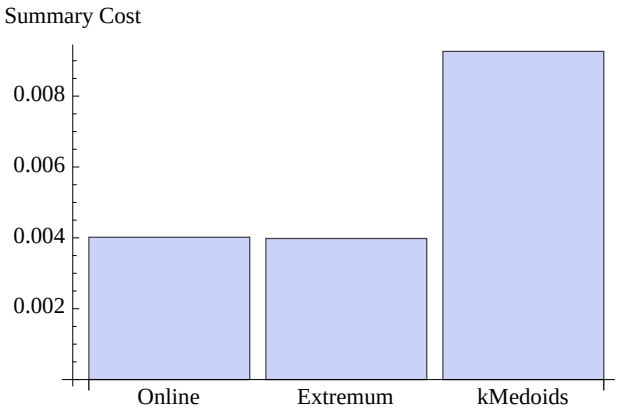


(a) Online



(b) Extremum

(c) k-Medoids



(d) Summary Cost

Figure 4.19: Location Based Summaries. The street view dataset contains 1255 geo-tagged images taken by a car as it goes around a city centre. The points in the dataset are shown using grey dots, and the points in the summary are shown using a red '+' sign. The summaries shown are generated using only the GPS data. (b) shows the extremum summary of these points. We see that the points are well distributed, and every point in the dataset is close to a summary point, whereas in the summary generated by k-medoids algorithm shown in (c), the points in the long tail of the dataset are much farther from the closest summary point. Figure (a) shows the evolution of the online summary generated using Algorithm 8. Points in blue correspond to portion of the path covered at that time. We start with all 8 summary points as the first 8 observed points. This is represented by the first column of (a). In the subsequent columns we see the evolution of the summary points as time progresses. Column 7 shows the final summary, which is similar to the extremum summary. (d) shows the summary cost of the three summaries, computed using Equation 4.7. Online summary performs similar to the offline summary, and both are much better than the k-medoids summary.

In such a scenario one can use Algorithm 8 to maintain a summary computed using location information. Each pick will corresponds to collecting the rock sample at that location, and removing a sample from the summary will correspond to dropping the rock. Figure 4.19 shows an example of such a summary. The subsampled dataset contains 1255 geo-tagged images taken by a car as it goes around a city centre. The points in the dataset are shown using grey dots, and the points in the summary are shown using a red '+' sign. Figure 4.19(b) shows the extremum summary of these points. We see that the points are well distributed, and every point in the dataset is close to at least one of the summary point, whereas in the summary generated by the k-medoids algorithm shown in Figure 4.19(c), the points in the long tail of the dataset are very far from the closest summary point.

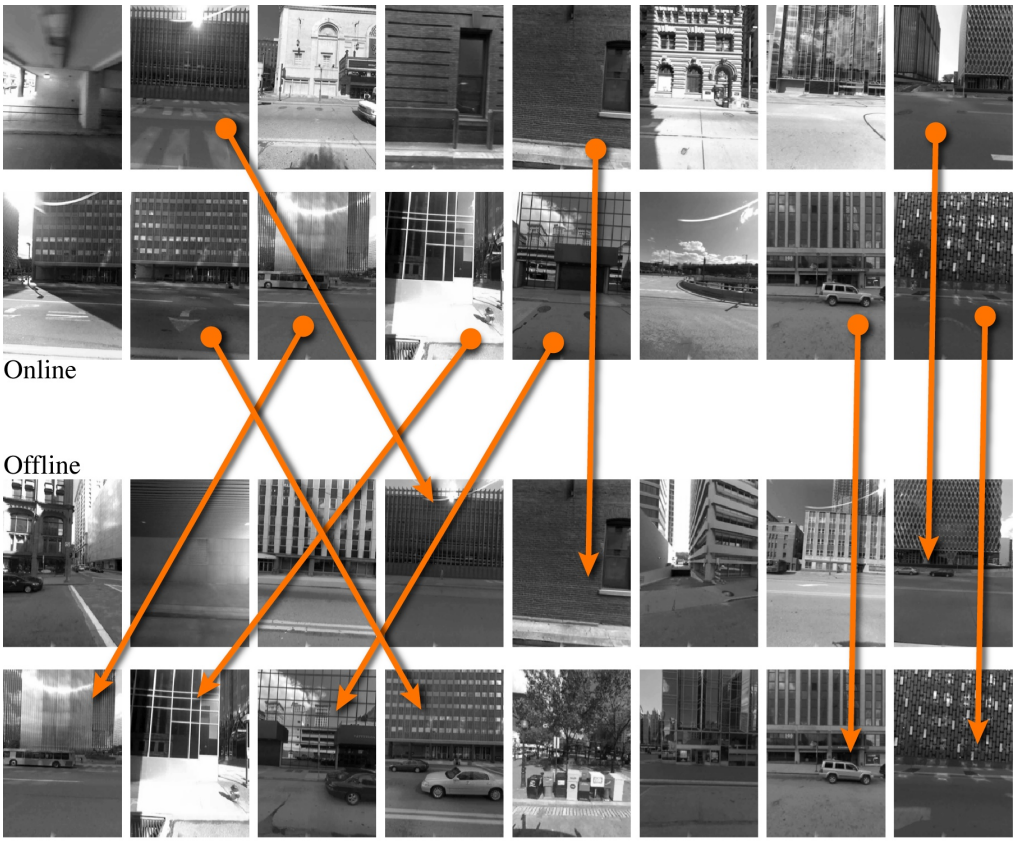
Figure 4.19(a) shows the evolution of the online summary generated using Algorithm 8. Points in blue correspond to portion of the path already traversed at that time. We start with all 8 summary points as the first 8 observed points. This is represented by the first column of Figure 4.19(a). Column 7 shows the final summary, which is similar to the extremum summary.

Figure 4.19(d) shows quantitatively the difference in performance of the different algorithms. The summary cost of the three summaries, was computed using Equation 4.7. We see that the online summary performs similar to the offline summary, and both are much better than the k-medoids summary. Both the online and the offline extremum summary have a cost of about 0.004, whereas the k-medoids summary has a much higher cost of about 0.009.

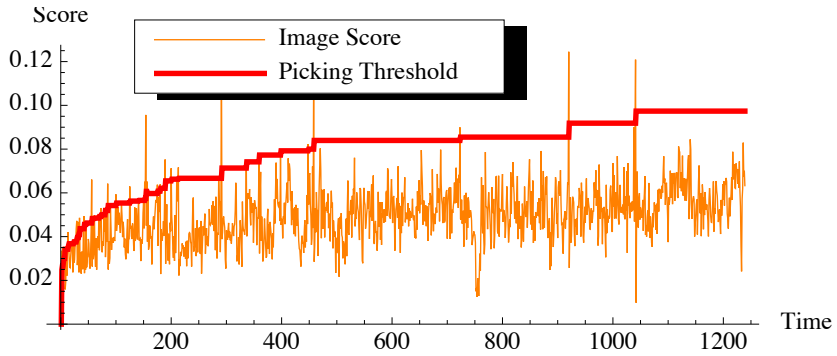
Visual Summaries

Figure 4.20 shows the summaries generated for the same street view dataset described in the previous section, however, instead of summarizing over GPS readings, we generate summaries using the image data. Each image was described using a bag of words histogram. We used a static vocabulary of size 1000, computed by clustering SURF[6] features extracted from all the images.

The bottom summary in Figure 4.20(a) shows the images selected by the extremum summary algorithm. Many of the images selected by the extremum summary algorithm correspond to images of building with different repetitive patterns, occupying large portions of the camera's field of view. Such images would be represented by histograms with a sharp



(a) Summaries



(b) Online Score

Figure 4.20: Summarizing Images in the Street View Dataset. The dataset contains 1255 geo-tagged images taken by a car as it goes around a city centre. The bottom summary in (a) shows images selected by the offline extremum summary algorithm, and the top summary in (a) shows images selected by the online summarization algorithm described in Algorithm 8. Large number of matches indicated by the arrows suggest the effectiveness of the online algorithm in picking the same outliers as the offline algorithm, even in high dimensional spaces.

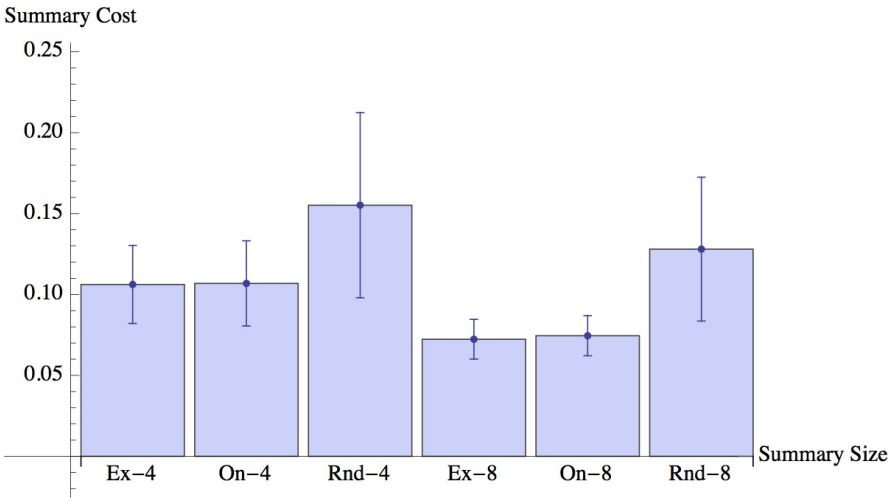


Figure 4.21: Mean Summary Cost of Street View Dataset. We took 20 random subsequences of length 100 from the street view dataset, and then computed the summary cost as defined in Equation 4.7. Mean cost of both the offline (Ex) and the online (On) summaries is shown for summaries of size 4 and 8. Error bars corresponds to 1 stddev. For comparison, we also show performance of a random algorithm (Rnd), which just chooses the summary samples randomly. We see that performance of the online algorithm is similar to the offline algorithm, and both perform much better than the random algorithm. Increasing the summary size decreases the cost, which is expected.

peaks at different locations when using the bag-of-words representation of an image. As a result, the KL divergence between these images is very high, and hence they are favored by the extremum summary algorithm.

The top summary in Figure 4.20(a) shows images selected by the online summarization algorithm described in Algorithm 8. We see that 9 out of 16 images in the online summary also exists in the offline summary. This indicates that the online algorithm is able to identify many of the same summary samples as the offline algorithm, even in the extremely high dimensional spaces corresponding to the bag of words histogram descriptions.

To quantify the measure of performance of the online summaries, we took 20 random subsequences of length 100 from the street view dataset, and then computed the summary cost as defined in Equation 4.7 for each of them. Mean cost of both; the offline, and the online summaries for these sequences is shown in Figure 4.21. For comparison, we also show performance of a random algorithm, which randomly chooses its summary samples. We see that for a summary size of 4, performance of the online algorithm is similar to the offline algorithm, and both have an average cost of about 0.106, whereas the random algorithm has

a much higher mean cost of about 0.155. For a summary size of 8, we see that the costs are lower for all the algorithms, which is expected since more summary points imply less distance between a sample and the closest summary point. Mean cost of the offline extremum summary for k=8 is about 0.068, and the online algorithm is slightly higher at about 0.071, and random is almost twice at about 0.137.

One of the fundamental difference between the offline and online algorithm is also the memory and computation cost. The offline algorithm requires the entire dataset to be in the memory, and hence as new data is added, it becomes slower linearly with time. The online algorithm does not have this problem, and has a constant memory footprint, proportional to the size of the desired summary, and a constant computation cost as new samples arrive.

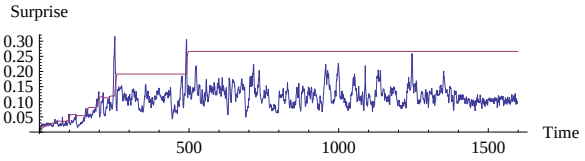
Terrain Summaries

We flew our aerial vehicle over the region shown in Figure 4.22(b). The downward looking video from a UAV was used to generate the summary. Figure 4.22(c) shows the evolution of the summary set over time. We start off with just the first observed image in the summary set, as indicated by the first row of Figure 4.22(c). For each new observation, we compute its surprise score given the images already in the summary set. If the surprise is above the threshold (mean surprise score), which is initially set to zero, we then include the image in the summary set. Each successive row of Figure 4.22(c) shows the state of the summary set after 3 modifications. The final row is the summary after observing the last image.

Figure 4.22(a) show the surprise of the incoming observations and acceptance threshold overtime. We see that initially, since the threshold is low, we rapidly pick several images and in the process the threshold also grows rapidly. This is also clear from Figure 4.22(c), where we see the initial rows are filled with similar looking images, which is result of a low threshold.

4.5 Summary

A navigation summary is a synopsis of observations made by a robot on a trajectory. In this chapter we explored batch and online solutions for producing navigation summaries that minimize the maximum surprise in observing the dataset. These summaries, which we refer to



(a) Surprise and selection threshold over time.

(b) Aerial photo of the entire region.

(c) Summary Evolution

Figure 4.22: Succession of 6-frame navigation summaries computed by the system at successive points in time during the flight of an aerial robot. Each row depicts an intermediate navigation summary state based on observations thus far. As time progresses we see that the frames describe increasingly varied types of images including frames that have land covering the North half of the image, or land covering only the South half of the image.

as extremum summaries, encourage picking outliers that often represent interesting samples in the dataset.

We pose the problem of generating navigation summaries as a sampling problem and considered three different strategies: k-medoids, k-cover, and kCenters. We demonstrate the difference between these strategies by experimenting with different datasets, and find that the k-cover summary with the right coverage ratio parameter, performs at least as good or better than the k-medoids or the extremum summary algorithm, in several different scenarios. The performance of a summary was measured in terms of its max surprise score measured over all the observations.

In general, we find that the kCenters based extremum summary algorithm focuses on picking outliers. These outliers could be genuinely interesting samples like in the case of street view dataset, or they could correspond to undesirable noisy samples like the ones present in the aerial view dataset.

On the other hand, the k-medoids summary algorithm focuses on picking samples representing the mean properties of the environment. These summaries could be useful if the dataset is simple and lacks and interesting outliers.

Using the k-cover algorithm, we can produce summaries which finds a balance between representing outliers and the mean properties of the environment. The desired coverage ration parameter γ allows us to tune out noise in the case that the dataset is full of undesirable outliers.

Extremum summaries are useful for providing mission updates, however, for this to happen efficiently, we must have the ability to update our summary online. Our proposed online algorithm approximates the solution of the batch k-centers problem. We showed that the proposed online algorithm matches the performance of the batch solution, when dealing with real world data consisting of images and location data.

Chapter 5

Hiring Secretaries for Sensor Dropping and Sample Collection

5.1 Introduction

Imagine a planetary rover charged with the task of picking locations for further investigation, either through drilling or by collecting rock samples to be returned to earth. There are only a finite number of locations that can be inspected, and due to lack of a positioning system or due to a dynamic environment, it is impossible to backtrack to a previously skipped location. Now upon arriving to a location, the robot makes an observation and must instantaneously and irrevocably decide whether to select this location or not. This problem can be considered as an instance of the classical secretaries hiring problem, where the secretaries correspond to observations, and the act of hiring a secretary corresponds to irrevocably picking the corresponding location for further investigation. We assume that we are given a scoring function that allows us to compare different observations based on their utility.

In Section 5.2 we will review the classical secretaries hiring problem. In Section 5.3 we will present a novel extension to the classic secretaries hiring problem where the goal is to optimize the probability of picking all top k secretaries. In Section 5.5 we adopt a previously known extension of the secretaries problem for use in the sample collection problem. In this variant of the problem, the goal is optimize the sum of score of the selected candidates. In

Section 5.6 we show an example application of picking image samples with high information content.

5.2 Secretaries Problem

The secretary problem has a long and varied history. Due to its broad relevance to different domains, it has been considered by different authors in several different contexts. Dynkin [28] appears to be the first one to solve the problem formally. For an interesting discussion on the origins of this problem see [30]. Here is a description of the problem in its simplest form.

You are given the task of hiring the best possible candidate for the job of a secretary. There are n applicants who have signed up for the interview. After interviewing each candidate you can rank them relative to all other candidates seen so far. You must either hire or reject the candidate immediately after the interview. You are not allowed to go back and hire a previously rejected candidate.

A typical strategy would be to just observe the first $r = \theta n$ candidates, $0 < \theta < 1$, without accepting any, then find the highest score among them, and then hire the first candidate with score higher than that. This is provably the optimal strategy for this problem. The problem now is to select the optimal value for θ.

Let success be defined iff we pick the highest scoring candidate. We then have:

$$\mathbf{P}\{\text{Success}\} = \sum_{i=r+1}^{n} P(S_i), \tag{5.1}$$

where S_i is the event that the ith candidate is the highest scoring candidate, and that our algorithm did not select any of the previous candidates. Hence we have:

$$\begin{aligned} \mathbf{P}\{\text{Success}\} &= \sum_{i=r+1}^{n} \frac{1}{n} \cdot \frac{r}{i-1} && (5.2)\\ &\approx \frac{r}{n} \int_{r}^{n} \frac{1}{i} di && (5.3)\\ &= \frac{r}{n}(\log n - \log r). && (5.4) \end{aligned}$$

Now to find optimal θ, we set the derivative of $\mathbf{P}\{\text{Success}\}$ to 0:

$$\begin{aligned} \frac{d}{dr}\mathbf{P}\{\text{Success}\} &= \frac{\log n - \log r}{n} - \frac{1}{n} = 0 && (5.5)\\ \implies r &= \frac{n}{e} && (5.6)\\ \implies \theta &= \frac{1}{e}. && (5.7) \end{aligned}$$

5.3 Hiring Top k Secretaries using Single Threshold.

Consider the generalization of the secretaries hiring problem where we would like to maximize the probability of hiring the top k candidates. Similar to the single secretary hiring problem, the general strategy is the following. Let $X_1, \cdots, X_n$ be the scores of the n candidates. Let

$$M = \max(X_1, \cdots, X_{\theta n}) \tag{5.8}$$

be the threshold score. We keep the first k candidates $\{X_i\}$ (if there are at least k), with $i > \theta n$ such that

$$X_i > M. \tag{5.9}$$

Observation 5.3.1. *The algorithm is invariant under monotone transformation, so we might as well assume that $X_1, \cdots, X_n$ are drawn independently from the uniform $[0,1]$ distribution.*
Fact 5.3.2. *If N is total number of candidates which exceed the threshold score M*

$$N = \sum_{\theta n < i \le n} \mathbf{1}_{[X_i > M]}, \tag{5.10}$$

then the probability of succeeding

$$\mathbf{P}\{\text{Success}\} = \sum_{j=k}^{n-\theta n} \mathbf{P}\{N = j\} \frac{1}{\binom{j}{k}}. \tag{5.11}$$

Proof. Given that j of the $X_{\theta n+1}, \cdots, X_n$ fall in $(M, 1]$, we only have probability $1/\binom{j}{k}$ that the k largest are the first k sampled. □

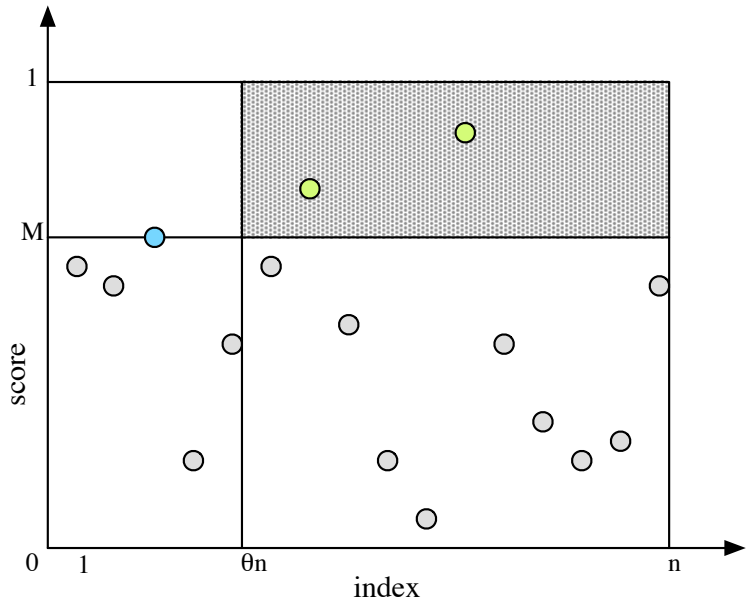


Figure 5.1: Scores of the candidates can be considered as points on a plane. The highest scoring candidate in the interval $1, \cdots, \theta n$ defines the threshold M.

Assuming $n \to \infty$, we the have

$$\lim_{n\to\infty} \mathbf{P}\{\text{Success}\} = \lim_{n\to\infty} \left(\sum_{j=k}^{n-\theta n} \mathbf{P}\{N=j\} \frac{1}{\binom{j}{k}} \right) \tag{5.12}$$

$$= \lim_{n\to\infty} \left(\sum_{j=k}^{\infty} \mathbf{P}\{N=j\} \frac{1}{\binom{j}{k}} \right). \tag{5.13}$$

This depends only upon k, and θ. Our goal then would be to maximize (5.13) with respect to θ.

Fact 5.3.3.

$$\lim_{n\to\infty} \mathbf{P}\{N=j\} = \theta(1-\theta)^j \tag{5.14}$$

Proof. Candidate scores can be considered as points in the plane: $(1, X_1), \cdots, (n, X_n)$, as shown in Figure 5.1. Now, N is the number of points in the green shaded rectangle, defined by the diagonal vertices $(\theta n, M)$ and $(1, 1)$ [In the example shown, two points fall in this region].

Assume now that we draw points from top down. Then the first j points drawn will be from the interval $(\theta n, n]$, and the next [$(j+1)$st] point will be from $[1, \theta n]$. It is like drawing from an urn.

$$\begin{aligned}
\mathbf{P}\{N = j\} &= \frac{n-\theta n}{n} \times \frac{n-\theta n-1}{n-1} \times \cdots \times \frac{n-\theta n-(j-1)}{n-(j-1)} \times \frac{\theta n}{n-j} && (5.15)\\
&= \frac{\theta n}{n-j} \prod_{i=0}^{j-1} \frac{n-\theta n-i}{n-i} && (5.16)\\
&= \frac{\theta n}{n-j} \prod_{i=0}^{j-1} \left(1 - \frac{\theta n}{n-i}\right) && (5.17)
\end{aligned}$$

Taking the limit $n \to \infty$ we get

$$\lim_{n\to\infty} \mathbf{P}\{N = j\} = \theta(1-\theta)^j. \tag{5.18}$$

□

Observation 5.3.4. *Setting $i =$*

$$\mathbf{P}\{N = j\} \leq \theta(1-\theta)^j \frac{n}{n-j} \tag{5.19}$$

and

$$\begin{aligned}
\mathbf{P}\{N = j\} &\geq \theta(1-\theta)^j \left(1 - \frac{j-1}{(1-\theta)n}\right)^j && (5.20)\\
&\geq \theta(1-\theta)^j \left(1 - \frac{j(j-1)}{(1-\theta)n}\right) && (5.21)
\end{aligned}$$

Fact 5.3.5.

$$\begin{aligned}
\lim_{n\to\infty} \mathbf{P}\{\text{Success}\} &= \sum_{j=k}^{\infty} \theta(1-\theta)^j \frac{1}{\binom{j}{k}} && (5.22)\\
&= \theta(1-\theta)^k \; {}_2F_1(1, 1; 1+k; 1-\theta), && (5.23)
\end{aligned}$$

were ${}_2F_1(a,b;c;z)$ is the Gauss Hypergeometric function, defined as

$$
\begin{aligned}
{}_2F_1(a,b;c;z) &= \sum_{i=0}^{\infty} \frac{(a)_i (b)_i}{(c)_i i!} z^i, && (5.24)\\
(a)_i &= a(a+1)(a+2)\cdots(a+i-1), (a)_0 = 1. && (5.25)
\end{aligned}
$$

Proof. We will prove this by showing that the upper bound is same as the lower bound.

1. *Lower Bound*:

$$
\begin{aligned}
\lim_{n\to\infty} \mathbf{P}\{\text{Success}\} &\geq \liminf_{n\to\infty} \mathbf{P}\{\text{Success}\} && (5.26)\\
&= \liminf_{n\to\infty} \sum_{j=k}^{\infty} \mathbf{P}\{N=j\} \frac{1}{\binom{j}{k}} && (5.27)
\end{aligned}
$$

   By Fatou's Lemma we have,

$$
\liminf_{n\to\infty} \sum_{j=k}^{\infty} \mathbf{P}\{N=j\} \frac{1}{\binom{j}{k}} \geq \sum_{j=k}^{\infty} \left(\liminf_{n\to\infty} \mathbf{P}\{N=j\} \right) \frac{1}{\binom{j}{k}}. \quad (5.28)
$$

   Since $\mathbf{P}\{N=j\}$ converges to a real value as $n \to \infty$, this implies

$$
\liminf_{n\to\infty} \mathbf{P}\{N=j\} = \lim_{n\to\infty} \mathbf{P}\{N=j\}. \quad (5.29)
$$

   Hence, we have

$$
\begin{aligned}
\lim_{n\to\infty} \mathbf{P}\{\text{Success}\} &\geq \sum_{j=k}^{\infty} \left(\lim_{n\to\infty} \mathbf{P}\{N=j\} \right) \frac{1}{\binom{j}{k}} && (5.30)\\
&= \sum_{j=k}^{\infty} \theta(1-\theta)^j \frac{1}{\binom{j}{k}} && (5.31)
\end{aligned}
$$

2. *Upper Bound*: Let $\epsilon > 0$ be arbitrary. Then,

$$
\sum_{j=\epsilon n}^{\infty} \mathbf{P}\{N=j\} \frac{1}{\binom{j}{k}} \leq \frac{1}{\binom{\epsilon n}{k}} \sum_{j=\epsilon n}^{\infty} \mathbf{P}\{N=j\} \quad (5.32)
$$

because $\binom{j}{k} \geq \binom{\epsilon n}{k}$, for $j \geq \epsilon n \geq 2k$.

Also,

$$\sum_{j=\epsilon n}^{\infty} \mathbf{P}\{N = j\} = \mathbf{P}\{N \geq \epsilon n\} \tag{5.33}$$

$$\leq (1-\theta)^{\epsilon n} \leq 1 \tag{5.34}$$

and thus,

$$\lim_{n\to\infty} \sum_{j=\epsilon n}^{\infty} \mathbf{P}\{N = j\} \frac{1}{\binom{j}{k}} = 0. \tag{5.35}$$

Finally, by (5.19),

$$\sum_{j=k}^{\epsilon n} \mathbf{P}\{N = j\} \frac{1}{\binom{j}{k}} \leq \sum_{j=k}^{\epsilon n} \theta(1-\theta)^j \frac{n}{n-j} \frac{1}{\binom{j}{k}} \tag{5.36}$$

$$\leq \left(\sum_{j=k}^{\epsilon n} \theta(1-\theta)^j \frac{1}{\binom{j}{k}} \right) \frac{1}{1-\epsilon} \tag{5.37}$$

$$\leq \left(\sum_{j=k}^{\infty} \theta(1-\theta)^j \frac{1}{\binom{j}{k}} \right) \frac{1}{1-\epsilon}, \tag{5.38}$$

and so we are done, since ϵ was arbitrary.

□

5.4 Optimal Observation Interval

To compute the optimal observation interval θ for k secretaries, we must maximize (5.23). There is however no known closed form solution to this optimization problem. Hence, we compute optimal θ for individual values of k. We summarize our results in Table 5.1, for the first 20 value of k.

Given the numerical values of θ, we computed an expression which approximates θ in

k	$\mathbf{P}\{\text{Success}\}$	θ
1	0.367879	0.367879
2	0.203632	0.284668
3	0.136	0.228475
4	0.100829	0.189087
5	0.0797197	0.160512
6	0.0657783	0.139087
7	0.0559294	0.122537
8	0.048619	0.109418
9	0.0429853	0.0987887
10	0.0385143	0.0900134
11	0.0348815	0.0826527
12	0.0318723	0.076394
13	0.0293394	0.0710093
14	0.0271784	0.0663288
15	0.0253131	0.0622238
16	0.0236869	0.0585949
17	0.0222566	0.0553642
18	0.020989	0.0524698
19	0.0198578	0.049862
20	0.0188421	0.0475004

Table 5.1: Optimal values for the training interval θ, and probability of success for first twenty values of k.

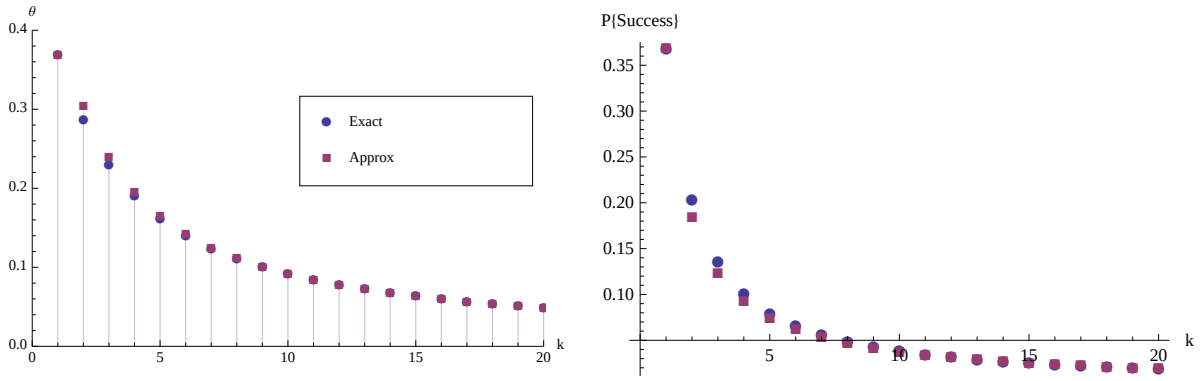


Figure 5.2: Optimal training interval θ and $\mathbf{P}\{\text{Success}\}$ as a function of number of secretaries k. The dots represent the actual training interval computed numerically, and the squares represent values computed using the closed form approximation.

closed form via curve fitting:

$$\theta \approx \frac{1}{ke^{1/k}} \tag{5.39}$$

Figure 5.2 shows a plot of this approximation. Note that for $k = 1$, we get $\theta = 1/e$, as was also proven in Section 5.2.

5.5 Other Extensions

There are many other extensions to the multiple secretaries hiring problem that have been explored in the literature. We discuss a few of them briefly.

Kleinberg [68] suggested an algorithm to maximize the expected sum of the scores of the candidates. The algorithm works by splitting the candidates into two roughly equal intervals, where the boundary is chosen randomly using a binomial distribution $B(n, 1/2)$. We then recursively apply the classic ($k = 1$) secretary algorithm to the first half of the candidates, choosing $l = \lfloor k/2 \rfloor$ candidates. While doing this we also find the lth highest scoring candidate from the first half and use this as a fixed threshold to select the remaining candidates in the second half.

Babaioff et al. [3, 2] suggest a simpler algorithm with the same goal of maximizing the expected sum of the scores of the selected candidates. A sliding threshold to choose the

candidates. Algorithm 9 describes this approach.

Find the top k scores in the first $r = \lfloor n/e \rfloor$ candidates, without selecting any. Call this list of thresholds, $T = \{t_1, .., t_k\}$.
foreach *remaining candidate* $(x_{r+1}, \cdots, x_n)$ **do**
  if *candidate has score higher than the minimum score in* T **then**
    Hire the candidate.
    Remove the minimum value from the set T.
  end
  if T *is empty* **then**
    break
  end
end

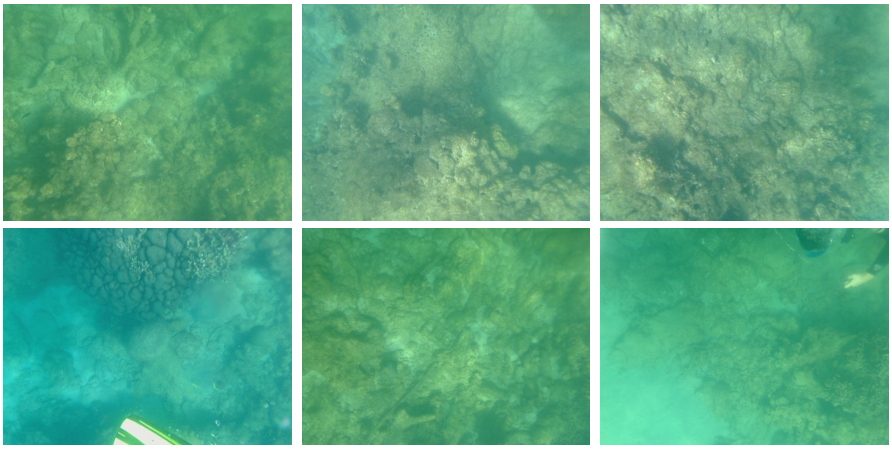
Algorithm 9: MaxExpectedSumScores($\{x_1, .., x_n\}$, k)

Some other variants of the secretaries hiring problem has been reviewed by Freeman [32].

5.6 Example Application: Picking Images with High Information Content

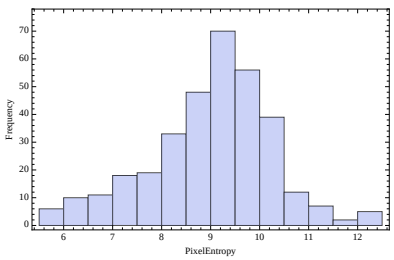
We are particularly interested in the problem of selecting sample images acquired from a robot, such that it can automatically identify the most informative locations in the world, for the purpose of placing static sensor nodes. To explore this idea further, we assume we have a robot (such as AQUA [91] discussed in Section 2.7.1), equipped with a camera and a set of k deployable static sensor nodes. As the robot moves around in the world, it takes visual measurements once every few seconds, and then gives each measurement a score. The goal then is to identify the highest scoring sites online, so that the sensors can be deployed.

For the purpose of this experiment, we use a very simple pixel entropy based measurement score. The score of a measurement image was defined as sum of pixel entropy for each of its RGB channels. This simple metric should in principal favor images with more colors than those that don't have as many colors. To do this, for each color channel, we computed an intensity histogram with 32 bins, and then computed its total entropy.

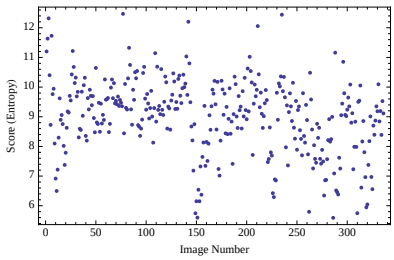


(a) Random

(b) Top 6



(c) Histogram of the image scores



(d) Scatter plot of image scores

Figure 5.3: Coral Reef Data-set consisting of 336 images. (a) Six randomly chosen images from the data set. (b) Top 6 highest scoring images in the coral reef data set, with scores of {11.72, 12.06, 12.20, 12.32, 12.44, 12.47 }. (c) Histogram of image scores (d) Scatter plot of the scores.

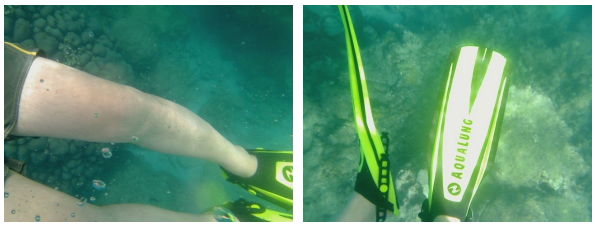


Figure 5.4: Highest scoring images chosen as a result of running the fixed threshold algorithm with $k = 6$. The corresponding scores are: {12.4692, 12.4398}. Since our scoring function is based on pixel entropy of the image, the algorithm skips over most of images of coral reef and instead picks more colorful *accidental* images of the swimmer.

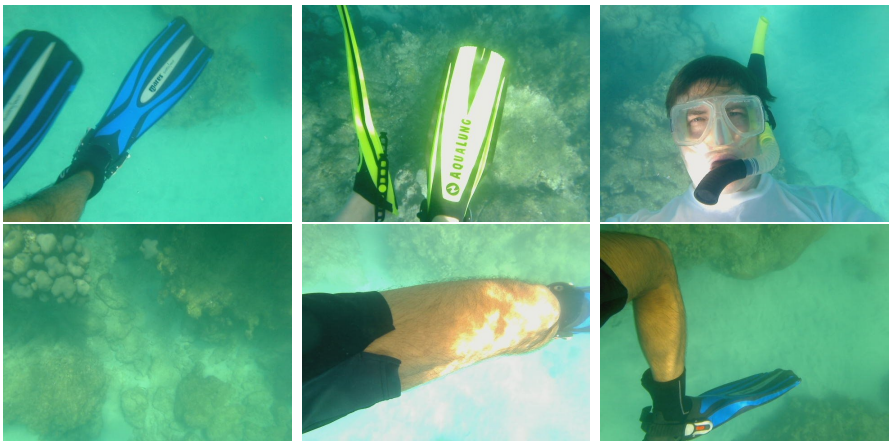


Figure 5.5: Highest scoring images chosen as a result of running the fixed threshold algorithm with $k = 6$, on reverse sequence of images in the coral reef data set. The corresponding scores are: {11.16, 12.44, 12.06, 11.02, 12.20, 11.03}.

Figure 5.3 shows a data set consisting of 336 images of a coral reef, collected by swimming on top of it with a camera pointing downwards. The images have a mean score of 9.00687, and variance of 1.57866. Figure 5.3(a) shows a few images selected randomly from the data, and (b) shows the top 6 highest scoring images. Figure 5.3(c,d) shows the scatter plot and histogram of their pixel entropy scores.

We ran our fixed threshold algorithm with $k = 6$, so that we get six or less images. Our algorithm only managed to return two images, however these two images were the two highest scoring images in our data set with score (pixel entropy) of 12.47 and 12.44 respectively. The optimal solution in this case is the six images with scores {11.72, 12.06, 12.20, 12.32, 12.44, 12.47} (shown in Figure 5.3(b)). Our algorithm picked only two of the images because the third-highest scoring image happens to fall quite early in the sequence and hence is chosen as

our threshold.

Inverting the sequence of images in our data set gives us a new data set without changing the temporal structure of the sequence. Hence, for the purpose of experimentation, we run our algorithm on the inverted sequence of images from the coral reef data set. This time we got six high scoring images (Figure 5.5), out of which three exist in the actual top six highest scoring images as seen in Figure 5.3(b).

5.7 Summary

In this chapter we have presented the solution to a new variant of the secretaries hiring problem which optimally chooses the k highest scoring samples in the data set, online without backtracking.

We discussed the relevance and utility of the secretary problem to robotic sampling. As a sample application, we applied our fixed threshold algorithm to the vacation snapshot problem, selecting images with high information content. Although we used a very simple image pixel entropy based scoring function, the picking algorithms presented in this chapter are valid for any scoring function where the score is independent of the previously selected samples.

The secretary hiring algorithm discussed in this chapter are only optimal when there is no prior information about the distribution of scores. For the problem of selecting sample images by a mobile robot, typically the adjacent samples will have some correlation. In practice these correlations are very hard to model or predict. For example in the presence of occlusion boundaries close to the camera, successive frames can be essentially uncorrelated. Our algorithm thus can be regraded as dealing with the worst case scenario. In addition when the video sequence is subsampled at a low frame rate, any existing correlation is likely to be drastically decreased.

For the purpose of selecting images that give the most information about the world, we would ideally like our samples to not only be highly informative, but also be independent with no mutual information between each other. We hope to deal with these issues in our future work.

Chapter 6

Curiosity Driven Exploration

6.1 Introduction

We define curiosity as the act of seeking information that improves our understanding of the world. Curiosity is the central driving force behind learning in humans and other animals. In the robotics context, ability to model curiosity in an autonomous robot can be helpful in many situations, making to robot suitable for a variety of previously undecided task such as: surprise detection, terrain and environment recognition. For example, if a robot has been exploring an environment driven by curiosity, it would likely be able to recognize a wider variety of objects, and it would be better at searching for a given object, even though it might not have specially been trained for it.

Curiosity can be viewed as unsupervised active modeling of the environment, with the goal of maximizing the amount of novel spatiotemporal phenomena that were observed. Information theoretically, this can be modeled as maximizing the entropy of the observations in the semantic space.

Our approach is the following. We assume the robot is equipped with a camera, and it continually streams images of the environment. We extract quantized visual features (words) from these images, and then use ROST (Chapter 3) to gives these observations semantic labels, allowing the robot to interpret what is being observed in a more abstract space. ROST enforces a Dirichlet prior over spatiotemporal regions in the video stream, encouraging

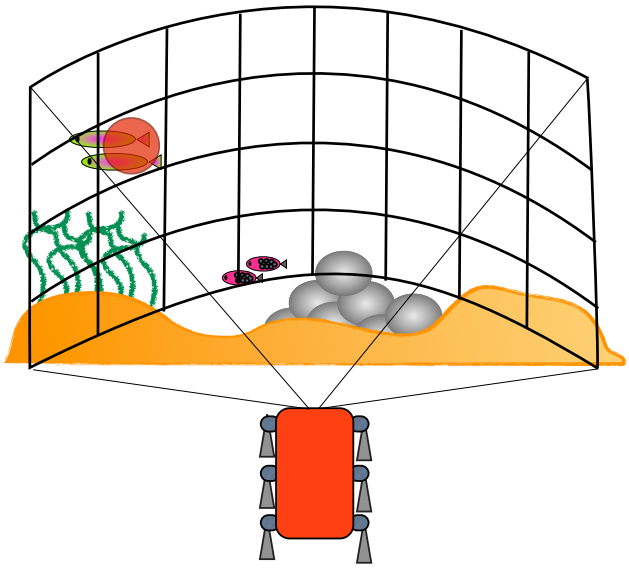
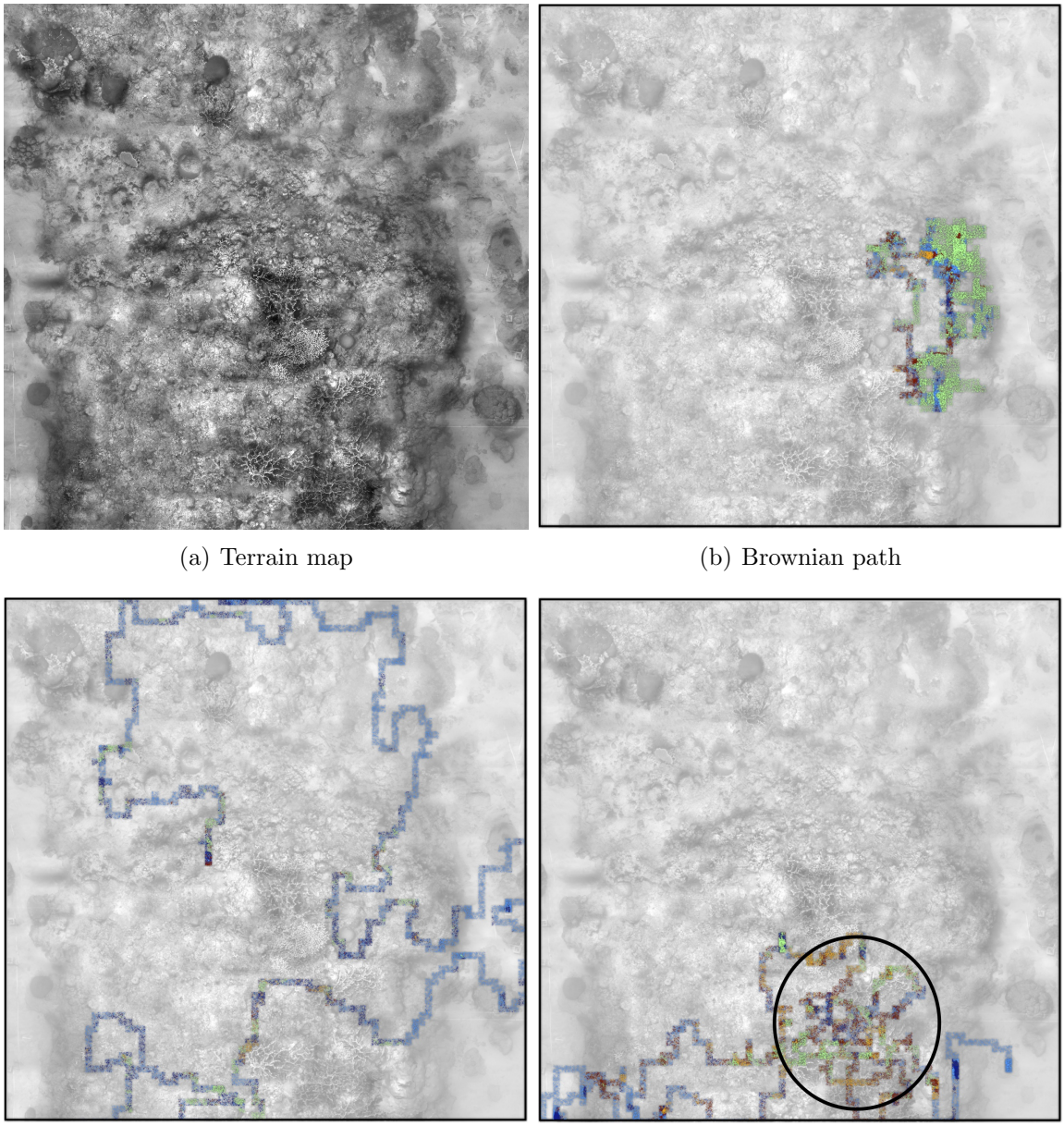


Figure 6.1: To model curiosity, the robot runs a realtime topic modeling framework, which allows it to perceive the world in semantic space, and then plans a path which tries to maximize the information gained. We approximate the optimal solution by greedily moving the robot in the direction which maximizes entropy.

the observations that frequently co-occur in space and time to cluster under the same topic label. A real-time Gibbs sampler keeps the topic model in a converged state even as we add new observations.

At each time step, we compute the perplexity of the observations from the neighboring locations. This perplexity score, along with a repulsive potential from previously visited locations, is then used to bias the probability of next step in the path. Since observations with high perplexity have high information gain, we claim that this approach would results in faster learning of the terrain topic model, which would imply shorter exploration paths for the same accuracy in predicting terrain labels for unseen regions.

Figure 6.2 shows an example of a typical path computed by the proposed exploration algorithms on a 2D map. We see that the proposed curiosity based algorithm favors locations with high information content, which in this case correspond to locations with corals.



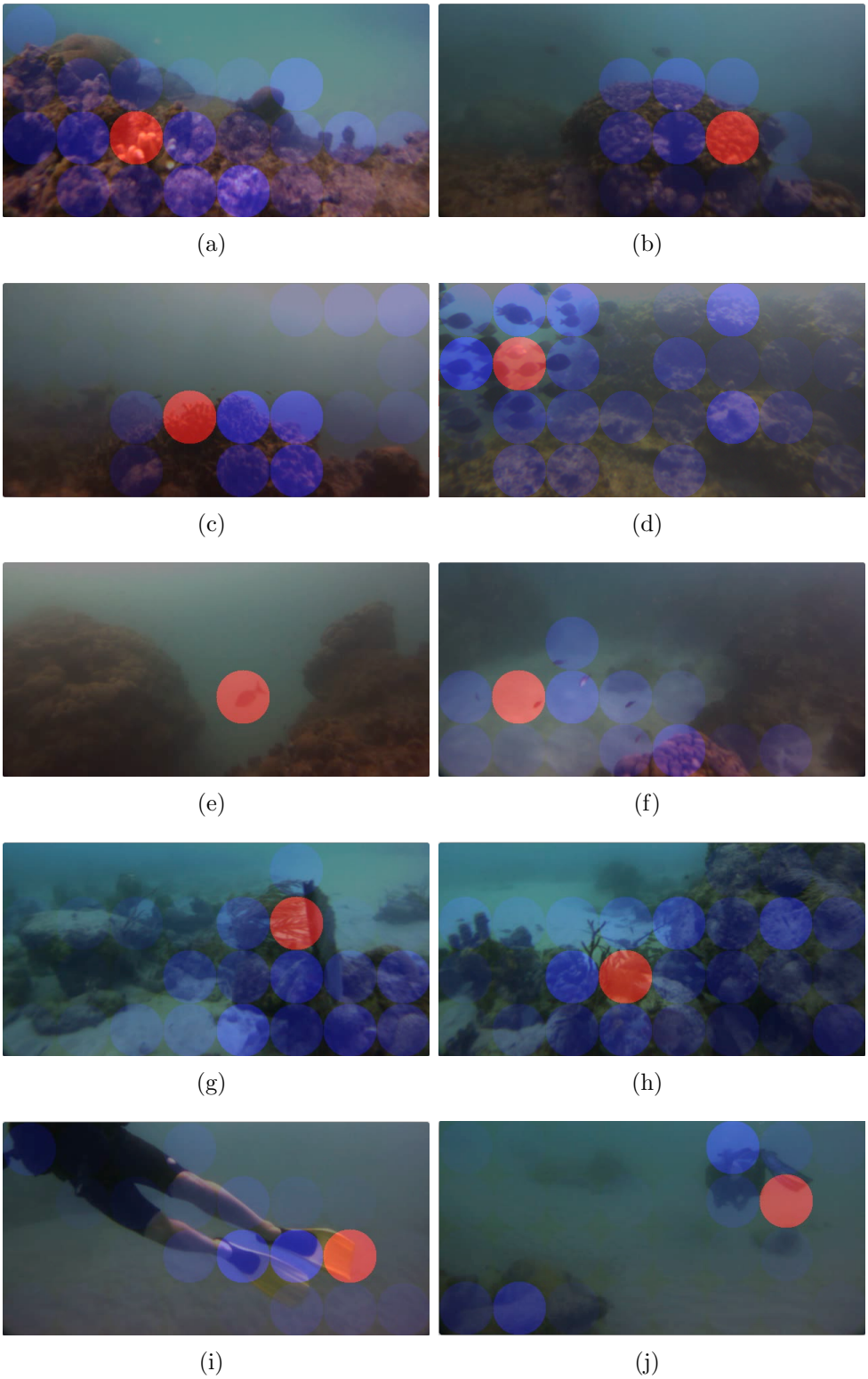
(a) Terrain map

(b) Brownian path

(c) Stochastic coverage paths

(d) Curiosity path (proposed)

Figure 6.2: Examples of different exploration paths of length 640 steps. Points on the paths are colored colored according to the topic labels of the words observed at that location. (a) Map used for the simulated exploration experiment. (b) A typical brownian motion exploration path with minimal coverage. (c) A typical stochastic coverage exploration path. We see that these paths are likely to encounter much more diversity of terrain. (d) A typical curiosity driven exploration path. We see that unlike stochastic coverage the proposed algorithm collects more data at locations where there is more information, such as the circled region containing corals.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 6.3: Examples of observations showing cells marked with their curiosity score. Red marks the cell with the highest score.

6.2 Curiosity based Exploration

We assume a cellular decomposition of the world, in which each cell $c \in C$ is connected to its neighboring cells $G(c) \subset C$. The world is composed of at most K different kinds of terrains or other high level visual objects (which we refer to as topics), each of which, when observed by a robot, can result in V different kinds of low level observations, where $V >> K$. Each topic k is described by a distribution ϕ_k over these V different types of observations, and for any cell c, $\phi_{G(c)}$ is the distribution of topics in and around the cell. The goal then is to plan a continuous path $P \subseteq C$, that allows us to learn the topic model $\Phi = \{\phi_k\}$ that best describes the world by labeling each observation at each location with a representative topic label.

At time t, let the robot be in cell $p_t = c$, and let $G(c) = \{g_i\}$ be the set of cells in its neighborhood. We would like to compute a weight value for each g_i, such that the probability of the robot taking a step in this direction is proportional to this weight.

$$\mathbf{P}(p_{t+1} = g_i) \propto \text{weight}(g_i). \tag{6.1}$$

In this work we consider four different weight functions, one that is completely unaware of of its surrounding, one that is only spatially aware and tries to cover the unexplored free space, and two that are both spatially and observationally aware.

1. Random Walk - Each cell in the neighborhood is equally likely to be the next step:

$$\text{weight}(g_i) = 1. \tag{6.2}$$

2. Stochastic Coverage - Use a potential function to repel previously visited locations:

$$\text{weight}(g_i) = \frac{1}{\sum_j n_j / d^2(g_i, c_j)}. \tag{6.3}$$

   where n_j is the number of times we have visited cell c_j, and $d(g_i, c_j)$ is the Euclidean distance between these two cells.

3. Word Perplexity - Bias the next step towards cells which have high word perplexity:

$$\text{weight}(g_i) = \frac{\text{WordPerplexity}(g_i)}{\sum_j n_j / d^2(g_i, c_j)}. \tag{6.4}$$

4. Topic Perplexity - Bias the next step towards cells which have high topic perplexity:

$$\text{weight}(g_i) = \frac{\text{TopicPerplexity}(g_i)}{\sum_j n_j / d^2(g_i, c_j)}. \tag{6.5}$$

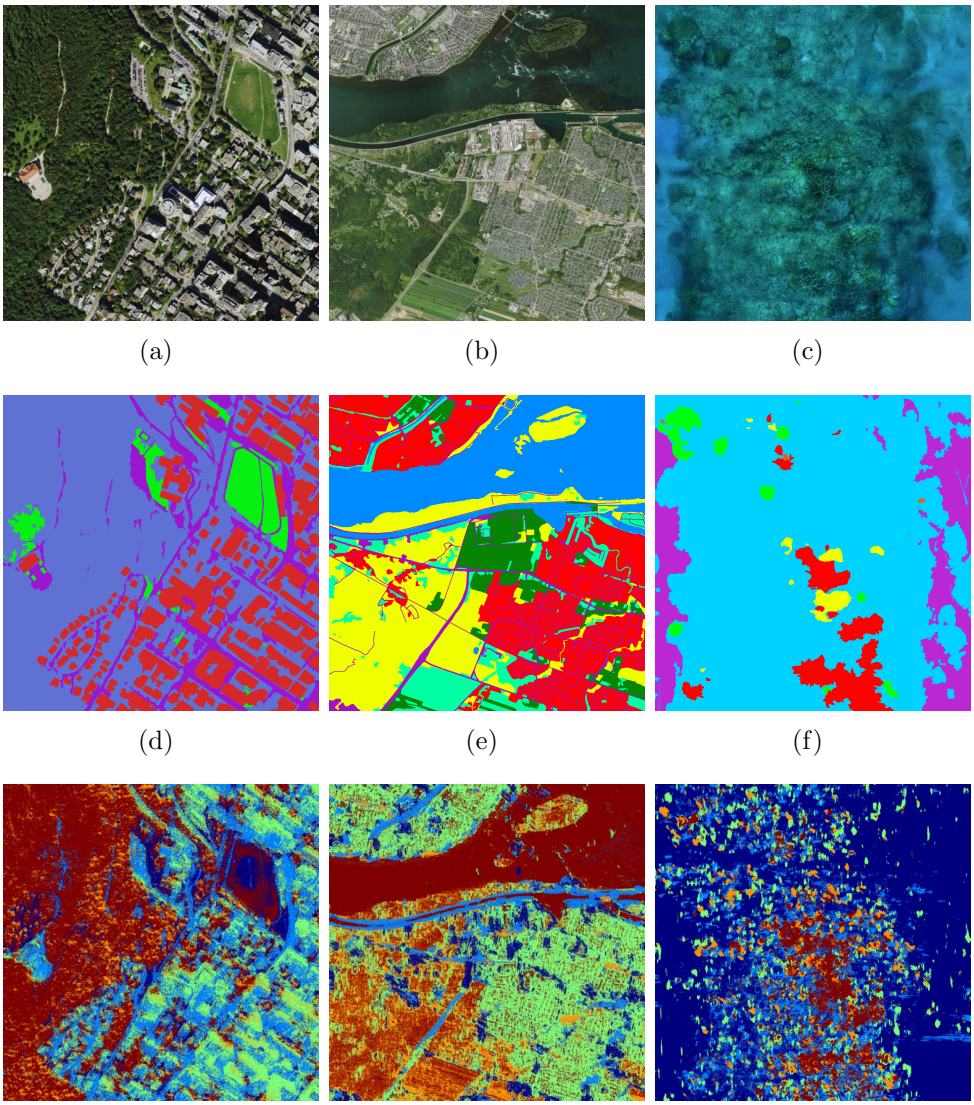
We compute the word perplexity of the words observed in g_i by taking the inverse geometric mean of the probability of observing the words in the cell, given the current topic model and the topic distribution of the path thus far.

$$\text{WordPerplexity}(g_i) = \exp\left(-\frac{\sum_i^W \log \sum_k \mathbf{P}(w_i = v|k)\mathbf{P}(k|P)}{W}\right), \tag{6.6}$$

where W is the number of words observed in g_i, $\mathbf{P}(w_i = v|k)$ is the probability of observing word v if its topic label is k, and $\mathbf{P}(k|P)$ is the probability of seeing topic label k in the path executed by the robot thus far.

To compute topic perplexity of the words observed in g_i, we first compute topic labels z_i for these observed words by sampling them from the distribution in Eq. 3.4, without adding these words to the topic model. These temporary topic labels are then used to compute the perplexity of g_i in topic space.

$$\text{TopicPerplexity}(g_i) = \exp\left(-\frac{\sum_i^W \log \mathbf{P}(z_i = k|P)}{W}\right). \tag{6.7}$$



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.4: Example of results of curiosity based exploration on a 2D dataset. (a)-(c) Input image used to generate observation data, (d)-(f) Groundtruth labeling. (g)-(i) Terrain labeling of the map using the topic model computed on the path.

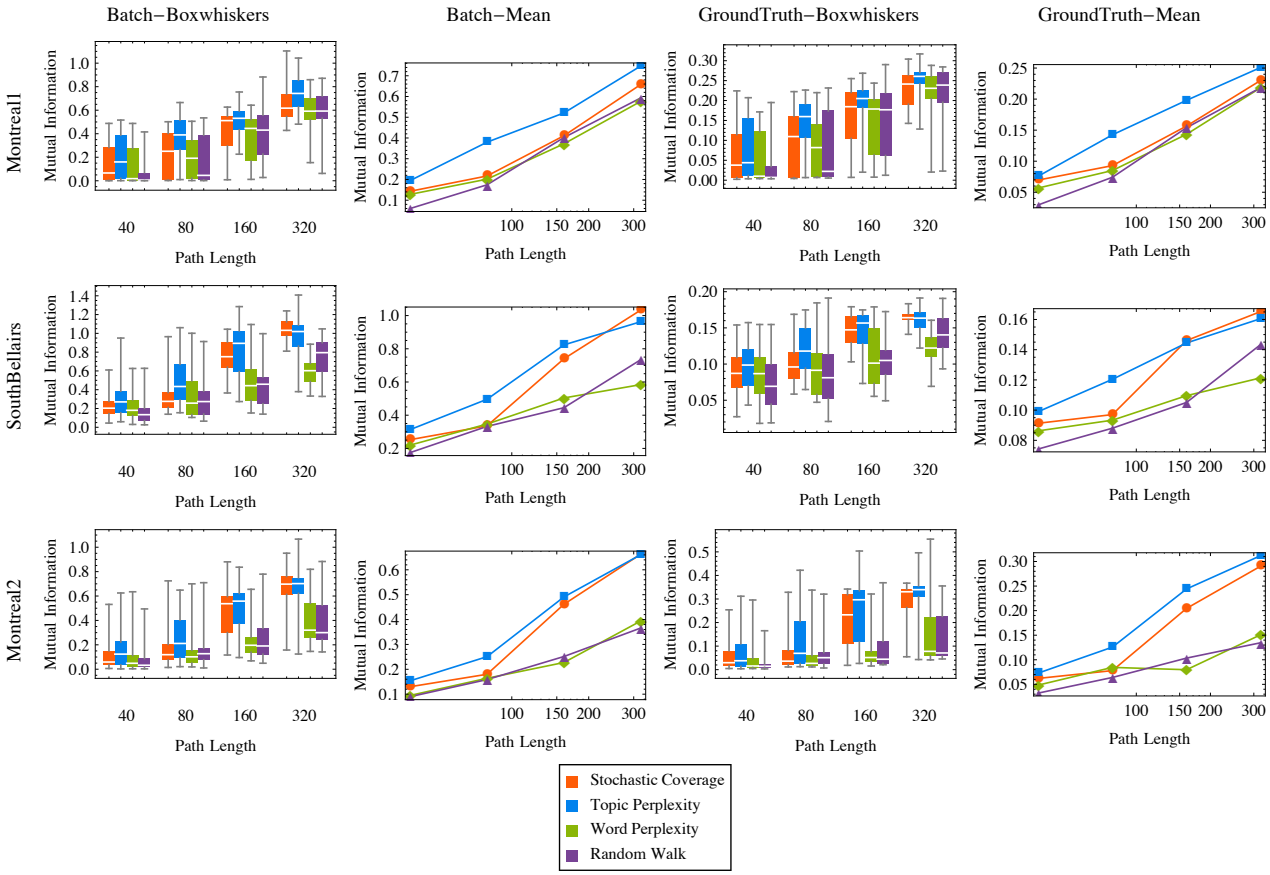


Figure 6.5: Evaluation of the proposed exploration techniques. The plots show mutual information between the maps labeling produced using the topic model computed online during the exploration, with maps labeled by a human, and maps labeled by batch processing of the data.

Dataset	width(px)	height(px)	n.cells	n.words
Montreal1 (aerial)	1024	1024	4096	3,239,631
Montreal2 (aerial)	1024	1024	4096	1,675,171
SouthBellairs (underwater)	2500	2500	6241	1,664,749

Table 6.1: Exploration dataset specifications

6.3 Experiments

6.3.1 Exploration on a 2D Map

Setup

To validate our hypothesis that biasing exploration towards high perplexity cells will result in a better terrain topic model of the environment, we conducted the following experiment. We considered three different maps: two aerial view, and one underwater coral reef map.

We extracted ORB words describing local features, and texton words describing texture at every pixel (every second pixel for the SouthBellairs underwater dataset). ORB [90] words had a dictionary size of 5000, and texton words had a dictionary size of 1000. The dictionary was computed by extracting features from a completely unrelated dataset.

Each of these maps were decomposed into square cells of width 16 pixels (32 for SouthBellairs). Now for each weight function, we computed exploration paths of varying length, with 20 different random restart locations for each case. Each time step was fixed at 200 milliseconds to allow the topic model to converge. We limited the path length to 320 steps, which is about $5\sqrt{|C|}$. Some basic statistics about the three datasets are given in Table 6.1

Each of these exploration runs returned a topic model Φ_p, which we then used to compute topic labels for each pixel in the map in batch mode. Let Z_p be these topic labels. An example of this labeling for each of the three dataset is shown in the last row of Figure 6.4. We compared this topic labeling with two other labelings: human labeled ground-truth Z_h, and labels computed automatically in batch mode Z_b, where we assume random access to the entire map.

We then computed the mutual information between Z_p and Z_h, Z_p and Z_b, and plotted the results as a function of path length, as shown in Figure 6.5.

Results

The results are both encouraging and surprising. As shown in Figure 6.5, we see that topic perplexity based exploration (shown with blue squares) performs consistently better than all other weight functions, when compared against ground truth, or the batch results.

For paths of length 80, which is close to the width of the maps, we see that mutual information between topic perplexity based exploration and ground truth is 1.51, 1.20 and 1.05 times higher respectively for the three datasets, compared to the next best performing technique.

For long path lengths (320 steps or more), stochastic coverage (shown with orange circles) based exploration matches the mean performance of topic perplexity exploration. This is expected because the maps are bounded, and as the path length increases, the stochastic coverage algorithm is able to stumble across different terrains, even without a guiding function.

For short path lengths (40 steps or less), we do not see any statistical difference between the performance of different techniques.

Marked with purple triangles, we see the results of exploration using Brownian random motion. Although this strategy has a probabilistic guarantee of asymptotically complete coverage, but it does so at a lower rate that stochastic coverage exploration startegy. A random walk in two dimensions is expected to travel a distance of $\sqrt{n}$ from start, where n is the number of steps. Hence it is highly likely that it never visits different terrains. The resulting topic models from these paths are hence unable to resolve between these unseen terrains.

The performance of word perplexity exploration (shown with green diamonds) is surprisingly poor in most cases. We hypothesis that this poor performance is due to the algorithm getting pulled towards locations with terrain described by a more complex word distribution. This will cause the algorithm to stay in these complex terrains, and not explore as much as the other algorithms. In comparison, the topic perplexity exploration is not affected by the complexity of the distribution describing the topic, and is only attracted to topic rarity.

6.3.2 Demonstration: Underwater Exploration

We implemented the proposed curiosity modeling system on Aqua amphibious robot [25, 91], and tested it in three different underwater scenarios as shown in the video located at: `http://cim.mcgill.ca/mrl/girdhar/rost/aqua_curiosity.mp4`. In this video we see the robot exploring its environment from two different points of view. We color the cells in robot's view with blue, and change the opacity based on the perplexity score. A cell marked with more opaque blue circle has higher topic perplexity score, and the cell with the highest score is marked with a red color. Figure 6.3 shows some examples of these high perplexity regions in observed images by the robot. For all our experiments, we fixed the number of topics to $K = 64$, and set Dirichlet hyper-parameters $\alpha = 0.1$, $\beta = 0.1$, refinement bias $\tau = 0.5$, and cell curiosity decay $\gamma = 07$.

Scenario 1: Exploring a coral head

In this trial, we started the robot near a coral head surrounded by monotonous sand. We see that the robot quickly gets attracted towards the coral head, and continues to bounce around over this structure while staying away from sand. We see the effect of curiosity decay variable γ, as the robot is successfully able to return back to the coral head several times after going over the much less interesting sandy regions.

Scenario 2: Interaction with a diver

Although our goal was to study the robot as it would interact with a fish, due to lack of cooperation with the fish, we were forced to conduct the experiment with a scuba diver instead. We see that as soon as the diver is in robot's view, it is the singular source of curiosity for the robot. We see the robot following the diver around, and hovering over the diver when he has stopped moving.

Scenario 3: Exploring the ocean floor

In this trial, we started the robot near the ocean floor, which was sparsely populated with sea plants and corals. We see the robot manages to keep its focus on sea life, while not wasting

time over sand.

6.4 Summary

In this chapter we have presented an exploration technique that aims to learn a observation model of the world by finding paths with high information content. The use of a realtime online topic modeling framework allows us to model incoming streams of low level observation data via the use of a latent variable representing the terrain. Given this online model, we measure the utility of the potential next steps in the path. We validated the effectiveness of the proposed exploration technique over candidate techniques by computing mutual information between the terrain maps generated through the use of the learned terrain model, and hand labeled ground truth, on three different datasets.

In our underwater video demonstration, we saw that the emergent behavior of the robot has a striking similarity to that of biological organisms. While the current work on automated exploration was not explicitly bio-inspired, the relationship between exploration by living agents and the behavior that emerges from this algorithm might be a fruitful direction for further research.

Chapter 7

Conclusion

7.1 Discussion

In this thesis we have examined several challenges involved in building intelligent data gathering robots: semantic perception of the environment, summarizing robot's experience concisely using a few interesting observations, making optimal online decisions to identify physical samples for collection or examination, and planning an exploration path that leads to better learning of perception models. We expect such robots to be useful in exploring and monitoring large and challenging environments such as ocean floor, forest canopies, and other planets.

For many data gathering tasks, it is important that the robot senses its environment at a higher level of abstraction than raw sensor readings. ROST, a realtime topic modeling framework presented in this thesis tackles this problem. Using ROST we can automatically discover high level visual concepts such as book shelf, sofa, or windows in an room; or coral, rock and sand in an ocean. Through experimentation with natural and artificially generated data we have demonstrated that ROST performs better than other competing techniques in learning such perception models, given realtime constraints.

Observation data collected by a robot as it traverses an environment is often too big for manual inspection, and it is easy to miss interesting sections of the data. Moreover, there is a need for online summarization of observation data that can be used to provide mission updates consisting of surprising observations to a human operator, over a low bandwidth

communication challenge. Summarization techniques such as k-means that aim to capture the mean properties of the data are not always useful in identifying interesting observations, which are often are the outliers in the data. Hence, we proposed batch and online summarization techniques that aim to identify observations that capture the variance of the collected data rather than the mean properties. We did this by posing the summarization problem as an instance of the k-centers problem, where we minimize the distance of the worst outlier to the closest observation in the summary. Our experiments with real and artificial data have shown that the generated summaries capture interesting observations well, and the proposed online summaries are competitive, and have performance that is statistically indistinguishable from batch summaries.

Physical sample collection and inspection, or alternatively dropping sensors at key locations, are important tasks for many exploration missions. Often we are given a scoring function to measure the utility of these samples or locations, and then during our exploration we need to irrevocably identify the best samples or locations. This problem is fundamentally different from identifying observations for a summary, because the act of picking a sample for summary if revokable. In this thesis we proposed a solution to this problem by formulating it as an instance of the secretaries hiring problem. The proposed analytic solution is optimal if the goal is to maximize the probability of finding all k top locations. We validated the effectiveness of the approach with visual data from an underwater mission, where the scoring function was the Shannon entropy of the image data, measuring the information content of the images. The results showed that although the algorithm is good at picking the top candidate location, often times it might not be able to identify all k locations.

Perhaps the most important goal of an exploration mission is to collect data about different physical phenomena that it might encounter. Using ROST we were able to describe the observations in topic space, which also allowed us to identify observations that are surprising, with high information content. We then proposed an exploration algorithm which biases the path of the robot towards locations with high information content in topic space. Our simulated exploration experiments on real observation data have shown that the topic model learned from the proposed exploration path is better at distinguishing between different terrains, and correlates well with hand labeled data, when compared with other exploration techniques such as free space exploration or random paths. Our evaluation was based on computing mutual information of the topic labels computed using the online topic model learned

during exploration, with hand labeled ground truth data, and with the topic labels computed in batch with random access to all the observation data (which is infeasible in reality).

We implemented ROST and the proposed exploration technique on the Aqua AUV and demonstrated its autonomous exploration behavior.

7.2 Future Directions

Information theoretic robotic exploration techniques can benefit from the developments in data modeling techniques such as topic modeling, and advancements in robotics such as better localization, mapping, and maneuverability.

7.2.1 Richer Perception Models

The topic model proposed in this thesis was flat, and of fixed complexity. ROST can be improved upon in many different ways.

Bayesian nonparametric techniques such as Hierarchical Dirichlet Process (HDP) [103, 104] are useful for building topic models that can grow automatically with the growth in size and complexity of the data, however their scalability and application to streaming data remains an open problem. Our preliminary work on extending ROST to use HDP has shown promise, however much more needs to be done to ensure convergence with large and streaming data.

Learning a topic hierarchy is possible through the use of techniques such as hierarchical LDA (hLDA) [10]. Unlike LDA and ROST, topics in hLDA correspond to a path in a tree. These topics become more specialized as they travel farther down from the root of the tree. Sivic et al. [96] have demonstrated that such topic models can be used to learn better hierarchical description of the scene.

More general learning representations have also been explored in the literature. Recent seminal works by Tenenbaum et al. [105, 67, 66, 48] on modeling the structure of data to understand human learning and reasoning have direct implications on building better semantic perception models. It is however currently unclear how to scale these techniques, and use

them under realtime constraints. Exploring these ideas could enable development of better surprise detection techniques, which is essential for building better exploration robots.

7.2.2 Exploration in Marine Environments

Most locations that still remain unexplored today are either underwater or are extra terrestrial. Although extra terrestrial exploration such as one done using the Curiosity rover [51] is very exciting, it is very expensive to use such missions for testing various exploration algorithms. Marine exploration remains an equally conceptually challenging problem due to the communication bottlenecks, and has many potential applications [31, 4, 19].

The underwater experiments in this thesis using the Aqua robot were done by fixing the depth of the robot, and only controlling the robot on a virtual 2D plane. There is ongoing work [33] on improving the maneuverability of the Aqua robot, with the final goal of making it fully maneuverable in 3D. Using such an autopilot, combined with obstacle avoidance, it would be possible for the robot to hug a 3D structure such as a large coral reef or ship wreck, thereby vastly improving the usability of the proposed algorithms.

While using Aqua to conduct the underwater exploration experiments we faced several challenges due to limited onboard computation, limited battery life, and lack of altitude (surface) sensing. Through the use of a larger AUVs such as the SeaBED [95], Puma, or Juaguar [73], we can hope to perform better and more realistic experiments to test various exploration techniques.

7.2.3 Exploration using Multiple Robots

Exploration is a highly parallelizable task, and use of multiple robots is a natural direction for future work. In our ongoing work [92, 44] we have demonstrated the use of a team of heterogeneous robots to do data collection tasks. Rendezvous of these robots in unknown environments without communications could be achieved by attempting to make rendezvous attempts at pre-specified time intervals [81].

Another challenge in extending ROST to multiple robots is to have a coherent topic model that is learnt in parallel across multiple robots. Apart from the physical and communication

challenges, there exist conceptual challenges in maintaining such coherent topic models across multiple agents. There is work on learning such distributed topic models [83] on a cluster of computers, however extending this to communication constrained multiple robots remains an interesting and unexplored problem.

7.3 Closing Remarks

As humans get closer to exhausting their physical capabilities and in directly exploring the unexplored universe, it is only through the use of autonomous robots that they can continue on their insatiable quest for knowledge and understanding. This thesis is but a small step in this direction that I hope will contribute to discovery of new science and old history in the near future.

Bibliography

[1] Javed A Aslam, Ekaterina Pelekhov, and Daniela Rus. The Star Clustering Algorithm for Static and Dynamic Information Organization. *J. Graph Algorithms Appl.*, 8(1):95–129, 2004.

[2] Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. Online auctions and generalized secretary problems. *SIGecom Exch.*, 7(2):1–11, 2008.

[3] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 434–443, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[4] Robert D. Ballard, Lawrence E. Stager, Daniel Master, Dana Yoerger, David Mindell, Louis L. Whitcomb, Hanumant Singh, and Dennis Piechota. Iron Age Shipwrecks in Deep Water off Ashkelon, Israel. *American Journal of Archaeology*, 106(2):151, April 2002.

[5] Arindam Banerjee and Sugato Basu. Topic Models over Text Streams: A Study of Batch and Online Unsupervised Learning. In *SIAM International Conference on Data Mining*, page 6, 2007.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. *European Conference on Computer Vision*, 2006.

[7] James G Bellingham and Kanna Rajan. Robotics in remote and hostile environments. *Science (New York, N.Y.)*, 318(5853):1098–102, November 2007.

[8] Asher Bender, Stefan B Williams, and Oscar Pizarro. Autonomous Exploration of Large-Scale Benthic Environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 390–396, 2013.

[9] Jonathan Binney, Andreas Krause, and Gaurav S. Sukhatme. Optimizing waypoints for monitoring spatiotemporal phenomena. *The International Journal of Robotics Research*, 32(8):873–888, July 2013.

[10] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. *Advances in neural information processing systems (NIPS)*, 16:106–114, 2004.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[12] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Scene Classification Via pLSA. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, volume 3954 of *Lecture Notes in Computer Science*, pages 517–530. Springer Berlin / Heidelberg, 2006.

[13] Frederic Bourgault, Alexei A Makarenko, Stefan B Williams, Ben Grocholsky, and Hugh F Durrant-Whyte. Information based adaptive robotic exploration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, pages 540—-545, 2002.

[14] Eric Bourque and Gregory Dudek. On the Automated Construction of Image-Based Maps. *Autonomous Robots*, 8(2):173–190, April 2000.

[15] Andrei Z Broder, Adam Kirsch, Ravi Kumar, Michael Mitzenmacher, Eli Upfal, and Sergei Vassilvitskii. The hiring problem and Lake Wobegon strategies. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1184–1193, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[16] Kevin R Canini, Lei Shi, and Thomas L Griffiths. Online Inference of Topics with Latent Dirichlet Allocation. *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 5(1999):65–72, 2009.

[17] Howie Choset and Philippe Pignon. Coverage path planning: The boustrophedon cellular decomposition. *Field and Service Robotics*, 1998.

[18] Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

[19] M Elizabeth Clarke, Nick Tolimieri, and Hanumant Singh. Using the Seabed AUV to Assess Populations of Groundfish in Untrawlable Areas. pages 357–372, 2009.

[20] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647, June 2008.

[21] Jnaneshwar Das, Frederic Py, Thom Maughan, Tom OReilly, Monique Messie, John Ryan, Gaurav S Sukhatme, and Kanna Rajan. Coordinated sampling of dynamic oceanographic features with underwater vehicles and drifters. *The International Journal of Robotics Research*, 31(5):626–646, April 2012.

[22] John G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, 1985.

[23] Scott C Deerwester, Susan T Dumais, Thomas K Landauer, George W Furnas, and Richard A Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[24] Bir Bikram Dey, Sandeep Manjanna, and Gregory Dudek. Ninja legs: Amphibious one degree of freedom robotic legs. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5622–5628. IEEE, November 2013.

[25] Gregory Dudek, Philippe Giguere, Chris Prahacs, Shane Saunderson, Junaed Sattar, Luz-Abril Torres-Mendez, Michael Jenkin, Andrew German, Andrew Hogue, Arlene Ripsman, Jim Zacher, Evangelos Milios, Hui Liu, Pifu Zhang, Marti Buehler, and Christina Georgiades. AQUA: An Amphibious Autonomous Robot. *Computer*, 40(1):46–53, 2007.

[26] Gregory Dudek and John-Paul Lobos. Towards Navigation Summaries: Automated Production of a Synopsis of a Robot Trajectories. In *2009 Canadian Conference on*

Computer and Robot Vision, pages 93–100, Kelowna, British Columbia, May 2009. IEEE.

[27] Gregory Dudek, Junaed Sattar, and Anqi Xu. A Visual Language for Robot Control and Programming: A Human-Interface Study. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2507–2513, Roma, April 2007. IEEE.

[28] Eugene Borisovich Dynkin. The optimum choice of the instant for stopping a Markov process. *Soviet Math. Dokl*, 4, 1963.

[29] Li Fei-Fei and Pietro Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531. IEEE, June 2005.

[30] Thomas S Ferguson. Who Solved the Secretary Problem? *Statistical Science*, 4(3):282–296, 1989.

[31] Brendan P Foley, Katerina Dellaporta, Dimitris Sakellariou, Brian S Bingham, Richard Camilli, Ryan M Eustice, Dionysis Evagelistis, Vicki Lynn Ferrini, Kostas Katsaros, Dimitris Kourkoumelis, Aggelos Mallios, Paraskevi Micha, David A Mindell, Christopher Roman, Hanumant Singh, David S Switzer, and Theotokis Theodoulou. The 2005 Chios Ancient Shipwreck Survey: New Methods for Underwater Archaeology. *Hesperia*, 78(2):269–305, June 2009.

[32] P R Freeman. The secretary problem and its extensions: A review. *International Statistical Review*, 1983.

[33] Philippe Giguere, Yogesh Girdhar, and Gregory Dudek. Wide-Speed Autopilot System for a Swimming Hexapod Robot. In *2013 International Conference on Computer and Robot Vision*, pages 9–15. IEEE, May 2013.

[34] Yogesh Girdhar and Gregory Dudek. Optimal Online Data Sampling or How to Hire the Best Secretaries. In *CRV '09: Proceedings of the 2009 Canadian Conference on Computer and Robot Vision*, pages 292–298, Kelowna, British Columbia, May 2009. IEEE Computer Society.

[35] Yogesh Girdhar and Gregory Dudek. Online Navigation Summaries. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5035–5040, May 2010.

[36] Yogesh Girdhar and Gregory Dudek. ONSUM: A System for Generating Online Navigation Summaries. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, pages 746–751, October 2010.

[37] Yogesh Girdhar and Gregory Dudek. Offline Navigation Summaries. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5769–5775, May 2011.

[38] Yogesh Girdhar and Gregory Dudek. Online Visual Vocabularies. In *CRV '11: Proceedings of the 2011 Canadian Conference on Computer and Robot Vision*, pages 191–196. IEEE Computer Society, May 2011.

[39] Yogesh Girdhar and Gregory Dudek. Efficient on-line data summarization using extremum summaries. In *2012 IEEE International Conference on Robotics and Automation*, pages 3490–3496, Saint Paul, MN, May 2012. IEEE.

[40] Yogesh Girdhar and Gregory Dudek. Exploring Underwater Environments with Curiosity. In *2014 Canadian Conference on Computer and Robot Vision*, pages 104–110, Montreal, May 2014. IEEE.

[41] Yogesh Girdhar, Philippe Giguere, and Gregory Dudek. Autonomous adaptive exploration using realtime online spatiotemporal topic modeling. *The International Journal of Robotics Research*, 33(4):645 – 657, November 2013.

[42] Yogesh Girdhar, David Whitney, and Gregory Dudek. Curiosity Based Exploration for Learning Terrain Models. In *IEEE International Conference on Robotics and Automation*, page 7, October 2014.

[43] Yogesh Girdhar, Anqi Xu, Bir Bikram Dey, Malika Meghjani, Florian Shkurti, Ioannis Rekleitis, and Gregory Dudek. MARE: Marine Autonomous Robotic Explorer. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5048 – 5053, San Francisco, USA, September 2011.

[44] Yogesh Girdhar, Anqi Xu, Florian Shkurti, Juan Camilo, Gamboa Higuera, Malika Meghjani, Philippe Giguere, Ioannis Rekleitis, and Gregory Dudek. Monitoring Marine Environments using a Team of Heterogeneous Robots. In *RSS 2012 Workshop on Robotics for Environmental Monitoring*, 2012.

[45] Yihong Gong and Xin Liu. Video summarization using singular value decomposition. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000*, volume 2, pages 174–180, Hilton Head Island, SC, 2000. IEEE Comput. Soc.

[46] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[47] Robert Grabowski, Pradeep Khosla, and Howie Choset. Autonomous exploration via regions of interest. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 2, pages 1691–1696. IEEE, 2003.

[48] Thomas L. Griffiths, Charles Kemp, and Joshua B. Tenenbaum. Bayesian models of cognition. *Cambridge handbook of computational cognitive modeling*, pages 59–100, 2008.

[49] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

[50] Simona E Grigorescu, Nicolai Petkov, and Peter Kruizinga. Comparison of texture features based on Gabor filters. *IEEE Transactions on Image processing*, 11(10):1160–1167, 2002.

[51] John P. Grotzinger, Joy Crisp, Ashwin R. Vasavada, Robert C. Anderson, Charles J. Baker, Robert Barry, David F. Blake, Pamela Conrad, Kenneth S. Edgett, Bobak Ferdowski, Ralf Gellert, John B. Gilbert, Matt Golombek, Javier Gómez-Elvira, Donald M. Hassler, Louise Jandura, Maxim Litvak, Paul Mahaffy, Justin Maki, Michael Meyer, Michael C. Malin, Igor Mitrofanov, John J. Simmonds, David Vaniman, Richard V. Welch, and Roger C. Wiens. Mars Science Laboratory Mission and Science Investigation. *Space Science Reviews*, 170(1-4):5–56, July 2012.

[52] Felix Hausdorff. *Grundzüge der Mengenlehre*. 1914.

[53] Thomas Hofmann. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42(1):177–196, 2001.

[54] Geoffrey A Hollinger, Brendan Englot, Franz S Hover, Urbashi Mitra, and Gaurav S Sukhatme. Active planning for underwater inspection and the benefit of adaptivity. *The International Journal of Robotics Research*, 32(1):3–18, November 2012.

[55] Wen-Lian Hsu and George L Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.

[56] David Hume. *A Treatise of Human Nature*. 1738.

[57] Richard Sewall Hunter. Photoelectric Color-Difference Meter. *Journal of the Optical Society of America (JOSA)*, 38(7):661, 1948.

[58] Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. *Vision Research*, 49(10):1295–1306, 2009.

[59] A K Jain and F Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[60] Bela Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, March 1981.

[61] Jefrey W. Kaeli. *Computational Strategies for Understanding Underwater Optical Image Datasets*. PhD thesis, Woods Hole Oceanographic Institution and Massachusetts Institute of Technology, 2013.

[62] R M Karp. Reducibility Among Combinatorial Problems. In R E Miller and J W Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[63] Hirokazu Kato. ARToolKit: library for Vision-Based augmented reality. *IEICE, PRMU*, pages 79—-86, 2002.

[64] L. Kaufman and P.J Rousseeuw. *Clustering by means of medoids*. North-Holland, 1987.

[65] Leonard Kaufman and Peter J. Rousseeuw. Partitioning Around Medoids (Program PAM). In *Finding Groups in Data*, pages 68—-125. John Wiley & Sons, Inc., 2008.

[66] Charles Kemp and Joshua B Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences of the United States of America*, 105(31):10687–92, August 2008.

[67] Charles Kemp, Joshua B Tenenbaum, Sourabh Niyogi, and Thomas L Griffiths. A probabilistic model of theory formation. *Cognition*, 114(2):165–96, February 2010.

[68] Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 630–631, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[69] Thomas Kollar and Nicholas Roy. Efficient Optimization of Information-Theoretic Exploration in SLAM. In *AAAI*, volume 8, pages 1369–1375, 2008.

[70] Thomas Kollar and Nicholas Roy. Trajectory optimization using reinforcement learning for map exploration. *The International Journal of Robotics Research*, 27(2)(June):175–197, 2008.

[71] Kurt Konolige, James Bowman, JD Chen, Patrick Mihelich, Michael Calonder, Vincent Lepetit, and Pascal Fua. View-based maps. *Robotics: Science and Systems Conference (RSS)*, 2009.

[72] Solomon Kullback. *Information theory and statistics*. John Wiley and Sons, NY, 1959.

[73] Clayton Kunz, Chris Murphy, Hanumant Singh, Claire Pontbriand, Robert A. Sohn, Sandipa Singh, Taichi Sato, Chris Roman, Ko-ichi Nakamura, Michael Jakuba, Ryan Eustice, Richard Camilli, and John Bailey. Toward extraplanetary under-ice exploration: Robotic steps in the Arctic. *Journal of Field Robotics*, 26(4):411–429, April 2009.

[74] Kenichi Kurihara, Max Welling, and Nikos Vlassis. Accelerated Variational Dirichlet Process Mixtures. In *Advances in Neural Information Processing Systems*, pages 761—-768, 2006.

[75] Jimmy Li, Anqi Xu, and Gregory Dudek. Graphical State Space Programming: A visual programming paradigm for robot task specification. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA '11)*, pages 4846–4853, 2011.

[76] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[77] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1967.

[78] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and Texture Analysis for Image Segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[79] Raphael Mannadiar and Ioannis Rekleitis. Optimal coverage of a known arbitrary environment. In *2010 IEEE International Conference on Robotics and Automation*, pages 5525–5530. IEEE, May 2010.

[80] Roar Marthiniussen, Karstein Vestgird, and Rolf Arne Klepaker. HUGIN-AUV concept and operational experiences to date. In *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, pages 846–850, 2004.

[81] Malika Meghjani and Gregory Dudek. Multi-robot exploration and rendezvous on graphs. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5270–5276. IEEE, October 2012.

[82] Ilya Nemenman, Fariel Shafee, and William Bialek. Entropy and inference, revisited. *ArXiv e-prints physics/0108025*, page 9, August 2001.

[83] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed Algorithms for Topic Models. *The Journal of Machine Learning Research*, 10:1801–1828, 2009.

[84] Christos H Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent Semantic Indexing: A Probabilistic Analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159—-168, 1998.

[85] Rohan Paul, Daniela Rus, and Paul Newman. How was your day? Online Visual Workspace Summaries using Incremental Clustering in Topic Space. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[86] James Philbin, Josef Sivic, and Andrew Zisserman. Geometric LDA: A Generative Model for Particular Object Discovery. In *In Proceedings of the British Machine Vision Conference*, 2008.

[87] Nicolas Plamondon. *Modeling and control of a biomimetic underwater vehicletle*. PhD thesis, McGill University, 2010.

[88] Plato. *Phaedo*. 360.

[89] Ananth Ranganathan and Frank Dellaert. Bayesian surprise and landmark detection. In *2009 IEEE International Conference on Robotics and Automation*, pages 2017–2023, Kobe, May 2009. Institute of Electrical and Electronics Engineers Inc., The, IEEE.

[90] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571, Barcelona, November 2011. IEEE.

[91] Junaed Sattar, Gregory Dudek, Olivia Chiu, Ioannis Rekleitis, Philippe Giguère, Alec Mills, Nicolas Plamondon, Chris Prahacs, Yogesh Girdhar, Meyer Nahon, and John-Paul Lobos. Enabling Autonomous Capabilities in Underwater Robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, pages 3628–3634, Nice, France, September 2008.

[92] Florian Shkurti, Anqi Xu, Malika Meghjani, Juan Camilo Gamboa Higuera, Yogesh Girdhar, Philippe Giguere, Bir Bikram Dey, Jimmy Li, Arnold Kalmbach, Chris Prahacs, Katrine Turgeon, Ioannis Rekleitis, and Gregory Dudek. Multi-Domain Monitoring of Marine Environments using a Heterogeneous Robot Team. In *Intelligent Robots and Systems (IROS)*, pages 1747–1753, 2012.

[93] Mark Sibenac, William J Kirkwood, Robert Mcewen, Farley Shane, and Richard Henthom. Modutar AUV for Routine Deep Water Science Operations. In *OCEANS '02 MTS/IEEE*, pages 167–172, 2001.

[94] Robert Sim, Gregory Dudek, and Nicholas Roy. Online control policy optimization for minimizing map uncertainty during exploration. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 1758–1763. IEEE, 2004.

[95] Hanumant Singh, Roy Armstrong, Fernando Gilbes, Ryan Eustice, Chris Roman, Oscar Pizarro, and Juan Torres. Imaging Coral I: Imaging Coral Habitats with the SeaBED AUV. *Subsurface Sensing Technologies and Applications*, 5(1):25–42, January 2004.

[96] Josef Sivic, Bryan C Russell, Andrew Zisserman, William T Freeman, and Alexei A Efros. Unsupervised discovery of visual object class hierarchies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2008.

[97] Josef Sivic and Andrew Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2, Nice, France, 2003. IEEE.

[98] Josef Sivic and Andrew Zisserman. Video Google: Efficient Visual Search of Videos. In Jean Ponce, Martial Hebert, Cordelia Schmid, and Andrew Zisserman, editors, *Toward Category-Level Object Recognition*, volume 4170 of *Lecture Notes in Computer Science*, pages 127–144. Springer Berlin / Heidelberg, 2006.

[99] Petr Slavík. A Tight Analysis of the Greedy Algorithm for Set Cover. *Journal of Algorithms*, 25(2):237–254, 1997.

[100] Ryan N. Smith, Mac Schwager, Stephen L. Smith, Burton H. Jones, Daniela Rus, and Gaurav S. Sukhatme. Persistent Ocean Monitoring with Underwater Gliders: Adapting Sampling Resolution. *Journal of Field Robotics*, 28(5):714–741, 2011.

[101] Cyrill Stachniss, G Grisetti, and W Burgard. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. *Robotics: Science and Systems*, 2, 2005.

[102] Daniel M. Steinberg, Ariell Friedman, Oscar Pizarro, and Stefan B. Williams. A Bayesian Nonparametric Approach to Clustering Data from Underwater Robotic Surveys. In *15th International Symposium on Robotics Research*, pages 1–16, Flagstaff, AZ, 2011.

[103] Yee Whye Teh and Michael I. Jordan. Hierarchical Bayesian Nonparametric Models with Applications. *Bayesian nonparametrics*, pages 158—-207, 2010.

[104] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, December 2006.

[105] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: statistics, structure, and abstraction. *Science (New York, N.Y.)*, 331(6022):1279–85, March 2011.

[106] Ba Tu Truong and Svetha Venkatesh. Video abstraction. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1), February 2007.

[107] Mathew A. Turk and Alex P. Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591. IEEE Comput. Sco. Press, 1991.

[108] Mark R Turner. Texture discrimination by Gabor functions. *Biological Cybernetics*, 55(2):71–82, 1986.

[109] Víctor Valdés and José M. Martínez. A framework for video abstraction systems analysis and modelling from an operational point of viewhttp://www.springerlink.com/content/j350425v67p66712/fulltext.pdf. *Multimedia Tools and Applications*, 49(1):7–35, October 2009.

[110] Xiaogang Wang and Eric Grimson. Spatial Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems*, volume 20, pages 1577–1584, 2007.

[111] David H Wolpert and David R Wolf. Estimating functions of probability distributions from a finite set of samples. *Phys. Rev. E*, 52(6):6841–6854, December 1995.

[112] Anqi Xu, Chatavut Viriyasuthee, and Ioannis Rekleitis. Optimal complete terrain coverage using an Unmanned Aerial Vehicle. In *2011 IEEE International Conference on Robotics and Automation*, pages 2513–2519. IEEE, May 2011.

[113] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE Comput. Soc. Press, 1997.