Adapting Automatic Image Annotation via Meta-learning

Ritendra Datta, Jai Li, and James Z. Wang

Department of Computer Science and Engineering
Technical Report
CSE-09-005

March 10, 2009
Adapting Automatic Image Annotation via Meta-learning

Ritendra Datta
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802, USA

Jia Li
Department of Statistics
The Pennsylvania State University
University Park, PA 16802, USA

James Z. Wang
College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802, USA

Abstract

Automatic image annotation is the task of learning to associate images with semantically meaningful tags. Virtually every previous approach has made the assumptions that ground-truth image tags come from a fixed vocabulary, are absolute over time, and are universally acceptable by various people. However, it is reasonable to believe and empirically demonstrable that these rigid assumptions make the models misrepresent the real world. In this paper, we explore learning algorithms for adapting automatic image annotation to different scenario changes. We propose a meta-learning framework called PLMfIT, which can augment a black-box annotation model to help provide the requisite adaptability to handle contextual changes, time evolution, and personalization. Our algorithms are inspired by inductive transfer and attempt to harness all available information, including the black-box model’s performance, previously assigned user tags, the image representations, and the WordNet ontology. Instead of re-training expensive annotation models, adaptability is achieved through efficient incremental learning of only the computationally lightweight meta-learning component. We conduct experiments with standard image datasets as well as large, real-world datasets. The strongly positive results suggest that meta-learning and inductive transfer are very promising approaches for this problem domain.

Keywords: Meta-learning, incremental learning, tagging, concept drift, personalization

1. Introduction

A large fraction of images on the Web exists without meta-data describing semantics, or with unreliable meta-data. Due to a lack of correspondence between low-level visual features and their semantics (Smeulders et al., 2000), meaningful search and organization of this large fraction remains elusive. Automatic image annotation\(^1\) is the task of producing tags for images based on their visual content. In the context of machine learning, automatic

---

\(^1\) The phrases ‘image annotation’ and ‘image tagging’ will be used interchangeably throughout this paper.
annotation falls into the class of learning tasks that involve making multiple binary decisions on each data point. If we could generate comprehensive, accurate, semantically meaningful tags for images, it would bring image organization up to roughly the level of text documents. Of late, many image annotation ideas have been proposed (Barnard et al., 2003; Carneiro et al., 2007; Chang et al., 2003; Feng et al., 2004; Gao et al., 2006; Jin et al., 2004; Li and Wang, 2008; Li et al., 2006; Liu and Tang, 2005; Monay and Gatica-Perez, 2003; Wang et al., 2006, 2008; Wong and Leung, 2008; Wu and Yang, 2006). Virtually all propositions are based on supervised learning, and take roughly the following form, (a) use a limited set of manually tagged images to train generative or discriminative models for associating visual features to tags, and (b) given a new image, use the model inferences on its visual content to assign a variable-size set of tags from its limited vocabulary. Performance is typically reported based on a training/testing split of one or more manually tagged image datasets. However, this may not accurately represent real-world settings for automatic tagging.

We argue for a fundamentally different notion of automatic image annotation in real-world settings involving users, whereby the goal is mainly to mimic the users in the tagging process as closely as possible. To elaborate, let us take the case of Yahoo! Flickr (2005), an online photo-sharing Website where collaborative image tagging, also referred to as *folksonomic tagging*, plays a key role in making the image collections meaningful, organizational by semantics and searchable by text (Volkmer et al., 2005). It is also an apt service platform for automatic image tagging. If the goal of the tagging system is to reduce human effort, or to assist humans tag photos more accurately and/or with less effort (e.g., by automatically suggesting tags to choose from), then a natural performance metric will be how closely the automatically generated tags mimic user-generated ones for the same images. If the goal is indeed to maximize performance defined in this manner, then it is easy to see how the standard procedure of splitting a dataset into training and test, and computing performance on the test set, can be misleading. What is problematic is the assumption that training and test cases are sampled from the same underlying distribution in the joint image-tag space. There are at least three factors that can affect the test time image-tag distribution and hence the generalization of model training to, say, a Flickr tagging scenario:

- **Context**: The Flickr users may tag the same photos differently than the users who tagged the training dataset. Some visual elements may be more commonly tagged in a different language, a local dialect, or slang, rather than their correct English language descriptors. The types of images uploaded by the users may also be different.

- **Time evolution**: The kinds of photos uploaded and the nature of user tags may evolve with time for various reasons, including news, current affairs, and changing political situations. Trends may spread through the network of users, and over time, get more pervasive. This trend of evolution over time has been observed in photo-sharing systems and reported in (Datta et al., 2007b; Dubinko et al., 2007).

- **User/community preferences**: The kinds of photos uploaded, the tags given to them, or the frequency distribution over tags, may vary to a great extent across individuals and small communities. This is particularly evident in Flickr, where the user has the option of choosing from her own set of previously used tags (Garg and Weber, 2008), thereby promoting locality in the tag space.
In the remainder of this paper, we will refer to all three of these commonly as a changed image tagging environment (ITE). In this sense, an annotation system which has the ability to adapt to changes in ITE can potentially be effective in all three scenarios.

1.1 A Challenging Setting for Learning

Given that generic images vary widely in their composition, learning a mapping function from the image space to the tag space is extremely challenging, as has been found in previous studies (Datta et al., 2008; Smeulders et al., 2000). Most proposed image annotation approaches therefore resort to complex statistical learning models in efforts to learn semantics from visual content, which involve computationally expensive training. We are unlikely to be able to reduce the complexity of these models significantly. On the other hand, we require efficient annotation adaptability to various ITE changes. If the ITE change involves expanding the vocabulary to include concepts that were trained under different tag names, we also wish to take advantage of the previously learned knowledge.

One obvious way to adapt an automatic annotation system to a changed ITE would be to re-train it using data sampled from that ITE. Unfortunately, with most systems this can take hours or even days (Barnard et al., 2003; Li and Wang, 2003), which means that for dynamically changing ITE, a significant amount of latency will be introduced, and training resources will stay blocked on a continual basis. When changes are frequent or the problem is scaled up, it will be impossible to keep up with the changes this way. Furthermore, most annotation systems are not well suited for incremental learning. We thus require a learning system with a unique need; given a core learning algorithm that is expensive to train, we require an augmentation which can quickly adapt to various kinds of ITE changes in a scalable manner, while taking advantage of previously learned knowledge. Additionally, ITE changes over short periods are typically localized, which presents an opportunity to train incrementally rather than having to re-train over entire datasets at each change point.

1.2 Overview of Our Approach

To meet the aforementioned needs, we propose a meta-learning layer above the core learning system which can adapt to evolving ITE incrementally, in a light-weight manner, without requiring a re-training of the core system. Suppose there is a core annotation black-box that, by some means (e.g., a one-time learning process), can analyze the visual content of images and generates sets of tags. Expectedly, these tags come from a fixed size vocabulary, and the association of visual features with these tags is intrinsic to the particular set of training images used. Our goal is to expand the capabilities of the black-box to allow adaptability to various kinds of ITE changes while maximizing the use of previously learned knowledge.

In order to make a case for meta-learning, let us draw analogy with a robot learning to solve math problems. Suppose robot X has initially learned to solve a fundamentally important pool of math problems. In this process, its fundamentals have been cleared and it has developed critical but limited skills. This X is the black-box system, and initial problem pool is an ITE. Now, X is required to solve harder, more diverse problems. Here, the problems are sampled from a larger source, and the same tactics may not work. We thus have a changed ITE, and X needs to get adapted to the new conditions and challenges. It may be too expensive to re-train the robot on an all-inclusive set of problems all over again,
and even that training may not be sufficient to solve a continuously evolving problem pool. Instead, if we have another robot $Y$ which is aware of the early-stage training of $X$, observes its response to new problems, analyzes its mistakes and is able to rectify it eventually by taking the output of $X$ and produces a new one, the combined $X + Y$ system can likely be more powerful. The assumption here is that the output of $X$ is still fundamentally in the right direction, and only needs refinement and perhaps a better representation of the solution. The knowledge acquired by $X$ can be used to solve problems in a new setting, similar in principle to inductive transfer or transfer learning. Thus, $Y$ here is equivalent to the meta-learning framework, in the sense that it sits and observes $X$, learns from mistakes, and eventually improves upon the output of $X$.

In terms of image tagging, the scenario is as follows. We have a fundamentally grounded black-box (which we assume to be any annotation system proposed and found to perform moderately), a meta-learning framework which we call PLMFFIT, which, for some fraction of the images in a changed ITE, can observe the tags generated by the black-box, as well as the ground-truth tags in those cases, and quickly learn to adapt to it. The change in ITE can be any of (a) change in context, (b) evolution over time, or (b) change in the people who the images belong to or are tagged by. Because the observations are received differently for the three cases, the PLMFFIT training process varies as well. For a change in context, we can train the PLMFFIT one-time using a batch of image samples collected from the new ITE. For changes over time, PLMFFIT can learn incrementally as new images get uploaded, assessed by the black-box, and tagged by users. For a particular person’s uploaded images, PLMFFIT can be trained on her previously tagged pool of images, if such a pool exists. We therefore first need a formulation for the meta-learning component PLMFFIT, and then the algorithms for training it to achieve adaptability under varied settings.

1.3 Related Work

There is a wealth of machine learning literature on meta-learning, incremental learning, and inductive transfer, concepts that our work is directly related to, although we are unaware of the use of these techniques for image tagging or related problem areas. Here, we briefly discuss literature most pertinent to this work. The term meta-learning has historically been used to describe the learning of meta-knowledge about learned knowledge. Research in meta-learning covers a wide spectrum of approaches and applications, as presented in a review (Vilalta and Drissi, 2002). One of the most popular meta-learning approaches, boosting is widely used in supervised classification (Freund and Schapire, 1996). Boosting involves iteratively adjusting weights assigned to data points during training, to adaptively reduce misclassification. In stacked generalization, weighted combinations of responses from multiple learners are taken to improve overall performance (Wolpert, 1992). The goal here is to learn optimal weights using validation data, in the hope of generalization to unseen data. Another research area under the meta-learning umbrella that bears relevance to our work is inductive transfer. Research in inductive transfer is grounded on the belief that knowledge assimilated about certain tasks can potentially facilitate the learning of certain other tasks (Caruana, 1997). A recent workshop (Silver et al., 2005) at the NIPS conference was devoted to discussing advances and applications of inductive transfer. Incremental learning deals with adapting predictions to contextual changes as new data enters the system.
Adapting to radical contextual changes via incremental learning was proposed by Widmer (1997). Incrementally learning support vectors as and when training data is encountered has been explored as a scalable supervised learning procedure by Cauwenberghs and Poggio (2001). The idea of partial instance memory, whereby only a relevant subset of the incoming stream of training samples are maintained (thereby saving memory) and used for incremental learning, was proposed and shown to be empirically effective (Maloof and Michalski, 2004). Authors Kolter and Maloof proposed a weighted ensemble of incremental learners for concept drift (Schlimmer and Granger, 1986), a formalization of the idea that concepts to learn change over time (Kolter and Maloof, 2007). In the case of image tagging, it is the mapping between tags and visual semantics that drifts over time.

Research in automatic image annotation can be roughly categorized into two different ‘schools of thought’: (1) Words and visual features are jointly modeled to yield compound predictors describing an image or its constituent regions. The words and image representations used could be disparate (Feng et al., 2004; Jin et al., 2004) or single vectored representations of text and visual features (Monay and Gatica-Perez, 2003; Liu and Tang, 2005). (2) Automatic annotation is treated as a two-step process consisting of supervised image categorization, followed by word selection based on the categorization results (Chang et al., 2003; Li and Wang, 2008; Carneiro et al., 2007). While the former approaches can potentially label individual image regions, ideal region annotation would require precise image segmentation, an open problem by itself in computer vision. While the latter techniques cannot label regions, they are typically more scalable to large image collections. Though less relevant to our work, approaches such as Wang et al. (2008) employ yet another philosophy for image tagging, i.e., to avoid learning and hence stay ‘model-free’. Specifically within the machine learning community, significant recent work in the domain of image categorization and annotation include Barnard et al. (2003), who proposed the use of latent Dirichlet allocation (Blei et al., 2003) for the purpose of associating images and tags, Chen and Wang (2004), who proposed a multiple-instance learning approach to image categorization, and Fleuret and Geman (2008), who explore the use of stationary visual features for the detection of cats in gray-scale images.

Evolution of image tags over time, or their variation across people in online communities, have only recently begun to get research focus. Researchers at Yahoo! have studied the problem of visualizing the evolution of salient tags popular among Flickr users (Dubinko et al., 2007). Assuming such an evolution of tags over time in general, Datta et al. (2007b) proposed a meta-learning approach to handling the evolution as part of an automatic image tagging system, and found the approach to be effective on traces obtained from the Alipr system (Alipr, 2006; Li and Wang, 2008). Image tag recommendation strategies for Flickr users was proposed by Sigurbjörnsson and van Zwol (2008) and found to be effective. On similar lines, personalized image tag recommendation was briefly explored by Garg and Weber (2008) using tag co-occurrences and tag history, but neither approach involved the exploitation of visual content of images for tagging purposes.

1.4 Our Contributions

Broadly speaking, the main contributions of this work are learning approaches and algorithms that greatly improve automatic image tagging in real-world settings. We take a
complex learning task, that of finding a mapping function from low-level image features to semantically meaningful tag sets, and create an appropriate meta-learner around it which is well-suited to incremental training, and designed to meet the needs of real-world usage. To the best of our knowledge, this is the first attempt at meta-learning and incremental learning for image tagging. Moreover, because image tagging is not simply formulated as a classification problem, existing meta-learning and incremental learning methods cannot be applied directly. We thus design a new statistical modeling approach, which also aims to achieve efficiency in real-world deployment. Specific contributions are summarized below.

- We propose a principled, lightweight, meta-learning framework for image tagging (PLMFIT), based on few simplifying assumptions, inspired by inductive transfer, and backed by experiments, which can augment any existing annotation system. This is the basic component that allows adaptation to ITE changes. Experimentally, we find that PLMFIT can adapt to new contexts\(^2\) very effectively, showing dramatic performance improvement over the core systems.

- We propose algorithms to make use of PLMFIT to adapt to concept drift (Schlimmer and Granger, 1986), or changes in ITE over time. Specifically, we propose fast algorithms for incremental/decremental learning over time that take advantage of the simplicity of the PLMFIT formulation. Two different memory models for incremental learning, persistent and transient, are explored. Experimentally, we find the algorithms to be highly effective in adapting over time on a real-world dataset.

- We propose algorithms for personalized tagging, adapting to ITE changes across people, assuming that some of their photos are already tagged. As part of this, we propose an approach to allow expansion of the tag vocabulary beyond the initial set, and incrementally updating model parameters for new users. Experiments on actual user data show that personalization greatly boosts tagging performance.

- Throughout, we use real-world data to justify each aspect of our approach, as well as to show that assumptions of changing ITE over time and across people hold true.

In all these cases, we assume the existence of a black-box annotation system that exhibits better-than-random performance. This is the only assumption we make about the black-box system, so most previously reported annotation algorithms qualify. Experiments are conducted using two different annotation systems. The datasets used for the experiments include the popular Corel image, two real-world image traces and user-feedback obtained from the Alipr system (Alipr, 2006), and a large set of collaboratively tagged images obtained from Yahoo! Flickr (2005), spanning hundreds of users.

The rest of this paper is arranged as follows. The technical details of PLMFIT are presented in Sec. 2, including model estimation and smoothing steps. In Sec. 3 we present the algorithms for using PLMFIT for adaptation over context, time, and people. Experimental results are presented in Sec. 4. We discuss results, limitations, and implications in Sec. 5. We conclude in Sec. 6.

\(^2\) One can think of a contextual change as a global change of ITE, e.g., when a model is trained using labeled Corel images, but applied to Flickr tagging.
2. PLMFIT: A Principled Meta-learning Framework for Image Tagging

In this section, we describe PLMFIT, our meta-learning framework that forms the backbone of this work. Consider any black-box image annotation system, such as (Barnard et al., 2003; Carneiro et al., 2007; Chang et al., 2003; Feng et al., 2004; Gao et al., 2006; Jin et al., 2004; Li and Wang, 2008; Li et al., 2006; Liu and Tang, 2005; Monay and Gatica-Perez, 2003; Wang et al., 2006; Wu and Yang, 2006; Wong and Leung, 2008), that takes an image as input and guesses one or more tags as its annotation. This paper does not deal with the algorithm behind the annotation black-box, but simply assumes that it captures, from visual analysis, the semantics of the images in the form of tags, to a better-than-random degree of reliability. Now let us assume that for a certain set of images not used in training the black-box, ground-truth tags are available. Clearly, for each such image, two sets of tags are available, (1) the ground-truth tags, and (2) the tags predicted by the black-box. Let us also assume that we have at our disposal a knowledge base such as WordNet (Miller, 1995), and the original images from which we can independently extract visual features. A high-level overview of a meta-learning framework that incorporates these components is show in Fig. 1.

As part of motivating the model, in what follows, we will graphically represent empirical evidence to support its various components. For this purpose, we have used 10,000 images randomly chosen from the Alipr dataset which is described in detail in Sec. 4. Each image is accompanied by human-provided tags as well as the corresponding machine predictions (Li and Wang, 2008; Alipr, 2006). In the remainder of this section, we begin with notation and an overview of the basic components of PLMFIT, then describe its formulation, and finally present estimation steps.
2.1 Notation and Basics

Let the black-box annotation system be known to have a tag vocabulary denoted by $V_{bbox}$. For a given ITE, let us denote the ground-truth vocabulary by $V_{gt}$. Let the full vocabulary of interest, $V$, be their union, i.e., $V = (V_{bbox} \cup V_{gt}) = \{w_1, \ldots, w_K\}$, where size $K = |V|$. Given an image $I$, the black-box predicts a set of tags to be its correct annotation. We introduce indicator variables $G_w \in \{0, 1\}, w \in V$, to denote if a guess $w$ is predicted or not. Similarly, for ground-truth, let $A_w \in \{0, 1\}$ denote the whether $w$ is a correct tag or not. The black-box can be denoted by a function $f_{bbox}$ mapping an image $I$ to a set of indicator variables, i.e., $f_{bbox}(I) = \{G_{w_1}, \ldots, G_{w_K}\}$, and ground-truth can be denoted analogously by function $f_{gt}(I) = \{A_{w_1}, \ldots, A_{w_K}\}$.

Regardless of the abstraction of visual content that the black-box uses for annotation, the pixel-level image representation is still available to the meta-learner. If some visual features, which can be cheaply extracted and hence are suitable for highly efficient incremental modeling, represent a different abstraction than what the black-box uses, they can help form a a different 'viewpoint' and thus can potentially complement semantics recognition. Suppose we have a D-dimensional vector representation for such visual features extracted from an image, denoted by $f_{vis}(I) = (h_1, \ldots, h_D)$. Note that though non-vector visual representations (e.g., variable sized sets of features) can be more powerful representations, we use this form here for computational advantages.

Furthermore, the English language semantic lexicon WordNet, which has been previously found useful for image one that has been found useful for automatic tagging (Jin et al., 2005; Datta et al., 2007a), is also available at our disposal. In particular, WordNet-based semantic relatedness measures (e.g., Leacock and Chodorow, 1998) have benefited annotation tasks. While this is a potentially useful external knowledge base, it is rendered useless for non-English words, proper nouns, contemporary slang, and incorrect usages that are commonly found among user tags. However, in cases for which WordNet based relatedness measures can be computed, we show how the transfer of learned knowledge can be better directed.

2.2 Model Formulation

We first describe the PLMFIT formulation, and then present the estimation steps. For each image $I$, a decision is taken on each word independently, based on all available information. To do so, we compute the following odds in favor of each word $w_j \in V$ to be a ground-truth tag, conditioned on pertinent information:

$$
\ell_{w_j}(I) = \frac{Pr(A_{w_j} = 1 \mid f_{bbox}(I), f_{vis}(I))}{Pr(A_{w_j} = 0 \mid f_{bbox}(I), f_{vis}(I))}
$$

In what follows, we will make simplifying assumptions to make this formulation tractable. Note that here $f_{bbox}(I)$ (and similarly, the other terms) denotes a joint realization of the corresponding random variables given the image $I$. Using Bayes' Rule, we can re-write:

$$
\ell_{w_j}(I) = \frac{Pr(A_{w_j} = 1, f_{bbox}(I), f_{vis}(I))}{Pr(A_{w_j} = 0, f_{bbox}(I), f_{vis}(I))}
$$
If realization of variable $A_{w_j}$ is denoted by $a_j \in \{0, 1\}$ and that of variables $G_{w_i}$ for each word $w_i$ are denoted by $g_i \in \{0, 1\}$, then using the chain rule of probability, and without loss of generality, we can re-write the following:

$$Pr\left(A_{w_j} = a_j, f_{\text{box}}(I), f_{\text{vis}}(I)\right)$$

$$= Pr\left(G_{w_j} = g_j, A_{w_j} = a_j, \bigcap_{i \neq j}(G_{w_i} = g_i), f_{\text{vis}}(I)\right)$$

$$= Pr\left(G_{w_j} = g_j\right) \times Pr\left(A_{w_j} = a_j \mid G_{w_j} = g_j\right)$$

$$\times Pr\left(\bigcap_{i \neq j}(G_{w_i} = g_i), A_{w_j} = a_j, G_{w_j} = g_j\right)$$

$$\times Pr\left(f_{\text{vis}}(I) \mid \bigcap_{i \neq j}(G_{w_i} = g_i), A_{w_j} = a_j, G_{w_j} = g_j\right)$$

The odds in Eq. 1 can now be factored using Eq. 2 and 3:

$$\ell_{w_j}(I) = \frac{Pr(A_{w_j} = 1 \mid G_{w_j} = g_j)}{Pr(A_{w_j} = 0 \mid G_{w_j} = g_j)}$$

$$= \frac{Pr(\bigcap_{i \neq j}(G_{w_i} = g_i) \mid A_{w_j} = 1, G_{w_j} = g_j) \times Pr(\bigcap_{i \neq j}(G_{w_i} = g_i) \mid A_{w_j} = 0, G_{w_j} = g_j) \times Pr(f_{\text{vis}}(I) \mid A_{w_j} = 1, \bigcap_{i \neq j}(G_{w_i} = g_i), G_{w_j} = g_j) \times Pr(f_{\text{vis}}(I) \mid A_{w_j} = 0, \bigcap_{i \neq j}(G_{w_i} = g_i), G_{w_j} = g_j)}{Pr(A_{w_j} = 1 \mid G_{w_j} = g_j)}$$

Note that the prior ratio $Pr(G_{w_j} = g_j)$ is 1, and hence is eliminated. The ratio $\frac{Pr(A_{w_j} = 1 \mid G_{w_j} = g_j)}{Pr(A_{w_j} = 0 \mid G_{w_j} = g_j)}$ is a sanity check on the black-box for each word. For $g_j = 1$, it can be paraphrased as “Given that word $w_j$ is guessed by the black-box for $I$, what are the odds of it being correct?”. Naturally, a higher odds indicates that the black-box has greater precision in guesses (i.e., when $w_j$ is guessed, it is usually correct). A similar paraphrasing can be done for $g_j = 0$, where higher odds implies lower word-specific recall in the black-box guesses. A useful annotation system should be able to achieve independently (word-specific) and collectively (overall) good precision and recall. These probability ratios therefore give the meta-learner indications about the black-box model’s strengths and weaknesses over its entire vocabulary. In Fig. 2, we have plotted empirical estimates of these probability terms for frequently occurring tags.

When $g_j = 1$, the ratio $\frac{Pr(A_{w_j} = 1 \mid G_{w_j} = g_j)}{Pr(A_{w_j} = 0 \mid G_{w_j} = g_j)}$ relates each correctly or wrongly guessed word $w_j$ to how every other word $w_i, i \neq j$ is guessed by the black-box. This component has strong ties with the concept of co-occurrence popular in the language modeling community, the difference being that here it models the word co-occurrence of the black-box’s outputs with respect to ground-truth. Similarly, for $g_j = 0$, it models how certain words do not co-occur in the black-box’s guesses, given the ground-truth. Since the meta-learner makes decisions about each word independently, it is intuitive to separate them out in this ratio as well. That is, the question of whether word $w_i$ is guessed or not, given that another word $w_j$ is correctly/wrongly guessed, is treated independently. Furthermore,
efficiency and robustness become major issues in modeling joint probability over a large number of random variables, given limited data. Considering these factors, we assume the guessing of each word $w_i$ conditionally independent of each other, given a correctly/wrongly guessed word $w_j$, leading to the following approximation:

$$Pr\left(\bigcap_{i \neq j}(G_{w_i}=g_i) \mid A_{w_i} = a_i, G_{w_j} = g_j\right) \approx \prod_{i \neq j} Pr(G_{w_i}=g_i \mid A_{w_i} = a_j, G_{w_j} = g_j)$$

The corresponding ratio term can then be written as

$$\frac{Pr(\bigcap_{i \neq j}(G_{w_i}=g_i) \mid A_{w_j} = 1, G_{w_j} = g_j)}{Pr(\bigcap_{i \neq j}(G_{w_i}=g_i) \mid A_{w_j} = 0, G_{w_j} = g_j)} = \prod_{i \neq j} \frac{Pr(G_{w_i}=g_i \mid A_{w_i} = 1, G_{w_j} = g_j)}{Pr(G_{w_i}=g_i \mid A_{w_i} = 0, G_{w_j} = g_j)}$$

A conditional multi-word co-occurrence model has been effectively transformed into that of pairwise co-occurrences, which is attractive in terms of modeling, estimation, and efficiency. While co-occurrence really happens when $g_i = g_j = 1$, the other combinations of values can also be useful, e.g., how the frequency of certain word pairs not being both guessed differs according to the correctness of these guesses. The contributions of the component ratio terms, namely $Pr(G_{w_i}=g_i \mid A_{w_j} = 1, G_{w_j} = g_j)$, can be understood by intuition. For the purpose of illustration, a visual plot of the ratio $\frac{Pr(G_{w_i}=1 \mid A_{w_j} = 1, G_{w_j} = 1)}{Pr(G_{w_i}=1 \mid A_{w_j} = 0, G_{w_j} = 1)}$ for a set of 30 most frequently occurring tags in the Alipr dataset is presented in Fig. 3. The following examples provide further insights:

- Some scene elements that are visually similar can often be mistakenly confused among themselves by the black-box. For example, if ‘sky’ is guessed for an image, and it is
a correct guess more often when 'water' is also guessed than when it is not, then the ratio will have a high value for \( w_i = 'water', g_i = 1, w_j = 'sky', g_j = 1 \). Here, the black-box prediction of one scene element ('water') reinforces belief in the existence of another ('sky') in the scene (See Fig. 3, location A).

- For some word \( w_j \), the black-box may not have learned anything due to lack of good training images, inability to capture apt visual properties, or simply its absence in \( V_{bbox} \). For example, consider images where ground-truth is \( g_j = 'feline' \) but black-box regularly guesses \( w_i = 'cat' \), only the latter being in its vocabulary. Here, \( g_j = 0 \) always, and the above ratio is high for \( g_i = 1 \). Here, the training on one tag induces guesses at another tag in the vocabulary (see also Fig. 3, location B).

Figure 3: Visualization of ratio \( \frac{Pr(\hat{G}_{wj} = 1 | A_{wj} = 1, G_{wj} = 1)}{Pr(\hat{G}_{wj} = 1 | A_{wj} = 0, G_{wj} = 1)} \) as obtained empirically with images from the Alipr dataset (see Sec. 4) for 30 most frequently occurring tags. Two interesting cases, that highlight the importance of these terms to meta-learning effectiveness, are marked. For example, the value at location (A) can be read as the ratio of probabilities of 'water' being guessed for an image by the black-box given that 'sky' is also guessed, correctly versus incorrectly.

Finally, \( \frac{Pr(I_{vis}(I)|A_{wj} = 1, \bigwedge_{k \neq j}(G_{wj} = g_k, G_{wj} = g_j))}{Pr(I_{vis}(I)|A_{wj} = 0, \bigwedge_{k \neq j}(G_{wj} = g_k, G_{wj} = g_j))} \) in Eq. 4 can be simplified, since \( f_{vis}(I) \), being the meta-learner's own visual representation, is independent of the black-box's visual...
abstraction. Therefore, we can re-write

$$\frac{Pr(f_{vis}(I)|A_{w_j}=1, \neg)}{Pr(f_{vis}(I)|A_{w_j}=0, \neg)} \approx \frac{Pr(h_1, \ldots, h_D|A_{w_j}=1)}{Pr(h_1, \ldots, h_D|A_{w_j}=0)}$$

(5)

which is the ratio of conditional probabilities of the meta-learner’s visual features extracted. We can think of this as a second, highly simplified image recognition black-box, that can be computed efficiently, and as described in Sec. 3.2, is suitable for incremental learning. In our experiments we have used LUV color space based histograms as features, described in Sec. 2.3. The main idea is that for some classes of images (see Fig. 8), even such simple features can add to performance without adding to complexity.

Putting the pieces together, and taking logarithm to get around issues of machine precision, we can re-write Eq. 4 as a logit:

$$\log \ell_{w_j}(I) = \log \frac{Pr(A_{w_j}=1 | G_{w_j}=g_j)}{1 - Pr(A_{w_j}=1 | G_{w_j}=g_j)} + \sum_{i \neq j} \log \frac{Pr(G_{w_i}=g_i | A_{w_j}=1, G_{w_j}=g_j)}{Pr(G_{w_i}=g_i | A_{w_j}=0, G_{w_j}=g_j)}$$

$$+ \log \frac{Pr(h_1, \ldots, h_D | A_{w_j}=1)}{Pr(h_1, \ldots, h_D | A_{w_j}=0)}$$

(6)

This logit is essentially the backbone of PLMFIT inference, where a higher value for a tag indicates greater support for its selection, for the image $I$. The final tags for image $I$ can then be based on any of the following selection methods:

- **Top r**: After ordering all words $w_j \in V$ in the increasing magnitude of $\log \ell_{w_j}(I)$ to obtain a rank ordering, we annotate $I$ using the top $r$ ranked words.

- **Threshold r%**: We can annotate $I$ by thresholding at the top $r$ percentile of the range of $\log \ell_{w_j}(I)$ values for the given image over all the words.

- **User Average**: For the personalization case, since there may be trends in the number of tags a particular user’s images have, we can compute the average number $r$ of tags in a given user’s training samples, and predict top $r$ tags for that user’s test cases.

In our experiments reported in Sec. 4, we have used all three methods and shed light on their performance differences, wherever applicable.

### 2.3 Model Estimation

For a given image $I$, predictions are first made by the black-box annotation system, which are then used for evaluating Eq. 6, to finally propose a set of tags. To facilitate evaluation of the equation, the probability terms need to be estimated using training data that mimics an ITE in question. These terms are estimated and indexed on, e.g., variables $A_{w_j}$ and $G_{w_j}$, for efficient evaluation. Let us consider the estimation of each term separately, given a training set of size $L$ for a particular ITE, consisting of images $\{I^{(1)}, \ldots, I^{(L)}\}$, the corresponding tags guessed by the black-box, $\{h_{box}(I^{(1)}), \ldots, h_{box}(I^{(L)})\}$, and the actual ground-truth tags, $\{f_{gt}(I^{(1)}), \ldots, f_{gt}(I^{(L)})\}$. To make the system lightweight, all estimation is based on empirical frequencies, making computation times deterministic.
The term $Pr(A_{w_j} = 1 \mid G_{w_j} = g_j)$ in Eq. 6 can be estimated from the size $L$ training data as follows:

$$\tilde{Pr}(A_{w_j} = 1 \mid G_{w_j} = g_j) = \frac{\sum_{n=1}^{L} I\{G_{w_j}^{(n)} = g_j \& A_{w_j}^{(n)} = 1\}}{\sum_{n=1}^{L} I\{G_{w_j}^{(n)} = g_j\}}$$  \hspace{1cm} (7)

Here, $I(\cdot)$ is the indicator function. A natural issue of robustness arises when the training set contains few or no samples for $G_{w_j}^{(n)} = 1$. Therefore, we perform interpolation-based smoothing using

$$\tilde{Pr}(A_{w_j} = 1 \mid G_{w_j} = g_j) = \begin{cases} \tilde{Pr}_{prior}(g_j) & m \leq 1 \\ \frac{1}{m} \tilde{Pr}_{prior}(g_j) + \frac{m}{m+1} \tilde{Pr}(A_{w_j} = 1 \mid G_{w_j} = g_j) & m > 1 \end{cases}$$

where $m = \sum_{n=1}^{L} I\{G_{w_j}^{(n)} = g_j\}$, the number of instances out of $L$ where $w_j$ is guessed (or not guessed, depending upon $g_j$), and the prior $\tilde{Pr}_{prior}(g_j)$ is estimated using

$$\tilde{Pr}_{prior}(g_j) = \frac{\sum_{i=1}^{K} \sum_{n=1}^{L} I\{G_{w_i}^{(n)} = g_j \& A_{w_i}^{(n)} = 1\}}{\sum_{i=1}^{K} \sum_{n=1}^{L} I\{G_{w_i}^{(n)} = g_j\}}$$  \hspace{1cm} (8)

where $g_j \in \{0, 1\}$. The prior term needs some explanation. For $g_j = 1$, $\tilde{Pr}_{prior}(g_j) = \frac{\# correct\ tag\ predictions\ overall}{\# tag\ predictions\ overall}$. Thus, this prior stands for the probability that whenever an arbitrary tag is predicted, what the chances are of it being correct. Similarly, for $g_j = 0$, the prior stands for the probability that an arbitrary tag, when not guessed, is actually correct. Given the total lack of training samples for a tag, these priors are optimal estimates.

The probability term $Pr(G_{w_j} = g_i \mid A_{w_j} = 1, G_{w_j} = g_j)$ in Eq. 6 can be estimated using the empirical frequency ratio

$$\tilde{Pr}(G_{w_i} = g_i \mid A_{w_j} = 1, G_{w_j} = g_j) = \frac{\sum_{n=1}^{L} I\{G_{w_i}^{(n)} = g_i \& G_{w_j}^{(n)} = g_j \& A_{w_j}^{(n)} = 1\}}{\sum_{n=1}^{L} I\{G_{w_j}^{(n)} = g_j\}}$$  \hspace{1cm} (9)

In this case, robustness is more critical, because many word pairs may appear together very infrequently among the black-box's guesses. Here, we describe and justify using the knowledge base WordNet (Miller, 1995) for robust smoothing of these probability estimates. If the vocabulary consists only of semantically meaningful tags such that WordNet-based word relatedness metrics are defined for all word pairs within it, we can take advantage of it to perform a different form of inductive transfer. Similarity based smoothing (Dagan et al., 1999), a method commonly used in word pair co-occurrence modeling, is appropriate here. Given a WordNet-based similarity measure $W(w_i, w_j)$ between $w_i$ and $w_j$, we smooth the frequency based estimates as follows:

$$\tilde{Pr}(G_{w_i} = g_i \mid A_{w_j} = 1, G_{w_j} = g_j) = \theta \cdot \tilde{Pr}(G_{w_i} = g_i \mid A_{w_j} = 1, G_{w_j} = g_j)$$

$$+ (1 - \theta) \cdot \sum_{k \neq j} \frac{W(w_j, w_k)}{Z} \tilde{Pr}(G_{w_i} = g_i \mid A_{w_k} = 1, G_{w_k} = g_k)$$  \hspace{1cm} (10)
where $Z$ is the normalization factor. A choice of $\theta = 0.5$ ensures that a particular smoothed estimate is 50% dependent on the original estimate, and is found to work well in our experiments. The remainder of the contribution comes from other tags weighted by semantic relatedness. In effect, for poorly estimated terms, probability estimates over semantically related terms 'substitute' for each other as part of smoothing.

Figure 4: Network depicting WordNet-based relations among the 60 most frequently occurring tags in the Alipr dataset (Sec. 4). An edge between a pair of words indicates that the relatedness measure LCH (Leacock and Chodorow, 1998) exceeds 1.7, roughly the mid-point of its [0.37, 3.58] range of values.

Let us elaborate further on the use of WordNet for smoothing, and its role in inductive transfer. The function $W(\cdot, \cdot)$ stands for the Leacock and Chodorow (LCH) word relatedness measure (Leacock and Chodorow, 1998), which takes values between 0.37 and 3.58, higher value meaning more semantically related. If we defined a pair of words $w_i$ and $w_j$ to be semantically related if $W(w_i, w_j) \geq 1.7$, then we would get a relatedness network among the most frequently occurring tags in the Alipr dataset as shown in Fig. 4. We observe that while some relationships make little sense, much of the network is meaningful, and hence the LCH measure can be generally trusted. The role played by this measure, based on Eq. 10, is to effectively transfer learned knowledge (estimates) among semantically related tags. To further validate that such knowledge transfer is practical, we reverse-engineered the problem by computing the averaged out absolute differences between the pre-smoothed empirical probability estimates among tags, and using a threshold to connect tag pairs with low average differences. The resultant network is shown in Fig. 5. Though not depicting the same underlying aspects, comparison with the network in Fig. 4 reveals interesting overlaps. In Fig. 6, we present a more direct attempt at assessing whether the use of semantic relatedness leads to better 'substitution' of estimates. We see that the use of strongly related tags lead to better estimates than weakly related tags for almost all cases.
Figure 5: Empirical evidence, based on Alipr dataset (Sec. 4) that WordNet can help with inductive transfer. Networking depicting proximity of pre-smoothed estimates of \( \Pr(G_{w_i}=1|A_{w_j}=a, G_{w_j}=1) \) for pairs of tags among the top 60 most frequently occurring ones. Specifically, there is an edge between a pair of tags \( w_{i1} \) and \( w_{j2} \) if
\[
\{\frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} [\Pr(G_{w_i}=1|A_{w_{j1}}=a, G_{w_{j1}}=1) - \Pr(G_{w_i}=1|A_{w_{j2}}=a, G_{w_{j2}}=1)]\}
\]
for \( a = 0 \) and 1 are both below 0.0025 (chosen to generate a less-cluttered but interesting graph). Compared to Fig. 4, while we see many overlaps, there are quite a few differences as well, by virtue of what the relations stand for.

When the vocabulary consist of many non-WordNet tags, such as slang, phrases, proper nouns, and foreign language terms, e.g., in the case of Flickr, the relatedness metric cannot be used. In this case we set \( W(w_i, w_j) \) to 1 in Eq. 10 for all cases, and normalize accordingly. When \( g(\cdot, \cdot) \) cannot be estimated due to lack of samples, a prior probability estimate, computed as in the previous case, is used in its place.

Finally, the parameters of \( \Pr(h_1, \ldots, h_D | A_{w_j}=a) \), \( a \in \{0, 1\} \), which models how simple visual features of image \( I \) (external to the black-box) differ when \( w_j \) is correct or incorrect, are estimated. The goal is to allow PLMFIT the opportunity to be directly influenced by visual features which can be efficiently and incrementally computed. Note that PLMFIT is already indirectly affected by visual aspects of \( I \) via the black-box. A formulation that we successfully incorporated into PLMFIT is now described. Each image is divided into 16 equally spaced, non-overlapping, orthogonal tiles. For each tile, the \( RGB \) color values are transformed into the \( LUV \) space, and the triplet of average \( L \), \( U \), and \( V \) values represent that block. Thus, each image is represented by a 48-dimensional vector \( (h_{1}, \ldots, h_{48}) \) of global visual features (see Fig. 7). For estimation, each of the 48 components are fitted with univariate Gaussians, which involves calculating the sample mean \( \mu_{j,d,a} \) and std. dev.
Figure 6: Empirical differences between estimates of $P_r(G_w=1|A_w=a, G_w=1)$ when using semantically related words as against unrelated ones, are shown. For each of the 30 most frequent tags $w_i$ in Alipr dataset, we compute

$\frac{1}{|\mathcal{V}|} \sum_{j=1}^{|\mathcal{V}|} \frac{1}{|\chi_j|} \sum_{w \in \chi_j} |P_r(G_w=1|A_w=a, G_w=1) - P_r(G_w=1|A_w=a, G_w=1)|,$

where $\chi_j$ is a set of either '10 most related' or '10 least related' tags of $w_i$, in the WordNet sense (defined previously). What we see is that out of 30 tags, 25 and 27 tags for $a = 0$ and $a = 1$ respectively have better estimates of the probability terms when substituted with semantically related terms, as against unrelated ones. This indicates that smoothing with relatedness weights is an attractive strategy.

$\delta_{j,d,a}$ over training data with $a = \{0,1\}$ on tag $w_j$. As before, smoothing is performed by interpolation with priors $\mu_{j,d,a}$ and $\sigma_{j,d,a}$ estimated over all tags. The joint probability is computed by treating each component conditionally independent given $w_j$:

$$\widetilde{P}_r(h_1, \ldots, h_{48} | A_w=a) = \prod_{d=1}^{48} \mathcal{N}(h_d | \mu_{j,d,a}, \sigma_{j,d,a})$$

(11)

Here, $\mathcal{N}(\cdot)$ denotes the Gaussian density function. Again, to provide intuition behind the role of these ratios in PLMFTT, we present $\mu_{j,d,a}$ values for two exemplary cases, estimated
with real-world data, in Fig. 8. With this, we have covered the estimation of each term in the PLMFIT model. We continue discussion on the application of this static meta-learning model to dynamic scenarios.

\[ (h_1, \ldots, h_{17}, \ldots, h_{33}, \ldots, h_{48}) \]

\[ L \quad U \quad V \]

Figure 7: The 48-dimensional visual feature extraction (LUV color space) for PLMFIT.

![Figure 7](image)

Figure 8: Estimated values of \( \hat{\theta}_{j,d,a} \), model parameters for \( Pr(h_1, \ldots, h_{48} \mid A_{w_j} = a) \), for the 48-dimensional global image features for two tags with observed differences, are shown. As in the case of 'space' and 'fruit', if differences are significant for \( A_{w_j} = 0 \) and \( 1 \), then this ratio contributes to the inference. An intuition behind the difference in case of 'fruit', for example, is that close-up shots of fruits tend to be brighter and more colorful than is typical.

3. Tagging Improvements with PLMFIT

With the PLMFIT fundamentals and estimation steps described, we proceed to discuss algorithms and setups for adaptation. As discussed before, three settings where PLMFIT can be applied for adaptation purposes are (1) Context, (2) Time evolution, and (3) Personalization. Contextual adaptation, which is achieved essentially by batch-learning the
PLMFIT using a sample set, is discussed first. The latter two entail new algorithms that employ PLMFIT, and therefore these are discussed in more details in following subsections.

3.1 Contextual Adaptation of Tagging

Contextual adaptation, with reference to automatic image tagging, is the application of image annotation models, trained with data generated in one ITE (image tagging environment), to a different ITE. This includes cases where (a) the test conditions vary only in sample selection (e.g., trained using one set of Corel images, and tested on a different set of Corel images), or (b) the test data is from an entirely different source as compared to the training data (e.g., training using Corel, testing on Flickr images). In this sense, the context is not changing on a continual basis, but is rather a one-time shift.

The underlying black-box model is trained using images from say ITE$_1$. The test scenario involves samples from ITE$_2$ which could represent either of the above mentioned cases. We assume the availability of $N$ tagged training samples from ITE$_2$, which we use to estimate the PLMFIT meta-learner as described in Sec. 2.3. For example, if the underlying black-box is Alipr (Li and Wang, 2008) trained using Corel images (ITE$_1$), and photo-sharing site Flickr (ITE$_2$) is our target application, we can use some user-tagged Flickr images to train PLMFIT, and then used the Alipr/PLMFIT combination to tag future images on Flickr. If the combination performed better than Alipr alone, it would be convincing that the meta-learning layer indeed adds to tagging performance. Empirical assessment of contextual adaptation strength of PLMFIT is presented in Sec. 4.1.

3.2 Adapting Tagging over Time

Figure 9: Time-ordered histograms of occurrence of the top 10 most frequent tags in the Alipr dataset (consisting of 20,000 images), computed over 2,000 image overlapping windows (except last one) with window starting points at 1,000 image intervals. Notice how tag popularity fluctuate over time, e.g., after a point, ‘wildlife’ diminishes in frequency while ‘animal’ gains prominence.
When an annotation black-box is deployed in an online environment such as Alipr or Flickr, where there is continuous image uploading and tagging over time, the user base, the kinds of photos uploaded by them, and the types of tags given to them may evolve with time (see evidence of this in Fig. 9). External impetus such as news and current affairs may further protract this (Datta et al., 2007b). To deal with this issue, we can employ PLMFIT to meta-learn and adapt itself as things change. However, continuously re-training the full model is computationally intensive. In this section we present algorithms that help adapt over time efficiently. A schematic view of this scenario and our approach to handling it is shown in Fig. 10.

![Diagram](image)

**Figure 10:** Tagging adaptation over time using a black-box augmented with PLMFIT.

In Flickr, images are publicly uploaded, and independently or collaboratively tagged, not necessarily at the time of uploading. In Alipr, feedback is solicited immediately upon uploading. In both these cases, ground-truth arrives into the system sequentially, giving an opportunity to learn from it to tag future uploads better. As described in Sec. 2.3, PLMFIT estimation is purposely designed to involve summation of instances only, followed by \(O(1)\) parameter computation. Inference steps are also lightweight in nature. We can take advantage of this to perform incremental/decremental learning, thereby eliminating the need for full-fledged re-estimation over time.

To start with, the PLMFIT needs to be estimated with seed images taken from the application ITE. Hence, over a certain initial period, the meta-learner stays inactive, collecting an \(L_{seed}\) number of user-tagged images. At this point, the meta-learner is trained, and starts tagging incoming images. After an \(L_{inter}\) number of new images has been received, the meta-learner is re-trained (see Fig. 11). The primary challenge here is to make use of the models already learned, so as not to redundantly train on the same data. Re-training can be of two types depending on the underlying `memory model':

- **Persistent Memory:** Here, PLMFIT accumulates new data into the current model, so that at steps of \(L_{inter}\), it learns from all data since the very beginning, inclusive of the seed data. Technically, this only involves incremental learning.
• **Transient Memory:** Here, while the model learns from new data, it also ‘forgets’ an equivalent amount of the earliest memory it has. Technically, this involves incremental and decremental learning, whereby at every $L_{\text{inter}}$ jump, PLMFIT is updated by (a) assimilating new data, and (b) ‘forgetting’ old data.

![Diagram showing the concept of transient memory involving data used to train meta-learner at $L_{\text{pt}}$ and the time points $L_{\text{prev}}$, $L_{\text{pt}}$, and $L_{\text{p}}$ for transient memory and persistent memory.

Figure 11: Overview of persistent/transient memory models for tagging adaption over time.

3.2.1 **INCREMENTAL/DECREMENTAL META-LEARNING**

The PLMFIT formulation makes incremental and decremental learning efficient. Let us denote ranges of image sequence indices, ordered by time, using the superscript $[\text{start} : \text{end}]$, and let the index of the current image be $L_{\text{cu}}$. We first discuss incremental learning, required for the case of *persistent memory*. Here, probabilities are re-estimated over all available data up to the current time, i.e., over $[1 : L_{\text{cu}}]$. This is done by maintaining summation terms, denoted $S(\cdot)$, computed in the most recent re-training at $L_{\text{pr}}$ (say), over a range $[1 : L_{\text{pr}}]$, where $L_{\text{pr}} < L_{\text{cu}}$. For the first term in Eq. 6, suppressing the irrelevant variables, we can write

$$\widehat{Pr}(A_{wj} \mid G_{wj})^{[1:L_{\text{cu}}]} = \frac{\sum_{n=1}^{L_{\text{cu}}} I \{ G_{wj}^{(n)} \land A_{wj}^{(n)} \}}{\sum_{n=1}^{L_{\text{cu}}} I \{ G_{wj}^{(n)} \}} $$

$$= S(G_{wj} \land A_{wj})^{[1:L_{\text{pr}}]} + \sum_{n=L_{\text{pr}}+1}^{L_{\text{cu}}} I \{ G_{wj}^{(n)} \land A_{wj}^{(n)} \} $$

$$= \frac{S(G_{wj} \land A_{wj})^{[1:L_{\text{pr}}]} + \sum_{n=L_{\text{pr}}+1}^{L_{\text{cu}}} I \{ G_{wj}^{(n)} \}}{S(G_{wj})^{[1:L_{\text{pr}}]} + \sum_{n=L_{\text{pr}}+1}^{L_{\text{cu}}} I \{ G_{wj}^{(n)} \}} $$

(12)

Therefore, updating and maintaining the summation values $S(G_{wj})$ and $S(G_{wj} \land A_{wj})$ suffices to re-train PLMFIT without using time/space on past data. The *priors* are also computed using these summation values in a similar manner, for smoothing. Since PLMFIT is re-trained at fixed intervals of $L_{\text{inter}}$, i.e., $L_{\text{inter}} = L_{\text{cu}} - L_{\text{pr}}$, only a fixed amount of time/space is required every time for getting the probability estimates, regardless of the
value of $L_{cu}$. The second term in Eq. 6 can also be estimated in a similar manner, by maintaining the summations, taking their quotient, and smoothing with re-estimated priors. For the last term related to visual features, the estimated mean $\hat{\mu}_{j,d,a}$ and std.dev. $\hat{\sigma}_{j,d,a}$ can also be updated with values of $(h_1, \ldots, h_48)$ for the new images by only storing summation values, made possible by our modeling choice of Gaussian distributions. Since $\sigma^2(X) = E(X^2) - (E(X))^2$,

$$\hat{\mu}_{j,d,a}^{[1:L_{cu}]} = \frac{1}{L_{cu}} \left( S(h_d)^{[1:L_{pr}]} + \sum_{n=L_{pr}+1}^{L_{cu}} h_d^{(n)} \right)$$

$$\hat{\sigma}_{j,d,a}^{[1:L_{cu}]} = \frac{1}{L_{cu}} \left( S(h_d)^{[1:L_{pr}]} + \sum_{n=L_{pr}+1}^{L_{cu}} (h_d^{(n)})^2 \right) - \left( \hat{\mu}_{j,d,a}^{[1:L_{cu}]} \right)^2$$

Here, $S(h_d)^{[1:L_{pr}]}$ is the sum-of-squares of the past values of feature $h_d$, to be maintained, and $E(.)$ denotes expectation. This justifies the simple visual representation we have, since it becomes convenient for incremental learning. Overall, this process continues to re-train PLMFIT, using the past summation values, and updating them at the end, as depicted in Fig. 11.

In the transient memory model, estimates need to be made over a fixed number of the most recent data instances. This can also be performed efficiently, avoiding redundancy, by combining incremental and decremental learning. We can again maintain summation values, but here we need to subtract the portion that is to be removed from consideration. Suppose the memory span is decided to be $L_{ms}$, meaning that at the current time $L_{cu}$, the model estimate must only be based on data over the range $[L_{cu} - L_{ms} : L_{cu}]$. Let $L_{old} = L_{cu} - L_{ms}$. Here, we show the re-estimation of $\hat{\mu}_{j,d,a}$. Here, along with summation $S(h_d)^{[1:L_{pr}]}$, we also require $S(h_d)^{[1:L_{old-1}]}$. Therefore,

$$\hat{\mu}_{j,d,a}^{[L_{old}:L_{cu}]} = \frac{1}{L_{ms} + 1} \sum_{n=L_{old}}^{L_{cu}} h_d^{(n)} = \frac{1}{L_{ms} + 1} \left( S(h_d)^{[1:L_{pr}]} + \sum_{n=L_{pr}+1}^{L_{cu}} h_d^{(n)} - S(h_d)^{[1:L_{old-1}]} \right)$$

Since $L_{ms}$ and $L_{inter}$ are decided a priori, we can pre-compute the values of $L_{old}$ for which $S(h_d)^{[1:L_{old-1}]}$ will be required, and store them along the way. Other terms in Eq. 6 can be estimated similarly.

In summary, a high-level version of tagging adaptation over time is presented in Algo. 1 starts with an initial training of PLMFIT using seed data of size $L_{seed}$. This could be accumulated online using the annotation system itself, or from an external source of images with ground-truth (e.g., Corel images). The process then takes one image at a time, annotates it, and when ground-truth is made available, it is stored for future meta-learning. After gaps of $L_{inter}$, the model is re-trained based on one of the two chosen strategies.

### 3.3 Personalized Tagging across People

In environments such as Flickr, where image tagging is typically performed by the owner and her connected relations, there is an opportunity for automatic annotation systems to
Algorithm 1 Adapting Tagging over Time with PLMFIT

Require: Image stream, Black-box, tagged image pool (seed)
Ensure: Annotation guesses for each new image

1: /* Learn an initial seed model using available tagged data */
2: Train PLMFIT using seed data.
3: repeat (T ← incoming image)
4: Annotate T using PLMFIT
5: if User tags image T then
6: \( L_{cu} \leftarrow L_{cu} + 1, L_{uc} \leftarrow T \)
7: \( \text{Dat}(L_{cu}) \leftarrow \text{Black-box guesses, user tags, etc.} \)
8: end if
9: if \( (L_{cu} - L_{seed}) \mod L_{inter} = 0 \) then
10: if Strategy = 'Persistent Memory' then
11: Re-train PLMFIT on Dat(1 : \( L_{cu} \))
12: /* Use incremental learning for efficiency */
13: else
14: Re-train PLMFIT on Dat(\( L_{cu} - L_{ms} : L_{cu} \))
15: /* Use incremental/decremental learning for efficiency */
16: end if
17: end if
18: until End of time

personalize the tagging process in order to improve performance. Different users upload different kinds of images and may also follow different tagging patterns. In fact, from Fig. 12, we can see that the distribution over the set of 50 most frequently occurring tags in the Flickr dataset varies greatly across users. To build personalization models, the following must be assumed; (a) there are tagged images available for the user in question, and (b) there is locality in the tag space of images belonging to the user, i.e., the tag distribution of an user's images does not represent a microcosm of the entire collection of images, but rather something characteristic of that particular user.

In Fig. 13, data obtained from Flickr shows strong evidence of the validity of those assumptions, focusing on (a) tag space locality, and (b) within-user tag similarity compared to across-user tag dissimilarity. One personalization approach could be that a separate black-box model is trained for every single individual. Given millions of users, who at a given time have a varying number of tagged images associated with them, this approach runs the risk of (a) being prohibitively expensive, (b) requiring re-training as more images are tagged, and (c) lacking sufficient data points in many cases. Instead, we can employ PLMFIT to personalize the tags in a lightweight manner, effectively using a prior model over all individuals, and incrementally incorporating new data points per individual by incurring low overhead. Algo. 2 presents a sketch of the personalization algorithm.

3.3.1 Incremental Update

In Algo. 2, when a tag \( w_k \) associated with user pool \( Q^U \) is already part of the vocabulary \( V^{pop} \), the personalization step is to update the parameters in model \( \text{PLMFIT}^U \) that are associated with \( w_k \). This can be done incrementally in a fashion very similar to Eq. 12 as described in Sec. 3.2. As before, summation values need to be maintained. The only difference is that instead of pooling over time, we pool tagged images specifically for user \( U \). Suppose \( Q^U = \{I_{U_1}, \ldots, I_{U_n}\} \), then \( \text{Pr}(A_{w_k} | G_{w_k})^U \), the first term of PLMFIT^U is
Figure 12: Distribution over the 50 most frequent tags in the Flickr dataset (see Sec. 4) for 18 randomly sampled users. Note how the distribution greatly varies, which reinforces our belief that personalization can give automatic image tagging a significant performance boost.

Estimated as

$$\overline{Pr}(A_{wk} \mid G_{wk}) = \frac{S(G_{wk} \& A_{wk})^{pop} + \sum_{q=1}^{n} T\{G_{wk}^{(q)} \& A_{wk}^{(q)}\}}{S(G_{wk}^{pop}) + \sum_{q=1}^{n} T\{G_{wk}^{(q)}\}}$$  (13)
where superscripts \( \text{pop} \) and \( U \) denote population and user \( U \) specific terms respectively, and \( S(\cdot) \) denotes summation terms, as before. The other terms in Eq. 6 can also be updated in a similar manner, as mentioned in Sec. 3.2. Intuitively, a larger size of the pool \( Q^U \) should have greater influence on PLMFIT, and hence the personalization should be more effective for users with more tagged images available.

3.3.2 Vocabulary Expansion

When a tag \( w_k \) associated with \( Q^U \) is not part of \( V^{\text{pop}} \), it needs to be added to vocabulary \( V^U \) for user \( U \), and corresponding probability terms need to be estimated from scratch. The first term \( Pr(A_{w_j} \mid G_{w_j})^U \) and the last term \( Pr(h_1, \ldots, h_{18} \mid A_{w_j})^U \) in Eq. 6 can each be estimated using Eq. 7 and Eq. 11 respectively, since the settings are essentially identical. Estimation of probabilities \( Pr(G_{w_k} = g_i \mid A_{w_j} = a_j, G_{w_j} = g_j)^U \), which make up the second term, needs more attention. We need to generate new estimates for terms of the form \( Pr(G_{w_k} = g_i \mid A_{w_j} = a_j, G_{w_j} = g_j)^U \) and \( Pr(G_{w_i} = g_i \mid A_{w_k} = a_k, G_{w_k} = g_k)^U \), for all
Algorithm 2 Personalized Tagging across People with PLMFT

Require: Black-box, Tagged image pool from population (seed)
Require: Previously tagged set of images for user \( U \), population vocabulary \( V^\text{pop} \)
Ensure: Personalized tagging model PLMFT\(^U\) for \( U \), with vocabulary \( V^U \)
1: /* Learn a seed model over the entire population */
2: Train PLMFT\(^{pop}\) using seed data (once for all users)
3: PLMFT\(^U\) ← PLMFT\(^{pop}\), \( V^U ← V^\text{pop} \)
4: \( Q^U \) ← Tagged image pool of user \( U \)
5: \( W ← \{ \text{user tags for } Q^U \} \cup (\text{PLMFT}\(^U\) tag predictions for } Q^U) \)
6: repeat \{ \( w_k ∈ W \), taken in arbitrary order \}
7: if \( w_k ∈ V^U \) then
8: /* Tag is in the vocabulary, so update model */
9: Perform incremental update of PLMFT\(^U\) using relevant subset of \( Q^U \)
10: else
11: /* Tag is not in current vocabulary, so add to model */
12: \( V^U ← V^U \cup \{ w_k \} \)
13: Perform vocabulary expansion of PLMFT\(^U\) using relevant subset of \( Q^U \)
14: end if
15: until all tags in \( W \) are covered

\( w_i, w_j ∈ V^\text{pop} \). For a new tag \( w_k \), \( G_{w_k} = 0 \) in all cases, so we can re-write the corresponding ratio terms as follows:

\[
\frac{Pr(G_{w_k} = g_k \mid A_{w_i} = 1, G_{w_i} = g_j)}{Pr(G_{w_k} = g_k \mid A_{w_j} = 0, G_{w_j} = g_j)} = 1, \text{ and} \\
\frac{Pr(G_{w_k} = g_k \mid A_{w_i} = 1, G_{w_i} = g_i)}{Pr(G_{w_k} = g_k \mid A_{w_i} = 0, G_{w_i} = g_k)} = \frac{Pr(G_{w_i} = g_i \mid A_{w_k} = 1)}{Pr(G_{w_i} = g_i \mid A_{w_k} = 0)}
\]

(14)

In plain words, this means that (a) new tags play no role in the prediction of the tags in the black-box vocabulary, and (b) since the new tags are never guessed by the black box, their prediction is based entirely upon guesses on the black-box vocabulary. This also shows that in Algo. 2, the order in which the new words are added to the vocabulary does not matter.

To summarize, for a new word \( w_k \), the logit in Eq. 6 can be simplified as follows:

\[
\log \ell_{w_k}(I) = \log \frac{Pr(A_{w_k} = 1)}{1 - Pr(A_{w_k} = 1)} + \sum_{i\neq k} \log \left( \frac{Pr(G_{w_i} = g_i \mid A_{w_k} = 1)}{Pr(G_{w_i} = g_i \mid A_{w_k} = 0)} \right) + \log \left( \frac{Pr(h_{11}, ..., h_D \mid A_{w_k} = 1)}{Pr(h_{11}, ..., h_D \mid A_{w_k} = 0)} \right)
\]

(15)

Vocabulary expansion in this manner is general enough that it can apply to tagging adaptation over time without any personalization. A combination of efficient adaptation over time and user-specific personalization is a natural extension, but we have not experimented with such combinations in this work.

4. Experimental Results

We perform image tagging experiments to validate the effectiveness of PLMFT in (1) contextual adaptation, (2) adaptation over time, and (3) personalization. Standard datasets as well as real-world data are used. First, we use the Corel Stock photos (Li and Wang, 2003) to compare our meta-learning approach with the state-of-the-art. This collection of
images is tagged with a 417 word vocabulary. Second, we obtain two real-world, temporally ordered traces from the Alipr Website (Alipr, 2006), each 10,000 in length, taken over different periods of time in the year 2006. Each trace consists of publicly uploaded images, the automatic annotations provided by Alipr, and the tags provided by users for them. The Alipr system provides the user with 15 guessed tags, and the user can opt to select the correct guesses and/or add new ones. The vocabulary for this dataset consists of 329 unique tags. Third, using the Flickr API (Flickr, 2005), we obtain up to 1000 public images belonging to each of 300 random users, totaling to 162,650 real-word images. After pruning tags that appeared less than 10 times in the entire dataset, we were left with a 2971 word vocabulary, with a mean of 7 (± 5) user tags per image, which we treat as ground-truth. As expected, we observed a great number of user-specific tags (e.g., names of people) in the dataset, although many of the less significant ones were eliminated in the aforementioned pruning process.

Two different black-box annotation systems, which use different algorithms for image tagging, are used in our experiments. A good meta-learner should fare well for different underlying black-box systems, which is what we set out to explore here. The first is Alipr (Li and Wang, 2008), which is a real-time annotation system, and the second is a recently proposed approach (Datta et al., 2007a) which was shown to outperform earlier algorithms. Both models generate tag guesses given an image, ordered by decreasing likelihoods. Annotation performance is gauged using three standard measures, namely precision, recall and F-score that have been used in the past. Specifically, for each image, precision = \#(tags guessed correctly) / \#(correct tags), recall = \#(tags guessed correctly) / \#(correct tags), and F-score = \#(Precision x Recall) / Precision + Recall (harmonic mean of precision and recall). Results reported in each case are averages over all images tested with.

The ‘lightweight’ nature of PLMFIT is validated by the fact that the (re-)training of each visual category in (Li and Wang, 2008) and (Datta et al., 2007a) are reported as 109 and 106 seconds respectively. Therefore, at best, re-training will take these times when the models are trained fully in parallel. In contrast, our meta-learner re-trains on 10,000 images in ~ 6.5 sec. on a single machine having equivalent configuration. Furthermore, the additional computation time due to the meta-learner during annotation is negligible.

### 4.1 Contextual Adaptation of Tagging

Our first set of experiment tests the notion that PLMFIT can adapt to contextual change, i.e., it can take a black-box annotation system trained on one type of sample and can improve its performance on a different dataset. In the work by Datta et al. (2007a), 24,000 Corel images, drawn from 600 image categories were used for training, and a separate 10,000 test images were used to assess performance. We use this system as black-box by obtaining the word guesses made by it, along with the corresponding ground-truth, for each image.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Precision</th>
<th>% Change</th>
<th>Recall</th>
<th>% Change</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Datta et al., 2007a)</td>
<td>1 in 4</td>
<td>-</td>
<td>2 in 5</td>
<td>-</td>
<td>31.3</td>
</tr>
<tr>
<td>PLMFIT (Top r)</td>
<td>1 in 3</td>
<td>+28%</td>
<td>3 in 4</td>
<td>+83%</td>
<td>45.2</td>
</tr>
<tr>
<td>PLMFIT (Threshold)</td>
<td>2 in 5</td>
<td>+50%</td>
<td>3 in 5</td>
<td>+50%</td>
<td>48.6</td>
</tr>
</tbody>
</table>
Our meta-learner PLMFIT uses an additional \( L_{\text{seed}} = 2,000 \) images (randomly chosen, non-overlapping) from the Corel dataset as the seed data. Therefore, effectively, (black-box + PLMFIT) uses 26,000 instead of 24,000 images for training. We present results on this case in Table 1. The PLMFIT performance is shown for both Top \( r \) (\( r = 5 \)) and Threshold \( r\% \) (\( r = 60 \)), as described in Sec. 2.2. The baseline results are those reported in (Datta et al., 2007a). Note the significant jump in performance with our meta-learner in both cases. Strictly speaking, this is not a contextual change, since the black-box was trained on Corel images and the testing was also done using Corel, although they were non-overlapping sets. This makes the results more surprising, in that simply adding PLMFIT to the mix makes a significant difference in performance.

Table 2: Contextual adaptation performance on 16,000 Alipr images

<table>
<thead>
<tr>
<th>Approach</th>
<th>Precision</th>
<th>% Change</th>
<th>Recall</th>
<th>% Change</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Li and Wang, 2008)</td>
<td>17 in 100</td>
<td>-</td>
<td>2 in 5</td>
<td>-</td>
<td>24.2</td>
</tr>
<tr>
<td>PLMFIT (Top ( r ))</td>
<td>22 in 100</td>
<td>+28%</td>
<td>1 in 2</td>
<td>+18%</td>
<td>30.3</td>
</tr>
<tr>
<td>PLMFIT (Threshold)</td>
<td>1 in 3</td>
<td>+55%</td>
<td>3 in 5</td>
<td>+42%</td>
<td>48.6</td>
</tr>
</tbody>
</table>

Next, we perform an experiment that truly tests the contextual adaptation power of PLMFIT. Real-world images obtained from Alipr do not share the typical characteristics of Corel images in terms of types of images uploaded and tags given to them. Therefore there is considerable challenge in using a Corel-trained black-box model to tag them, as will also become evident from the baseline performance presented here. We use both Alipr traces consisting of 10,000 images each. It turns out that given the Alipr Website, most people provided feedback by selection, and a much smaller fraction typed in new tags. As a result, the recall is by default very high for the black-box system, but it also yields poor precision. For each Alipr trace, our meta-learner is trained on \( L_{\text{seed}} = 2,000 \) seed images, and tested on the remaining 8,000 images. In Table 2, averaged-out results using PLMFIT for both Top \( r \) (\( r = 5 \)) and Threshold \( r\% \) (\( r = 75 \)), as described in Sec. 2.2, are presented alongside the baseline (Li and Wang, 2008) performance on the same data (top 5 guesses). Again we observe significant performance improvements over the baseline in both cases. As is intuitive, a lower percentile cut-off for threshold, or a higher number \( r \) of top words both lead to higher recall, at the cost of lower precision. Therefore, either number can be adjusted according to the specific needs of the application. Given these results, we can conclude that PLMFIT layer provides a statistically significant boost to image tagging performance of a black-box system under contextual changes.

4.2 Adapting Tagging over Time

We now test whether the PLMFIT layer is effective at adapting to changes over time or not. Because the Alipr data was generated by a real-world process with real users, it makes an apt dataset for this test. Again, the black-box here is the Alipr system, which provides guessed tags, and the Website users provide ground-truth tags. First, we experiment with the two data traces separately. For each trace, a seed data consisting of the first \( L_{\text{seed}} = 1,000 \) images (in temporal order) is used to initially train PLMFIT. Re-training is performed in intervals of \( L_{\text{inter}} = 200 \). We test on the remaining 9,000 images of the trace for (a) static
tagging - PLMFit is not further re-trained after seed training, and (b) tagging over time - PLMFit is re-trained over time, using (a) Top r (r = 5), and (b) Threshold r% (r=75) in each case. For these experiments, the persistent memory model is used. Comparison is made using precision and F-score, with the baseline performance being that of Alipr, the black-box. These results are shown in Figs. 14 and 15. The scores shown are moving averages over 500 images (or less, in the case of the initial 500 images). We observe that seed training considerably boosts performance over the baseline, and this performance keeps getting better over time.

Next, we explore how the persistent and transient memory models fare against each other. The main motivation for transient learning is to ‘forget’ earlier training data that may have become irrelevant, due to concept drift or otherwise. Because we observed such a shift between Alipr traces #1 and #2 (being taken over distinct time-periods), we merged
Figure 15: Performance (precision and F-score) of adaptation over time for Alipr trace #2.

thom together to obtain a single 20,000 image trace to emulate a scenario of shifting trend in image tagging. Performing a seed learning over images 4,001 to 5,000 (part of trace #1), we test on the trace from 5,001 to 15,000. The results obtained using the two memory models, along with the static and baseline cases, are presented in Fig. 16. Observe the performance dynamics around the 10,000 mark where the two traces are merged. While the persistent and transient models follow each other closely till around this mark, the latter performs better after it (by up to 10%, in precision), verifying our hypothesis that under significant changes over time, ‘forgetting’ helps PLMFT get adapted better.

A strategic question to ask, on implementation, is ‘How often should we re-train PLMFT, and at what cost?’ To analyze this, we experimented with the 10,000 images in Alipr trace #1, varying the interval $L_{\text{inter}}$ between re-training while keeping everything else identical, and measuring the F-score. In each case, the computation durations are noted
and normalized by the maximum time incurred, i.e., at $L_{inter} = 100$. These results are presented in Fig. 17. Note that with increasing gaps in re-training, F-score decreases to a certain extent, while computation time hits a lower bound, which is the amount needed exclusively for tagging. There is a clear trade-off between computational overhead and the F-score achieved. A graph of this nature can therefore help decide on this trade-off for a given application.

Finally, in Fig. 18, we show a sampling of images from a large number of cases (which we found via eye-balling) where annotation performance improves meaningfully with PLMFIT re-training over time. Specifically, at time 0 we show the top 5 tags given to the image by Alipr. This is followed by PLMFIT’s guesses after training with 1000 and 3000 temporally
Adapting Automatic Image Tagging via Meta-learning

Figure 17: Comparing F-score and computation time with varying $L_{inter}$.

Figure 18: Sample annotation results found to improve over time with PLMFIT adaptation.

ordered images. Clearly, more correct tags are pushed up by the meta-learning process, which improves with more re-training data.

4.3 Personalized Tagging across People

Our final set of experiments are aimed at measuring the effectiveness of PLMFIT meta-learning for the purpose of user-wise personalization. Our experiments here are all based on
162,650 public Flickr images belonging to 300 real users. The black-box annotation system (baseline) used here is the Alipr algorithm (Li and Wang, 2008). Of its vocabulary of 329, we find that 108 do not appear even once in the Flickr dataset. Aside from performance improvement with personalization, some other key aspects of interest were (a) vocabulary expansion, described in Sec. 3.3, for expanding the tag vocabulary of the black-box, (b) the effect of the user average method of tag selection, on personalization, and (c) the variation of performance with the amount of per-user data samples used.

Table 3: Personalization performance on 162,650 Flickr images (300 users)

<table>
<thead>
<tr>
<th>Approach</th>
<th>Precision</th>
<th>% Change</th>
<th>Recall</th>
<th>% Change</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Li and Wang, 2008)</td>
<td>1 in 60</td>
<td>-</td>
<td>1 in 25</td>
<td>-</td>
<td>2.3</td>
</tr>
<tr>
<td>PLMFIT (seed only)</td>
<td>1 in 14</td>
<td>+359%</td>
<td>1 in 10</td>
<td>+149%</td>
<td>8.5</td>
</tr>
<tr>
<td>PLMFIT (personalized - VE)</td>
<td>1 in 7</td>
<td>+778%</td>
<td>1 in 8</td>
<td>+203%</td>
<td>13.04</td>
</tr>
<tr>
<td>PLMFIT (personalized + VE)</td>
<td>2 in 9</td>
<td>+1270%</td>
<td>2 in 5</td>
<td>+881%</td>
<td>27.98</td>
</tr>
</tbody>
</table>

First, we computed the overall performance of different PLMFIT settings as compared to the baseline. A set of 5000 images, sampled from the full set of users, is set aside, and used as seed. For the case of PLMFIT, the top 10 tags are used for annotation prediction. For the purpose of personalization, the images from each user are divided into 75% training (images previously uploaded by the user) and 25% (new images uploaded) testing. The results are presented in Table 3. We see that Alipr performance is significantly improved in all cases of PLMFIT use. For 287 out of the 300 users (i.e., 95.7%), PLMFIT with personalization leads to higher precision as well as recall than the baseline. When a random seed is used alone, performance improvement is relatively lower than with personalization, which can be partly explained by the graphs in Fig. 13. This justifies the need for personalization as against simply using one PLMFIT model for every user, i.e., contextual adaptation 3.1 helps but can be further improved with personalization. Of course, personalization in this manner is not possible for new users entering the system.

Second, we explore how the final tag selection strategy affects personalization performance. The setting remains the same as the previous case, except as specified. In Fig. 19, metrics are shown for the baseline, for PLMFIT with top 10, 15, or 20 tags being selected, and when the number of tags that PLMFIT predicts is based on the average for the given user. As expected, we see a clear trend whereby more tags predicted lead to higher recall at the cost of lower precision. More interestingly, the strategy of picking the average number of tags in the user training data for future prediction seems to be the winning strategy in terms of F-score. While this can be thought of as an additional personalization step, it was found to be much less effective for users with high variance in tag counts. A more effective strategy can possibly be built around this fact.

Finally, we set out to test the effect of different mixes of seed data and user-specific data for personalization. Settings remain same as before, except that the PLMFIT model is trained differently. In these experiments, we only consider users which have over 800 images, so that upto 800 of them can be used for training. In our dataset, there were 83 such users, with a collective pool of 78,928 images. We mix randomly drawn seed images with these user-specific samples, to always come up with 800 training images. We train
Figure 19: Graph showing precision, recall, and F-score for a number of settings, including Alipr’s tagging of the Flickr images (baseline), PLMFiT adaptation without personalization (seed), and personalization with different numbers of tags predicted. The case of 'User Average' uses the average number r of tags for a given user’s training data and predicts top r tags for the test cases.

PLMFiT with this set, and compute performance metrics on the remainder of the images for that user, not used in training. Results, averaged over all users, are plotted in Fig. 20. At $x = 0$ in this graph, no user-specific data is used, and hence performance improvement is not dramatic. As the user-specific sample share gets larger, performance keeps improving, eventually flattening out at about the 0.6 mark. This can be justified by the fact that the non-Alipr tags tend to be less generic and more localized, thus more user data means improved estimation on these tags. We also observe that adding more user data seems to improve performance more significantly with vocabulary expansion than without it.

Figure 20: Graph showing variation of precision and recall with a varying proportion of seed data to user-specific samples, used for training the PLMFiT for personalization.
5. Discussion

The need to bring automatic image tagging to real-world applicability inspired us to employ a meta-learning approach. The ability to adapt to various kinds of scenario changes in a scalable manner led us to incorporate incremental learning into the approach. The results with our approach were found to be strongly positive, with large jumps in tagging performance, without any significant addition to the image analysis sophistication. That the results would be positive is very encouraging but not surprising. At the inception of work on this problem, we held the belief that even without extracting more profound semantics from visual features, there is a certain unknown amount of performance gain to be got for 'free'. During analysis of the datasets, the rates of changes of tagging trends over time and across people that we observed exceeded our expectations, and further strengthened this belief. The more surprising result was how much exactly this amount turned out to be. A small amount of additional computation leads to large chunks of performance gain.

While we are much excited about the proposed approach, here we discuss some of its issues and limitations, and make some general comments:

- The PLMFIT approach is based on the assumption that the underlying black-box annotation system does, to some extent, learn semantics from visual features. A black-box system that maps image features to tags randomly is unlikely to benefit from meta-learning. The effectiveness of inductive transfer also depends on the nature of the black-box system.

- Unlike many other machine-learning problems, automatic image tagging approaches have historically been reported to perform moderately at best, and the semantic gap (Smeulders et al., 2000) has been most often cited as the main challenge in this learning task. PLMFIT, being dependent on an underlying annotation system, is thus bounded by the same. Despite the significant jumps in performance with our approach, the absolute values of the performance metrics leave much to be desired. Regardless, in this problem domain, a moderate precision and recall can still prove to be very useful in real-world applications.

- The choice of black-box annotation systems for testing PLMFIT was based on practical considerations. When possible, we would like to experiment with other annotation systems as well.

- The experimental results are based on real-world samples, but those samples do not represent every possible dynamism. We are unable to conclude on the adaptability of our approach under, for example, extreme changes.

- The 'new user' problem of personalization exists here as well. We use a generic 'prior' model for new users, estimated from all users, but performance gain is not as significant as when some of the user's data is used in training. There is a possibility of exploiting the local neighborhood around a new user to obtain a more specific prior model, but we have not experimented with this idea.

- While adaptation over time and across people have been treated as separate scenarios, it is practical to also analyze adaptation performance under simultaneous change of
time and users. We have skipped this combined analysis in order to keep the focus on the main contributions.

- In our Flickr dataset, tags exhibit a heavy-tailed distribution. Prior to pruning the tags by frequency, over 73% of the tags appeared only once. Models cannot be trained with them, which means that the heavy-tail imposes an upper-bound on recall.

- If the tag vocabulary consists of mostly proper nouns and non-English terms, as in the case of our Flickr dataset, an annotation system may well be predicting semantically correct tags, but there is no easy way to assess performance. Such a system will be useful for semantic image organization, but not in mimicking user tagging. The role of WordNet is also greatly diminished in such cases.

- While the proposed models and algorithms are designed specifically for the image tagging problem, they can generally apply to any learning task that involves making multiple binary decisions on each data point.

Despite these limitations, the results should encourage a greater use of meta-learning and a greater focus on the real-world applicability of automatic image tagging.

6. Conclusions and Future Work

In this paper, we have proposed the use of meta-learning and incremental learning to make automatic image tagging applicable in the real-world. We have proposed a principled, lightweight, meta-learning framework for image tagging (PLMFit), inspired by inductive transfer, which can augment existing annotation systems, to allow adaptation to changes of any nature, such as contextual changes. We have proposed an algorithm to make use of PLMFit to improve and adapt tagging over the time domain, and showed it to be effective. Finally, an algorithm that uses PLMFit for personalized tagging has been proposed. The personalized version of PLMFit is found to produce significantly better results than the generic version, showing improved tagging precision as well as recall for over 95% of the users in the Flickr dataset. In achieving this goal, methods to allow expansion of the system's tag vocabulary beyond that of the initial version, has been presented. In all cases, efficiency is achieved via incremental learning, and the methods have been validated using large real-world datasets. In general, we can conclude that the meta-learning approach to image tagging appears to have a number of attractive properties.

For future extensions of this work, we plan to couple PLMFit with a wider array of black-box systems to ensure that these results generalize well. It is also one of our goals to combine tagging adaptation over time with personalization, so as to have a comprehensive meta-learning layer that can truly and effectively adapt to all forms of changes in real-world environments. For the 'new user' problem of personalization, it might help to exploit the network of neighbors surrounding a user, to build better prior models under limited data.
Acknowledgments

The material is based upon work supported by the National Science Foundation under Grant Nos. IIS-0347148, DMS-0705210, and CNS-0202007. Discussions with Dhiraj Joshi of the Eastman Kodak Research Labs were very fruitful.

References


