HAL
open science

A Framework for Efficient Structured Max-Margin Learning of High-Order MRF Models

Nikos Komodakis, Bo Xiang, Nikos Paragios

To cite this version:

Nikos Komodakis, Bo Xiang, Nikos Paragios. A Framework for Efficient Structured Max-Margin Learning of High-Order MRF Models. [Research Report] RR-8645, Ecole de Ponts-ParisTech; Ecole Centrale de Paris; Inria Saclay Ile de France; INRIA. 2014, pp.1 - 34. hal-01090971

HAL Id: hal-01090971

https://inria.hal.science/hal-01090971v1

Submitted on 4 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Framework for Efficient Structured Max-Margin Learning of High-Order MRF Models

Nikos Komodakis , Bo Xiang, Nikos Paragios

RESEARCH REPORT

N° 8645

December 2014

Project-Teams GALEN

ISSN 0249-6399 ISRN INRIA/RR--8645--FR+ENG

A Framework for Efficient Structured Max-Margin Learning of High-Order MRF Models*

Nikos Komodakis [†‡], Bo Xiang [§], Nikos Paragios[§¶]

Project-Teams GALEN

Research Report n° 8645 — December 2014 — 31 pages

Abstract: We present a very general algorithm for structured prediction learning that is able to efficiently handle discrete MRFs/CRFs (including both pairwise and higher-order models) so long as they can admit a decomposition into tractable subproblems. At its core, it relies on a dual decomposition principle that has been recently employed in the task of MRF optimization. By properly combining such an approach with a max-margin learning method, the proposed framework manages to reduce the training of a complex high-order MRF to the parallel training of a series of simple slave MRFs that are much easier to handle. This leads to a very efficient and general learning scheme that relies on solid mathematical principles. We thoroughly analyze its theoretical properties, and also show that it can yield learning algorithms of increasing accuracy since it naturally allows a hierarchy of convex relaxations to be used for loss-augmented MAP-MRF inference within a max-margin learning approach. Furthermore, it can be easily adapted to take advantage of the special structure that may be present in a given class of MRFs. We demonstrate the generality and flexibility of our approach by testing it on a variety of scenarios, including training of pairwise and higher-order MRFs, training by using different types of regularizers and/or different types of dissimilarity loss functions, as well as by learning of appropriate models for a variety of vision tasks (including high-order models for compact pose-invariant shape priors, knowledge-based segmentation, image denoising, stereo matching as well as high-order Potts MRFs).

Key-words: Markov Random Fields, Conditional Random Fields, structured prediction, parameter estimation, graphical models, convex optimization

* This work was partially supported by the ERC Starting Grant DIOCLES (ERC-STG-259112).
† Universite Paris-Est, Ecole des Ponts ParisTech, France
‡ Laboratoire d'Informatique Gaspard Monge, UMR 8049, CNRS, France
§ CVN - Center for Visual Computing, École Centrale de Paris, France
¶ Equipe GALEN, INRIA Saclay - Île de France, Orsay, France

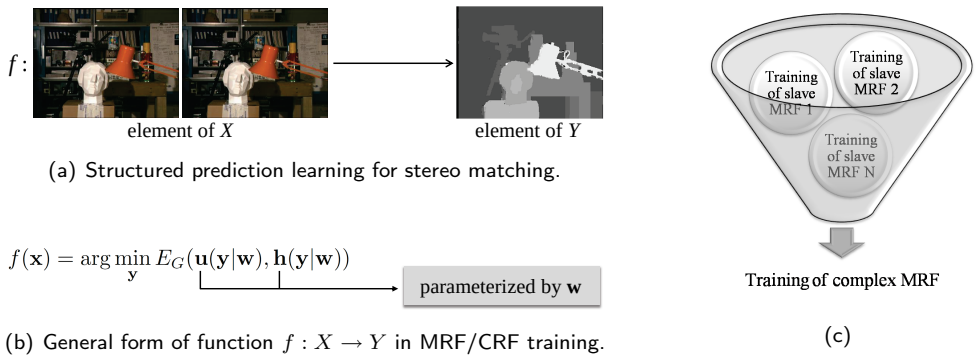
RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE

Parc Orsay Université
4 rue Jacques Monod
91893 Orsay Cedex

Un cadre efficace pour l'apprentissage structuré à marge maximale (SVM) de modèles de MRF d'ordres supérieurs

Résumé : Nous présentons un algorithme très général pour l'apprentissage structuré capable de gérer des MRFs ou CRFs discrets (pouvant contenir des hyper-arêtes d'ordre supérieur) tant que ces derniers admettent une décomposition en sous-problèmes optimisables. Fondamentalement, l'algorithme utilise le principe de la décomposition duale qui a été employée récemment pour l'optimisation de MRF. En combinant intelligemment cette approche avec une méthode d'apprentissage à marge maximale, on arrive à réduire l'entraînement d'un MRF d'ordre supérieur à l'entraînement parallèle d'une série de sous-problèmes simples bien plus facile à gérer. Nous avons alors un cadre pour l'apprentissage général et très efficace, qui s'appuie sur des principes mathématiques solides. Nous étudierons en détails ses propriétés théoriques et montrerons que l'on peut obtenir des algorithmes d'apprentissage de plus en plus précis car il admet naturellement une hiérarchie de relaxations convexes utilisables pour l'inférence de MAP-MRF avec augmentation de la fonction de perte (loss-augmented inference), le tout dans une approche d'apprentissage à marge maximale. De plus, l'approche peut être facilement adaptée pour utiliser la structure particulière d'une classe de MRFs. Nous démontrerons la généralité et la flexibilité de notre approche en la testant sur une variété de scénarios dont l'entraînement de MRF d'ordre 1 puis supérieurs, l'entraînement avec différents types de régularisations et différents type de fonctions de perte, et l'apprentissage de modèles appropriés pour de nombreuses tâches concernant la vision (ordres supérieurs pour des aprioris de forme compacts et invariants par rapport à la pose, segmentation, débruitage d'images, appariement stéréo et MRFs avec des potentiels de Potts d'ordres supérieurs).

Mots-clés : Champs aléatoires de Markov, champs conditionnels aléatoires, prédiction structurée, estimation de paramètre, modèle graphique, optimisation convexe



(a) Structured prediction learning for stereo matching.

(b) General form of function $f : X \rightarrow Y$ in MRF/CRF training.

(c)

Fig. 1: (a) In MRF/CRF training, one aims to learn a mapping $f : X \rightarrow Y$ between a typically high-dimensional input space X and an output space of MRF/CRF variables Y. In stereo matching, for instance, the elements of the input space X correspond to stereoscopic images, and the elements of the output space Y correspond to disparity maps. **(b)** In general, the mapping $f(\mathbf{x})$ is defined as minimizing the energy $E_G(\mathbf{u}(\mathbf{y}|\mathbf{w}), \mathbf{h}(\mathbf{y}|\mathbf{w}))$ of an MRF/CRF model whose unary and higher-order potentials $\mathbf{u}(\mathbf{y}|\mathbf{w})$, $\mathbf{h}(\mathbf{y}|\mathbf{w})$ are parameterized by $\mathbf{w}$ (the potentials also depend on $\mathbf{x}$, but this is omitted here to simplify notation). Therefore, to fully specify this mapping it suffices to estimate $\mathbf{w}$, which is what parameter learning aims to achieve in this case. **(c)** Our framework reduces, in a principled manner, the training of a complex MRF model into the parallel training of a series of easy-to-handle slave MRFs. The latter can be freely chosen so as to fully exploit the problem structure, which, in addition to efficiency, contributes a sufficient amount of flexibility and generality to our method.

1 Introduction

Markov Random Fields (MRFs), and their discriminative counterparts Conditional Random Fields (CRFs)[1] [36], are ubiquitous in computer vision and image analysis [5, 37]. They have been used with great success in a variety of applications so far, including both low-level and high-level problems from the above domains [14, 22, 32, 71]. Due to this fact, algorithms that perform MAP estimation for models of this type have attracted a significant amount of research interest in the computer vision community over the past years [8, 19, 29, 30, 31, 58]. However, besides the ability to accurately minimize the energy of a MRF model, another extremely crucial issue is how to actually select this energy in the first place, such that the resulting model yields an accurate representation of a specific problem that one aims to solve (a MAP-MRF solution is of little value if the used MRF model does not properly represent the problem at hand). It turns out that one of the most successful and principled ways for achieving this goal is through learning. In such a context, one proceeds by parameterizing the potentials of a MRF model by a vector of parameters $\mathbf{w}$, and, then, these parameters are estimated automatically by making use of training data that are given as input. For many cases in vision, this is, in fact, the only viable solution as the existing parameters can often be too many to tune by hand (*e.g.*, deformable parts-based models for object detection can have thousands of parameters to estimate).

As a result, learning algorithms for MRF parameter estimation play a fundamental role in successfully applying MRF models to computer vision problems. However, training these models poses a task that is quite challenging. This is because, unlike standard machine learning tasks where one must learn functions predicting simple true-false answers or scalar values (as in classification and regression), the goal, in this case, is to learn models that predict answers

[1] The terms Markov Random Fields (MRFs) and Conditional Random Fields (CRFs) will be used interchangeably throughout.

much more complex consisting of multiple interrelated variables. In fact, this is a characteristic example of what is known as *structured prediction learning*, where one uses a set of input-output training pairs $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{1 \leq k \leq K} \subseteq X \times Y$ to estimate a function $f : X \rightarrow Y$ that has the following characteristics: both the input and output spaces X, Y are high-dimensional, and, furthermore, the variables in Y are interrelated, *i.e.*, each element $\mathbf{y} \in Y$ carries out some structure (for instance, it can represent a graph). In the particular case of MRF parameter estimation, X is representing the space where the observations (*e.g.*, the input visual data) reside, whereas Y is representing the space of the variables of the MRF model (see Fig. 1(a), 1(b)).

In fact, the difficulty of the above task becomes even greater due to the computational challenges that are often raised by computer vision applications with regard to learning. For instance, many of the MRFs used in vision are of large scale. Also, the complexity and diversity of vision tasks often require the training of MRFs with complex potential functions. On top of that, over the last years the use of high order MRFs is becoming increasingly popular in vision since such MRFs are often found to considerably improve the quality of estimated solutions. Yet, most of the MRF learning methods proposed in the vision literature so far focus mainly on models with pairwise potentials or on specific classes of high-order models for which they need to derive specifically tailored algorithms [1, 2, 34, 40, 44, 53, 59].

The goal of this work is to address the above mentioned challenges by proposing a general learning method that can be directly applicable to a very broad class of problems. To achieve this goal the proposed method makes use of some recent advances made on the MRF optimization side [27, 28], which it combines with a max-margin approach for learning [63]. More specifically, it makes use of a dual decomposition approach [28] that has been previously used for MAP estimation. Thanks to this approach, it essentially manages to reduce the task of training a complex MRF to that of training in parallel a series of simpler slave MRFs that are much easier to handle within a max-margin framework (Fig. 1(c)). The concurrent training of the slave MRFs takes place in a principled way through an efficient projected subgradient algorithm. This leads to a powerful learning framework that makes the following contributions compared to prior art:

1. It is able to efficiently handle not just pairwise log-linear MRF models but also high-order ones as long as the latter can admit a decomposition into tractable subproblems, in which case no other restriction needs to be imposed on the topology of the underlying MRF graph or on the type of MRF potentials.

2. Thanks to the parallel training of a series of easy-to-handle submodels in combination with the used projected subgradient method, it leads to a highly efficient learning scheme that is scalable even to very large problems. Moreover, unlike prior cutting-plane or primal subgradient descent methods for max-margin learning, which require performing loss-augmented MAP-MRF inference to completion at every iteration, the proposed scheme is able to jointly optimize both the vector of parameters and the loss-augmented MRF inference variables.

3. It allows a hierarchy of convex relaxations for MAP-MRF estimation to be used in the context of learning for structured prediction (where this hierarchy includes all the commonly used LP relaxations for MRF inference), thus leading to structured prediction learning algorithms of increasing accuracy.

4. It is sufficiently flexible and extendable, as it only requires providing a routine that computes an optimizer for the slave MRFs. As a result, it can be easily adapted to take advantage of the special structure that may exist in a given class of MRF models to be trained.

The present paper is based on our previous work [24]. Compared to that work, here we also provide a more detailed mathematical and theoretical analysis of our method as well as a significantly extended set of experimental results, including results for learning pose invariant models, for knowledge-basesd segmentation (both on 2D and 3D cases), for training using high-order loss functions, as well as for training using sparsity inducing regularizers.

2 Related work

Over the past years, structured prediction learning has been a topic that has attracted a significant amount of interest both from the vision and machine learning community. There is, therefore, a substantial body of related work in this area.

Many approaches on this topic can essentially be derived from, or are based on, the so-called *regularized risk minimization* paradigm, where one is given a set of training samples $\big\{(\mathbf{x}_k, \mathbf{y}_k)\big\}_{1\leq k\leq K} \subseteq X \times Y$ (assumed to be generated by some unknown distribution on $X \times Y$) and seeks to estimate the parameters $\mathbf{w}$ of a graphical model, such as a Markov Random Field, by minimizing an objective function of the following form

$$\min_{\mathbf{w}} R(\mathbf{w}) + C \sum_{k=1}^{K} \mathcal{L}(\mathbf{y}^k, \widehat{\mathbf{y}}^k(\mathbf{x}^k|\mathbf{w})) \ . \tag{1}$$

In the above, $\mathbf{y}^k$ denotes the desired (i.e., ground truth) MRF labeling of the k-th training sample, $\widehat{\mathbf{y}}^k(\mathbf{x}^k|\mathbf{w})$ denotes the corresponding labeling that results from minimizing an MRF instance constructed from the input $\mathbf{x}^k$ and parameterized by $\mathbf{w}$, and $\mathcal{L}(\cdot,\cdot)$ is a loss function used for incurring a penalty if there exist differences between the two solutions $\mathbf{y}^k$ and $\widehat{\mathbf{y}}^k(\mathbf{x}^k|\mathbf{w})$. In view of this notation, the second term in (1) represents essentially an empirical risk that is used for approximating the true risk, which cannot be computed due to the fact that the joint distribution on the input-output pairs $(\mathbf{x}, \mathbf{y}) \in X \times Y$ is not known. The above approximation of the true risk is equal to the average of the loss on the input training samples, which is combined in (1) with a regularizer $R(\mathbf{w})$, whose main role is essentially to prevent overfitting (the relative importance of the two terms, *i.e.*, the regularizer and the empirical risk, is determined by the regularization constant C in (1)).

Depending on the choice made for the loss function $\mathcal{L}(\cdot,\cdot)$, different types of structured prediction learning methods can be recovered, including both generative (e.g., maximum-likelihood) and discriminative (e.g., max-margin) algorithms, which comprise the two most general and widely used learning approaches. In the case of maximum-likelihood learning, one maximizes (possibly along with an L2 norm regularization term) the product of posterior probabilities of the ground truth MRF labelings $\prod_k P(\mathbf{y}^k|\mathbf{w})$, where $P(\mathbf{y}|\mathbf{w}) \propto \exp\big(-E(\mathbf{y}|\mathbf{w})\big)$ denotes the probability distribution induced by an MRF model with energy $E(\mathbf{y}|\mathbf{w})$. This leads to a convex differentiable objective function that can be optimized using gradient ascent. However, in the case of log-linear models, it is known that computing the gradient of this function involves taking expectations (of some appropriate feature functions) with respect to the MRF distribution $P(\mathbf{y}|\mathbf{w})$. This, therefore, requires performing probabilistic MRF inference, which is, in general, an intractable task. As a result, approximate inference techniques (such as the loopy belief propagation algorithm [45]) are often used for approximating the MRF marginals required for the estimation of the gradient. This is, e.g., the case in [53], where the authors demonstrate how to train a CRF model for stereo matching, as well as in [34], where a comparison with other MRF training methods such as the pseudo-likelihood [4], [35] and MCMC-based contrastive divergence [18] are included as well. A disadvantage, of course, of having to use approximate

probabilistic inference techniques is that the estimation of the gradient is incorrect and so it is difficult for these methods to provide any theoretical guarantees.

Besides maximum-likelihood, another widely used class of structured prediction learning techniques, the so-called max-margin learning methods, can be derived from (1) by choosing a hinge-loss term as the loss function $\mathcal{L}(\cdot,\cdot)$. In this case, it turns out that the goal of the resulting optimization problem is to adjust the MRF parameters $\mathbf{w}$ so that, ideally, there is at least a non-negative margin attained between the energy attained by the ground truth solution of a training sample and the energy of any other solution.

When $R(\mathbf{w}) = ||\mathbf{w}||^2$, such a problem is equivalent to a convex quadratic program (QP) with an exponential number of linear inequality constraints. One class of methods [12, 38, 69] try to solve this QP by use of a cutting-plane approach. These methods rely on the core idea that only a very small fraction of the exponentially many constraints will actually be active at an optimal solution. Therefore, they proceed by solving a small QP whose number of constraints increases at each iteration. The increase, in this case, takes place by finding and adding the most violated constraints each time (still, the total number of constraints can be shown to be polynomially upper-bounded). However, one drawback of such an approach relates to the fact that computing the most violated constraint requires solving at each iteration a loss-augmented MAP-MRF inference problem that is, in general, NP-hard. Therefore, one still has to resort to approximate MAP inference techniques. This can lead to the so-called *under-generating* or *over-generating* approaches depending on the type of approximate inference used during this step. The former approaches rely on algorithms that consider only a subset of all possible solutions for the loss-augmented MAP-MRF inference step. As a consequence, solutions that are not considered do not get penalized during training. In contrast, the latter approaches make use of algorithms that consider a superset of the valid solutions. This typically means also penalizing fractional solutions corresponding to a relaxation of the loss-augmented MAP-MRF inference problem, thus promoting the extraction of a valid integral solution at test time. Due to this fact, overgenerating approaches are typically found to have much better empirical performance [12].

Crucially, however, both undergenerating and overgenerating approaches typically impose great computational cost during training, especially for problems of large scale or high order that are frequently encountered in computer vision, due to the fact that the MAP inference process has to be performed at the level of full size MRFs at each iteration. Note that this a very important issue that appears in other existing methods as well, e.g., [51]. An exception perhaps is the special case of submodular MRFs, for which the authors of [2] have shown how to express the exponential set of constraints in a compact form, thus allowing for a more efficient MRF training to take place under this setting.

The method proposed in this paper aims to address the aforementioned shortcomings. It belongs to the class of overgenerating training methods. Among other methods of this type, the approach closest to our work is [40], where the authors choose to replace the structured hinge-loss for pairwise MRFs by a convex dual upper bound that decomposes over the MRF cliques (the specific dual bound that has been used in this case is the one that was first employed in the context of the max-sum diffusion algorithm [65]). That work, however, focuses on the training of pairwise MRFs, but it can potentially be extended to higher-order models by properly adapting the dual bound of [65] and deriving corresponding block-coordinate dual ascent methods. Our method, on the other hand, handles *directly* in a *unified*, elegant and modular manner high-order models, models that employ tighter relaxations for improved accuracy, higher-order loss functions, as well as models with any type of special characteristics (*e.g.*, submodularity). Furthermore, [40] is theoretically valid, and thus applicable, only to problems with a strictly convex regularizer such as the squared l_2-norm. In contrast, our approach handles any convex regularizer (including ones based on sparsity inducing norms - *e.g.*, l_1 - that have often proved to be very useful

during learning), offering guaranteed convergence in all cases. Moreover, an additional advantage compared to [40] is that our method is parallelizable, as it allows all of the optimizers for the slave MRFs to be computed concurrently (instead of sequentially). One other max-margin training method that replaces the loss-augmented inference step by a compact dual LP relaxation is the approach proposed in [13] . However, this is done only for a restricted class of MRF problems (those with a strictly trivial equivalent), for which the LP relaxation is assumed to be equivalent to the original MRF optimization. An additional CRF learning method that makes use of duality is [17], which proposes an approximation for the CRF structured-prediction problem based on a local entropy approximation and derives an efficient message-passing algorithm with guaranteed convergence. Similarly to our method and [40], the method proposed in [17] breaks down the classical separation between inference and learning, and tries to directly formulate the learning problem via message passing operations, but uses different dual formulations and optimization techniques.

It should be mentioned at this point that, over the last years, additional types of structured prediction training methods have been proposed that can make use of various other types of learning objective functions and losses, as well as optimization algorithms [10, 15, 39, 41, 42, 47, 49, 50, 60, 62]. This also includes recent cases such as the inference-machines framework proposed in [43], as well as various types of randomized models such as the "Perturb-and-MAP" framework [48] or the "randomized optimum models" described in [61]. Also, a pseudo-max approach to structured learning (inspired by the pseudo-likelihood method) is proposed in [57], where the authors also analyze for which cases such an approach leads to consistent training. Furthermore, learning algorithms that can handle graphical models with hidden variables have been recently proposed as well, in which case it is assumed that only partial ground truth labelings are given as input during training [11, 25, 33, 55, 70]. Last, but not least, another strand of work focuses on developing learning approaches for the case of continuously valued MRF problems [52].

The remainder of this paper is structured as follows. We begin by briefly reviewing the dual decomposition method for MAP estimation in §3. We also review the max-margin structured prediction approach in §4. We describe in detail our MRF learning framework and also thoroughly analyze various aspects of it in §5-§7. We show experimental results for a variety of different settings and tasks in §8. Finally, we present our conclusions in §9.

3 MRF Optimization via Dual Decomposition

Let $\mathcal{L}$ denote a discrete label set, and let $G = (\mathcal{V}, \mathcal{C})$ be a hypergraph consisting of a set of nodes $\mathcal{V}$ and a set of hyperedges[2] $\mathcal{C}$. A discrete MRF defined on the hypergraph G is specified by its so-called unary and higher-order potential functions $\mathbf{u} = \{u_p\}_{p\in\mathcal{V}}$ and $\mathbf{h} = \{h_c\}_{c\in\mathcal{C}}$ respectively (where, for every $p \in \mathcal{V}$ and $c \in \mathcal{C}$, $u_p : \mathcal{L} \to \mathbb{R}$ and $h_c : \mathcal{L}^{|c|} \to \mathbb{R}$). If $\mathbf{y} = \{y_p\}_{p\in\mathcal{V}} \in \mathcal{L}^{|\mathcal{V}|}$ represents a labeling of the nodes in $\mathcal{V}$, the values $\mathbf{u}(\mathbf{y}) = \{u_p(y_p)\}_{p\in\mathcal{V}}$ and $\mathbf{h}(\mathbf{y}) = \{h_c(\mathbf{y}_c)\}_{c\in\mathcal{C}}$ of the above potential functions (where $\mathbf{y}_c$ denotes the set $\{y_p | p \in c\}$) define the MRF energy of $\mathbf{y}$ as

$$E_G(\mathbf{u}(\mathbf{y}), \mathbf{h}(\mathbf{y})) := \sum_{p\in\mathcal{V}} u_p(y_p) + \sum_{c\in\mathcal{C}} h_c(\mathbf{y}_c) \ . \tag{2}$$

In MRF optimization the goal is to find a labeling $\mathbf{y}$ that attains the minimum of the above energy function, which amounts to solving the following task

$$\min_{\mathbf{y}\in\mathcal{L}^{|\mathcal{V}|}} E_G(\mathbf{u}(\mathbf{y}), \mathbf{h}(\mathbf{y})) \ . \tag{3}$$

[2] A hyperedge (or clique) c of a hypergraph $G = (\mathcal{V}, \mathcal{C})$ is simply a subset of the nodes $\mathcal{V}$, *i.e.*, $c \subseteq \mathcal{V}$.

The above problem is, in general, NP-hard. One common way to compute approximately optimal solutions to it is by making use of convex relaxations. The dual decomposition framework in [21,23,28] provides a very general and flexible method for deriving and solving tight dual relaxations in this case. According to this framework, a set $\{G_i = (\mathcal{V}_i, \mathcal{C}_i)\}_{1 \leq i \leq N}$ of sub-hypergraphs of the original hypergraph $G = (\mathcal{V}, \mathcal{C})$ is first chosen such that $\mathcal{V} = \cup_{i=1}^N \mathcal{V}_i$, $\mathcal{C} = \cup_{i=1}^N \mathcal{C}_i$. The original hard optimization problem $\min_{\mathbf{y}} E_G(\mathbf{u}(\mathbf{y}), \mathbf{h}(\mathbf{y}))$ (also called the *master*) is then decomposed into a set of easier to solve subproblems $\{\min_{\mathbf{y}} E_{G_i}(\mathbf{u}^i(\mathbf{y}), \mathbf{h}(\mathbf{y}))\}_{1 \leq i \leq N}$ (called the *slaves*), which involve optimizing local MRFs defined on the chosen sub-hypergraphs $\{G_i\}_{1 \leq i \leq N}$. As can be seen, each slave MRF inherits[3] the higher-order potentials $\mathbf{h}$ of the master MRF, but has its own unary potentials $\mathbf{u}^i = \{u_p^i\}_{p \in \mathcal{V}_i}$. Its energy function is thus given by

$$E_{G_i}(\mathbf{u}^i(\mathbf{y}), \mathbf{h}(\mathbf{y})) := \sum_{p \in \mathcal{V}_i} u_p^i(y_p) + \sum_{c \in \mathcal{C}_i} h_c(\mathbf{y}_c) \ .$$

The condition that the above unary potentials $\mathbf{u}^i$ have to satisfy is the following

$$\sum_{i \in \mathcal{I}_p} u_p^i = u_p \ , \ \forall p \in \mathcal{V} \ , \tag{4}$$

where $\mathcal{I}_p$ denotes the set of indices of all sub-hypergraphs containing node p, *i.e.*,

$$\mathcal{I}_p = \{i | p \in \mathcal{V}_i\} \ . \tag{5}$$

The above property simply expresses the fact that the sum of the unary potentials of the slaves should give back the unary potentials of the master MRF. Due to this property, the sum of the minimum energies of the slaves can be shown to always provide a lower bound to the minimum energy of the master MRF, *i.e.*, it holds

$$\sum_{i=1}^N \min_{\mathbf{y}} E_{G_i}(\mathbf{u}^i(\mathbf{y}), \mathbf{h}(\mathbf{y})) \leq \min_{\mathbf{y}} E_G(\mathbf{u}(\mathbf{y}), \mathbf{h}(\mathbf{y})) \ . \tag{6}$$

Maximizing the lower bound appearing on the left-hand side of (6) by adjusting the unary potentials $\{\mathbf{u}^i\}_{1 \leq i \leq N}$ (which play the role of dual variables in this case) gives rise to the following dual relaxation for problem (3)

$$\text{DUAL}_{\{G_i\}}(\mathbf{u}, \mathbf{h}) = \max_{\{\mathbf{u}^i\}_{1 \leq i \leq N}} \sum_{i=1}^N \min_{\mathbf{y}} E_{G_i}(\mathbf{u}^i(\mathbf{y}), \mathbf{h}(\mathbf{y})) \tag{7}$$

$$\text{s.t.} \sum_{i \in \mathcal{I}_p} u_p^i = u_p \ , \ (\forall p \in \mathcal{V}) \ . \tag{8}$$

By simply choosing different decompositions $\{G_i\}_{1 \leq i \leq N}$ of the hypergraph G, one can derive different convex relaxations to problem (3). These include the standard marginal polytope LP relaxation for pairwise MRFs, which is widely used in practice, as well as alternative relaxations that can be much tighter[4].

[3] Slave MRFs could also have non-inherited high-order potentials. Here we consider only the case where just the unary potentials are non-inherited to simplify notation.

[4] We should note, though, that none of these relaxations are guaranteed to be exact in the general case.

4 Max-margin Markov Networks

Let us now return to the central topic of the paper, which is the training of MRF/CRF models. To that end, let $\left\{\mathbf{x}^k, \mathbf{y}^k\right\}_{1\leq k\leq K} \in X \times Y$ be a training set of K samples, where $\mathbf{x}^k$, $\mathbf{y}^k$ represent the input observations and the label assignments of the k-th sample, respectively. We assume that the MRF instance associated with the k-th sample is defined on a hypergraph[5] $G = (\mathcal{V}, \mathcal{C})$, and both the unary potentials $\mathbf{u}^k = \left\{u_p^k\right\}_{p\in\mathcal{V}}$ and the higher-order potentials $\mathbf{h}^k = \left\{h_c^k\right\}_{c\in\mathcal{C}}$ of that MRF are parameterized linearly in terms of a vector of parameters $\mathbf{w}$ we seek to estimate, *i.e.*,

$$u_p^k(y_p|\mathbf{w}) = \mathbf{w}^T \cdot \phi_p(y_p, \mathbf{x}^k), \;\; h_c^k(\mathbf{y}_c|\mathbf{w}) = \mathbf{w}^T \cdot \phi_c(\mathbf{y}_c, \mathbf{x}^k) \;\; , \tag{9}$$

where $\phi_p(\cdot,\cdot)$, $\phi_c(\cdot,\cdot)$ represent known vector-valued feature functions that are extracted from the corresponding observations $\mathbf{x}^k$ (and are application-specific). Note that, by properly zero-padding these vector-valued features $\phi_p(\cdot,\cdot)$ and $\phi_c(\cdot,\cdot)$, the above formulation allows us to use separate parameters for each different node, clique or even label[6].

Let $\Delta(\mathbf{y}, \mathbf{y}')$ represents a dissimilarity measure between any two MRF labelings $\mathbf{y}$ and $\mathbf{y}'$ (that satisfies $\Delta(\mathbf{y}, \mathbf{y}') \geq 0$ and $\Delta(\mathbf{y}, \mathbf{y}) = 0$). In a maximum margin Markov network [63] one ideally seeks a vector of parameters $\mathbf{w}$ such that the MRF energy of the desired ground-truth solution $\mathbf{y}^k$ is smaller by a margin $\Delta(\mathbf{y}, \mathbf{y}^k)$ than the MRF energy of any other solution $\mathbf{y}$, *i.e.*,

$$(\forall \mathbf{y}), \;\; E_G(\mathbf{u}^k(\mathbf{y}^k|\mathbf{w}), \mathbf{h}^k(\mathbf{y}^k|\mathbf{w})) \leq E_G(\mathbf{u}^k(\mathbf{y}|\mathbf{w}), \mathbf{h}^k(\mathbf{y}|\mathbf{w})) - \Delta(\mathbf{y}, \mathbf{y}^k) \;\; . \tag{10}$$

To account for the fact that there might be no vector $\mathbf{w}$ satisfying all of the above constraints, a slack variable ξ_k per sample is introduced that allows some of the constraints to be violated

$$(\forall \mathbf{y}), \;\; E_G(\mathbf{u}^k(\mathbf{y}^k|\mathbf{w}), \mathbf{h}^k(\mathbf{y}^k|\mathbf{w})) \leq E_G(\mathbf{u}^k(\mathbf{y}|\mathbf{w}), \mathbf{h}^k(\mathbf{y}|\mathbf{w})) - \Delta(\mathbf{y}, \mathbf{y}^k) + \xi_k \;\; . \tag{11}$$

Ideally, ξ_k should take a zero value. In general, however, it can hold $\xi_k > 0$ and so the goal, in this case, is to adjust $\mathbf{w}$ such that the sum $\sum_{k=1}^K \xi_k$ (which represents the total violation of constraints (10)) takes a value that is as small as possible. This leads to solving the following constrained minimization problem, where a regularization term $R(\mathbf{w})$ has been also added so as to prevent the components of $\mathbf{w}$ from taking too large values

$$\min_{\mathbf{w}} \; R(\mathbf{w}) + C \sum_{k=1}^{K} \xi_k \tag{12}$$

$$\text{s.t. } \xi_k \geq E_G(\mathbf{u}^k(\mathbf{y}^k|\mathbf{w}), \mathbf{h}^k(\mathbf{y}^k|\mathbf{w})) - \big(E_G(\mathbf{u}^k(\mathbf{y}|\mathbf{w}), \mathbf{h}^k(\mathbf{y}|\mathbf{w})) - \Delta(\mathbf{y}, \mathbf{y}^k)\big), \;\; (\forall \mathbf{y}) \tag{13}$$

The term $R(\mathbf{w})$ can be chosen in several different ways (for instance, it is often set as a squared Euclidean norm $\frac{1}{2}||\mathbf{w}||^2$, or as a sparsity inducing norm like$||\mathbf{w}||_1$).

It is easy to see that at an optimal solution of problem (12) each variable ξ_k should equal to

$$\xi_k = E_G(\mathbf{u}^k(\mathbf{y}^k|\mathbf{w}), \mathbf{h}^k(\mathbf{y}^k|\mathbf{w})) - \min_{\mathbf{y}} \big(E_G(\mathbf{u}^k(\mathbf{y}|\mathbf{w}), \mathbf{h}^k(\mathbf{y}|\mathbf{w})) - \Delta(\mathbf{y}, \mathbf{y}^k)\big) \, . \tag{14}$$

Furthermore, assuming that the dissimilarity measure $\Delta(\mathbf{y}, \mathbf{y}^k)$ decomposes in the same way as the MRF energy, *i.e.*, it holds

$$\Delta(\mathbf{y}, \mathbf{y}^k) = \sum_{p\in\mathcal{V}} \delta_p(y_p, y_p^k) + \sum_{c\in\mathcal{C}} \delta_c(\mathbf{y}_c, \mathbf{y}_c^k) \;\; , \tag{15}$$

[5] In general, each MRF training instance can be defined on a different hypergraph $G^k = (\mathcal{V}^k, \mathcal{C}^k)$, but here we assume $G^k = G, \forall k$ in order to reduce notation clutter.

[6] For instance, if $u_p^k(y_p|\mathbf{w}) = \mathbf{w}_{p,y_p}^T \cdot \tilde{\phi}_p(y_p, \mathbf{x}^k)$ and $h_c^k(\mathbf{y}_c|\mathbf{w}) = \mathbf{w}_{c,\mathbf{y}_c}^T \cdot \tilde{\phi}_c(\mathbf{y}_c, \mathbf{x}^k)$, we can define $\mathbf{w}$ as the concatenation of all vectors $\left\{\mathbf{w}_{p,y_p}\right\}$ and $\left\{\mathbf{w}_{c,\mathbf{y}_c}\right\}$, in which case each feature vector $\phi_p(y_p, \mathbf{x}^k)$ should be defined as a properly zero-padded extension of $\tilde{\phi}_p(y_p, \mathbf{x}^k)$ that has the same size as $\mathbf{w}$ (and similarly for $\phi_c(\mathbf{y}_c, \mathbf{x}^k)$).

we can define the following *loss-augmented* MRF potentials $\bar{\mathbf{u}}^k(\mathbf{y}|\mathbf{w})$, $\bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w})$

$$\bar{u}_p^k(\cdot\,|\mathbf{w}) = u_p^k(\cdot\,|\mathbf{w}) - \delta_p(\cdot\,, y_p^k) \tag{16}$$
$$\bar{h}_c^k(\cdot\,|\mathbf{w}) = h_c^k(\cdot\,|\mathbf{w}) - \delta_c(\cdot\,, \mathbf{y}_c^k)\ , \tag{17}$$

which allow expressing the slack variable in (14) as $\xi_k = L_G^k(\mathbf{w})$, with $L_G^k(\mathbf{w})$ being defined as the following hinge loss term

$$L_G^k(\mathbf{w}) := E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \min_{\mathbf{y}} E_G(\bar{\mathbf{u}}^k(\mathbf{y}|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))\ . \tag{18}$$

Therefore, problem (12) finally reduces to the following unconstrained optimization task

$$\min_{\mathbf{w}} R(\mathbf{w}) + C \sum_{k=1}^{K} L_G^k(\mathbf{w})\ , \tag{19}$$

which shows that maximum-margin learning essentially corresponds to using the hinge loss term $L_G^k(\mathbf{w})$ as the loss $\mathcal{L}(\mathbf{y}^k, \widehat{\mathbf{y}}^k(\mathbf{w}))$ in the empirical minimization task (1). Intuitively, this term $L_G^k(\mathbf{w})$ expresses the fact that the loss will be zero only when the loss-augmented MRF with potentials $\bar{\mathbf{u}}^k$, $\bar{\mathbf{h}}^k$ attains its minimum energy at the desired solution $\mathbf{y}^k$.

5 Learning via Dual Decomposition

Unfortunately, even evaluating (let alone minimizing) the loss function $L_G^k(\mathbf{w})$ is going to be intractable in general. This is because it is NP-hard to compute the term $\min_{\mathbf{y}} E_G(\bar{\mathbf{u}}^k(\mathbf{y}|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))$ involved in the definition of $L_G^k(\mathbf{w})$ in (18). To address this fundamental difficulty, here we propose to approximate the above term (that involves computing the optimum energy of a loss-augmented MRF with potentials $\bar{\mathbf{u}}^k$, $\bar{\mathbf{h}}^k$) with the corresponding optimum of a convex relaxation $\text{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k)$ that is derived based on dual decomposition.

To accomplish that, as explained in section 3, we must first choose an arbitrary decomposition of the hypergraph $G = (\mathcal{V}, \mathcal{C})$ into sub-hypergraphs $\{G_i = (\mathcal{V}_i, \mathcal{C}_i)\}_{1 \le i \le N}$. Then, for the k-th training sample and for each sub-hypergraph G_i, we define a slave MRF on G_i that has its own unary potentials $\mathbf{u}^{k,i}$ while inheriting the higher-order potentials $\bar{\mathbf{h}}^k$. These slave MRFs are used for approximating $\min_{\mathbf{y}} E_G(\bar{\mathbf{u}}^k(\mathbf{y}|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))$ (*i.e.*, the minimum energy of the loss-augmented MRF of the k-th training sample) with the following convex relaxation $\text{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k)$

$$\text{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k) = \max_{\{\mathbf{u}^{k,i}\}_{1 \le i \le N}} \sum_{i=1}^{N} \min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w})) \tag{20}$$
$$\text{s.t.} \quad \sum_{i:\, p \in \mathcal{V}_i} u_p^{k,i} = \bar{u}_p^k\ ,\ \ (\forall p \in \mathcal{V}). \tag{21}$$

If we now replace in (18) the optimum $\min_{\mathbf{y}} E_G(\bar{\mathbf{u}}^k(\mathbf{y}|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))$ with the optimum of the

above convex relaxation $\text{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k)$, we get the following derivation

$$L_G^k(\mathbf{w}) = E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \min_{\mathbf{y}} E_G(\bar{\mathbf{u}}^k(\mathbf{y}|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w})) \tag{22}$$

$$\approx E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \text{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k) \tag{23}$$

$$= E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \max_{\{\mathbf{u}^{k,i}\}_{1\le i\le N}} \sum_{i=1}^{N} \min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w})) \tag{24}$$

$$= \min_{\{\mathbf{u}^{k,i}\}_{1\le i\le N}} \Big(E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \sum_{i=1}^{N} \min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w})) \Big), \tag{25}$$

where in (25) we have made use of the following identity that holds for any function f

$$-\max_{\{\mathbf{u}^{k,i}\}_{1\le i\le N}} f(\{\mathbf{u}^{k,i}\}) = \min_{\{\mathbf{u}^{k,i}\}_{1\le i\le N}} (-f(\{\mathbf{u}^{k,i}\})) \ .$$

Due to the fact that the dual variables $\mathbf{u}^{k,i}$ have to satisfy constraint (21), *i.e.*, $\bar{u}_p^k = \sum_{i:\, p\in\mathcal{V}_i} u_p^{k,i}$, the following equality stands in this case

$$E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) = \sum_{i=1}^{N} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}^k), \bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) \ . \tag{26}$$

By substituting this equality into (25), we finally get

$$L_G^k(\mathbf{w}) \approx \min_{\{\mathbf{u}^{k,i}\}_{1\le i\le N}} \sum_{i=1}^{N} \Big(E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}^k), \bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w})) \Big) \ . \tag{27}$$

Therefore, if we define

$$L_{G_i}^k(\mathbf{w}, \mathbf{u}^{k,i}) := E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}^k), \bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w})) \tag{28}$$

equation (27) translates into

$$L_G^k(\mathbf{w}) \approx \min_{\{\mathbf{u}^{k,i}\}_{1\le i\le N}} \sum_i L_{G_i}^k(\mathbf{w}, \mathbf{u}^{k,i}) \ . \tag{29}$$

Note that each $L_{G_i}^k(\mathbf{w}, \mathbf{u}^{k,i})$ corresponds to a hinge loss term that is exactly similar to $L_G^k(\mathbf{w})$ except from the fact that the former relates to a slave MRF on G_i with potentials $\mathbf{u}^{k,i}$, $\bar{\mathbf{h}}^k$, whereas the latter relates to an MRF on G with potentials $\bar{\mathbf{u}}^k$, $\bar{\mathbf{h}}^k$.

The final function to be minimized results from substituting (29) into (19), thus leading to the following optimization problem

$$\min_{\mathbf{w}, \{\mathbf{u}^{k,i}\}_{1\le k\le K, 1\le i\le N}} R(\mathbf{w}) + C \sum_{k=1}^{K} \sum_{i=1}^{N} L_{G_i}^k(\mathbf{w}, \mathbf{u}^{k,i}) \tag{30}$$

$$\text{s.t.} \quad \sum_{i:\, p\in\mathcal{V}_i} u_p^{k,i} = \bar{u}_p^k \ , \ \ (\forall k \in \{1, 2, \ldots, K\}, p \in \mathcal{V}). \tag{31}$$

As can be seen, the initial objective function (19) (which was intractable due to containing the hinge losses $L_G^k(\cdot)$) has now been decomposed into the hinge losses $L_{G_i}^k(\cdot)$ that are a lot easier to handle.

If a *projected subgradient* method [3] is used for minimizing the resulting convex function, we get algorithm 1, for which the following theorem holds true:

Theorem 1. *If multipliers $\alpha_t \geq 0$ satisfy $\lim_{t\to\infty} \alpha_t = 0$, $\sum_{t=0}^{\infty} \alpha_t = \infty$, then, the iterative updates in Algorithm 1 converge to an optimal solution of problem (30).*

Proof. We are going to make use of the following auxiliary variables $\boldsymbol{\lambda}^{k,i} = \{\lambda_p^{k,i}\}_{p\in\mathcal{V}_i}$, which are defined in terms of the variables $\mathbf{u}^{k,i} = \{u_p^{k,i}\}_{p\in\mathcal{V}_i}$ as follows

$$\lambda_p^{k,i} = u_p^{k,i} - \frac{u_p^k}{|\mathcal{I}_p|} \quad , \tag{32}$$

where $1 \leq k \leq K$, $1 \leq i \leq N$, and $\mathcal{I}_p = \{i | p \in \mathcal{V}_i\}$. In this case, constraints (31) map into constraints $\{\boldsymbol{\lambda}^{k,i}\}_{1\leq k\leq K, 1\leq i\leq N} \in \boldsymbol{\Lambda}$, where the set $\boldsymbol{\Lambda}$ is given by

$$\boldsymbol{\Lambda} = \left\{ \{\boldsymbol{\lambda}^{k,i}\}_{1\leq k\leq K, 1\leq i\leq N} | \sum\nolimits_{i\in\mathcal{I}_p} \lambda_p^{k,i} = 0 \right\} \quad . \tag{33}$$

To prove the theorem, we will proceed by showing that Algorithm 1 corresponds to applying the projected subgradient method to problem (30) with respect to variables $\mathbf{w}$, $\{\boldsymbol{\lambda}^{k,i}\}_{1\leq k\leq K, 1\leq i\leq N}$. According to the projected subgradient method, variables $\mathbf{w}$, $\left\{\boldsymbol{\lambda}^{k,i}\right\}_{1\leq k\leq K, 1\leq i\leq N}$ must be updated at each iteration using the following scheme [3, 56]

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha_t \cdot d\mathbf{w} \tag{34}$$

$$\boldsymbol{\lambda}^{k,i} \leftarrow \mathrm{Proj}_{\boldsymbol{\Lambda}}(\boldsymbol{\lambda}^{k,i} - \alpha_t \cdot d\boldsymbol{\lambda}^{k,i}) \quad . \tag{35}$$

In the above, $d\mathbf{w}$ and $\left\{d\boldsymbol{\lambda}^{k,i}\right\}_{1\leq k\leq K, 1\leq i\leq N}$ denote the components of a subgradient of the objective function (30) (with respect to $\mathbf{w}$ and $\left\{\boldsymbol{\lambda}^{k,i}\right\}_{1\leq k\leq K, 1\leq i\leq N}$, respectively), and $\mathrm{Proj}_{\boldsymbol{\Lambda}}(\cdot)$ denotes projection onto the feasible set $\boldsymbol{\Lambda}$.

To compute these components, a subgradient of function $L_{G_i}^k(\mathbf{w}, \mathbf{u}^{k,i})$ must be computed first, which in turns requires computing a subgradient of the term $-\min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))$, which is the only non-differentiable term used in the definition of $L_{G_i}^k(\mathbf{w}, \mathbf{u}^{k,i})$. This can be done by making use of the following well known lemma[7]

Lemma. *Let $f(\mathbf{x}) = \max_{m=1,\ldots,M} f_m(\mathbf{x})$, with f_m being convex and differentiable functions of $\mathbf{x}$. A subgradient of f at $\mathbf{x}_0$ is given by $\nabla f_{\hat{m}}(\mathbf{x}_0)$, where $\widehat{m} = \arg\max_m f_m(\mathbf{x}_0)$.*

Based on the above lemma, a subgradient of the term $-\min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))$ is given by the gradient vector $-\nabla E_{G_i}(\mathbf{u}^{k,i}(\widehat{\mathbf{y}}^{k,i}), \bar{\mathbf{h}}^k(\widehat{\mathbf{y}}^{k,i}|\mathbf{w}))$, where $\widehat{\mathbf{y}}^{k,i}$ denotes a minimizer for the energy $E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))$ of the slave MRF defined on sub-hypergraph G_i. This gradient vector has the following components:

$$\begin{aligned}
-\frac{\partial E_{G_i}(\mathbf{u}^{k,i}(\widehat{\mathbf{y}}^{k,i}), \bar{\mathbf{h}}^k(\widehat{\mathbf{y}}^{k,i}|\mathbf{w}))}{\partial \mathbf{w}} &= -\frac{\partial}{\partial \mathbf{w}} \left(\sum_{p\in\mathcal{V}_i} u_p^{k,i}(\widehat{y}_p^{k,i}) + \sum_{c\in\mathcal{C}_i} \bar{h}_c^k(\widehat{\mathbf{y}}_c^{k,i}|\mathbf{w}) \right) \\
&\overset{(32)}{=} -\frac{\partial}{\partial \mathbf{w}} \left(\sum_{p\in\mathcal{V}_i} \left(\frac{u_p^k(\widehat{y}_p^{k,i}|\mathbf{w})}{|\mathcal{I}_p|} + \lambda_p^{k,i}(\widehat{y}_p^{k,i}) \right) + \sum_{c\in\mathcal{C}_i} \bar{h}_c^k(\widehat{\mathbf{y}}_c^{k,i}|\mathbf{w}) \right) \\
&= -\frac{\partial}{\partial \mathbf{w}} \left(\sum_{p\in\mathcal{V}_i} \frac{u_p^k(\widehat{y}_p^{k,i}|\mathbf{w})}{|\mathcal{I}_p|} + \sum_{c\in\mathcal{C}_i} \bar{h}_c^k(\widehat{\mathbf{y}}_c^{k,i}|\mathbf{w}) \right)
\end{aligned}$$

[7] The proof of this lemma follows from the fact that $f(\mathbf{x}) \geq f_{\widehat{m}}(\mathbf{x}) \geq f_{\widehat{m}}(\mathbf{x}_0) + \nabla f_{\widehat{m}}(\mathbf{x}_0)^\top (\mathbf{x} - \mathbf{x}_0) = f(\mathbf{x}_0) + \nabla f_{\widehat{m}}(\mathbf{x}_0)^\top (\mathbf{x} - \mathbf{x}_0)$.

$$\overset{(9)}{=} -\sum_{p\in\mathcal{V}_i} \frac{\phi_p(\widehat{y}_p^{k,i},\mathbf{x}^k)}{|\mathcal{I}_p|} - \sum_{c\in\mathcal{C}_i} \phi_c(\widehat{\mathbf{y}}_c^{k,i},\mathbf{x}^k) \ ,$$

$$-\frac{\partial E_{G_i}(\mathbf{u}^{k,i}(\widehat{\mathbf{y}}^{k,i}),\bar{\mathbf{h}}^k(\widehat{\mathbf{y}}^{k,i}|\mathbf{w}))}{\partial\lambda_p^{k,i}(l)} = -\frac{\partial E_{G_i}(\mathbf{u}^{k,i}(\widehat{\mathbf{y}}^{k,i}),\bar{\mathbf{h}}^k(\widehat{\mathbf{y}}^{k,i}|\mathbf{w}))}{\partial u_p^{k,i}(l)} \cdot \frac{\partial u_p^{k,i}(l)}{\partial\lambda_p^{k,i}(l)}$$

$$= -\frac{\partial u_p^{k,i}(\widehat{y}_p^{k,i})}{\partial u_p^{k,i}(l)} \cdot \frac{\partial u_p^{k,i}(l)}{\partial\lambda_p^{k,i}(l)} \overset{(32)}{=} -\frac{\partial u_p^{k,i}(\widehat{y}_p^{k,i})}{\partial u_p^{k,i}(l)} = -\left[\widehat{y}_p^{k,i}=l\right] \ ,$$

where $[\,\cdot\,]$ denotes the Iverson bracket, *i.e.*, it equals 1 if the expression in square brackets is satisfied, and 0 otherwise. Based on the above result, it is easy to verify that the components $d\mathbf{w}$, $\left\{d\boldsymbol{\lambda}^{k,i}\right\}_{1\le k\le K, 1\le i\le N}$ of a total subgradient of the objective function (30) are given by

$$d\mathbf{w} = \nabla R(\mathbf{w}) + C\left(\sum_{k,i,p} \frac{\phi_p(y_p^k,\mathbf{x}^k) - \phi_p(\widehat{y}_p^{k,i},\mathbf{x}^k)}{|\mathcal{I}_p|} + \sum_{k,i,c}\left(\phi_c(\mathbf{y}_c^k,\mathbf{x}^k) - \phi_c(\widehat{\mathbf{y}}_c^{k,i},\mathbf{x}^k)\right)\right) \quad (36)$$

$$d\lambda_p^{k,i}(l) = C\left(\left[y_p^k=l\right] - \left[\widehat{y}_p^{k,i}=l\right]\right) \ . \quad (37)$$

Furthermore, after the update $\boldsymbol{\lambda}^{k,i} \leftarrow \boldsymbol{\lambda}^{k,i} - \alpha_t \cdot d\boldsymbol{\lambda}^{k,i}$, eq. (35) also requires projecting the resulting $\boldsymbol{\lambda}^{k,i}$ onto the feasible set $\boldsymbol{\Lambda}$. This projection is equivalent to subtracting $\left(\sum_{i\in\mathcal{I}_p}\lambda_p^{k,i}\right)/|\mathcal{I}_p|$ from each $\lambda_p^{k,i}$ such that the sum $\sum_{i\in\mathcal{I}_p}\lambda_p^{k,i}$ remains equal to zero as required by the definition of $\boldsymbol{\Lambda}$. Based on this observation and the definition of $d\boldsymbol{\lambda}^{k,i}$ given in eq. (37), the combined update (35) reduces to

$$\lambda_p^{k,i}(l) \mathrel{+}= \alpha_t C\left(\left[\widehat{y}_p^{k,i}=l\right] - \frac{\sum_{j\in\mathcal{I}_p}\left[\widehat{y}_p^{k,j}=l\right]}{|\mathcal{I}_p|}\right). \quad (38)$$

All the above lead to the pseudocode of algorithm 1.

The proof now follows directly from the fact that the subgradient updates are known to converge to an optimal solution if the multipliers α_t satisfy the conditions stated in the theorem (see Proposition 2.2 in [46]). Interestingly, the key quantity to proving the convergence of the subgradient method is not the objective function value (which may have temporary fluctuations), but the Euclidean distance to an optimal solution, which is guaranteed to decrease per iteration. The proof of this is based on the fact that the angle between the current subgradient and the vector formed by the difference of the current iterate with an optimal solution is less than 90 degrees. □

Let us pause for a moment to see what we have been able to accomplish so far. Comparing objective functions (19) and (30), we can immediately see that, thanks to the use of dual decomposition, we have managed to replace each term L_G^k, which is the hinge loss of a possibly difficult-to-solve high-order MRF on G, with the sum of the terms $\left\{L_{G_i}^k\right\}_{1\le i\le N}$ that are the hinge losses of a series of simpler slave MRFs on sub-hypergraphs $\left\{G_i^k\right\}_{1\le i\le N}$. In this manner, we have essentially been able to reduce the difficult task of training a complex high-order MRF to the much easier task of training *in parallel* a series of simpler slave MRFs.

At a high level, to achieve this goal, the resulting learning algorithm operates by allowing each slave MRF to have its own unary potentials $\mathbf{u}^{k,i}$ and by properly adjusting these potentials such that the estimated minimizer $\widehat{\mathbf{y}}^{k,i}$ of each slave MRF coincides with the desired ground truth solution $\mathbf{y}^k$ restricted on the nodes of G_i. This is essentially done via updates (38) and (34). To get a better intuition for the role of these updates, note that the aim of the former updates is to modify $\mathbf{u}^{k,i}$ such that the minimizers of different slave MRFs are consistent with

Algorithm 1 Pseudocode of learning via dual-decomposition.

Input:

Training samples $\{\mathbf{x}^k, \mathbf{y}^k\}_{1\leq k\leq K}$, hypergraph $G{=}(\mathcal{V},\mathcal{C})$, regularization constant C
Unary and high-order feature functions $\{\phi_p(\cdot,\cdot)\}_{p\in\mathcal{V}}$, $\{\phi_c(\cdot,\cdot)\}_{c\in\mathcal{C}}$

Learning procedure:

Choose decomposition $\{G_i{=}(\mathcal{V}_i,\mathcal{C}_i)\}_{1\leq i\leq N}$ of hypergraph G

$\forall k,i,p \;\; \lambda_p^{k,i} \leftarrow \mathbf{0} \;\;,\;\; u_p^{k,i} \leftarrow \lambda_p^{k,i} + \frac{u_p^k}{|\mathcal{I}_p|}$

repeat

$\mathcal{K} = \{1,2,\ldots,K\}$ // ***for stochastic learning, use*** $\mathcal{K} = \{$*pick randomly single index* $k \in \{1,2,\ldots,K\}\}$ *(see section 7)*

// ***compute minimizers of slave MRFs***

$\forall k \in \mathcal{K}, i, \quad \widehat{\mathbf{y}}^{k,i} = \arg\min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))$ // ***optimize MRF with potentials*** $\mathbf{u}^{k,i}$***,*** $\bar{\mathbf{h}}^k$

// ***update*** $\mathbf{w}$

$d\mathbf{w} \leftarrow \nabla R(\mathbf{w}) + C\left(\sum_{k\in\mathcal{K},i,p} \frac{\phi_p(y_p^k,\mathbf{x}^k) - \phi_p(\widehat{y}_p^{k,i},\mathbf{x}^k)}{|\mathcal{I}_p|} + \sum_{k,i,c}\left(\phi_c(\mathbf{y}_c^k,\mathbf{x}^k) - \phi_c(\widehat{\mathbf{y}}_c^{k,i},\mathbf{x}^k)\right)\right)$

$\mathbf{w} \leftarrow \mathbf{w} - \alpha_t \cdot d\mathbf{w}$

// ***update*** $\mathbf{u}^{k,i}$

$\forall k \in \mathcal{K}, i, p, l, \;\; \lambda_p^{k,i}(l) \mathrel{+}= \alpha_t C\left(\left[\widehat{y}_p^{k,i}{=}l\right] - \frac{\sum_{j\in\mathcal{I}_p}\left[\widehat{y}_p^{k,j}{=}l\right]}{|\mathcal{I}_p|}\right)$

$\forall k \in \mathcal{K}, i, p, \quad u_p^{k,i} \leftarrow \lambda_p^{k,i} + \frac{u_p^k}{|\mathcal{I}_p|}$

until convergence

each other (*i.e.*, they agree for the labels that are assigned to common nodes). Indeed, it is easy to verify that the right hand side of (38) equals to zero (which means that, as a result of this update, no change is applied to $\lambda_p^{k,i}$ and thus to $u_p^{k,i}$ as well) only if all minimizers of slave MRFs assign a common label to node p. On the contrary, if node p (contained, say, in only 2 sub-hypergraphs G_i, G_j) is assigned 2 different labels by the corresponding minimizers $\widehat{\mathbf{y}}^{k,i}$, $\widehat{\mathbf{y}}^{k,j}$ at the current iteration (*i.e.*, it holds $\widehat{y}_p^{k,i} \neq \widehat{y}_p^{k,j}$), then, update (38) results into the following updates for $u_p^{k,i}$ (where $\epsilon = \alpha_t C/2$)

$$u_p^{k,i}(\widehat{y}_p^{k,i}) \mathrel{+}= \epsilon \;\;,\;\; u_p^{k,j}(\widehat{y}_p^{k,i}) \mathrel{-}= \epsilon \;\;, \tag{39}$$

$$u_p^{k,i}(\widehat{y}_p^{k,j}) \mathrel{-}= \epsilon \;\;,\;\; u_p^{k,j}(\widehat{y}_p^{k,j}) \mathrel{+}= \epsilon \;\;. \tag{40}$$

The above updates can be seen as trying to encourage the slave MRF minimizers computed at the next iteration to satisfy $\widehat{y}_p^{k,i} = \widehat{y}_p^{k,j}$, *i.e.*, to assign a common label to node p. Furthermore, the role of the second updates (34) is exactly to encourage this common label to actually coincide with the ground truth label y_p^k.

6 Choice of decompositions $\{G_i\}_{1\leq i\leq N}$ and tighter approximations

The only requirement imposed by the above learning framework is that one should be able to compute the minimizers for the slave MRF subproblems. If this condition is satisfied, the previously described algorithm can automatically take care of the entire MRF training process. As a result of this fact, the proposed framework provides a great amount of flexibility. For

instance, a user is freely allowed to utilize different decompositions $\{G_i\}_{1\leq i\leq N}$. As we will explain next, this fact can be utilized for improving the learning algorithm in various ways.

To that end, let $\mathcal{F}_0$ denote the minimum of the original regularized loss function (19) and let $\mathcal{F}_{\{G_i\}}$ denote the minimum of loss function (30) that results from using decomposition $\{G_i\}$. The following theorem holds true

Theorem 2. *Loss $\mathcal{F}_{\{G_i\}}$ upper bounds $\mathcal{F}_0$,* i.e., $\mathcal{F}_0 \leq \mathcal{F}_{\{G_i\}}$

Proof. By definition (19) it holds that

$$\begin{aligned}
\mathcal{F}_0 &= \min_{\mathbf{w}} R(\mathbf{w}) + C\sum_{k=1}^{K} L_G^k(\mathbf{w}) \\
&\overset{(18)}{=} \min_{\mathbf{w}} R(\mathbf{w}) + C\sum_{k=1}^{K}\Big(E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}),\bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \min_{\mathbf{y}} E_G(\bar{\mathbf{u}}^k(\mathbf{y}|\mathbf{w}),\bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))\Big) && (41)\\
&\leq \min_{\mathbf{w}} R(\mathbf{w}) + C\sum_{k=1}^{K}\Big(E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}),\bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \mathrm{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k,\bar{\mathbf{h}}^k)\Big) && (42)\\
&= \min_{\mathbf{w}} R(\mathbf{w}) + C\sum_{k=1}^{K}\Big(E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}),\bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \max_{\{\mathbf{u}^{k,i}\}_{1\leq i\leq N}}\sum_{i=1}^{N}\min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}),\bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))\Big)\\
&= \min_{\mathbf{w}} R(\mathbf{w}) + C\sum_{k=1}^{K}\min_{\{\mathbf{u}^{k,i}\}_{1\leq i\leq N}}\Big(E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}),\bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \sum_{i=1}^{N}\min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}),\bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))\Big)\\
&= \min_{\mathbf{w},\{\mathbf{u}^{k,i}\}_{1\leq k\leq K,1\leq i\leq N}} R(\mathbf{w}) + C\sum_{k=1}^{K}\sum_{i=1}^{N}\Big(E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}^k),\bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \min_{\mathbf{y}} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}),\bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))\Big) && (43)\\
&= \min_{\mathbf{w},\{\mathbf{u}^{k,i}\}_{1\leq k\leq K,1\leq i\leq N}} R(\mathbf{w}) + C\sum_{k=1}^{K}\sum_{i=1}^{N} L_{G_i}^k(\mathbf{w},\mathbf{u}^{k,i}) = \mathcal{F}_{\{G_i\}}\ , && (44)
\end{aligned}$$

where inequality (42) is true because $\mathrm{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k,\bar{\mathbf{h}}^k)$ is a convex relaxation of problem $\min_{\mathbf{y}} E_G(\bar{\mathbf{u}}^k(\mathbf{y}|\mathbf{w}),\bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))$, while equality (43) is satisfied due to that $E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}),\bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) = \sum_{i=1}^{N} E_{G_i}(\mathbf{u}^{k,i}(\mathbf{y}^k),\bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w}))$ since $\sum_{i\in\mathcal{I}_p} u_p^{k,i} = \bar{u}_p^k$. □

The above theorem implies that, by minimizing $\mathcal{F}_{\{G_i\}}$, one can also guarantee that the original loss $\mathcal{F}_0$ will decrease as well. Not only that but, by appropriately choosing the hypergraph decomposition $\{G_i\}$, we can also improve the approximation $\mathcal{F}_{\{G_i\}}$ to the true loss $\mathcal{F}_0$. This is true because the tightness of the convex relaxation $\mathrm{DUAL}_{\{G_i\}}$ depends crucially on the choice of decomposition $\{G_i\}$. More specifically, we will say that decomposition $\{\tilde{G}_j\}$ is stronger than decomposition $\{G_i\}$ (and we will denote this by $\{G_i\} < \{\tilde{G}_j\}$) if the convex relaxation from $\{\tilde{G}_j\}$ is tighter than the relaxation from $\{G_i\}$, *i.e.*, it always holds $\mathrm{DUAL}_{\{G_i\}} < \mathrm{DUAL}_{\{\tilde{G}_j\}}$. Under this notation, the following theorem is true

Theorem 3. *If $\{G_i\}<\{\tilde{G}_j\}$ then $\mathcal{F}_0\leq\mathcal{F}_{\{\tilde{G}_j\}}<\mathcal{F}_{\{G_i\}}$,* i.e., *$\mathcal{F}_{\{\tilde{G}_j\}}$ is a better approximation to $\mathcal{F}_0$ than $\mathcal{F}_{\{G_i\}}$.*

Proof. By definition (30) it holds that

$$\mathcal{F}_{\{G_i\}} = \min_{\mathbf{w},\{\mathbf{u}^{k,i}\}_{1\le k\le K, 1\le i\le N}} R(\mathbf{w}) + C\sum_{k=1}^{K}\sum_{i=1}^{N} L^k_{G_i}(\mathbf{w},\mathbf{u}^{k,i}) \tag{45}$$

$$= \min_{\mathbf{w}} R(\mathbf{w}) + C\sum_{k=1}^{K}\left(E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}),\bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \mathrm{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k,\bar{\mathbf{h}}^k)\right), \tag{46}$$

where the equality (46) is derived using a similar reasoning as in the proof of theorem 2 above.

Similarly, the following equality can be shown to hold true

$$\mathcal{F}_{\{\tilde{G}_j\}} = \min_{\mathbf{w}} R(\mathbf{w}) + C\sum_{k=1}^{K}\left(E_G(\bar{\mathbf{u}}^k(\mathbf{y}^k|\mathbf{w}),\bar{\mathbf{h}}^k(\mathbf{y}^k|\mathbf{w})) - \mathrm{DUAL}_{\{\tilde{G}_j\}}(\bar{\mathbf{u}}^k,\bar{\mathbf{h}}^k)\right). \tag{47}$$

By assumption it also holds $\{G_i\} < \{\tilde{G}_j\}$, which means that the relaxation $\mathrm{DUAL}_{\{\tilde{G}_j\}}(\bar{\mathbf{u}}^k,\bar{\mathbf{h}}^k)$ is tighter than the relaxation $\mathrm{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k,\bar{\mathbf{h}}^k)$, which in turn implies that

$$\mathrm{DUAL}_{\{G_i\}}(\bar{\mathbf{u}}^k,\bar{\mathbf{h}}^k) < \mathrm{DUAL}_{\{\tilde{G}_j\}}(\bar{\mathbf{u}}^k,\bar{\mathbf{h}}^k) \ . \tag{48}$$

The theorem now follows directly by combining equations (46), (47) and (48). □

Given any MRF graph, it is always possible to choose a decomposition $G_{\text{single}} = \{G_c\}_{c\in\mathcal{C}}$, which contains a sub-hypergraph $G_c = (\mathcal{V}_c, \mathcal{C}_c)$ for each clique $c \in \mathcal{C}$ where $\mathcal{V}_c = \{p|p \in c\}$ and $\mathcal{C}_c = \{c\}$. In this case, each slave MRF consists of a single high-order clique. Due to this fact, such slaves are often very easy to optimize regardless of the complexity of the original MRF. As a result, the derived learning algorithm can have wide applicability. Furthermore, the convex relaxation $\mathrm{DUAL}_{G_{\text{single}}}(\bar{\mathbf{u}}^k,\bar{\mathbf{h}}^k)$ resulting from G_{single} can be shown to coincide with the LP relaxation of the following integer programming formulation of MRF optimization [26]:

$$\min_{\mathbf{z}} \sum_{p}\sum_{y_p} \bar{u}^k_p(y_p|\mathbf{w}) z_p(y_p) + \sum_{c}\sum_{\mathbf{y}_c} \bar{h}^k_c(\mathbf{y}_c|\mathbf{w}) z_c(\mathbf{y}_c) \tag{49}$$

$$\text{s.t.} \sum\nolimits_{y_p} z_p(y_p) = 1\,, \quad \forall p \in \mathcal{V} \tag{50}$$

$$\sum\nolimits_{\mathbf{y}_c : y_p = l} z_c(\mathbf{y}_c) = z_p(l)\,, \quad \forall c \in \mathcal{C},\ p \in c,\ l \in \mathcal{L} \tag{51}$$

$$z_p(\cdot), z_c(\cdot) \in \{0,1\}\ , \tag{52}$$

In the above, $z_p(y_p)$ and $z_c(\mathbf{y}_c)$ are binary indicator variables that exist respectively for each label y_p of node p and each labeling $\mathbf{y}_c$ of clique c. Note that such a relaxation extends the marginal polytope relaxation [7, 54], which is commonly used for pairwise MRFs, to the case of higher-order MRF models.

Of course, one can also choose decompositions $\{\tilde{G}_j\}$ that are stronger than G_{single}. Based on theorem 3 above, this can lead to using better approximations of the loss $\mathcal{F}_0$. This can be achieved, for instance, by taking advantage of the special structure that may exist in certain classes of MRFs. One characteristic example appears in [26] for the case of MRFs with the so-called pattern-based potentials. More generally, the following theorem holds true:

Theorem 4. *$\mathcal{F}_{\left\{\tilde{G}_j\right\}}$ is a better approximation to $\mathcal{F}_0$ than $\mathcal{F}_{G_{\text{single}}}$ only if decomposition $\left\{\tilde{G}_j\right\}_{1\leq j\leq N}$ has at least one sub-hypergraph $\tilde{G}_j$ for which slave MRFs on $\tilde{G}_j$ do not have the integrality property*[8]*.*

Proof. As mentioned above, the MRF optimization problem $\min_{\mathbf{y}} E_G(\bar{\mathbf{u}}^k(\mathbf{y}|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))$ can be equivalently formulated as the following linear integer program:

$$\min_{\mathbf{z}} \sum_{p\in\mathcal{V}}\sum_{y_p} \bar{u}_p^k(y_p|\mathbf{w})z_p(y_p) + \sum_{c\in\mathcal{C}}\sum_{\mathbf{y}_c} \bar{h}_c^k(\mathbf{y}_c|\mathbf{w})z_c(\mathbf{y}_c) \tag{53}$$

$$\text{s.t} \quad \mathbf{z} \in Z(G) \ , \tag{54}$$

where the feasible set $Z(G)$ is defined for any hypergraph $G = (\mathcal{V}, \mathcal{C})$ as follows

$$Z(G) = \left\{\mathbf{z} \in \bar{Z}(G) \mid z_p(\cdot), z_c(\cdot) \in \{0,1\}, \ \forall\, p \in \mathcal{V}, \ c \in \mathcal{C}\right\} \ , \tag{55}$$

$$\bar{Z}(G) = \left\{ \mathbf{z} \ \middle| \ \begin{array}{ll} \sum\limits_{y_p} z_p(y_p) = 1, & \forall\, p \in \mathcal{V} \\ \sum\limits_{\mathbf{y}_c : y_p = l} z_c(\mathbf{y}_c) = z_p(l), & \forall\, c \in \mathcal{C}, \ p \in c, \ l \in \mathcal{L} \\ z_p(\cdot) \geq 0, z_c(\cdot) \geq 0, & \forall\, p \in \mathcal{V}, c \in \mathcal{C} \end{array} \right\} \ .$$

Let $\left\{\tilde{G}_j = (\tilde{\mathcal{V}}_j, \tilde{\mathcal{C}}_j)\right\}_{1\leq j\leq N}$ be a hypergraph decomposition of G (*i.e.* $\cup_{j=1}^N \tilde{\mathcal{V}}_j = \mathcal{V}$, $\cup_{j=1}^N \tilde{\mathcal{C}}_j = \mathcal{C}$, $\tilde{\mathcal{C}}_j \cap \tilde{\mathcal{C}}_{j'} = \emptyset, \forall j \neq j'$) and let $\{\mathbf{u}^{k,j}\}_{1\leq j\leq N}$ be the corresponding set of unary potentials for the slave MRFs of the k-th training sample, which satisfy equation (31), *i.e.*

$$\sum_{j\in\mathcal{I}_p} u_p^{k,j} = \bar{u}_p^k, \tag{56}$$

where $\mathcal{I}_p = \{j | p \in \tilde{\mathcal{V}}_j\}$ (*e.g.* $\mathbf{u}^{k,j}$ can be chosen as $u_p^{k,j} = \bar{u}_p^k/|\mathcal{I}_p|$). Using these potentials, the above linear integer program (53) can be equivalently expressed as

$$\min_{\mathbf{z},\{\mathbf{z}^j\}_{1\leq j\leq N}} \sum_{j=1}^N \left(\sum_{p\in\tilde{\mathcal{V}}_j}\sum_{y_p} u_p^{k,j}(y_p) z_p^j(y_p) + \sum_{c\in\tilde{\mathcal{C}}_j}\sum_{\mathbf{y}_c} \bar{h}_c^k(\mathbf{y}_c|\mathbf{w}) z_c^j(\mathbf{y}_c) \right) \tag{57}$$

$$\text{s.t.} \quad \mathbf{z}^j \in Z(\tilde{G}_j) \ , \ \forall j \in \{1, 2, \ldots, N\} \tag{58}$$

$$z_p^j = z_p \, , \qquad \forall j \in \{1, 2, \ldots, N\}, \ p \in \mathcal{V} \ . \tag{59}$$

The convex relaxation $\text{DUAL}_{\left\{\tilde{G}_j\right\}}(\bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k)$ is derived by relaxing constraints (59) and then solving the resulting Lagrangian relaxation. Therefore, $\text{DUAL}_{\left\{\tilde{G}_j\right\}}(\bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k)$ results from the above problem (57) by simply replacing constraints (59) with the constraints $\mathbf{z}^j \in \text{CH}\left(Z(\tilde{G}_j)\right)$, where $\text{CH}(\cdot)$ denotes the convex hull of a set.

If we now assume that all slave MRFs corresponding to decomposition $\left\{\tilde{G}_j\right\}$ have the integrality property then, by definition, this implies that $\text{CH}\left(Z(\tilde{G}_j)\right) = \bar{Z}(\tilde{G}_j)$ (*i.e.* we can safely

[8] We say that an MRF has the integrality property if and only if the corresponding LP relaxation of (49) is tight.

ignore the integrality constraints in (55)) and so $\text{DUAL}_{\{\tilde{G}_j\}}(\bar{\mathbf{u}}^k, \bar{\mathbf{h}}^k)$ further reduces to

$$\min_{\mathbf{z},\{\mathbf{z}^j\}_{1\leq j\leq N}} \sum_{j=1}^{N} \left(\sum_{p\in\tilde{\mathcal{V}}_j} \sum_{y_p} u_p^{k,j}(y_p) z_p^j(y_p) + \sum_{c\in\tilde{\mathcal{C}}_j} \sum_{\mathbf{y}_c} \bar{h}_c^k(\mathbf{y}_c|\mathbf{w}) z_c^j(\mathbf{y}_c) \right) \tag{60}$$

$$\text{s.t.} \quad \mathbf{z}^j \in \bar{Z}(\tilde{G}_j) \ , \quad \forall j \in \{1,2,\ldots,N\} \tag{61}$$

$$z_p^j = z_p \, , \qquad \forall j \in \{1,2,\ldots,N\}, \ p \in \mathcal{V} \ . \tag{62}$$

Due to constraints (56) and (62), the objective function (60) above is equal to the objective function (49). Furthermore, constraints (61) are equivalent to constraints (54) after the integrality constraints in the latter have been replaced by non-negativity constraints on the $\mathbf{z}$ variables. Therefore, problem (60) is equivalent to the LP relaxation of the linear integer program (49), which, as mentioned earlier, corresponds to the dual relaxation derived from decomposition G_{single}. This concludes the proof of the theorem. □

Based on the above theorem, one can, for instance, provably derive better learning algorithms for pairwise MRF models just by using decompositions containing loopy subgraphs of small tree width (MRFs on such subgraphs can still be efficiently optimized via, *e.g.*, the junction tree algorithm).

However, besides improving the accuracy of a learning algorithm, an appropriate choice of a decomposition $\{G_i\}$ can also improve the computational efficiency of that algorithm. For instance, consider the case of pairwise MRFs and a decomposition $G_{\text{tree}} = \{T_i\}$ that consists entirely of spanning trees T_i of an MRF graph G. Although in this case the accuracy of learning is not improved compared to G_{single} (due to the fact that it holds $\text{DUAL}_{G_{\text{tree}}} = \text{DUAL}_{G_{\text{single}}}$ and thus $\mathcal{F}_{G_{\text{tree}}} = \mathcal{F}_{G_{\text{single}}}$ [28]), the speed of convergence does improve in practice. The reason is because, when using convex relaxation $\text{DUAL}_{G_{\text{tree}}}$, each slave MRF now covers a much larger number of nodes, which allows information to propagate faster across the whole graph G during the MRF dual decomposition updates.

More generally, computational efficiency can be significantly improved simply by choosing a decomposition that is specifically adapted to the class of MRFs we aim to train. For instance, if part of the energy of a MRF is known to be submodular we can take advantage of this fact simply by using that part as a slave. The very fast graph-cut based optimizers that exist for submodular energies can be used directly and will greatly reduce the computational cost of learning in this case.

7 Incremental and stochastic subgradient

To further improve computational efficiency we can also use an incremental subgradient method, which is well suited to objective functions that can be expressed as a sum of components, like in the case of objective function (30) that is given by $R(\cdot) + C\sum_{k=1}^{K}\sum_{i=1}^{N} L_{G_i}^k(\cdot)$, where K is the number of training samples and N the number of sub-hypergraphs in the decomposition. At each iteration of the incremental subgradient method, a step is taken along the subgradient of only one component, where this component can be picked either deterministically (by repeatedly visiting all components in a fixed order) or uniformly at random. A component for the above objective function can have the following form:

$$R(\cdot) + C\sum_{i\in S} L_{G_i}^k(\cdot) \ ,$$

where S denotes a subset of the set $\{1, 2, \ldots, N\}$. In other words, at each iteration we need to consider the hinge losses $L^k_{G_i}(\cdot)$ for only a subset of slave MRFs corresponding to the indices in S (in this case updates are similar to (38), (34) but they need to take into account only a subset of slaves).

For instance, when using a randomized version of this scheme, we can decide to pick at each iteration the index of a training sample k randomly from $\{1, \ldots, K\}$, and then also pick a subset S randomly from a predefined partition of the slave indices of the k-th sample. If S is always chosen to contain all slave indices of the k-th sample then this is essentially equivalent to the more well known stochastic subgradient algorithm, in which case the total subgradient with respect to $\mathbf{w}$ is computed analogously to (36) as follows

$$d\mathbf{w} = \nabla R(\mathbf{w}) + C \left(\sum_{i,p} \frac{\phi_p(y^k_p, \mathbf{x}^k) - \phi_p(\widehat{y}^{k,i}_p, \mathbf{x}^k)}{|\mathcal{I}_p|} + \sum_{i,c} \left(\phi_c(\mathbf{y}^k_c, \mathbf{x}^k) - \phi_c(\widehat{\mathbf{y}}^{k,i}_c, \mathbf{x}^k) \right) \right). \quad (63)$$

To implement this learning scheme, we can modify Algorithm 1 by simply choosing $\mathcal{K}$ as a randomly picked index from $\{1, 2, \ldots, K\}$ at each iteration. Just like the subgradient method, the resulting algorithm is guaranteed to converge to an optimal solution since a theorem similar to Thm. 1 is known to also hold true for the incremental subgradient method [46].

8 Experimental results

To demonstrate the generality and flexibility of our approach, in this section we present experiments and show results for a variety of test cases and scenarios. Such experiments include the training of pairwise and higher-order MRFs, the training by using different types of regularizers (including sparsity inducing ones) and/or different types of dissimilarity loss functions $\Delta(\cdot, \cdot)$, as well as the learning of appropriate models for a variety of vision tasks (including high-order models for pose-invariant knowledge-based segmentation, image denoising, stereo matching, as well as high-order Potts MRFs).

8.1 Image denoising

We begin by presenting experiments related to image denoising. For this purpose, we have created training and testing datasets consisting of synthetic piecewise constant images that have been corrupted by gaussian noise (see Fig. 3). To denoise these images we will make use of a pairwise MRF model whose unary potential is given by $u_p(l) = |l - I_p|$, where I_p denotes image intensity at pixel p, and its pairwise potential $h_{pq}(\cdot, \cdot)$ is assumed to have the following form

$$h_{pq}(l_p, l_q) = V(|l_p - l_q|).$$

Our goal, in this case, is to learn the underlying function $V(\cdot)$, based on which the pairwise potentials are defined. This requires estimating a vector $\mathbf{w} = (w_j)_{1 \leq j \leq 256}$ of size 256, each component of which corresponds to one value of $V(\cdot)$, *i.e.*, $V(j) = w_j, \forall j \in \{1, 2, \ldots, 256\}$. Fig. 2(a) shows the resulting function $V(\cdot)$ as estimated when applying our method to a training set of 10 images, using the hamming loss as the dissimilarity function $\Delta(\cdot, \cdot)$. As can be observed, although $V(\cdot)$ has been learnt automatically from training data, it looks very much like a truncated linear function, which fully agrees with the common practice of using this type of discontinuity preserving potentials when dealing with piecewise constant images.

Fig. 2(b) shows how the primal objective function (30) varies during the course of our algorithm, thus demonstrating how quickly convergence takes place. We also compare to two other

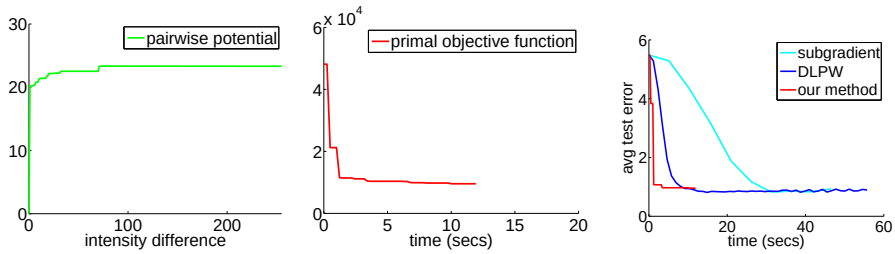
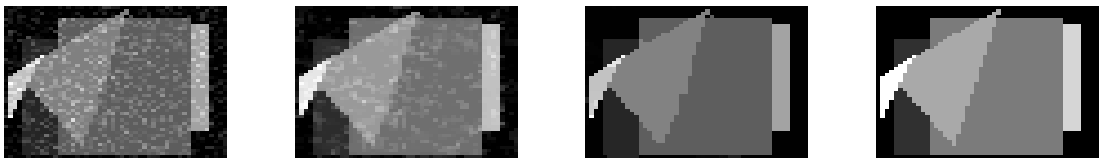


Fig. 2: **(a)** Learnt pairwise potential $V(\cdot)$, **(b)** primal objective (30), **(c)** and average test error as a function of time for the image denoising problem.



(a) Energy: 6.29×10^4 (b) Energy: 8.42×10^4 (c) Energy: 5.71×10^4 (d) Energy: 6.08×10^4

Fig. 3: **(a)** Noisy test image **(b)** Denoised image when using a function $V(\cdot)$ estimated during the course of the learning algorithm **(c)** Denoised result when using the final $V(\cdot)$ **(d)** Ground truth image. We also show below each image the corresponding MRF energy computed using the final estimated $V(\cdot)$.

methods: the subgradient algorithm from [51][9] and the DLPW algorithm from [40]. Fig. 2(c) shows the average test error (for a test set of 10 noisy synthetic images) as a function of time for each algorithm. Our method manages to reduce the test error faster than DLPW. Similarly, it is a lot more efficient than the subgradient method [51]. The inefficiency of the algorithm [51], which relies on applying the subgradient method to the problem formulation (19), comes from the fact that the computation of a subgradient is much more expensive than in our method as it requires solving fully an LP-relaxation of problem $\min_{\mathbf{y}} E_G(\bar{\mathbf{u}}^k(\mathbf{y}|\mathbf{w}), \bar{\mathbf{h}}^k(\mathbf{y}|\mathbf{w}))$, *i.e.*, a relaxation that involves an MRF defined on the whole graph. We also show in Fig. 3 a sample result produced when denoising a test image using the function $V(\cdot)$ learnt by our method.

8.2 Stereo matching

We next test our method on stereo matching, which is a task that requires estimating a per-pixel disparity map between two images of a stereoscopic pair. For this purpose, we will use a pairwise MRF with unary potentials given by $u_p(l) = |I_p^{\text{left}} - I_{p-l}^{\text{right}}|$, where l represents discretized disparity, and I^{left}, I^{right} denote the left and right images, respectively. A very commonly used pairwise potential in this case is a gradient-modulated Potts model of the following form:

$$h_{pq}(l_p, l_q) = f(|\nabla I_p^{\text{left}}|)[l_p \neq l_q], \tag{64}$$

where p, q are neighboring pixels and $\nabla I_p^{\text{left}} = I_p^{\text{left}} - I_q^{\text{left}}$ represents the gradient of the left image at p. Our goal is to automatically learn the function $f(\cdot)$ that is used in the above formula (64) for assigning a discontinuity penalty based on the magnitude of the image gradient. Function $f(\cdot)$

[9] When applying method [51], warm-starting has been used for the successive subgradient computations.

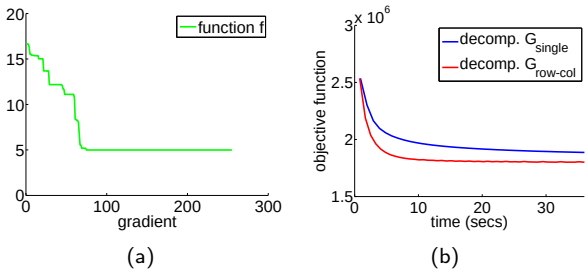
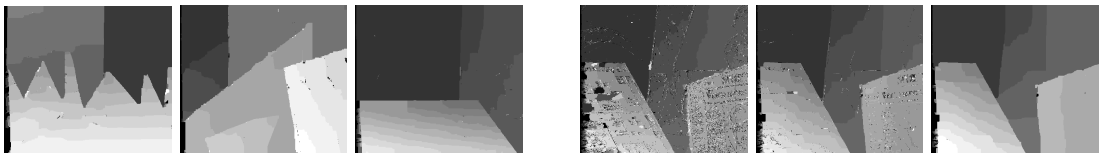


Fig. 4: (a) Learnt function $f(\cdot)$ and **(b)** primal objective (30) as a function of time for two different graph decompositions in the case of stereo-matching.

can take 256 different values assuming integer intensities and so the positive vector $\mathbf{w}$ that we need to estimate will be of size 256 with $w_i = f(i)$. Furthermore, we also impose the restriction that vector $\mathbf{w}$ should belong to the set $\mathcal{W} = \{\mathbf{w} \geq 0 | w_i \geq w_{i+1}\}$, thus reflecting the a priori knowledge that $f(\cdot)$ should be a decreasing function. To accommodate this into our method, we must simply include an additional projection step[10] $\mathbf{w} \leftarrow \text{Proj}_{\mathcal{W}}(\mathbf{w})$ at the end of each iteration of the projected subgradient algorithm, which is the only modification required. We show in Fig. 4(a) the resulting function $f(\cdot)$ that was estimated by our learning algorithm when using the hamming loss as the dissimlarity $\Delta(\cdot,\cdot)$ and a training set of two stereo pairs from the middlebury stereo dataset (the 'Tsukuba' and the 'Map' pairs were used). Using this function, we computed disparity maps for the 'Venus', 'Sawtooth', 'Bull' and 'Poster' stereo pairs from the middlebury dataset (see Fig. 5(a), Fig. 5(b)). The corresponding disparity error rates were 4.9%, 4.4%, 2.8%, 3.7% respectively. Fig. 5(b) also shows 3 different disparity maps that were computed for one of these test images using the function $f(\cdot)$ as estimated at 3 different iterations of our learning algorithm. Notice how the errors in the disparity are reduced as the algorithm converges. Fig. 4(b) shows how the primal objective (30) varies as a function of time during learning. Notice again that our method manages to successfully reduce this objective function very fast. On the contrary, the subgradient method [51] is not very practical to use due to the large size of the MRF problems, which has as a result a considerable increase of the training time in this case. We should note at this point that the goal here is not to show that max-margin learning can achieve state-of-the-art performance on stereo-matching, but to demonstrate that our algorithmic framework provides a flexible and efficient way for MRF parameter estimation that is applicable to many different contexts.. For instance, state-of-the-art stereo matching methods (`http://vision.middlebury.edu/stereo/eval/`) typically make use of very elaborate unary potentials, utilize left-right consistency terms, take into account occlusions in their model, etc., whereas here we use a very basic model for stereo matching with very simple unary terms and features without even taking color information into account (grayscale images are used).

We also compare in Fig. 4(b) what happens when using two different decompositions during learning. Decomposition $G_{\text{row-col}}$ uses each row and column of the MRF grid as subgraphs for the slave MRFs, whereas G_{single} uses each edge separately. As mentioned in section §6, $G_{\text{row-col}}$ is expected to lead to a faster convergence (due to the fact that each slave MRF now covers a larger part of the graph G, thus allowing information to propagate faster during the dual-decomposition updates), which is indeed what is observed in practice.

[10]The projection onto $\mathcal{W} = \{\mathbf{w} \geq 0 | w_i \geq w_{i+1}\}$ is computed very fast via the so-called cyclic projection algorithm [6], where we iteratively project (in a cyclic manner) onto the sets $\mathbf{w} \geq 0$, $w_i \geq w_{i+1}, \forall i \in \{1, 2, \ldots, 255\}$ until convergence.



(a) (b)

Fig. 5: (a) Disparity maps for the 'Sawtooth', 'Poster' and 'Bull' stereo pairs. **(b)** Three disparity maps computed for the stereo pair 'Venus' using functions $f(\cdot)$ estimated at different iterations of our learning algorithm (the final result is the one shown on the right).

8.3 Higher order sparse MRF knowledge-based segmentation

We next apply our framework to the problem of knowledge-based image segmentation, focusing at the same time on the challenging task of learning *sparse, pose invariant* shape priors. For reasons of flexibility and generality, we use, in this case, a shape representation based on a point distribution model $\mathbf{y} = \{y_1, \cdots, y_n\}$ that consists of a set $\mathcal{V} = \{1, \cdots, n\}$ of n control points distributed on the boundary of the object of interest, where y_p ($p \in \mathcal{V}$) denotes the coordinates of the pth point. Additionally, we associate this model with a clique set $\mathcal{C} = \{(p, q, r) | p, q, r \in \mathcal{V} \text{ and } p \neq q \neq r\}$ consisting of all possible combinations of three points.

Considering a triplet clique $c = \{p, q, r\} \in \mathcal{C}$, the geometric shape of the clique $\mathbf{y}_c = (y_p, y_q, y_r)$ is characterized in a pose invariant manner by the measurement of two inner angles $(\alpha_c(\mathbf{y}_c), \beta_c(\mathbf{y}_c))$ defined as follows

$$\alpha_c(\mathbf{y}_c) = \cos^{-1} \frac{\overrightarrow{y_p y_q} \cdot \overrightarrow{y_p y_r}}{\|y_p y_q\| \, \|y_p y_r\|} \ , \ \beta_c(\mathbf{y}_c) = \cos^{-1} \frac{\overrightarrow{y_q y_r} \cdot \overrightarrow{y_q y_p}}{\|y_q y_r\| \, \|y_q y_p\|} \ , \tag{65}$$

where notation $\overrightarrow{y_p y_q}$ denotes the vector $y_p - y_q$. Given a training set of K shape instances $\{\mathbf{y}^k\}_{1 \leq k \leq K}$, we assume that point correspondences exist between the point distribution models within the training set (without assuming that these shapes have been brought to the same referential). Each triplet c is thus associated with K instances $\{(\alpha_c(\mathbf{y}_c^1), \beta_c(\mathbf{y}_c^1)), \cdots, (\alpha_c(\mathbf{y}_c^K), \beta_c(\mathbf{y}_c^K))\}$ used for estimating a probability density $p_c(\alpha_c, \beta_c)$ of triplet c (where a standard probabilistic model based on a Gaussian distribution is employed for the angles (α_c, β_c)).

A shape prior can then be constructed with the accumulation of all triplet clique constraints. To accomplish this, we incorporate into the MRF energy the following higher-order potentials $h_c(\mathbf{y}_c)$

$$h_c(\mathbf{y}_c) = -w_c \log p_c(\alpha_c(\mathbf{y}_c), \beta_c(\mathbf{y}_c)) \ . \tag{66}$$

As can be seen, these potentials are parameterized through a vector $\mathbf{w} = \{w_c\}_{c \in \mathcal{C}}$ containing one component w_c per clique c. Based on the above model, a clique c is essentially ignored if the corresponding element is close to zero, *i.e.*, if it holds $w_c \approx 0$. Therefore, the use of $\mathbf{w}$ allows us to reduce the otherwise excessive number of higher order cliques, thus also reducing the computational cost of inference. Furthermore, given that the significance of the different triplets towards capturing the observed deformations of the training set is not the same, the role of the introduced vector $\mathbf{w}$ is to also weigh the contribution of those triplets that are retained in the model. Note that such a shape prior inherits pose invariance so that neither training samples nor testing shape needs to be aligned in a common coordinates frame. Moreover, it can capture shape variations even with a small number of training examples.

When applying our max-margin learning method to this problem, we opt to make use of a sparsity inducing l_1-norm regularizer, *i.e.*, $R(\mathbf{w}) = ||\mathbf{w}||_1$. Such a choice serves the above

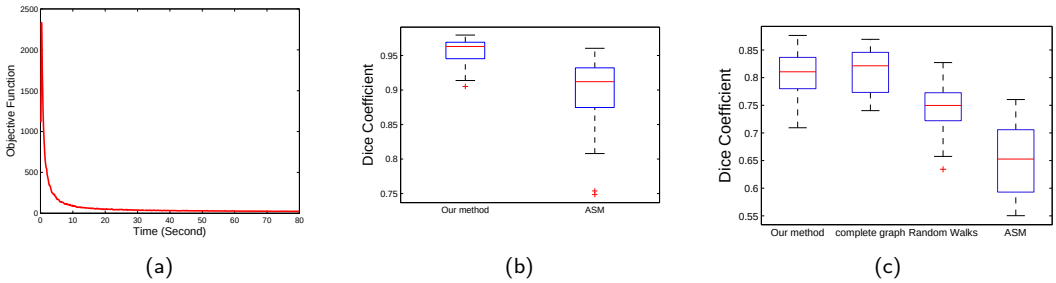


Fig. 6: (a) Learning objective function during MRF training with the hand dataset. Boxplots of Dice coefficients for **(b)** 2D hand segmentation, and **(c)** 3D left ventricle segmentation (the Dice coefficient is a similarity measure between sets X and Y, defined as $\frac{2|X\cap Y|}{|X|+|Y|}$).

purpose of compressing the size of the graph by eliminating as many redundant cliques as possible (through setting their corresponding weights to zero), thus producing a compact and efficient representation.

One important issue that the learning needs to address, in this case, relates to the pose-invariant properties of the learnt shape representation. In other words, it should be able to account for the fact that if $\mathbf{y}^k$ is a ground truth shape, then, any transformed shape instance $T(\mathbf{y}^k)$, where $T(\cdot)$ represents a similarity transformation, is an equally good solution and should not be penalized during training. To accomplish that by our method, we make use of a dissimilarity function $\Delta(\mathbf{y},\mathbf{y}')$ that satisfies the following conditions, which ensure that pose invariance is indeed taken into account during the training process

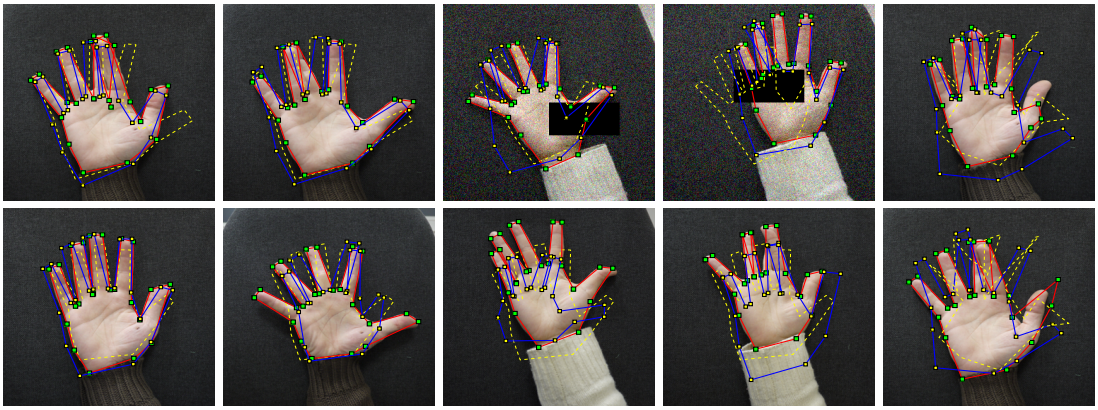
$$(\forall \mathbf{y}),\ \ \Delta(T(\mathbf{y}),\mathbf{y}) = 0\ . \tag{67}$$

The specific function that we use for this purpose decomposes into high-order terms as follows $\Delta(\mathbf{y},\mathbf{y}') = \sum_{c\in\mathcal{C}} \delta_c(\mathbf{y}_c,\mathbf{y}'_c)$, where

$$\delta_c(\mathbf{y}_c,\mathbf{y}'_c) = \begin{cases} 0 & \text{if the triplets of points } \mathbf{y}_c \text{ and } \mathbf{y}'_c \text{ connect by a similarity transform} \\ 1 & \text{otherwise}\ . \end{cases} \tag{68}$$

The aforementioned L1 sparse higher-order shape model is used in conjunction with knowledge-based segmentation. In this context, following [68] we consider on top of the above third-order cliques, pair-wise cliques delineating the object boundaries in 2D, or third-order cliques corresponding to the object surface. The use of generalized Stokes theorem from differential geometry allows to convert regional integrals to surface ones and therefore combine edge-based terms with regional ones [67] (thus integrating both edge information and regional statistics towards seeking the separation in terms of intensity statistical means between the object and the background). During training, a decomposition that assigns a single clique per slave has been used, in which case enumeration is applied for solving the resulting slave problems.

Our learning method was evaluated in this context using two different examples, a 2D hand data-set and a 3D medical imaging example (CT segmentation of the Left Ventricle). The 2D hand dataset contains 40 right hand examples (20 used for training and 20 used for testing), showing different poses (*i.e.*, translations, rotations, and scales) and also movements between the fingers. Manual segmentations on the database are available and used as ground truth, while a number of 23 control points is used in the point distribution model. Fig. 6(a) shows how the learning objective function varies during training. As can be seen, the algorithm converges very



(a) Red contours: our results. Blue contours: ASM. Yellow contours: initialization.

(b) Our results on video images with cluttered background.

Fig. 7: 2D hand segmentation results.

fast despite having to deal with a large number of parameters. Moreover, it leads to a sparse model since the estimated $\mathbf{w}$ comprises only 5.6% non-zero components.

Some visual test results are shown in Fig 7 where red contours outline the results, and yellow contours represent initializations. As can be observed, thanks to the learnt model, satisfactory results are obtained even when noise or occlusions are present in the images. For example, in the first two images of the first row, the fingers are partially self-occluded which corresponds to shapes not seen at all during training, and yet our result is reasonable as it can still correctly localize the shape. In the 3rd and 4th images of the first row, noise and occlusions are present in the testing images: with a larger weight on prior term, our model shows its robustness. The first two rows also provide qualitative comparisons of our method with the Active Shape Models (ASM) segmentation algorithm [9] (blue contours). The third row shows results from additional tests that were conducted on a set of video images. For both quantitative and comparison purposes, we also plot in Fig 6(b) the Dice coefficients of our approach and ASM.

Regarding the 3D segmentation dataset, in our tests we have used one that contains 20 3D CT cardiac images. The volumes from different subjects have an approximate mean size of $512 \times 512 \times 250$ voxels and the voxel size is about $0.35 \times 0.35 \times 0.5mm^3$. The point distribution model consists of 88 control points both on the myocardium surface as well as the atrium surface, while the coarse triangulated mesh is made up of 172 triangle faces. Some indicative results are shown in Fig 8, where the yellow contours correspond to our method, while the green contours represent the results from ASM models. As can be observed, the resulting model exhibits good accuracy on the boundary (the first two columns) and robustness to the papillary muscles in the blood pool (the last column). In addition, we provide in Fig. 6(c) a quantitative comparison of our algorithm with the ASM algorithm [9], the Random Walks algorithm [16], and the method [66] as applied to the above 3D dataset (we present the corresponding Dice coefficients that were obtained). Note that method [66] does not learn the cliques and uses the complete graph (it

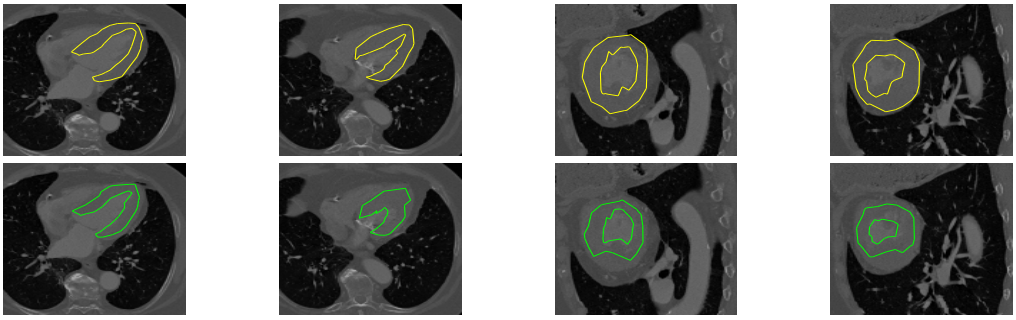


Fig. 8: 3D Segmentation results on cardiac CT volumes. (Top row) Our results. (Bottom row) ASM results.

is thus very computationally expensive taking more than 1 hour for one volume segmentation). On the contrary, we use a very sparse graph since the estimated $\mathbf{w}$ by our method contains only 0.9% non-zero components (the computation time in this case is about 3 minutes).

8.4 High-order Potts model

Last, we conclude the experimental part of the paper by showing some additional results on synthetic problems that are related to high-order MRF learning. More specifically, we applied our method to MRFs with $\mathcal{P}^n$ Potts high-order potentials [20], which have the following form:

$$h_c(\mathbf{y}_c) = \begin{cases} \beta_l^c & \text{if } y_p = l, \ \forall p \in c \\ \beta_{\max}^c & \text{otherwise} \ , \end{cases} \tag{69}$$

where l denotes any label from a discrete set of labels $\mathcal{L}$. We assume that each value β_l^c is equal to the dot product of a vector of parameters $\mathbf{w}_l$ with a feature vector $\mathbf{x}_l^c$, *i.e.*, $\beta_l^c = \mathbf{w}_l \cdot \mathbf{x}_l^c$, and the goal of learning includes estimating all vectors $\mathbf{w}_l$. For this we use synthetic data: we randomly sample unary potentials as well as feature vectors $\{\mathbf{x}_l^c\}$ and then we generate the values β_l^c of the high-order potentials based on a specified set of vectors $\{\mathbf{w}_l\}$. We then approximately minimize the resulting MRF using the method from [26], and the solution that we obtain is used as the ground truth for the current sample (we repeat this process to generate as many samples as we want). For the corresponding MRF hypergraph we assume that its nodes are arranged in a 2D grid and there exists a high-order clique for each subrectangle of size $s \times s$ in that grid. Our learning algorithm is applied by using a hamming loss for the dissimilarity function $\Delta(\cdot,\cdot)$, as well as a decomposition that assigns one clique per slave. Note that the minimization of each slave is very efficient as it takes time $O(|\mathcal{L}|)$ *regardless of the size of the high-order clique*. Fig. 9 shows an example of how fast the learning objective function decreases in this case (where we used a grid of size 50×50, the clique size was 3×3, $|\mathcal{L}| = 5$, and we had 100 training samples). The main point we want to emphasize here is the efficiency of our method even when high-order MRF terms are present. It is also worth mentioning that the duality gap during optimization was typically small in this case as there was a large amount of agreement between the local solutions of the slave MRFs. This was true to a lesser extent for other experiments in the paper that involved more difficult high-order MRFs (e.g., pose-invariant knowledge based segmentation).

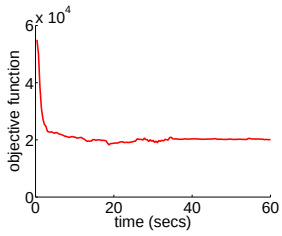


Fig. 9: Primal objective function during training of high-order Potts MRFs.

9 Conclusions

In this paper we have presented a general algorithmic framework for MRF/CRF training. Such a framework essentially manages to reduce the training of a complex high-order MRF model to the parallel training of a set of simple slave submodels. We have demonstrated that the derived learning scheme is sufficiently efficient and flexible (*e.g.*, it can be applied to both pairwise and high-order models, it requires no submodularity assumptions, and it is easily adapted to the structure of a given class of MRFs). Moreover, it relies on solid mathematical principles and enjoys good theoretical properties. Due to all of the above, and given that learning problems are becoming increasingly important and challenging for a great variety of applications these days, we believe that our method can find use in a broad class of image analysis and computer vision tasks. It is also worth noting that this framework can be extended to handle the task of parameter estimation for latent CRF models [25], in which case a subset of variables remain unknown (*i.e.*, hidden) during both training and test time. Although not thoroughly discussed in the present work, these, too, comprise a very important class of models in computer vision, with an increasingly large number of applications [11, 64].

References

[1] Karteek Alahari, Chris Russell, and Philip Torr. Efficient piecewise learning for conditional random fields. In *CVPR*, 2010.

[2] Dragomir Anguelov, Ben Taskar, Vassil Chatalbashev, Daphne Koller, Dinkar Gupta, Geremy Heitz, and Andrew Ng. Discriminative learning of markov random fields for segmentation of 3d scan data. In *CVPR*, 2005.

[3] Dimitri Bertsekas. *Nonlinear Programming*. 1999.

[4] Julian Besag. Efficiency of pseudolikelihood estimation for simple Gaussian fields. *Biometrika*, 64(3):616–618, December 1977.

[5] A. Blake, P. Kohli, and C. Rother. *Advances in Markov random fields for vision and image processing*. MIT Press, 2011.

[6] Yair Al Censor and Stavros A. Zenios. *Parallel Optimization: Theory, Algorithms and Applications*. Oxford University Press.

[7] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *SODA*, 2001.

[8] Bruno Conejo, Nikos Komodakis, Sebastien Leprince, and Jean-Philippe Avouac. Inference by learning: Speeding-up graphical model optimization via a coarse-to-fine cascade of pruning classifier. In *Advances in Neural Information Processing Systems 27*, pages 1–9. 2014.

[9] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59, January 1995.

[10] Justin Domke. Learning graphical model parameters with approximate marginal inference. *in IEEE TPAMI*, 2013.

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 2009.

[12] T. Finley and T. Joachims. Training structural svms when exact inference is intractable. In *ICML*, 2008.

[13] Vojtech Franc and Bogdan Savchynskyy. Discriminative learning of max-sum classifiers. *JMLR*, 2008.

[14] Ben Glocker, Nikos Komodakis, Georgios Tziritas, Nassir Navab, and Nikos Paragios. Dense image registration through MRFs and efficient linear programming. *Medical Image Analysis*, 12(6):731 – 741, 2008.

[15] Stephen Gould. Max-margin learning for lower linear envelope potentials in binary mrfs. In *ICML*, 2011.

[16] Leo Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006.

[17] Tamir Hazan and Raquel Urtasun. A primal-dual message-passing algorithm for approximated large scale structured prediction. In *NIPS*, pages 838–846, 2010.

[18] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002.

[19] Jörg H. Kappes, Bjoern Andres, Fred A. Hamprecht, Christoph Schnörr, Sebastian Nowozin, Dhruv Batra, Sungwoong Kim, Bernhard X. Kausler, Jan Lellmann, Nikos Komodakis, and Carsten Rother. A comparative study of modern inference techniques for discrete energy minimization problem. In *CVPR*, 2013.

[20] Pushmeet Kohli, Pawan Kumar, and Philip Torr. P3 and beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[21] N. Komodakis and N. Paragios. Beyond loose lp-relaxations: Optimizing mrfs by repairing cycles. 2008.

[22] N. Komodakis and G. Tziritas. Image completion using global optimization. 2006.

[23] Nikos Komodakis. Towards more efficient and effective lp-based algorithms for mrf optimization. In *ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*, pages 520–534. 2010.

[24] Nikos Komodakis. Efficient training for pairwise or higher order CRFs via dual decomposition. In *CVPR*, 2011.

[25] Nikos Komodakis. Learning to cluster using high order graphical models with latent variables. In *ICCV*, 2011.

[26] Nikos Komodakis and Nikos Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *CVPR*, 2009.

[27] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007.

[28] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF energy minimization and beyond via dual decomposition. *PAMI*, 2010.

[29] Nikos Komodakis and Georgios Tziritas. Approximate labeling via graph-cuts based on linear programming. 2007.

[30] M Pawan Kumar and Philip H.S. Torr. Fast memory-efficient generalized belief propagation. *the Proceedings of the Ninth European Conference on Computer Vision 2006*, 2006.

[31] M Pawan Kumar and Philip H.S. Torr. Efficiently solving convex relaxations for map estimation. *Proceedings International Conference of Machine Learning (ICML)*, 2008.

[32] M.P. Kumar, H. Turki, D. Preston, and D. Koller. Learning specific-class segmentation from diverse data. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[33] Pawan Kumar, Benjamin Packer, and Daphne Koller. Modeling latent variable uncertainty for loss-based learning. In *ICML*, 2012.

[34] Sanjiv Kumar, Jonas August, and Martial Hebert. Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In *EMMCVPR*, 2005.

[35] Sanjiv Kumar and Martial Hebert. Discriminative random fields. *International Journal of Computer Vision*, 2006.

[36] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[37] Stan Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2009.

[38] Yunpeng Li and Daniel P. Huttenlocher. Learning for stereo vision using the structured support vector machine. In *CVPR*, 2008.

[39] David A. McAllester, Tamir Hazan, and Joseph Keshet. Direct loss minimization for structured prediction. In *NIPS*, 2010.

[40] Ofer Meshi, David Sontag, Tommi Jaakkola, and Amir Globerson. Learning efficiently with approximate inference via dual losses. In *ICML*, 2010.

[41] Kevin Miller, M. Pawan Kumar, Benjamin Packer, Danny Goodman, and Daphne Koller. Max-margin min-entropy models. 2012.

[42] Kevin Miller, M. Pawan Kumar, Benjamin Packer, Danny Goodman, and Daphne Koller. Max-margin min-entropy models. In *AISTATS*, volume 22 of *JMLR Proceedings*, pages 779–787. JMLR.org, 2012.

[43] Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. Stacked hierarchical labeling. In *ECCV*, 2010.

[44] Daniel Munoz, J. Andrew (Drew) Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual classification with functional max-margin markov networks. In *CVPR*, 2009.

[45] K. Murphy, Y. Weiss, and M. Jordan. Loopy-belief propagation for approximate inference: An empirical study. In *UAI*, 1999.

[46] A. Nedic and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM J. on Optimization*, 2001.

[47] Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, and Pushmeet Kohli. Decision tree fields. In *ICCV*, 2011.

[48] G. Papandreou and A. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *ICCV*, 2011.

[49] Kyoungup Park and Stephen Gould. On learning higher-order consistency potentials for multi-class pixel labeling. In *ECCV*, 2012.

[50] Patrick Pletscher and Pushmeet Kohli. Learning low-order models for enforcing high-order statistics. In *AISTATS*, 2012.

[51] Nathan Ratliff, J. Andrew (Drew) Bagnell, and Martin Zinkevich. (online) subgradient methods for structured prediction. In *AISTATS*, 2007.

[52] Kegan G. G. Samuel and Marshall F. Tappen. Learning optimized map estimates in continuously-valued mrf models. In *CVPR*, 2009.

[53] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *CVPR*, 2007.

[54] M.I. Schlesinger and V.V. Giginyak. Solution to structural recognition (MAX,+)-problems by their equivalent transformations. *Control Systems and Computers*, 2007.

[55] Alexander G. Schwing, Tamir Hazan, Marc Pollefeys, and Raquel Urtasun. Efficient structured prediction with latent variables for general graphical models. In *ICML*, 2012.

[56] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated subgradient solver for svm. In *ICML*, 2007.

[57] David Sontag, Ofer Meshi, Tommi Jaakkola, and Amir Globerson. More data means less inference: A pseudo-max approach to structured learning. In *NIPS*, pages 2181–2189, 2010.

[58] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):1068–1080, June 2008.

[59] Martin Szummer, Pushmeet Kohli, and Derek Hoiem. Learning CRFs using graph cuts. In *ECCV*, 2008.

[60] Marshall F. Tappen, Kegan G. G. Samuel, Craig V. Dean, and David M. Lyle. The logistic random field - a convenient graphical model for learning parameters for mrf-based labeling. In *CVPR*, 2008.

[61] Daniel Tarlow, Ryan Adams, and Richard Zemel. Randomized optimum models for structured prediction. In *AISTATS*, 2012.

[62] Daniel Tarlow and Richard S. Zemel. Structured output learning with high order loss functions. In *AISTATS*, 2012.

[63] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. In *NIPS*, 2004.

[64] Huayan Wang, Stephen Gould, and Daphne Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*, 2010.

[65] Tomas Werner. A linear programming approach to max-sum problem: A review. *PAMI*, 2007.

[66] B. Xiang, C. Wang, J.F. Deux, A. Rahmouni, and N. Paragios. 3d cardiac segmentation with pose-invariant higher-order mrfs. In *ISBI 2012*, pages 1425–1428. IEEE, 2012.

[67] Bo Xiang, Chaohui Wang, Jean-François Deux, Alain Rahmouni, and Nikos Paragios. Tagged cardiac mr image segmentation using boundary & regional-support and graph-based deformable priors. In *ISBI*, pages 1706–1711, 2011.

[68] Bo Xiang, Chaohui Wang, Jean-François Deux, Alain Rahmouni, and Nikos Paragios. 3d cardiac segmentation with pose-invariant higher-order mrfs. In *ISBI*, pages 1425–1428, 2012.

[69] Payman Yadollahpour, Dhruv Batra, and Greg Shakhnarovich. DivMCuts: Faster Training of Structural SVMs with Diverse M-Best Cutting-Planes. In *AISTATS*, 2013.

[70] Chun-Nam John Yu and Thorsten Joachims. Learning structural SVMs with latent variables. In *ICML*, 2009.

[71] Yipin Zhou and Nikos Komodakis. A map-estimation framework for blind deblurring using high-level edge priors. In *ECCV 2014*, volume 8690, pages 142–157, 2014.

Contents

1 Introduction **3**

2 Related work **5**

3 MRF Optimization via Dual Decomposition **7**

4 Max-margin Markov Networks **9**

5 Learning via Dual Decomposition **10**

6 Choice of decompositions $\{G_i\}_{1 \leq i \leq N}$ and tighter approximations **14**

7 Incremental and stochastic subgradient **18**

8 Experimental results **19**
8.1 Image denoising 19
8.2 Stereo matching 20
8.3 Higher order sparse MRF knowledge-based segmentation 22
8.4 High-order Potts model 25

9 Conclusions **26**

RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE

Parc Orsay Université
4 rue Jacques Monod
91893 Orsay Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399