Weakly supervised structured output learning for semantic segmentation

Citation for published version:
Vezhnevets, A, Ferrari, V & Buhmann, JM 2012, Weakly supervised structured output learning for semantic segmentation. in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. Institute of Electrical and Electronics Engineers (IEEE), pp. 845-852. https://doi.org/10.1109/CVPR.2012.6247757

Digital Object Identifier (DOI):
10.1109/CVPR.2012.6247757

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published in:
Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Abstract

We address the problem of weakly supervised semantic segmentation. The training images are labeled only by the classes they contain, not by their location in the image. On test images instead, the method must predict a class label for every pixel. Our goal is to enable segmentation algorithms to use multiple visual cues in this weakly supervised setting, analogous to what is achieved by fully supervised methods. However, it is difficult to assess the relative usefulness of different visual cues from weakly supervised training data. We define a parametric family of structured models, where each model weighs visual cues in a different way. We propose a Maximum Expected Agreement model selection principle that evaluates the quality of a model from the family without looking at superpixel labels. Searching for the best model is a hard optimization problem, which has no analytic gradient and multiple local optima. We cast it as a Bayesian optimization problem and propose an algorithm based on Gaussian processes to efficiently solve it. Our second contribution is an Extremely Randomized Hashing Forest that represents diverse superpixel features as a sparse binary vector. It enables using appearance models of visual classes that are fast at training and testing and yet accurate. Experiments on the SIFT-flow data-set show a significant improvement over previous weakly supervised methods and even over some fully supervised methods.

1. Introduction

In this paper we consider the problem of semantic segmentation, where a label must be predicted for every pixel in an image (e.g. "dog", "car" or "road"). This is a fundamental and challenging problem in computer vision. The standard approach is to train with full supervision, where every pixel is manually labeled [1, 2, 3, 4, 5]. Producing this annotation is very time-consuming. Recently, a few weakly supervised methods have emerged [6, 7], which can train from image labels indicating which classes are present, but without pixel-level labels. This setting increases the challenge, as pixel labels for the training set have to be inferred before a method is ready to label a novel test image.

Visual classes are intrinsically varied and complex. In fully supervised semantic segmentation, models that have complex structure [1, 2] or that leverage a diverse and large set of visual features [8] achieve state-of-the-art performance. Also for fully supervised object detection, [9] reported outstanding results by using multiple kernel learning to integrate diverse feature sets into one model. Features are usually integrated in a weighted sum [8, 9], where weights correspond to the usefulness of features. Weights are estimated on the training set by minimizing the discrepancy between the model output and ground-truth annotations. However, in the weakly supervised case pixel labels are not available, which makes it impossible to directly adapt the weights.

Our goal is to enable weakly supervised algorithms to benefit from a rich and diverse set of visual features and structured models. We formulate semantic segmentation as a pairwise CRF, as in other works [1, 7, 5, 6]. The task of the model is to infer latent superpixel labels in training images and learn appearance models of classes. We consider a parametric family of CRF models. In this family, different models give different mixing weights to different visual similarity metrics between superpixels (color, texture, e.t.c.), and also have a different weighting of the pairwise vs. unary potentials. We propose a model selection criterion that evaluates the quality of each model in the family, without looking at superpixel labels. Finding the best model is a difficult optimization problem with many local maxima and no analytic gradient. We cast this problem into a Bayesian optimization framework and propose an efficient method for solving it based on Gaussian Processes.

Our second contribution is an improved representation of the appearance models of semantic classes. On one hand, appearance models should be flexible and leverage a diverse set of visual features. On the other hand, learning and prediction must be efficient, because during the model selection phase they are performed at every optimization step. To satisfy both requirements, we propose the Extremely Randomized Hashing Forest (ERHF), which is capable of mapping almost any feature space into a sparse, binary representation. This choice enables us to use a very simple and
Figure 1. Schematic illustration of Expected Agreement evaluation. It proceeds by 1. estimating parameters $\theta$ from weak labels $\{Y^j\}_{\tau_1}$ of data-set $\tau_2$. 2. using $\theta$ predicting labels $\{\hat{y}_i^j\}_{\tau_1}$ for superpixels $\{x_i^j\}_{\tau_1}$ from data-set $\tau_1$. 3. inferring labels $\{y_i^j\}_{\tau_1}$ from weak labels $\{Y^j\}_{\tau_1}$ of data-set $\tau_1$. 4. comparing $\{y_i^j\}_{\tau_1}$ and $\{\hat{y}_i^j\}_{\tau_1}$. Notice, that on step 2, data-set $\tau_1$ is treated as a test set - labels $\{Y^j\}_{\tau_1}$ are not used.

Our experiments on the challenging SIFT-flow dataset [10] show that the above contributions significantly improve semantic segmentation accuracy over a state-of-the-art weakly supervised method [6] and even over a fully supervised method [5].

2. Weakly supervised multiple features integration

We start by setting up the problem and notation. Images are represented by their superpixels, obtained by an oversegmentation algorithm [11]. Let $\tau = \{I^j = \{(x_i^j)_{i=1}^{N_j}, Y^j\}_{j=1}^N\}$ be the training set, where image $I^j$ consists of superpixels $x_i^j$. For each image we are given a label set $Y^j \subset \mathcal{Y}$, which is a subset of the set of all possible labels $\mathcal{Y} = \{1, \ldots, C\}$ corresponding to classes. Each superpixel $x_i^j$ has an associated latent label $y_i^j \in Y^j$. The image label set $Y^j$ is the union of the (unknown) labels of all superpixels inside it ($Y^j = \bigcup y_i^j$). The task of weakly supervised learning is to recover the latent labels $y_i^j$ and to learn appearance models for the classes. These models will later help to predict superpixel labels in new test images.

2.1. Segmentation model

We model the weakly labeled training set as a CRF, where nodes correspond to latent superpixel labels. CRF is a widely adopted model for semantic segmentation, both in fully [1, 2, 5] and weakly [6, 7] supervised approaches. The total energy $\mathcal{E}$ of the model is a function of the superpixel labels $y_i^j$, the appearance model parameters $\theta$ and the weights $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_k)$:

\[
\mathcal{E}(\{y_i^j\}, \theta, \alpha) = \alpha_0 \sum_{x_i^j \in I^j, j \in \tau} \psi(y_i^j, x_i^j, \theta, Y^j) + \\
(1 - \alpha_0) \sum_{k=1}^K \alpha_k \sum_{(y_i^j, y_i^j') \in E_k} \phi_k(y_i^j, y_i^j', x_i^j, x_i^j')
\]

The unary potential $\psi(y_i^j, x_i^j, \theta, Y^j)$ measures how well the appearance of $x_i^j$ matches the appearance model $\theta_{y_i^j}$ of class $y_i^j$, and whether $y_i^j$ belongs to the given image label set $Y^j$.

Each pairwise potential $\phi_k$ encourages connected superpixels to take the same label if their appearance similarity is high

\[
\phi_k(y_i^j, y_i^j', x_i^j, x_i^j') = \begin{cases}
1 - D_k(x_i^j, x_i^j') & y_i^j \neq y_i^j' \\
0 & y_i^j = y_i^j'
\end{cases}
\]

where $D_k(x_i^j, x_i^j') : \mathbb{R} \rightarrow [0, 1]$ is a similarity metric.

Each $D_k$ corresponds to a different measure of visual similarity such as colour, texture, etc. Note that the $\phi_k$ potentials are submodular, since $1 - D_k(x_i^j, x_i^j') \geq 0$ holds always.

The model in (1) has a set of pairwise potentials $\{\phi_k\}_{k=1}^K$, each weighted by $\alpha_k \geq 0$, with $\sum_{k=1}^K \alpha_k = 1$. The weight $\alpha_0 \in [0, 1]$ controls the overall balance between pairwise and unary potentials. The pairwise potentials $\{\phi_k\}_{k=1}^K$ are defined on their own set of edges $E_k$. Since any $\alpha_k$ can be set to zero, $\alpha$ also controls the structure of the model.

The pairwise potentials are typically used to encourage smooth segmentations, by connecting neighbouring superpixels in individual images [4, 2, 12]. In the weakly supervised setting, recently [6] has shown that it is useful to also introduce pairwise potentials connecting superpixels between different training images that share a label. Such potentials encourage superpixels with similar appearance to assume the same label values. The potentials have been shown to facilitate inferring latent superpixel labels and to regularize learning of appearance models. Overall, they are important for the accuracy of weakly supervised methods.

The notation we introduce above subsumes both type of potentials, depending on the definition of the edge set $E_k$.

Learning appearance models $\theta$ and recovering $\{y_i^j\}$. If the weights $\alpha$ are fixed, then $\theta$ and $\{y_i^j\}$ can be obtained by
alternating optimization [6]: i) fix labeling \( \{ y_i^j \} \) and learn \( \theta \); ii) fix \( \theta \) and infer \( \{ y_i^j \} \). The first step corresponds to supervised learning of appearance models and can be solved efficiently for a wide range of appearance models [5, 4, 1]. The second step is a discrete, multi-label submodular optimization problem, which can be solved to a good approximation by alpha-expansion [13].

### 2.2. Expected Agreement criterion for selecting \( \alpha \)

In this section we present our novel criterion for selecting weights \( \alpha \) in the weakly supervised setting. Direct optimization of eq. (1) over \( \alpha \) yields a trivial solution. We must set \( \alpha_0 = 1 \), so that the pairwise potentials are completely ignored. The same phenomenon arises in other domains, for example, if we try to select the weight \( C \) of the regularizer in a SVM, or \( k \) in k-means, by optimizing their loss on the training set. This selection would always prefer the least possible regularization and the largest data (over)-fit.

In this paper, we take a model selection view on this problem and we propose an Expected Agreement criterion, inspired by clustering validation works [14] in unsupervised learning. Each different setting of \( \alpha \) defines a model for which a method to learn \( \theta \) and infer \( \{ y_i^j \} \) is known (see previous subsection). The goal is to select the best model among the family defined by all possible \( \alpha \). The challenge is how to evaluate the quality of a model without looking at superpixels labels. Our answer is: a model is better if it produces consistent results on different subsets of the training data. More precisely, the superpixel labeling produced by training on a subset \( \tau_1 \) should be as close as possible to the labeling obtained by training on another subset \( \tau_2 \) and then ‘testing’ on \( \tau_1 \). Both \( \tau_1 \) and \( \tau_2 \) are weakly supervised i.i.d samples from the same distribution. During testing, the image-level labels of \( \tau_1 \) are concealed.

Let \( L : (\alpha, \{ x_i^j \}_{\tau_1}, \{ Y_j \}_{\tau_1}) \rightarrow (\theta, \{ y_i^j \}_{\tau_1}) \) be the learning algorithm, that given \( \alpha \) and a weakly supervised training subset \( \tau_1 \) learns appearance model \( \theta \) and recovers superpixel labels \( \{ y_i^j \}_{\tau_1} \). Let \( f : (\theta, \alpha, \{ x_i^j \}_{\tau_2}) \rightarrow \{ \hat{y}_i^j \}_{\tau_2} \) be a prediction function, that given parameters \( \theta, \alpha \) predicts superpixel labels \( \{ \hat{y}_i^j \}_{\tau_2} \) for another subset \( \tau_2 \). The Expected Agreement induced by \( \alpha \) is:

\[
A(\alpha) = \mathbb{E}_{\tau_1, \tau_2} \frac{1}{|Y|} \sum_{l \in Y} \left( \frac{1}{\#\{ y_i^j = l \}} \sum_{i \in \tau_1, y_i^j = l} \mathbb{I}\{ \hat{y}_i^j = y_i^j \} \right)
\]

\[
\text{s.t. } \{ y_i^j \} = L(\alpha, \{ x_i^j \}_{\tau_1}, \{ Y_j \}_{\tau_1}) \quad (3)
\]

\[
\theta' = L(\alpha, \{ x_i^j \}_{\tau_2}, \{ Y_j \}_{\tau_2}), \{ \hat{y}_i^j \} = f(\theta', \alpha, \{ x_i^j \}_{\tau_2})
\]

This expression measures the expectation of the average per-class accuracy of the model trained on \( \tau_2 \) and tested on \( \tau_1 \), as if the labels recovered by training on \( \tau_1 \) were the ground-truth. In practice, \( \tau_1 \) and \( \tau_2 \) are random disjoint subsets of a training set. Figure illustrates the process.

It is important to note how the criterion we propose to evaluate a model does not involve any additional parameter, which would otherwise defeat its purpose. We choose the average per-class accuracy because it avoids bias toward classes that cover larger image areas, such as sky or grass. It also naturally penalizes a model that maximizes agreement simply by predicting very few labels overall.

### 2.3. Gaussian Processes for optimization

How can we find \( \alpha \) that maximizes \( A(\alpha) \) from eq. (3)? Notice that \( A(\alpha) \) has no analytic gradient and typically has multiple local maxima. We follow the Bayesian optimization framework [15] and define a distribution over possible realizations of \( A(\alpha) \) using Gaussian Processes (GP) [16]

\[
A(\alpha) \sim \mathcal{GP}(m(\alpha), k(\alpha, \alpha'))
\]

where \( m(\alpha) \) is a mean function and \( k(\alpha, \alpha') \) is a covariance function. Here the mean function is zero \( m(\alpha) = 0 \) and the covariance (kernel) is squared exponential:

\[
k(\alpha, \alpha') = \gamma \exp\left( -\frac{1}{2}(\alpha - \alpha')^T \text{diag}(\nu)^{-2}(\alpha - \alpha') \right)
\]

where \( \nu \) is a vector of hyperparameters, which regulates the influence of each element of \( \alpha \) on the output of the kernel; \( \gamma \) regulates the overall scale (signal variance).

Suppose we have already evaluated \( A \) for \( t \) different \( \alpha_i \), thereby acquiring pairs \( \{ \alpha_i, s_i \} \), where \( s_i = A(\alpha_i) \). Let \( K \) be a kernel matrix \( K_{ij} = k(\alpha_i, \alpha_j) \). Consider now a new point \( \alpha' \), for which \( A(\alpha') \) is unknown. Let \( k = [k(\alpha', \alpha_1), \ldots, k(\alpha', \alpha_t)] \). Then a predictive distribution for \( \alpha' \) is:

\[
A(\alpha') = \mathcal{N}(\mu_t(\alpha'), \sigma_t^2(\alpha'))
\]

where

\[
\mu_t(\alpha') = k^T K^{-1} s_{1:t}
\]

\[
\sigma_t^2(\alpha') = k(\alpha', \alpha') - k^T K^{-1} k
\]

We are now ready to formulate the optimization strategy, known as upper confidence bound (GP-UCB) [17]

\[
\alpha_{t+1} = \arg \max_\alpha (\mu_t(\alpha) + \beta \sigma_t^2(\alpha))
\]

The expectation \( \mu_t(\alpha) \) represents the estimate of the function value \( s_t \) at point \( \alpha_t \). Variance \( \sigma_t^2(\alpha_{t+1}) \) is an inverse of certainty of the estimate. By looking at both the mean \( \mu_t(\alpha_{t+1}) \) and variance \( \sigma_t^2(\alpha_{t+1}) \) GP-UCB trades off exploitation and exploration - a point is queried if its expected value is high or if certainty is low, thus dealing with the problem of local maxima (fig. 2).
Figure 2. Illustration of AdaGP-UCB for one-dimensional function maximization. Each panel corresponds to one iteration. The three curves represent the true function (black), the current GP estimate (blue), and the upper confidence bound (UCB, green). A red cross represents a queried point, a red circle the point with the maximum UCB, which is the next query. At iteration t+1 algorithm gets stuck in local maximum due to false certainty of current approximation. The kernel is then scaled at t+2 and in the end at iteration T the true optimum is found.

Finding hyperparameters $\nu$ and $\gamma$. In the traditional application of GPs for regression, $\nu$ is estimated by maximizing the likelihood of the training data [16]. This concept is applied in a straightforward manner to our case by re-estimating $\nu$, $\gamma$ after each new measurement arrives. However, our samples $\{\alpha_{1:t}, s_{1:t}\}$ are not i.i.d, because they are actively selected. Hence maximum likelihood estimate is not reliable. GP-UCB is especially vulnerable in the beginning, when only a few samples have been observed. In this case, a very smooth function (small values of $\gamma$) yields high likelihood and, therefore, it results in false certainty and low values of $\sigma_t(\alpha_{t+1})$, leading to pure exploitation. In consequence, the algorithm gets stuck, returning the same value $\alpha_{t+1} = \alpha_t$ over and over again. We propose a simple heuristic to avoid this behavior. Whenever $\alpha_{t+1} = \alpha_t$, we scale $\gamma$ by a fixed factor, lowering the confidence and stimulating exploration. We call this procedure Adaptive GP-UCB (AdaGP-UCB) and summarize it in Alg. 1. The algorithm is run either for a given number of iteration or until $k$ consecutive queries $\alpha_{n:n+k}$ return the expected values $s_{n:n+k} = \mu(\alpha_{n:n+k})$. The latter case means that the function value is well-approximated and no new information is being gained.

Algorithm 1 AdaGP-UCB

<table>
<thead>
<tr>
<th>INPUT:</th>
<th>Initial measurements ${\alpha_{1:t_0}, s_{1:t_0}}$, covariance function $K$, $\beta$, scaling factor $\lambda$, maximum number of evaluations $T$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUT:</td>
<td>$\alpha^*$</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1: for $t = t_0 + 1 : T$ do</td>
<td></td>
</tr>
<tr>
<td>2: estimate $\nu_t, \gamma_t$ using maximum likelihood</td>
<td></td>
</tr>
<tr>
<td>3: $\alpha_{t+1} := \arg\max_{\alpha_t} (\mu_t(\alpha) + \beta \sigma_t^2(\alpha))$</td>
<td></td>
</tr>
<tr>
<td>4: while $\alpha_{t+1} = \alpha_t$ do</td>
<td></td>
</tr>
<tr>
<td>5: scale $\gamma_t := \lambda \gamma_t$</td>
<td></td>
</tr>
<tr>
<td>6: $\alpha_{t+1} := \arg\max_{\alpha} (\mu_t(\alpha) + \beta \sigma_t^2(\alpha))$</td>
<td></td>
</tr>
<tr>
<td>7: end while</td>
<td></td>
</tr>
<tr>
<td>8: end for</td>
<td></td>
</tr>
</tbody>
</table>

3. Appearance models via Extremely Randomized Hashing Forest

Here we detail our classifier and feature representations for the appearance models used in unary potential in $\psi$. Every iteration of Alg. 1 involves training and inference for a model defined by $\alpha_t$. In turn, this involves several iterations of estimating $\theta$ and inferring $\{y_i^t\}$ (sec. 2). Therefore the estimation of the appearance model parameters $\theta$ must be computationally efficient. On the other hand, the visual variability of semantic classes demands rich features and a flexible appearance model [8]. To satisfy both requirements, we propose to use a Naive Bayes classifier on top of an intermediate representation obtained by our novel method - Extremely Randomized Hashing Forests (ERHF).

Unlike a regular random forest, we employ ERHF for feature representation, not for classification. It hashes instances $x$ from the original feature space into buckets corresponding to its leaves.

ERHF is an ensemble of decision trees. Each internal node splits the space of data in half. At test time, an instance $x$ is passed through each tree and the indices of leaves it reaches are recorded. The instance $x$ is then represented by a sparse binary vector $b$, where the entry $b[l] = 1$ if $x$ reaches leaf $l$ (and $b[l] = 0$ otherwise). Here $l$ indexes leaves of all trees, thus $b$ has as many elements as the total number of leaves in all trees (see Figure 3).

During training, we build the forest in a completely randomized way, without looking at any class labels. Binary split functions are chosen at random for each node. Data is used only to avoid trivial splits, that have zero samples on one side. This protocol avoids all issues related to weak supervision, since no class labels are used. Training is performed only once and the ERHF is kept fixed. We can use any initial representation of $x$, as long as we can define appropriate binary functions on it. We can combine arbitrary heterogeneous features such as colour, texture and SIFT histograms, superpixel area and location, GIST, and so on. ERHF will transform them into a convenient sparse representation.
Naive Bayes appearance model. Training a Naive Bayes classifier on top of the binary feature representation output by ERHF is very efficient. Our goal is to learn matrix $\theta$, where a sample of class $c$ will reach leaf $l$. Let $B$ be an ERHF representation matrix, where each column $B[i, :]$ is a binary representation of superpixel $x_i$. Matrix $C$ is a binary matrix, where each column is associated with a superpixel and each row with a label; $C[i, :] = 1$ if a superpixel $x_i$ is currently labeled by class $c$. Then $\tilde{\theta} = BC^T$. Since both matrices are very sparse, their multiplication and storage is very efficient. To obtain final parameters $\theta$ we normalize rows of $\tilde{\theta}$.

The appearance model is defined as:

$$f(c, x_i, \theta) = P(c) \prod_{l : B[l, i] = 1} P(B[l, i] | c). \quad (9)$$

Then $\Psi = B^T \text{diag}(P(c))\theta$, where $\Psi_{i, c} = g(c, x_i, \theta)$.

Initial superpixel features are taken from [8]. They constitute a wide range of visual characteristics of a superpixel, like texture, colour, keypoints, size, location, GIST and so on. We choose binary functions for ERHF to be linear functions for similar texture, colour, keypoints, size, location, GIST and so on.

4. Generalized MIM

In this section we detail the particular segmentation model we use - the Generalized Multi-Image Model (GMIM), which generalizes the original Multi-Image Model [6].

$$\mathcal{E}\left(\{y_i^t\}, \alpha, \theta\right) = \alpha_0 \sum_{x_i^t \in \mathcal{T}} \left( \psi(y_i^t, x_i^t, \theta) + \pi(y_i^t, Y_j) \right) +$$

$$+ (1 - \alpha_0) \sum_{k=1}^{K} \alpha_k \left( \sum_{(y_i^t, y_j^t) \in E_k} \phi_k(y_i^t, y_j^t, x_i^t, x_j^t) \right) \quad (10)$$

As most of the other CRFs for segmentation [1, 2, 3, 4, 5, 6], GMIM fits in the general class captured by eq. (1). For clarity, here we decompose the unary potential in two. The first unary potential $\psi(y, x, \theta) = -\log g(y, x, \theta)$ corresponds to appearance models parameterized by $\theta$ and described in the previous section.

The second unary potential $\pi(y_i^t, Y_j)$ enforces a superpixel to take a label from the label set $Y_j$ of the image

$$\pi(y_i^t, Y_j) = \begin{cases} \infty & y_i^t \notin Y_j \\ 0 & y_i^t \in Y_j \end{cases} \quad (11)$$

The edge set $E_k$ over which the pairwise potential is defined is built by a simplified algorithm from the original MIM work [6]. For each $y_i^t$, the algorithm selects the $p$ most similar superpixels from each image $I^t$ which shares a label with the image $I^t$ where $y_i^t$ comes from: $I^t : Y_i^t \cap Y_j^t \neq \emptyset$. This list is then pruned to the $q$ most similar superpixels overall. Finally, $y_i^t$ is connected to the $y$ variables of these $q$ superpixels. The reasons for this procedure is to keep the edge set down to a manageable size, while sacrificing little in terms of modeling accuracy. In fact, the pairwise potential (11) gives very low energy to superpixels of dissimilar appearance regardless of their labels, and therefore only connections between similar superpixels matter. As in [18], we can interpret $E_k$ as a model for the manifolds formed by superpixels in the space defined by the similarity metric $D_k$.

Pairwise potentials penalize labelling $\{y_i^t\}$ that cut through these manifolds.

The original MIM [6] has only one pairwise potential ($K = 1$). In GMIM instead we have a set of pairwise potentials $\{\phi_k\}$. Each is defined on a different set of edges $E_k$, corresponding to $l$ different similarity metrics ($K = l$).

All metrics are based on the $\chi^2$ distance for histograms of quantized i) SIFT [19] ii) colour and ii) texture features integrated over the superpixel. We use the code released by [8] to compute these histograms.

4.1. Segmenting a test image.

Assuming all the parameters of GMIM $\left(\alpha^* \theta^*, \{y_i^t\}\right)$ have been learned, a new image $I^t$ can be segmented in the same way as for the original MIM [6]. First, the few training images most globally similar to $I^t$ are retrieved, using a pre-trained multiple kernel metric [20]. We then derive an estimation of image-level label probabilities for $I^t$, called image-level prior (ILP) [4], by histogramming the labels of the retrieved training images. Finally, the following energy is minimized to label the superpixels $y_i^t$ of $I^t$.
Table 1. Results on SIFT-flow data-set [10]. All methods that, to our knowledge, reported results on this data-set are presented, both weakly and fully supervised. Our GMIM ranks third, surpassing fully supervised TextonBoost [5] and reaching close to SIFT-flow based [10].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>supervision</td>
<td>full</td>
<td>full</td>
<td>full</td>
<td>weak</td>
<td>weak</td>
</tr>
<tr>
<td>average acc.</td>
<td>13</td>
<td>24</td>
<td>29</td>
<td>14</td>
<td>21</td>
</tr>
</tbody>
</table>

The first unary potential \( \psi (y_i^t, x_i^t, \theta^*) \) measures how well \( x_i^t \) matches the appearance model of class \( \theta^*_i \). The second unary potential \( \mu \) is ILP, which can be seen as a soft version of \( \pi \) in eq. (10). The pairwise potentials \( \phi_k \) connect each test image superpixel \( y_i^t \) to its 3 most similar (according to \( D_k \)) superpixels in each retrieved training image. Note how superpixel labels \( y_i^t \) from the training images are fixed, which facilitates optimization.

5. Related work

Our proposed GMIM generalizes previous works on weakly supervised semantic segmentation [7, 6, 21]. The latent aspect model [7] is a special case of GMIM with prefixed \( \alpha \), no pairwise potentials between images, and a specific way of estimating \( \theta \) and inferring \( \{y_i^t\} \). The Multi-Image Model [6] (MIM) is a special case of GMIM with a single pairwise potential between images (constructed via a Semantic Texton Forest from [21]). Moreover, in this work the balance between the unary and pairwise potentials, regulated by \( \alpha_0 \) is also learned using our model selection principle from weakly supervised data. Instead, in previous weakly supervised works it was set heuristically or based on a labeled validation set [7, 6].

GMIM is also related to fully supervised semantic segmentation approaches. In [8] multiple visual cues (SIFT, colour and position cues) were used, but only for unary potentials. In GMIM multiple visual cues are used in both unary and pairwise potentials. In supervised learning in general, the weights \( \alpha \) of a CRF are often learned using structured SVM [22]. However, these cannot be applied in the weakly supervised case, as the loss function would need to observe superpixel labels in the training set. Instead, in our work annotation comes only in the form of image labels. Because the CRF itself is used for inferring the labeling \( \{y_i^t\} \) of the training set, its weights \( \alpha \) have to be selected by a meta-principle.

The Expected Agreement principle is inspired by model selection for clustering [14]. Out of a parametric set of clustering models (e.g. choosing \( k \) for k-means) these techniques prefer a model that produces consistent results on resampled versions of the data.

Using Gaussian Processes for global optimization of “black-box” functions [15] is a recent idea that has been used in problems where no analytic gradient is available and the function is expensive to evaluate, e.g. when solving bandits problems [17] or learning user preferences [23]. The hyperparameters \( \psi , \gamma \) of the GP are usually assumed to be given beforehand. One strategy [15] is Bayesian integration over the hyperprior, but it is only computationally feasible for very low dimensional domains. The dimensionality of \( \psi \) plus \( \gamma \) in our experiments is 7, which is already prohibitive for that approach. Instead of integrating results over all possible \( \psi \) and \( \gamma \), our AdaGP-UCB first commits to the parameters with the highest likelihood. Whenever the algorithm gets stuck, \( \gamma \) is scaled. This corresponds to switching to a less smooth prior over possible functions and have higher uncertainty, hence stimulating exploration.

Random Forests [24] have recently gained popularity in computer vision [4, 25, 26]. Our use of RF is different from common practice. The structure of the forest is trained in an unsupervised way and is used for reformatting the representation of the image features, not for classifying them. ERHF presents a view on RF as hashing, i.e. samples are hashed into buckets corresponding to leaves. This allows fast (re)-training of appearance models, which is valuable in weakly supervised learning. In principle, we could use non-parametric approach from [8] to integrate multiple features into appearance models, but its computational complexity is prohibitive.

6. Experimental results

We present experiments on the LabelMe subset introduced in [10] (called the SIFT-flow data-set). This data-set with 2668 images and 33 classes is very challenging. It is best suited for our task, as all classes are labeled in all images and the significant co-occurrence between classes. Moreover, the large size of the data-set allows to perform model validation without suffering from a small sample size. The standard performance measures in semantic segmentations are the total measure (percentage of correctly classified pixels) and the average per-class measure (percentage of correctly classified pixels for a class, averaged over all classes) [1, 2, 3, 4, 5]. The average measure is preferable as it gives equal contribution to classes, regardless of how large they appear in the images (e.g. dog vs sky). [6, 4]

Experimental protocol. We use standard train/test split [8, 10], consisting of 2488/200 images. We split the training data into two equal parts \( \tau_1 \) and \( \tau_2 \) and use them
for model selection (sec. 2) to obtain the weights $\alpha$, using AdaGP-UCB for optimization. When $\alpha$ is set, we use the recovered superpixel labels $\{y_i^k\}$ from both $\tau_1$ and $\tau_2$ to estimate the final appearance models $\theta^*$ (sec. 3). After training, we apply the model to segment all images in the test set (sec. 4.1) and report the accuracy. In a summary, all parameters of our GMIM model are automatically selected by looking only at image labels in the training set. Pixel labels are used only to report final accuracy on test set.

Results. Results on the test set are reported in table 1. GMIM substantially outperforms the best existing weakly supervised approach [6], which in turn was demonstrated on the easier MSRC21 data-set [5] to outperform earlier weakly supervised methods [7, 21]. Moreover, GMIM surpasses the fully supervised TextonBoost [5], and reaches a performance comparable to the modern fully supervised non-parametric method [10]. However, the very recent state-of-the-art fully supervised method [8] achieves even higher performance. Overall, this comparison shows that GMIM can perform in the range of some fully supervised approaches on a highly challenging data-set, despite training from weak supervision. As a side note, on the training set we recover superpixel labels with 56% per class accuracy.


Here we study the influence of our proposed novel components: i) learning the weights of multiple similarity metrics $\alpha$ ii) learning balance $\alpha_0$ between unary and pairwise term iii) ERHF.

Baselines and protocol. We run grid search over $\alpha$. Vector $\alpha$ lives in a 7 dimensional space, where $\alpha_0 \in [0,1]$ and $\sum_{k=1}^{7} \alpha_k = 1$, $\alpha_{1:k} \geq 0$. We produce a regular grid over the simplex for $\alpha_{1:k}$ and over the $[0,1]$ interval for $\alpha_0$ at 0.1 steps. Each point on the grid defines a model. We choose the model using the Expected Agreement criterion and compare to the following baselines. As a first baseline, we set $\alpha$ to simply average all similarity metrics, and set $\alpha_0 = 0.5$, giving equal important to unary and pairwise potentials. The second baseline uses information of the superpixel labels $\{y_i^k\}$ derived from the training set. We choose 'best' (found by grid search) values of whole $\alpha$, by looking at average accuracy of GMIM when training on $\tau_1$ and testing on $\tau_2$ and vice versa. To investigate the influence of $\alpha_0$, third baseline averages similarity metrics, but sets $\alpha_0$ to the best value.

This is performed for two families of GMIM models, one with unary potential based on ERHF for all superpixel features taken from [8] and the other based only on histogram features from [8] (e.g. SIFT, color, textons), to which Naive Bayes can be applied directly.

Results. Table 2 summarizes the results. All of our novel components contribute to the final result. We significantly outperform flat averaging. Moreover, selecting $\alpha$ by looking at true accuracy of GMIM is no better than selecting it using Expected Agreement. Using ERHF consistently improves results both for baselines and proposed method. Notice, how selecting correct $\alpha_0$ improves results from 6% to 17% even for blind averaging.

6.2. Bayesian optimization

Here we investigate the effectiveness of our AdaGP-UCB optimization. As baselines, we use random search and GP-UCB [17] without adaptation. As a point of reference, we use the optimum found by grid search described above.

In AdaGP-UCB we set $\beta = 0.01$ and $\lambda = 1.1$. The results are presented in fig. 4. AdaGP-UCB matches the maximum found by grid search in just 66 iterations, while regular GP-UCB gets stuck after 15 iterations in local optimum. Grid search made 41136 evaluations of $A(\alpha)$, in contrast to 66 by AdaGP-UCB, making grid search 623 times slower.

7. Conclusion

This paper addresses the problem of integrating multiple visual cues for weakly supervised semantic segmentation. Results show a substantial improvement over previous weakly supervised approaches and further bridge the gap between weakly and fully supervised methods. Our main contribution is a MEA model selection principle for
this problem. Given a parametric family of models that utilize different visual cues, we use it to evaluate the quality of models without looking at superpixel labels. We show that the optimization problem associated with finding the best model can be efficiently solved by Bayesian optimization. Our second contribution is ERHF, which enables us to map diverse, heterogeneous superpixel features into a common sparse binary representation. This allows us to use class appearance models which are efficient to train.

Acknowledgments
A. Vezhnevets was supported by the SNSF under grant #200021-117946. V. Ferrari was supported by a SNSF Professorship.

References