

Content-Based Image Retrieval Through a Multi-Agent Meta-Learning Framework

Abraham Bagherjeiran, Ricardo Vilalta, Christoph F. Eick
{abagherj, vilalta, ceick}@cs.uh.edu
University of Houston
Department of Computer Science
Houston, TX 77204-3010

Abstract

The objective of a general-purpose content-based image retrieval system is to find images in a database that match an external measure of relevance. Since users follow different and inconsistent relevance measures, processing queries in a task-specific manner has shown to be an effective approach. Viewing specialized image retrieval algorithms as agents, we propose a general-purpose image retrieval system that uses a new multi-agent meta-learning framework. The framework adapts a distance function defined over both image distance weights and image queries to identify clusters of algorithms that produce similar solutions to similar problems. Experiments compare our approach with a traditional information retrieval algorithm; results show that our framework provides better average relevance scores.

1. Introduction

The goal of a content-based image retrieval algorithm is to predict which images in a database are relevant to an arbitrary query. Many retrieval algorithms learn distance function weights based on feedback on a set of related queries [10]. Recent work has shown evidence that no single set of weights is best for queries associated with different tasks [5].

It is now well acknowledged that an image retrieval system for arbitrary images should specialize for particular classes of queries. Specializing learning algorithms for classes of tasks is a subject of meta-learning or bias learning [3]. Many meta-learning algorithms learn distance function weights that apply to problems of a known class distribution. This solution is not entirely applicable in image retrieval, however, because relevance is often subjectively defined and not available during training. In image retrieval, the meta-learning approach must be extended to first discover common relevance classes, to determine which re-

trieval algorithms best serve them, and finally to adapt to changes in relevance measures and learning algorithms. We approach this problem as a multi-agent meta-learning problem and propose a new framework for its solution.

The framework, *Artificial Government*, solves the problem by viewing agents as meta-learning agents in a single task space. In our application to image retrieval, weight-learning retrieval algorithms are the learning agents, and we model human users as evaluation agents. As the retrieval algorithms respond to image query tasks, they receive an evaluation of their performance on the task in the form of relevance feedback. The objective of the framework is to determine the best match between image query tasks and retrieval algorithms by adapting a distance function defined over the learning agents that aims to maximize their evaluations.

This paper is organized as follows. Section 2 describes work in meta-learning directly relevant to Artificial Government. Artificial Government is described in Section 3. Section 4 describes our experiments. Section 5 presents and discusses our results. Section 6 describes additional related work and conclusions.

2. Meta-Learning

We begin by reviewing related work in meta-learning that bears a direct connection to Artificial Government. Additional related work is discussed in Section 6.

2.1. Meta-Learning

In meta-learning, a learning algorithm L learns from a family of potentially related tasks T_L rather than just a single task $t \in T_L$. The objective of the meta-learning algorithm is to find the hypothesis space $H_L(T_L) \in \mathcal{H}$ that contains the hypotheses that minimize the expected loss over all examples in all tasks faced by the learning algorithm [3] is:

$$H_L(T_L) = \arg \min_{H \in \mathcal{H}} \int_{t \in T} \inf_{h \in H} \int_{X \times Y} \lambda(h(x), y) dP(x, y) dQ(t)$$

where a task t is a sample of input-output pairs (x, y) , $P(x, y)$ is the probability of drawing the pair (x, y) , $Q(t)$ is the probability of drawing that sample, and $\lambda(h(x), y)$ is the loss for classifying x as $h(x)$ given that its true class is y . The idea is that if the hypothesis space is specialized for a task set T_L , the learning algorithm can find all locally optimal hypotheses for every conceivable task in the task set. One can describe a meta-learning algorithm L as the pair $\{T_L, H_L(T_L)\}$ meaning that it has learned the optimal hypothesis space $H_L(T_L)$ for the set of tasks T_L on which it has trained. One important concept to understand is that the globally optimal hypothesis space over all tasks may not, in practice, contain the locally optimal hypothesis for each task. The fundamental assumption behind meta-learning is that, the globally optimal hypothesis space is “optimal enough” for each separate task. The justification for the assumption is that since many hypotheses are equally optimal, then one should exist that is reasonably good for all tasks.

The justification for the multi-task optimality of a hypothesis relies on the additional assumption that tasks are actually related. Given the description of a learning algorithm as a set of tasks and the optimal hypothesis space for the task, we can consider the distribution of the set of all learning algorithms in this joint task-hypothesis space. In this space, the probability corresponds to the event that a given hypothesis space is optimal for a task. The assumption that a single optimal hypothesis space exists for a set of related tasks means that there is a single peak in this probability distribution in a region of similar hypotheses and similar tasks.

2.2. The Multi-Agent Meta-Learning Problem

The single-peak task distribution assumption behind meta-learning is suitable for single-agent learning algorithms because one expects a single-agent learning algorithm to encounter only a small subset of all possible tasks. The presence of multiple agents presents new challenges that requires a different approach.

The multi-agent meta-learning problem assumes a set of N heterogeneous meta-learning agents. Each agent finds a hypothesis space that minimizes the loss over a set of tasks; heterogeneous agents have different initial hypotheses and task spaces. In the multi-agent meta-learning problem, the system of agents looks for a constant number $k > 1$ of locally optimal hypothesis spaces. These hypothesis spaces correspond to each of k peaks that are assumed to exist in the task-hypothesis probability distribution. The solution to the multi-agent meta-learning problem is a correspondence between the set of the k subsets of tasks $\{T_1, \dots, T_k\} \in \mathcal{T}$ and k hypothesis spaces $\{H_1, \dots, H_k\} \in \mathcal{H}$ such that the aggregate loss is minimized over all possible tasks for all possible optimal hypotheses.

The assumption here is not only that there are $k > 1$

optimal hypothesis spaces but that there are $1 < k \leq N$ spaces for N agents. The assumption implies that there are common task groups among the agents such that agents in these groups form similar solutions to similar problems. A multi-agent system should exploit this shared information so that agents producing similar solutions receive similar problems.

3. Artificial Government

Artificial Government (AG) is a multi-agent meta-learning framework designed to exploit similar solutions to similar problems. This section provides an overview and some motivation behind the framework. It describes the shared task-hypothesis space of the agents. In addition, it describes the roles of the agents and their interactions.

Our framework presumes that a group of intelligent, autonomous, and heterogeneous agents has agreed to collaborate on a set of related tasks. The proposed perspective is that in every collaboration, heterogeneous agents with different goals will tend to develop two distinct roles. Producer agents will work on solving individual tasks from scratch. Consumer agents will provide new tasks and use their solutions as components for other, more complex problems. We can imagine this two-role agent system as follows: consumers will seek to find solutions for their tasks while producers will seek to acquire problem-solving expertise by solving new tasks.

An important question is now addressed: how do consumers find the solutions to their tasks and how do producers know what tasks to solve? Artificial Government addresses this question by introducing a third agent: the mediator. The mediator discovers the tasks that consumers wish to solve and the solutions that producers are capable of producing. It guides producers toward those solutions that are most beneficial to consumers. As problem-solving learning agents, producers will continue to solve problems, but the goal of the mediator is to help each producer solve the problems for which the consumer will provide positive feedback.

3.1. MP Space

Section 2 described the status of a learning algorithm by the current task and hypothesis space presumed to hold a solution to the task. This description bears little value for heterogeneous meta-learning agents because they use different learning algorithms to explore potentially incompatible hypothesis spaces. We introduce the model-policy (MP) space as an abbreviated description of producers that is independent of the learning algorithm. Producers use a function $f_{\mathcal{M}}$ to form *models* that describe their tasks and a function $f_{\mathcal{P}}$ to form *policies* that describe their hypothesis spaces. Together, a producer’s model and policy form

a point in the MP space for a finite number n_T of tasks $T \in \mathcal{T}^{n_T}$ and a finite number n_H of hypothesis spaces $H \in \mathcal{H}^{n_H}$ at a particular moment in time.

Definition: MP Space Let \mathcal{M} be the set of models, \mathcal{P} be the set of policies, and $f_{\mathcal{M}}$ and $f_{\mathcal{P}}$ be functions. The set $\mathcal{M} \times \mathcal{P}$ is a model policy space if and only if $f_{\mathcal{M}} : \mathcal{T}^{n_T} \rightarrow \mathcal{M}$ and $f_{\mathcal{P}} : \mathcal{H}^{n_H} \rightarrow \mathcal{P}$.

3.2. Producers

Each producer P_i has access to a set of tasks $T_{P_i} \subseteq \mathcal{T}$ and generates an optimal hypothesis space $H_{P_i}(T_{P_i}) \in \mathcal{H}$. Following the preceding discussion, a model-policy pair $\{f_{\mathcal{M}}(T_{P_i}), f_{\mathcal{P}}(H_{P_i}(T_{P_i}))\}$ will be used to describe each producer. For simplicity, we assume that the number of tasks used to describe the producers is the same for all producers as with the number of hypotheses.

While learning to solve many tasks to gain problem-solving experience, a producer explores a region in the model-policy space. The tasks it solves will determine the direction and area of exploration. As a specialized meta-learning algorithm, the producer should concentrate on specific regions in the model-policy space.

3.3. Consumers

A consumer is characterized by a pair $C_i = \{T_{C_i}, \lambda_{C_i}\}$ where $T_{C_i} \subseteq \mathcal{T}$ is a subset of a common set of tasks and λ_{C_i} is a consumer-specific loss (reward) function that the consumer will use to evaluate a potential hypothesis from a producer. The consumer seeks to find a hypothesis that minimizes the consumer-specific loss for any task that it provides. Several consumers will provide different subsets T_{C_i} of tasks and loss functions. Under the single-agent meta-learning framework, only one consumer exists. It provides the one set of tasks \mathcal{T} and the one loss function λ . Artificial Government assumes the existence of multiple, different consumers each with different loss functions λ_{C_i} .

The consumer defines an evaluation function over the MP space and selects which producers should receive its tasks. It provides a producer-level, $C_{i,1}$, and task-level, $C_{i,2}$, evaluation for the producer that performed the task. For function $C_{i,1}(MP) \in [0, 1]$, a value of 1 indicates that the consumer is completely satisfied with the model-policy pair, and 0 indicates that the consumer is totally dissatisfied. The function $C_{i,2}$ is a task-specific function and will be defined later in the context of image retrieval.

When selecting a producer to solve its task, a consumer can only select one of the k locally optimal hypothesis spaces. It randomly selects one according to a distribution

$$I = \{i_1, \dots, i_k\}$$

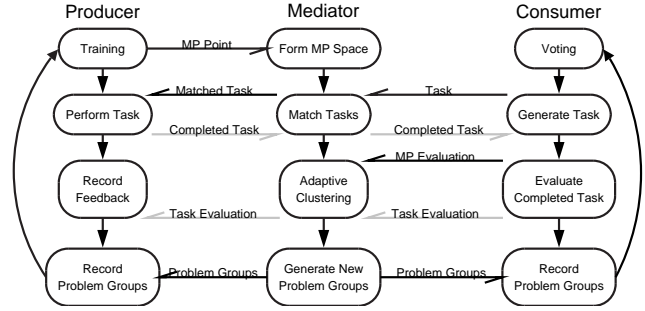


Figure 1. The Artificial Government interaction process.

where $\sum_{i_i \in I} i_i = 1$ and $\forall_{i_i \in I} i_i \geq 0$

that represents the probability that the consumer will choose 1 of the k locally optimal hypothesis spaces. The consumer updates the probabilities to emphasize spaces that have worked well in the past while still permitting exploration.

3.4. Mediator

The goal of the mediator is to find a correspondence between consumers' tasks and producers' MP points, and it follows the sequence of steps illustrated in Figure 1. As producers seek optimal hypothesis spaces, they receive new tasks that the mediator expects to positively contribute to their meta-learning processes. In our current implementation we have made some assumptions on the distribution of tasks and the MP space. We assume that the distributions are mixtures of models, where each model is a Normal or Gaussian density, and there is a fixed number, k , of models.

The mediator employs a variant of the k -means clustering algorithm to group MP points [8]. This calls for a distance function d that quantifies the distance between points. The function $d_{MP} : MP \times MP \rightarrow \mathbb{R}$ is called a *model-policy distance function*. The distance function enables us to identify the k locally optimal hypothesis spaces as problem groups which are mutually exclusive and exhaustive clusters in the MP space. A problem group has a representative called the prototype that is the centroid of the points in the group.

The consumer returns a scalar value in $[0, 1]$ that we view as a weighting of the points in the MP space. With the application of clustering to these weighted points, the cluster centroids will emphasize points that most satisfy the consumers.

The mediator defines two different sets of problem groups. The consumers' problem groups are computed with

an unweighted distance function but weighted MP points such that the mean of the MP points

$$\mu_{g_{C,j}} = \frac{1}{\sum_{i=1}^N C_{i,1}} \sum_{i=1}^N C_{i,1} (MP_i) MP_i$$

is weighted by a consumer’s evaluation of the producer that occupies the MP point in the problem group $g_{C,j}$. The computation assumes that each MP point has exactly one consumer evaluation. The producers’ problem groups are a clustering of the MP space using a weighted distance function:

$$d_{MP}(MP_i, MP_j) = \sum_{l=1}^L v_l |MP_{i,l} - MP_{j,l}|$$

where L is the length of an MP point, and the weights v_l sum to 1. Letting $G_C = \{g_{C,1}, \dots, g_{C,k}\}$ be the prototypes of the consumers and $G_P = \{g_{P,1}, \dots, g_{P,k}\}$ be the prototypes of the producers and assuming that the number of prototypes k is the same for the two groups, the distance between the groups is

$$d_{MP}(G_C, G_P) = \frac{1}{k^2} \sum_{i=1}^k \sum_{j=1}^k d_e(g_{C,i}, g_{P,j})$$

the average Euclidean distance (d_e) between the centroids in both groups.

We use an adaptive clustering algorithm that explores the space of distance weights by adjusting the weight of a single attribute upon encountering a new clustering [1]. It updates the distance function until it finds a clustering with a minimum inter-group distance. The problem groups G_C and G_P with this minimum distance are the correspondence that is the solution to the multi-agent meta-learning problem and the goal of the mediator.

4. Experiments

The purpose of the experimental evaluation of Artificial Government is to show that consumers receive a better quality of service with rather than without the mediator. In the multi-agent retrieval domain, the 10 retrieval producers will serve a total of 1000 queries from 10 consumers each having different query images and relevance scores. A producer retains its learned weights between tasks in both experiments. Without the mediator, the consumer has no choice over its producer; with the mediator, the agents follow the previously described interaction process.

A producer sorts its specific database in increasing order of the weighted distance to the query image. The producer employs the relevance feedback algorithm to learn the best set of distance weights [10]. The weights are randomly ini-

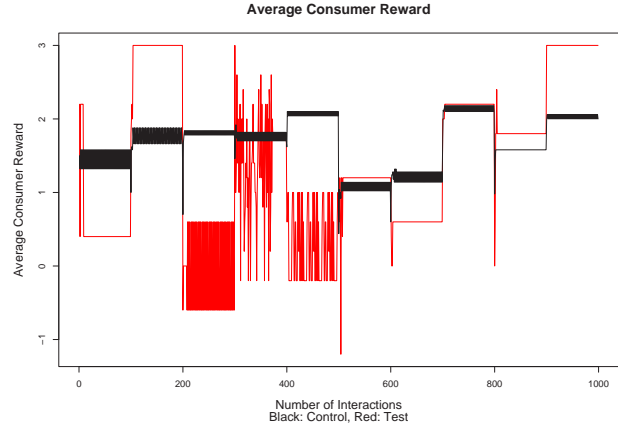


Figure 2. Average consumer reward.

tialized as seeds for different learning experiences.

In this model-policy (MP) space, the model is the average distance between each pair of images in the last query result set using the current weight vector. The policy is the vector of distance function weights.

Each consumer has a randomly selected relevance score and maintains a probability distribution over the set of prototypes. The score ranges from $\{-3, \dots, 3\}$ where -3 indicates least relevance and 3 indicates most relevance. The evaluation over the MP space is the average of the relevance scores normalized to be within the range $[0, 1]$. After the consumer evaluates its solution, it will revise its probabilities to favor prototypes that have done well in the past.

5. Results

The results for the experiments were generated after each interaction with the mediator¹. Figure 2 shows the average consumer reward for all 10 consumers for the duration of the experiment. The vertical axis is the value of the evaluation or reward function $C_{i,1}$ which is in the range $[-3, 3]$. The horizontal axis is the number of interactions with the mediator.

Figure 2 shows that on average, the consumers receive better performance with the mediator than without it. The graph shows a few shorter drops in the reward plots with each new query image (100 iterations). This implies that the producer that serves the consumer for the query image does not have to relearn as much as it did without the mediator. For approximately 50% of the query images, the consumers had higher average relevance with the mediator than without it.

¹A detailed account of the experiments is available through the authors.

6. Related Work

This section provides a brief overview relation of previous work to Artificial Government. We briefly touch on blackboard systems, hierarchical reinforcement learning, multi-agent systems, and meta-learning.

Blackboard systems provide a distributed architecture to solve knowledge-intensive problems [7]. As in blackboard systems, producers and consumers collaborate on a problem by updating entries in the *MP* space, but AG learns the tasks on which producers should learn to better satisfy the consumers.

In hierarchical reinforcement learning (HRL) algorithms, an agent can operate through a user-provided decomposition of the environment into behavioral patterns of primitive actions [6, 9]. Similarly, the goal of the mediator in AG is to discover producers that learn similar solutions to similar tasks and then to give the producers similar tasks that better satisfy the consumers.

Extensions to HRL learn the hierarchical structure by clustering and extend reinforcement learning (RL) to multi-agent systems [2, 4]. Artificial Government applies an adaptive clustering method to learn the structure of the interactions between the producers and consumers in the form of problem groups. Like multi-agent RL algorithms, AG learns on higher-level shared information and then distributes to the producers and consumers the tasks a producer should solve.

Markets have been used as coordination mechanisms for multi-agent systems [11]. Artificial Government draws on the motivation behind market: the idea of a globally equitable allocation of tasks to learning algorithms. The allocation balances the supply and demand for similar solutions to similar problems.

6.1. Conclusion

The goal of a general-purpose content-based image retrieval system is to find images in a database that match an external measure of relevance. This paper proposes a multi-agent image retrieval system that supports the tailoring of retrieval algorithms to cope with different relevance measures. Creating this system requires solving a multi-agent meta-learning problem in which the goal is to find which specialized learning algorithms are best able to solve new tasks

We call our solution *Artificial Government* because it provides the roles necessary for agents in the multi-agent system to solve the meta-learning problem *i.e.*, to govern themselves. Following these roles, meta-learning producer agents are image retrieval algorithms that learn weights for an image distance function for query tasks. Consumer agents provide the tasks and feedback. A mediator agent uses an adaptive clustering algorithm to identify the com-

mon tasks and solutions that best satisfy the consumers and then controls which producer serves which query task. The experimental results show that the consumers receive better results with the mediator than without it.

The main contributions of this paper are that the proposed multi-agent meta-learning framework learns a correspondence between image retrieval algorithms and tasks for consumers with different relevance measures and that using this framework provides greater average relevance to the consumers. Existing work in image retrieval largely favors manual selection of image distance function weights for particular classes of queries or relearning the weights from scratch for new queries. Our system learns this information by observing and controlling the interactions between the producers and consumers in the retrieval system.

References

- [1] A. Bagherjeiran, C. F. Eick, and R. Vilalta. Adaptive clustering: Better representatives with reinforcement learning. Technical Report UH-CS-05-06, University of Houston Computer Science Department, 2005.
- [2] B. Bakker and J. Schmidhuber. Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In *Proc. of the 8th Conference on Intelligent Autonomous Systems*, 2004.
- [3] J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.
- [4] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Transition-independent decentralized markov decision processes. In *Proc. of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 41–48, New York, NY, USA, 2003. ACM Press.
- [5] T. Deselaers, D. Keysers, and H. Ney. Features for image retrieval: A quantitative comparison. In *DAGM 2004, Pattern Recognition, 26th DAGM Symposium*, pages 228–236, 2004.
- [6] T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.
- [7] B. Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3):251–321, 1985.
- [8] J. McQueen. Some methods for the classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Math. Stat., Prob.*, number 1, pages 281–297, 1967.
- [9] R. Parr and S. Russell. Reinforcement learning with hierarchies of machines. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, pages 1043–1049. MIT Press, 1998.
- [10] Y. Rui, T. S. Huang, and S. Mehrotra. Relevance feedback techniques in interactive content-based image retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 25–36, 1998.
- [11] M. P. Wellman. A market-oriented programming environment and its application to distributed multi-commodity flow problems. *Journal of Artificial Intelligence Research*, 1:1–23, 1993.