A Concept for Guiding the Learning of Conditional Random Fields for the Classification of Urban Areas in SAR Images

TESSIO NOVACK[1], OLIVER MAKSYMIUK[1] & UWE STILLA[1].

In dieser Arbeit wird ein Konzept zur Klassifikation urbaner Strukturen in SAR Intensitätsbildern vorgestellt. Die Methode basiert dabei auf der Verwendung von Conditional Random Fields (CRF), deren Netzwerkstruktur und Parametrisierung automatisch durch das Optimieren einer Zielfunktion gelernt werden soll. Allerdings handelt es sich um ein schlecht gestelltes Problem, was deutliche Herausforderungen an die Algorithmen stellt. Wir zeigen einen effektiven Weg, um dennoch eine Optimierung erreichen zu können und welchen Vorteil regelbasierte Verfahren, die durch den Nutzer definieret werden, in diesem Zusammenhang haben.

1 Introduction

Regarding the classification of high-resolution SAR imagery, the consideration of the spatial context is practically indispensable. This can be done by (1) calculating texture attributes, (2) possibly using these and/or other attributes as inputs to an image segmentation procedure and (3) considering contextual attributes that relate neighbouring image regions. Probabilistic Graphical Models (PGMs) are powerful probabilistic frameworks that allow the modelling of complex contextual relations between neighbouring image regions (or pixels) regarding their attributes and their possible classes. As a specific PGM type, Conditional Random Fields (CRFs) focus directly on estimating the posterior probability distribution of the possible classifications of the scene given the observed images attributes (KOLLER & FRIEDMAN, 2009). The automatic learning of the CRF network's structure and its parameters based on sample data is known to produce models of better performance and potentially reveal unsuspected dependencies between the observed and unobserved variables (GANAPATHI, 2008). Nevertheless, the automatic learning of CRFs models very easily becomes computationally intractable, due to the necessity of running inference over the current CRF network at each step of the optimization procedure and, if discriminative training is carried out, for each data sample. This problem is worsened by the fact that very often remote sensing image regions may contain a large number of attributes with large cardinality. In this paper, we show a framework for the automatic learning of the CRF network's structure and parameters in a tractable way. This framework was customized aiming its application on remote sensing image classification and having two guidelines in mind: (1) reduce as much as possible computational costs and (2) allow the inclusion of image interpretation knowledge in the learning process. We give a brief explanation on the theoretical background of our framework and exemplify how it would be applied for the classification of Urban Structure Types based on high-resolution (In)SAR imagery.

1) Photogrammetrie & Fernerkundung, Technische Universität München, Arcisstrasse 21, 80333 München, http://www.pf.bv.tum.de

2 Some theory behind the proposed framework

2.1 Probabilistic Graphical Models and CRFs

Probabilistic Graphical Models (PGMs) enable the representation of complex joint distributions with considerably fewer parameters. Such representation is given in the form of a network whose nodes are random variables and the links between them represent the probabilistic dependencies and conditional independencies among the variables of the network. A fully connected sub-network is called a factor. A potential is associated to every possible assignment of the variables values in a factor. It should be noted that in most cases the variables are modelled as being discrete. Most effective inference algorithms only work for discrete variables. A PGM is fully defined when its network structure and potentials are defined. The problem of defining the structure and the parameters of a PGM is the problem of defining how the joint distribution factorizes and which the potentials from each factor are. Each factor of a CRFs model must contain at least one unobserved variable. In the context of remote sensing image classification, the observed variables are the image attributes and the unobserved variables are the classes of interest.

2.2 Log-linear Models

Log-linear models are a compact representation of undirected PGMs and CRFs which are defined in terms of a set of feature functions $f_k(X_k)$. The variables X_k in every feature k are exactly the ones in one of the factors of the network. Given a set of feature functions $F = \{f_k\}$, the parameters of the log-linear model are weights $\theta = \{\theta_k : f_k \in F\}$. The overall distribution is then defined as:

$$P_\theta(x) = \frac{1}{Z(\theta)} \exp\left(\sum_{f_k \in F} \theta_k f_k(x_k)\right) \tag{1}$$

where x_k is the assignment to X_k within x, and $Z(\theta)$ is the partition function that ensures that the distribution is normalized. The term $f_k(X_k)$ can be any function that defines a numerical value for each assignment x_k. A commonly used function is the indicator function, which returns value 1 in case of certain x_k assignment and 0 otherwise. The weight θ_k becomes than the potential for this assignment. Commonly, one such indicator function is created for each possible assignment from each factor.

2.3 Parameter Learning

A few approaches have been proposed for learning the parameters of undirected PGMs (KOLLER & FRIEDMAN, 2009), the maximization of the log-likelihood function being the simplest variant of this task. The log-likelihood objective function has the form:

$$l(M, \theta : D) = \sum_{f_k \in F} \theta_k \left(\sum_m f_k(\xi[m])\right) - M \log Z(\theta) \tag{2}$$

where M is the number of samples and m is a single sample from the training data set. This function is a convex one, which means it has no local optima but a global one. However, it has

no closed-form solution and because of that its global optimum has to be found through an iterative optimization process (like gradient descent, for example) which uses the gradient between the empirical counts of the features (calculated directly from the sample data set) and their marginal probability given a parameter set θ to guide the optimization. Formally we want that for each feature *k*,

$$(\sum_{m} f_k(\xi[m])) - ME_{x\sim P_\theta}[f_k(x)] = 0 \quad (3)$$

In order to calculate the second term of this expression (i.e. the marginal probability of a certain feature), inference should be performed over the current model. Because usually several iterations are needed until convergence is reached, inference is run many times, what makes this process computationally very costly.

2.4 Structure Learning

The problem of finding the best structure of the CRF network is equivalent to the problem of finding the best set of features *F* given the set of observed and unobserved variables. Including the L1-regularization term to equation (2) and maximizing it has the effect of forcing many features to have weight zero when convergence is achieved. At the end of the optimization process, the features that have weights different than zero represent the factors of the CRF network, i.e. the edges and nodes of the CRF structure. This L1-regularized log-likelihood objective function has the form:

$$\max_{\theta} = \left[\sum_{f_k \in F} \theta_k (\sum_{m} f_k(\xi[m])) - M \log Z(\theta) - \sum_{k} \beta_k |\theta_k| \right] \quad (4)$$

where the first two terms are exactly the ones at equation (2). The third term is the L1-regularization term, which is nothing more than the sum of the model's parameters multiplied by the hyper-parameter β. This objective function is also a convex one, so when the convergence point is achieved, we are guaranteed that the CRF network and its parameters are the best ones[2]. As equation (2), this objective has no closed-form solution and its maximum is also achieved by reducing the gradient of each feature to zero. Formally, we want that for each feature $f_k(X)$

$$((\sum_{m} f_k(\xi[m])) - ME_{x\sim P_\theta}[f_k(x)]) - \tfrac{1}{\beta} sign(\theta_k) = 0 \quad (5)$$

The effect of biasing many of the parameters to zero happens to integrate the network learning and the parameter optimization problems into a single objective function. Features with weight zero represent discarded edges in the network, whereas features with non-zero weights represent variables (i.e. nodes) that are connected with each other in the network. Hence, we are solving

[2] Although this solution is the optimal one, it may not be unique. Several redundant parameterizations might be at the convergence point.

both the parameter and the structure learning problems by optimizing a single objective function at a computational cost that is not much higher than learning the parameters alone.

2.5 Discriminative training

Equations (4) and (5) are used for learning a CRF model in a generative way. We may learn a CRF model in a generative way as long as each of the possible features involves at least one of the unobserved variables. Another way to train such a model is in the discriminative way. In this setting, our training set consists of pairs $D = \{(y[m], x[m])\}_{m=1}^{M}$ specifying assignments to Y (unobserved variables) and X (observed variables). Here we want to optimize the likelihood of each $y[m]$ given the corresponding $x[m]$. This is also a convex function (actually a sum of convex log-likelihood functions, i.e. one for each data sample) with a convex region of global optima, i.e. redundant optimal parameterizations. The gradient in the discriminative training case has the form:

$$\sum_{m=1}^{M} (f_i(y[m], x[m]) - \mathrm{E}_{\theta}[f_i \mid x[m]]) \tag{6}$$

Whereas in the generative training each gradient step required only a single execution of inference, training a model in the discriminative way is more cumbersome because we have to execute inference for every data sample conditioned on $x[m]$. On the other hand, the inference is executed on a simpler model, since conditioning on evidence in a Markov network can only reduce the computational costs. Inserting the L1-regularization term in the computation of the gradient of each feature leads to the form:

$$\sum_{m=1}^{M} (f_i(y[m], x[m]) - \mathrm{E}_{\theta}[f_i \mid x[m]]) - \frac{1}{\beta} sign(\theta_k) = 0 \tag{7}$$

Discriminative training is considered to be better for image classification tasks, whereas on the other hand models can be trained with fewer samples in the generative way (KOLLER & FRIEDMAN, 2009).

2.6 The Learning Procedure

As mentioned, inserting the L1-regularization term in the parameter learning optimization process allow us to also learn the CRF network structure using the same objective function and at a not much higher computational cost. In theory, all the possible features can be submitted at once to this optimization process and at convergence the features that have weight zero (the vast majority of them) are discarded and the features that have weight non-zero induce probabilistic dependencies between the variables they involve, representing in this way the CRF network. However, inserting all possible features at once in the optimization process would give rise to a highly dense network, what makes parameter inference imprecise and prohibitively costly. Because of that, we must resort to a feature introduction scheme, where the most pertinent features are introduced first in order to achieve convergence faster. As the objective function is a convex one, any feature introduction scheme leads to the global optimum, but inserting unimportant features first increases the computational costs (even if they are eventually discarded) and decreases the accuracy of the calculated gradients (LEE et al., 2007). A few

feature evaluation measures have been proposed (DELLA PIETRA et al., 1997; PERKINS et al., 2003) and can be used for deciding which feature to introduce next in the optimization process. These measures evaluate quantitatively each individual features, what increases computational costs.

The learning proceeds in the following way: (1) among all possible features, a few of them are allowed to have weight non-zero (the so-called active features), (2) we optimize the weights of these features reducing their gradients to zero (equation 7); many of the features continue to have weight zero, due to the sparseness effect of the L1-regularization term; following, (3) each feature with weight zero (the so-called inactive features) is evaluated by some quantitative measure; (4) the best feature is inserted in the active features group and steps 2 to 4 are repeated until convergence is achieved. Each time the procedure finishes step 2, the convergence criteria are checked. The criteria are that the gradients of all active features should equal zero and the gradients of all inactive features should be in module smaller than $1/2\,\beta$ (LEE et al. 2007).

3 Guiding the introduction of features

As mentioned, the global optimum is not a point representing a unique solution, but a region with possibly many equivalent optimal solutions. We want to achieve this region as fast as possible and, more specifically, we want to pick a model that represents our expert knowledge regarding the interpretation of high-resolution SAR imagery. We now show the building blocks of our framework for guiding the CRF learning towards this goal. This framework is a flexible expert system (COWELL et al. 1999) for the gradual introduction of log-linear features in the optimization of the L1-regularized log-(conditional)-likelihood objective function. The expert system is a rule-based one composed of *if... then...* rules and has as inference engine the quantitative measure proposed by DELLA PIETRA et al. (1997) for ranking the gain of inserting each of the available log-linear features in the optimization process.

3.1 Context Range Definition

As the first step, we must define the distance range of the possible contextual relations between image segments. This can be defined in Euclidean distance or as *k*-nearest-neighbours (*k*-NN). The range is then divided into distance categories and the segments in each category are labelled accordingly. In case of the *k*-NN approach, we divide the maximum *k* into *n* categories and label the segments accordingly. Figure 1 shows an example of such a categorization for the two above mentioned cases. The centroids of the segments are considered for the distance computations.

3.2 Categorization and semantic association of log-linear features

Once we defined the context's distance range, we automatically define all the possible log-linear features that can be created with this universe of image segments (given that the observed image attributes are also defined). All these available log-linear features can be categorized regarding, for instance, their number of observed variables, number of unobserved variables, number of observed and unobserved variables in it, number of observed variables in the distance (or *k*-NN) category, number of unobserved variables in the distance (or *k*-NN) category, number of variables *lengthness*, number of variables *squareness*, number of variables of *texture*, number of

variables of *intensity* etc. These categories are not mutually exclusive and they can be sub-divided into many other categories. The types and number of categories are defined by the user. After the creation of all possible features and their categorization, the semantic association of the feature categories takes place. This can be represented as a semantic net where the arrows indicate the association of log-linear feature categories. Figure 2 (a) shows an example of such a semantic net when the features are grouped into nine categories.

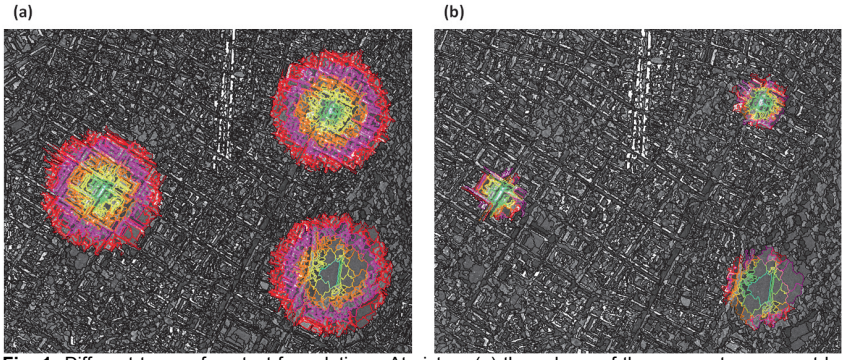


Fig. 1: Different types of context formulations. At picture (a) the colours of the segments represent how distant they are (in meters) from the central segment. At picture (b) the colours of the segment represent nearest-neighbours (NN) categories, e.g. 5-NN, 6 to 10-NN, 11 to 20-NN.

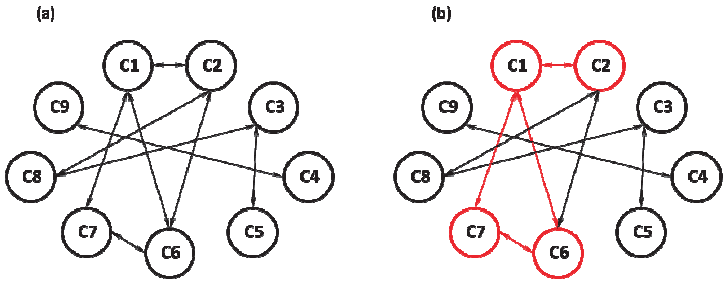


Fig. 2: Semantic net representing the association of log-linear feature categories (a) and graphical representation of the categories whose features should be evaluated next in case we insert a feature from category C1 at a certain stage of the optimization/feature introduction process (b).

Once we have a net representing the semantic association of features, we can use it to decide which ones should be evaluated at each point of the optimization process. Relying on this semantic model avoids us to evaluate each of the features each time the current model is optimized and we have to insert a new feature into the active group. At the first optimization

iteration we can evaluate all possible features and detect to which category the winning feature belongs to. Then, at the next iteration we will only evaluate features from these categories. So that the process does not get stuck on certain categories, we can at some point evaluate all features again. If the assertions and actions, i.e. if the semantic net and the rules are good, we will achieve an optimum solution soon, otherwise it will take longer. However, we will never need a longer time as we would by inserting features blithely relying on the convex property of the objective function.

3.3 Inference for computing the gradients

At the end of each optimization step we must check the converge criteria in order to know if an optimal solution has been found, what requires the computation of the gradients from the active and the inactive features. According to LEE et al. (2006), the best way to do that when using the Loopy Belief Propagation (MURPHY et al. 1999) inference algorithm to calculate the gradients (what almost always is inevitable) is to create a cluster graph which contains all the variables that the model may have. In this way, it is possible to extract clique trees out of the calibrated cluster graph and use them to infer the marginal distributions of the inactive features. The marginal distributions of the active features can be computed directly out of the calibrated cluster graph. Figure 3 shows what would be the simplest set of active log-linear features (factors of the graph) when starting the optimization process. This network structure gets altered according to what features get inserted and maintained in the active group. However, its basic structure must persist so that the gradients of all possible features can be calculated.

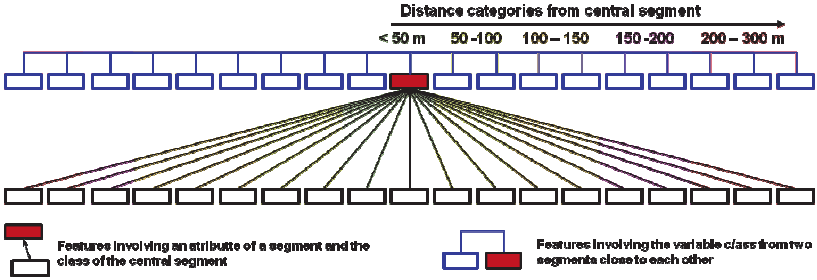


Fig. 3: Initial CRF network structure. This is the initial set of active log-linear features. Constructing a cluster-graph out of this network enables the computation of the gradient from each possible log-linear feature that can be considered with this set of variables.

4 Learning CRFs models for the classification of Urban Structure Types

Classifying Urban Structure Types (UST) (HEIDEN et al., 2012) requires considering long-range contextual relations between the observed attributes and the possible classes of the image segments. Hence, the context's distance range (section 3.1) should be defined accordingly. The mean or maximum area of the city's urban blocks may be considered for defining the range.

Also, log-linear feature categories and its semantic net might be edited so that features containing the observed variables *area*, *squareness* and *lengthness* at different range categories be privileged in the search for the optimum solution. USTs are complex and their complexity is inevitably reflected on the complexity of the features and the network. Furthermore, SAR image regions have to be classified based on its neighbouring regions classes besides its observed attributes. These conditions threat tractability. Good remedies are (1) the radical reduction of the observed variables cardinalities and (2) to avoid as much as possible over-segmentation of the SAR scene.

5 Summary and future work

In this paper we summarized how the network structure and the parameters of a CRF model can be learned in a tractable way by optimizing the L1-regularized log-likelihood objective function. We showed how the training can be carried out in a generative or discriminative way. Besides, the main elements of a rule-based system for guiding the insertion of log-linear features in the optimization process were shown. This system enables the insertion of expert knowledge in the CRF learning procedure and is expected to accelerate the convergence to a global optimal solution. In the next works we will test these hypotheses by applying this methodology for classifying UST using high-resolution space-borne InSAR data.

6 References

COWELL, R.G.; DAWID, A. P.; LAURITZEN, S.L. AND SPIEGELHALTER, D.J., 1999: Probabilistic Networks and Expert Systems. Springer Verlag: New York. P.321.

DELLA PIETRA, S.; DELLA PIETRA, V.J. & LAFFERTY, 1997: Inducing features of random fields. IEEE Transactions on Pattern Analysis and Machine Learning, **19**(4), P.380-393.

GANAPATHI V, VICKREY D, DUCHI JC, KOLLER D., 2008: Constrained Approximate Maximum Entropy Learning of Markov Random Fields. In: UAI 2008, Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence, Helsinki, Finland, July 9-12, 2008: P.196-203.

HEIDEN, U., HELDENS, W., ROESSNER, S., SEGLL, K., ESCH, T. & MUELLER, A., 2012: Urban structure type characterization using hyperspectral remote sensing and height information. Landscape and Urban Planning, **105**, P.361-375.

KOLLER, D. & FRIEDMAN, N., 2009: Probabilistic Graphical Models – Principles and Techniques. The MIT Press: Cambridge: USA. P.1231.

LEE S, GANAPATHI, V. & KOLLER, D., 2007: Efficient structure learning of Markov networks using L1-regularization. In: NIPS (Neural Information Processing Systems).

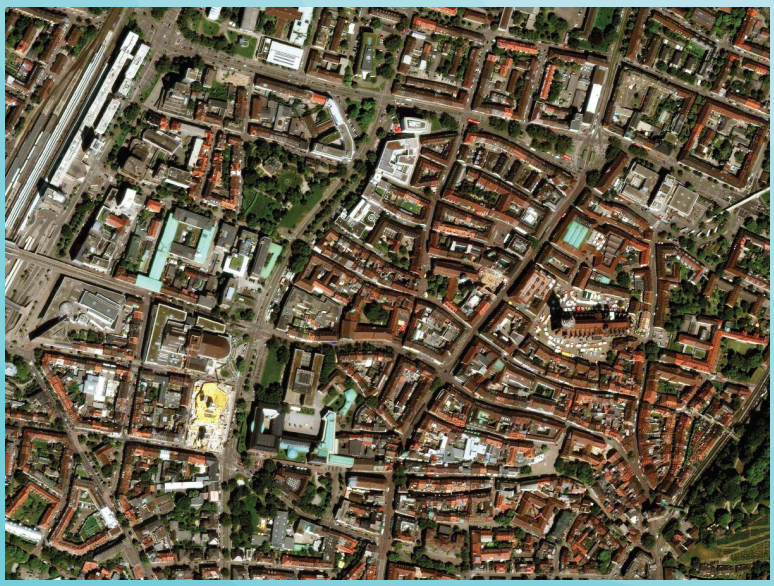
MURPHY, K., WEISS, Y. & JORDAN. M., 1999: Loopy belief propagation for approximate inference: an empirical study. In UAI, 1999.

PERKINS, S.; LACKER, K. & THEILER, J., 2003: Grafting: Fast, incremental feature selection by gradient descent in function space. **3**, P.1333-1356.

Publikationen der Deutschen
Gesellschaft für Photogrammetrie,
Fernerkundung und Geoinformation e.V.

DGPF

Band 22 **2013**



Vorträge

33. Wissenschaftlich-Technische Jahrestagung der DGPF

27. Februar – 1. März 2013
in Freiburg i. B.

Dreiländertagung D - A - CH

ISSN 0942-2870

Eckhardt Seyfert, Hrsg.