Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Continuous Conditional Random Fields for Efficient Regression in Large Fully Connected Graphs

Kosta Ristovski,[1] **Vladan Radosavljevic,**[1] **Slobodan Vucetic,** and **Zoran Obradovic**
Center for Data Analytics and Biomedical Informatics, Temple University Philadelphia, PA 19122 USA
{kosta, vladan, vucetic, zoran.obradovic}@temple.edu

Abstract

When used for structured regression, powerful Conditional Random Fields (CRFs) are typically restricted to modeling effects of interactions among examples in local neighborhoods. Using more expressive representation would result in dense graphs, making these methods impractical for large-scale applications. To address this issue, we propose an effective CRF model with linear scale-up properties regarding approximate learning and inference for structured regression on large, fully connected graphs. The proposed method is validated on real-world large-scale problems of image denoising and remote sensing. In conducted experiments, we demonstrated that dense connectivity provides an improvement in prediction accuracy. Inference time of less than ten seconds on graphs with millions of nodes and trillions of edges makes the proposed model an attractive tool for large-scale, structured regression problems.

Introduction

In a typical regression setting, we are given a data set with N training examples, $D = (\mathbf{x}_i, y_i), i = 1, \ldots, N$, where $\mathbf{x}_i \in X \subset \mathcal{R}^M$ is an M-dimensional vector of explanatory variables, and $y_i \in \mathcal{R}$ is a real-valued output variable. The objective is to learn a mapping $f : X \rightarrow \mathcal{R}$ that predicts output y as accurately as possible given $\mathbf{x}$. This setup is appropriate when examples are independently and identically distributed (IID). The IID assumption is often violated in structured data, where examples exhibit sequential, temporal, spatial, spatio-temporal, or some other dependencies. In such cases, the traditional supervised learning approaches could result in a weak model with low prediction accuracy (Neville et al. 2012).

This issue can be addressed by employing a structured learning model. In structured learning, the model represents mapping $f : X^N \rightarrow \mathcal{R}^N$, which simultaneously predicts all outputs given all inputs. If dependencies among output variables exist, the use of a structured approach often results in accuracy improvements over unstructured approaches that predict independently for each example.

Copyright © 2013, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]These authors contributed equally

In learning from structured data, Markov Random Fields (MRF) (Solberg and Jain 1996) and more recently proposed Conditional Random Fields (CRF) (Lafferty and Pereira 2001) are among the most popular models. Originally, CRFs were designed for classification of sequential data (Lafferty and Pereira 2001). However, using CRF for regression is a less explored topic which was initially addressed by Continuous CRF (CCRF) (Qin et al. 2008) that was used for document retrieval. A method related to CRF for regression of sequential data was proposed in (Kim and Pavlovic 2009). In (Tappen, Adelson, and Freeman 2007), a continuous valued CRF was applied on image denoising applications whereas CRF for remote sensing was described in (Radosavljevic, Vucetic, and Obradovic 2010).

Due to computational tractability, traditional CRFs model interactions among examples in a local neighborhood (e.g., temporal in time sequences or spatial in images), thus enforcing local smoothness in data. A more expressive fully-connected CRF allows us to relax or even drop the notion of local neighborhood and model interactions across the entire data. However, with such a large set of interactions, exact learning and inference algorithms would take $\mathcal{O}(N^3)$ computational time, which is prohibitive when the size of data grows. Approximation algorithms, including variants of belief propagation, graph-cut based methods, and mean field approximation (Kolmogorov 2006; Payet and Todorovic 2010; Toyoda and Hasegawa 2008) are limited to medium-scale data because of computational complexity proportional to $\mathcal{O}(N^2)$.

Significant efforts have been recently devoted to accelerating learning and inference in fully-connected CRFs for classification using fast filter-based methods. Recent studies (Krahenbuhl and Koltun 2011; Zhang and Chen 2012; Vineet, Warrell, and Torr 2012) strongly suggest that mean field techniques together with recent advances in data structures that support fast filtering offer opportunity for highly efficient $\mathcal{O}(N)$ modeling of large structured data. Interestingly, apart from mean field approximation of continuous valued MRF (Schelten and Roth 2012) with $\mathcal{O}(N^2)$ time complexity, regression models with continuous valued outputs have not been much explored in this setting.

In this work, we propose a novel CRF based structured regression model (FF-CCRF) with linear learning and inference times in terms of N on fully-connected graphs. The

FF-CCRF model is based on a fast Gaussian filtering applied on mean field approximation of the CCRF model in which the pairwise interactions were weighted by Gaussian kernel-distances between examples. Our proposed approach is similar to (Krahenbuhl and Koltun 2011) but with important differences. First, our method is targeted to regression problems with continuous outputs, where one faces new computational challenges that cannot be solved by a straightforward extension of the existing methods developed for classification. Second, unlike (Krahenbuhl and Koltun 2011), which used cross-validation, we developed a gradient ascent approach for learning model parameters.

To demonstrate the convenience and power of the FF-CCRF model, we applied it on two real-world large-scale structured regression applications. The experimental results indicate that the FF-CCRF improves prediction accuracy by efficiently utilizing information from fully connected structured data.

Fully Connected Continuous Conditional Random Fields

Continuous conditional random fields are used to model conditional distribution $P(\mathbf{y}|X), \mathbf{y} = (y_1 \dots y_N)$ (Qin et al. 2008), as

$$P(\mathbf{y}|X) = \frac{1}{Z(X, \boldsymbol{\alpha}, \boldsymbol{\beta})} \exp\left(\phi(\mathbf{y}, X, \boldsymbol{\alpha}, \boldsymbol{\beta})\right) \qquad (1)$$

where the term in the exponent $\phi(\mathbf{y}, X, \boldsymbol{\alpha}, \boldsymbol{\beta})$, and normalization constant $Z(X, \boldsymbol{\alpha}, \boldsymbol{\beta})$ are defined as

$$\begin{aligned} \phi(\mathbf{y}, X, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \sum_{i=1}^{N} A(\boldsymbol{\alpha}, y_i, X) + \sum_{i \sim j} I(\boldsymbol{\beta}, y_i, y_j, x), \\ Z(X, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \int_{\mathbf{y}} \exp\left(\phi(\mathbf{y}, X, \boldsymbol{\alpha}, \boldsymbol{\beta})\right) d\mathbf{y}. \end{aligned} \qquad (2)$$

To have a feasible structured regression model, Z must be integrable. On the other hand, models for classification are always feasible because Z is finite and defined as a sum over finitely many values of $\mathbf{y}$.

The output y_i is associated with explanatory variables X by a real-valued function called the association potential $A(\boldsymbol{\alpha}, y_i, X)$, where $\boldsymbol{\alpha}$ is a K-dimensional set of parameters. In general, A takes as input X, which could be any useful combination of explanatory variables from data set D. To model interactions between outputs $y_i \sim y_j$, a real valued function called the interaction potential $I(\boldsymbol{\beta}, y_i, y_j, X)$ is used, where $\boldsymbol{\beta}$ is an L dimensional set of parameters.

In CRF applications, A and I are often conveniently defined as linear combinations of a set of feature functions f and g in terms of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ (Lafferty and Pereira 2001),

$$\begin{aligned} A(\boldsymbol{\alpha}, y_i, X) &= \sum_{k=1}^{K} \boldsymbol{\alpha}_k f_k(y_i, X), \\ I(\boldsymbol{\beta}, y_i, y_j, x) &= \sum_{l=1}^{L} \boldsymbol{\beta}_l g_l(y_i, y_j, X). \end{aligned} \qquad (3)$$

The use of feature functions is convenient because it allows modeling of arbitrary relationships between inputs and outputs. In this way, any potentially relevant feature function could be included to the model and its degree of relevance would be determined automatically by the learning algorithm.

Feature functions

Construction of appropriate feature functions in CRF is a manual process that depends on prior beliefs of a practitioner about what features could be useful. The choice of features is often constrained to reduce the complexity of learning and inference from CRF. In general, to evaluate $P(\mathbf{y}|X)$ during learning and inference, one would need to use time consuming sampling methods (Xin et al. 2009). However, if A and I are defined as quadratic functions of $\mathbf{y}$, learning and inference can be accomplished in a computationally efficient manner (Qin et al. 2008). Let us assume we are given K unstructured models, $R_k(X), k = 1, \dots K$, that predict single output y_i taking into account X (as a special case, only x_i can be used as X). The quadratic feature functions for the association potential can be written as

$$f_k(y_i, X) = -(y_i - R_k(X))^2, \quad k = 1 \dots K. \qquad (4)$$

These feature functions follow the basic principle for association potentials in that their values are large when predictions and outputs are similar. The quadratic feature functions for the interaction potential can be written as

$$g_l(y_i, y_j, X) = -k_l(\mathbf{p}_i^{(l)}, \mathbf{p}_j^{(l)})(y_i - y_j)^2, \; l = 1 \dots L, \quad (5)$$

where each $k_l (k_l > 0)$ is some similarity measure between feature vectors $\mathbf{p}_i^{(l)}$ and $\mathbf{p}_j^{(l)}$ in arbitrary feature space that is a subset of the set of explanatory variables. These feature functions impose that outputs y_i and y_j have similar values if they are close in k_l. Similarity measures k_l provide (a potentially useful) rich set of interactions between each pair of outputs producing fully connected CCRF.

Approximation of Fully-connected CCRF Using Fast Gaussian Filtering

Exact inference on fully connected CCRF requires computational time of $\mathcal{O}(N^3)$ due to the need for calculations of inverse matrices (Radosavljevic, Vucetic, and Obradovic 2010). As the first step to reduce computational time we need to approximate conditional distribution $P(\mathbf{y}|X)$ by mean field theory (Koller and Friedman 2009). The objective is to approximate distribution $P(\mathbf{y}|X)$ with distribution $Q(\mathbf{y}|X)$ that can be expressed as a product of independent marginals $Q(\mathbf{y}|X) = \prod_{i=1}^{N} Q_i(y_i|X)$. Under mean field theory, the best approximation of P is such a distribution Q which minimizes Kullback-Leibler (KL) divergence between P and Q. The solution for Q has form (Bishop 2006)

$$\log(Q_i(y_i|X)) = E_{j \neq i}[\log P(\mathbf{y}|X)] + const, \qquad (6)$$

where $E_{j \neq i}$ denotes an expectation under Q distributions over all variables y_j for $j \neq i$. Combining (1), (4), (5), and

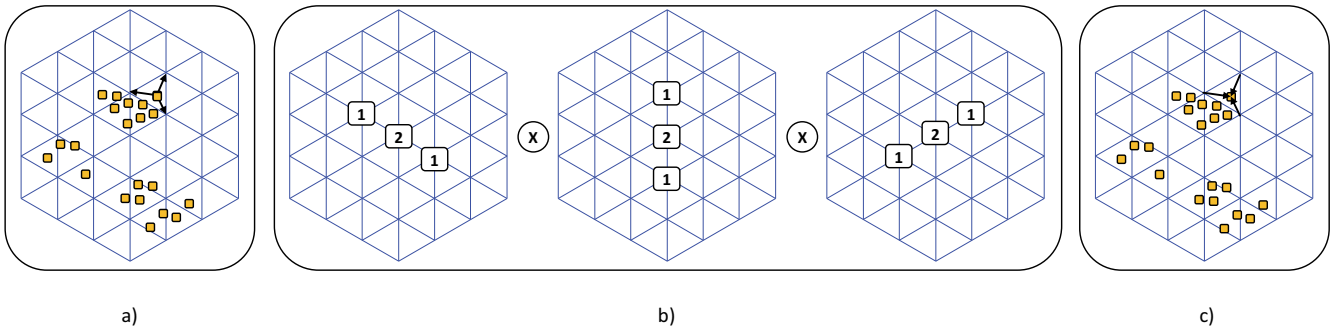

Figure 1: Stages of fast Gaussian filtering: a) splatting onto vertices of permutohedral lattice b) filtering convolution of discrete Gaussian kernels [1 2 1] along each axis c) slicing (interpolation) to original positions.

(6) we obtain

$$\log(Q_i(y_i|X)) = -\sum_{k=1}^{K}\alpha_k(y_i^2 - 2y_iR_k(X)) \\ -2\sum_{l=1}^{L}\beta_l\sum_{j\neq i}k_l(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)})(y_i^2 - 2y_iE[y_j]) + const, \quad (7)$$

Each $\log Q_i(y_i|X)$ is a quadratic form with respect to y_i so it can be represented as Gaussian distribution whose mean and variance are

$$\mu_i = \frac{\sum_{k=1}^{K}\alpha_kR_k(X) + 2\sum_{l=1}^{L}\beta_l\sum_{j\neq i}k_l(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)})\mu_j}{\sum_{k=1}^{K}\alpha_k + 2\sum_{l=1}^{L}\beta_l\sum_{j\neq i}k_l(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)})}, \quad (8)$$

$$\sigma_i^2 = \frac{1}{2(\sum_{k=1}^{K}\alpha_k + 2\sum_{l=1}^{L}\beta_l\sum_{j\neq i}k_l(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)}))}. \quad (9)$$

To calculate all $\mu_i, i = 1\dots N$ we need to solve a linear system of equations in (8) which can be done iteratively (Bishop 2006) until: (1) a pre-specified number of iterations I is exceeded or (2) guaranteed convergence is reached. If $I << N$ an iterative approach will reduce computational time from $\mathcal{O}(N^3)$ to $\mathcal{O}(IN^2)$ by doing summation over all $j \neq i$ for each i in every iteration. The calculation of σ_i has a closed form solution (9) and also requires summation over all $j \neq i$ for each i, thus having $\mathcal{O}(N^2)$ computational time. To speed up this process, the time-consuming summations can be expressed as

$$\sum_{j\neq i}k_l(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)})\mu_j = \sum_{j}k_l(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)})\mu_j - \mu_i, \quad (10)$$

$$\sum_{j\neq i}k_l(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)}) = \sum_{j}k_l(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)}) - 1. \quad (11)$$

If k_l is a Gaussian kernel

$$k(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)}) = \exp\left(-\frac{1}{2}(\mathbf{p}_i^{(l)} - \mathbf{p}_j^{(l)})^T\Lambda_l(\mathbf{p}_i^{(l)} - \mathbf{p}_j^{(l)}\right) \quad (12)$$

where symmetric, positive-definite matrix $\Lambda^{(l)}$ defines kernel shape, then sums over all examples j in (10) and (11) can be computed using signal processing techniques. From a signal processing point of view, each example i can be represented as a point in d dimensional space with coordinates $\mathbf{p}_i^{(l)}$ and value $\mathbf{s}_i$. Having such representation, $\sum_j k_l(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)})\mathbf{s}_i$ for all examples i can be calculated at once by convolving Gaussian kernel G with a whole set $S = \{\mathbf{s}_i, i = 1\dots N\}$ (an approach know as Gaussian filtering $G \otimes S$). The value of resulting Gaussian filtering at point i with coordinates $\mathbf{p}_i^{(l)}$ is

$$(G \otimes S)(\mathbf{p}_i^{(l)}) = \sum_{j}k_l(\mathbf{p}_i^{(l)},\mathbf{p}_j^{(l)})\mathbf{s}_i. \quad (13)$$

However, straightforward application of Gaussian filtering still requires iterating over all examples j for each example i, giving $\mathcal{O}(N^2)$ computational time.

Fast High-Dimensional Gaussian Filtering

By projecting data onto carefully designed uniform structures that are composed of vertices, which tessellate feature space, filtering can be approximately computed by

- *splatting*: each $\mathbf{s}_i$ with coordinates $\mathbf{p}_i^{(l)}$ is projected onto vertices from enclosing structure (Figure 1a) using appropriate weights.
- *filtering*: convolution is performed on vertices using discrete Gaussian kernels [1 2 1] along each direction in the structure independently as shown in Figure 1b.
- *slicing*: final filtering result is obtained by interpolation from vertices to the positions $\mathbf{p}_i^{(l)}$ (Figure 1c).

One of the uniform structures that can be used for fast filtering is a regular grid, proposed in (Paris and Durand 2009; Adams et al. 2009). The filtering algorithm developed on regular grid has complexity of $\mathcal{O}(2^dN)$ because the projection of $\mathbf{s}_i$ onto 2^d vertices of d-dimensional hypercube requires $\mathcal{O}(2^d)$ time. A novel approach (Adams, Baek, and Davis 2010) offers implementation of fast filtering on a permutohedral lattice, where algorithm complexity was reduced to $\mathcal{O}(d^2N)$. An example of filtering on a two-dimensional permutohedral lattice is shown in Figure 1. As suggested in (Adams, Baek, and Davis 2010) a permutohedral lattice

is preferable if $d \in [4, 8]$, while regular grid should be the structure of choice if $d \in [2, 3]$. Both structures are equally good for $d = 1$. The choice of the structure depends on the application, so our FF-CCRF algorithm uses fast Gaussian filter implementation on a permutohedral lattice.

Fast inference

The inference task is to find the outputs $\mathbf{y}$ for given inputs X, such that the conditional probability $P(\mathbf{y}|X)$ is maximized. As we approximate conditional distribution with product of independent Gaussian marginals, an estimate of each y_i is obtained as the expected value μ_i of the corresponding Gaussian Q_i

$$y_i = \arg\max_{y_i}(Q_i(y_i|X)) = \mu_i. \tag{14}$$

Using fast Gaussian filter in (8) computational time of iterative calculation of all μ_i is upper bounded by $O(Id^2N)$ where I is a pre-specified maximum number of iterations (I set to 200 in our applications).

Fast learning

The learning task is to choose $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ to maximize the conditional log-likelihood

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}}(\mathcal{L}(\boldsymbol{\theta})), \text{ where } \mathcal{L}(\boldsymbol{\theta}) = \log P(\mathbf{y}|X) \tag{15}$$

$\boldsymbol{\theta}$ can be learned by the gradient-based optimization. To apply it, we need to find the gradient of the conditional log-likelihood

$$\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = \frac{\partial\phi(\mathbf{y}, X, \boldsymbol{\theta})}{\partial\boldsymbol{\theta}} - \frac{\partial(logZ)}{\partial\boldsymbol{\theta}} + const. \tag{16}$$

$\phi(\mathbf{y}, X, \boldsymbol{\theta})$ is linear with respect to $\boldsymbol{\theta}$ so its partial derivative is straightforward from (2) and (3). The derivative of $logZ$ has more complex form

$$\begin{aligned}\frac{\partial(logZ)}{\partial\boldsymbol{\theta}} &= \frac{1}{Z}\frac{\partial}{\partial\boldsymbol{\theta}}\int_{\mathbf{y}}\exp\left(\phi(\mathbf{y}, X, \boldsymbol{\theta})\right)d\mathbf{y} \\ &= \int_{\mathbf{y}}\phi(\mathbf{y}, X, \boldsymbol{\theta})P(\mathbf{y}|X)d\mathbf{y} = E_P[\phi(\mathbf{y}, X, \boldsymbol{\theta})].\end{aligned} \tag{17}$$

To ensure that the model is feasible we impose the constraint that all elements of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are greater than 0. To convert this constraint problem to the an unconstrained optimization, we adopt a technique used in (Qin et al. 2008) that applies the exponential transformation of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to guarantee that they are positive

$$\alpha_k = \exp(u_k), \;\; \beta_l = \exp(v_l). \tag{18}$$

Using the chain rule we find derivatives of log-likelihood with respect to u_k and v_l

$$\frac{\partial\mathcal{L}}{\partial u_k} = \frac{\partial\mathcal{L}}{\partial\alpha_k}\exp(u_k), \;\; \frac{\partial\mathcal{L}}{\partial v_l} = \frac{\partial\mathcal{L}}{\partial\beta_l}\exp(v_l). \tag{19}$$

Exact calculation of (19) requires an inversion of a large matrix, which takes $\mathcal{O}(N^3)$ time in fully connected model (Qin et al. 2008). To speed up learning, we calculate approximate gradients of the log-likelihood $\partial\hat{\mathcal{L}}/\partial\alpha_k$ and $\partial\hat{\mathcal{L}}/\partial\beta_l$ instead of exact ones (Vishwanathan et al. 2006). By substituting $P(\mathbf{y}|X)$ in (16) and (17) with $Q(\mathbf{y}|X) = \prod_{i=1}^{N} Q_i(y_i|X)$ we obtain the expressions for $\partial\hat{\mathcal{L}}/\partial\alpha_k$ and $\partial\hat{\mathcal{L}}/\partial\beta_l$

$$\begin{aligned}\frac{\partial\hat{\mathcal{L}}}{\partial\alpha_k} &= \sum_{i=1}^{N}(E_Q[(y_i - R_k(X))^2] - (y_i - R_k(X))^2), \\ \frac{\partial\hat{\mathcal{L}}}{\partial\beta_l} &= \sum_{i=1}^{N}\sum_{j\neq i} k_l(\mathbf{p}_i^{(l)}, \mathbf{p}_j^{(l)})(E_Q[(y_i - y_j)^2] - \\ &\quad - (y_i - y_j)^2).\end{aligned} \tag{20}$$

We can use the fact that $E_Q(y_i) = \mu_i$, $E_Q(y_i^2) = \mu_i^2 + \sigma_i^2$, $E_Q(y_iy_j) = \mu_i\mu_j$ together with fast Gaussian filtering to efficiently compute derivatives in (20). To calculate all μ and σ we need $\mathcal{O}(Id^2N)$ time. Finding derivatives with respect to $\boldsymbol{\alpha}$ requires summation over N variables, which takes $\mathcal{O}(KN)$ time. Finding derivatives with respect to $\boldsymbol{\beta}$ requires summation over all $j \neq i$ examples for each i and would have $\mathcal{O}(N^2)$ time if implemented naively. However, by close inspection of (20) all terms are weighted by k_l and thus they can be calculated from the single call of a fast Gaussian filter with $\mathbf{s_j} = (E_Q(y_j), E_Q(y_j^2), y_j, y_j^2, 1)$. Computational time to find derivatives with respect to $\boldsymbol{\beta}$ becomes $\mathcal{O}(LId^2N)$. Learning procedure is summarized in Algorithm 1 with total complexity $\mathcal{O}(T(Id^2 + K + LId^2)N)$ where T is the number of gradient-ascent based iterations.

Algorithm 1 Learning of FF-CCRF

Input: X, $R(X)$, $\mathbf{p}$
Initialize: u, v
repeat
  Calculate α and β using (18).
  Calculate all μ_i and σ_i^2 from (8) and (9) by applying fast Gaussian filtering.
  Update u and v by gradient ascent algorithm using derivatives in (20) and (19).
until Convergence

Experiments

To demonstrate the power of FF-CCRF we preformed experiments on two large-scale real-world applications: (1) remote sensing and (2) denoising of large images. All experiments were done on Intel Core i7-3770 @3.4GHz. Main code was written in Matlab while filtering was performed in C++, which was integrated with Matlab through mex interface.

Remote sensing application

Our remote sensing application is related to prediction of aerosol optical depth (AOD) in the atmosphere from satellite observations. It has been recognized that one of the main challenges in climate research is to characterize and quantify the effect of aerosols on earth's radiation budget (Hansen, Sato, and Ruedy 1997). Thus, AOD prediction plays an important role in modern climate research. Our

Table 1: Root mean squared error (RMSE) of fast fully connected CRF (FF-CCRF), fully-connected CCRF, and domain-knowledge based algorithm (C005) for aerosol optical depth prediction on different test set sizes N.

SIZE	$N = 3,000$	$N = 20,000$
FF-CCRF	$\mathbf{0.117 \pm 0.001}$	$\mathbf{0.115 \pm 0.005}$
C005	0.126 ± 0.010	0.129 ± 0.010
CCRF	$\mathbf{0.116 \pm 0.001}$	N/A

dataset contained 53,895 examples equally distributed over two years. The dataset was composed of ground truth ($\mathbf{y}$), satellite observations represented by five reflectances ($X = \{x_1, x_2, x_3, x_4, x_5\}$), and an AOD prediction from domain-knowledge based operational algorithm named C005 (Remer and Kaufman 2005) (unstructured predictor $R_1(X)$). We created $L = 5$ kernels defined separately for each reflectance ($\mathbf{p}^{(l)} = x_l, l = 1 \dots 5$). Shape factor $\Lambda^{(l)}$ for each kernel l was set to standard deviation of the corresponding reflectance.

Accuracy of FF-CCRF vs. accuracy of exact fully-connected CCRF. In order to compare accuracy of FF-CCRF to exactly learned fully-connected CCRF with the same feature functions, we had to perform experiments on a smaller datasets due to the time-consuming learning algorithm for CCRF. We randomly selected 3,000 data points from one year for training and 3,000 data points from another year for testing. We repeated this experiment 100 times and report average and standard deviation of root mean square error (RMSE) in Table 1 (lower RMSE is better). We notice that there is only a slight decrease in accuracy of approximative FF-CCRF when compared to the CCRF, while both approaches outperform C005 algorithm.

FF-CCRF on large-scale remote-sensing data. To test computational efficiency of FF-CCRF algorithm, we performed experiments on a large-scale dataset where CCRF was not applicable. We trained FF-CCRF on all examples (∼25,000) from one year and tested it on randomly selected 20,000 examples from another year. We repeated this procedure 100 times to obtain results reported in Table 1, column $N = 20,000$. We see that there is ∼10% increase in accuracy in favor of FF-CCRF as compared to C005. Also, FF-CCRF was able to utilize information from a large number of interactions and to slightly improve accuracy when compared to the results obtained on the smaller datasets, although the larger dataset contained more noise (domain-based C005 performed worse on the large dataset than it did on the small dataset).

Denoising of large images

We applied our FF-CCRF on the problem of restoring an image y from its noisy observation x (denoising). Image denoising is a corner-stone of image processing and has been an active research field for a long time. In recent years, a major progress in the field has been made using the non-local means (NLM) algorithm. NLM represents a pixel

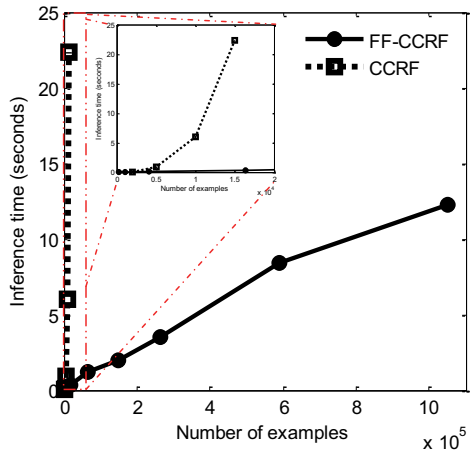

Figure 2: FF-CCRF vs CCRF inference time as a function of the test data size (CCRF was infeasible for large data). The small plot contains zoomed area of the bigger plot for small number of examples.

$i, i = 1 \dots N$ with a feature vector composed of noisy observations x in $r \times r$ neighborhood (patch) centered around pixel i. A denoised estimate $\hat{y}_i$ of pixel i is then a weighted average of the entire set of noisy pixels x_j where weights are proportional to the similarity between i's and j's feature vectors.

Such an exhaustive averaging at each pixel leads to $\mathcal{O}(N^2)$ computational time, which is infeasible for large images. Therefore, due to computational issues, averaging has been restricted to a square region of modest size $R \times R$ ($R < 30$) (Salmon and Strozecki 2012) centered around pixel i. This restriction usually performs poorly on pixels in textured areas because their patches may only have a few similar patches in an $R \times R$ region. To alleviate this problem, we used a fully connected FF-CCRF model on top of NLM that allowed for matching of similar patches across the whole image in linear time.

As an unstructured predictor $R_1(X)$, we used a denoised estimate provided by recently proposed improved NLM algorithm (Buades, Coll, and Morel 2011). We created $L = 1$ kernel with features $\mathbf{p}$ derived from 3×3 local neighborhood around each pixel. In order to reduce the dimensionality we projected feature vectors to a lower dimensional subspace by Principal Component Analysis (PCA) as in (Van De Ville and Kocher 2011). It was shown that projecting patch based feature vectors to $d = 6$ dimensions by PCA gives satisfying denoising results on typical images. Setting appropriate shape $\Lambda^{(1)}$ was also explored in (Tasdizen 2009). Shape depends on patch size (the larger the patch the larger the scaling). We used $\Lambda^{(1)}$ proportionally to the patch size 3×3.

FF-CCRF for large-scale image denoising. We evaluated the FF-CCRF on a standard benchmark dataset for image denoising (Roth and Black 2005). The training set of noisy/noise-free images was constructed over 40 gray-level

Figure 3: A qualitative example of denoising on an image from test set. a) Noise-free, b) Noisy ($\sigma = 25$), c) Non-local means estimate (PSNR = 27.73), d) FF-CCRF estimate (PSNR = 28.15). The FF-CCRF model preserves the texture in the image.

Table 2: Average root mean squared error (RMSE) over 68 test images as result of denoising by FF-CCRF and NML (lower RMSE is better).

NOISE LEVEL	10	15	20	25
NLM	6.20	8.11	9.39	10.68
FF-CCRF	**6.05**	**7.85**	**9.17**	**10.43**

Table 3: Average peak signal-to-noise ratio (PSNR) over 68 test images as result of denoising by FF-CCRF, NML, and FoE (higher PSNR is better).

NOISE LEVEL	10	15	20	25
FOE	32.68	**30.50**	28.78	27.60
NLM	32.46	30.14	28.95	27.83
FF-CCRF	**32.70**	30.43	**29.15**	**28.02**

training images (Roth and Black 2005) with 154,401 pixels each. White Gaussian noise with $\sigma = 10, 15, 20,$ or 25 was added to noise-free images to construct noisy images. Similarly, the test set was constructed over 68 gray-level test images (Roth and Black 2005), also with 154,401 pixels each. The denoising quality is usually assessed by the root mean square error (RMSE) and the peak signal-to-noise ratio (PSNR) (Van De Ville and Kocher 2011). Our results are shown in Tables 2 and 3. FF-CCRF improved both RMSE and PSNR as compared to NML approach for all noise levels. In Figure 3 we present qualitative comparison of FF-CCRF and NML on an image from test set. We also compared PSNR of FF-CCRF to PSNR available in literature of the state-of-the-art Field of Experts (FoE) denoising model (Roth and Black 2009). FF-CCRF outperforms FoE for large noise levels while the results are comparable for low noise levels.

Computational efficiency of FF-CCRF. In order to evaluate computational efficiency of the learned FF-CCRF model, we generated a set of synthetic images of various sizes. We took a synthetic image (upper half contains black pixels while lower half contains white pixels) and varied its size from 16×16 (256 pixels with ten thousand interactions) to 1024×1024 (a million of pixels with a trillion interactions). Noisy synthetic images were corrupted by white Gaussian noise with standard deviation 25. We tested learned FF-CCRF on such a dataset. Simulation result is presented in Figure 2, which shows that inference time scales linearly with test data size comparing to cubic time of CCRF. On data with a millions of examples and a trillions of interactions, inference can be done in less than ten seconds. Moreover, FF-CCRF performed consistently better in terms of PSNR than NML on all synthetic images having an average PSNR = 42.15 while NML achieved PSNR = 40.01.

Conclusion

We proposed a new Fast Fully Connected Continuous CRF (FF-CCRF) model that is able to combine the outputs of nonstructured regression models and exploit the correlation between output variables in a fully connected model. The FF-CCRF uses mean field approximation and fast high dimensional filtering to achieve linear computational cost of both inference and learning. We demonstrated that inference can be performed in seconds on data with a million of examples and a trillion of interactions. The proposed method is also readily applicable to other large-scale regression applications where there is a need for knowledge integration, data fusion, and exploitation of correlation among outputs.

Acknowledgement

We are greatful to Nemanja Djuric for his valuable comments. This work is supported in part by DARPA Grant FA9550-12-1-0406 negotiated by AFOSR, and NSF grant IIS-1117433.

References

Adams, A.; Baek, J.; and Davis, M. A. 2010. Fast High-Dimensional Filtering Using the Permutohedral Lattice. *Computer Graphics Forum* 29(2):753–762.

Adams, A.; Gelfand, N.; Dolson, J.; and Levoy, M. 2009. Gaussian KD-trees for fast high-dimensional filtering. *ACM Transactions on Graphics* 28(3):1.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Buades, A.; Coll, B.; and Morel, J.-M. 2011. Non-Local Means Denoising. *Image Processing On Line* 2011.

Hansen, J.; Sato, M.; and Ruedy, R. 1997. Radiative forcing and climate response. *Journal of Geophysical Research: Atmospheres* 102(D6):6831–6864.

Kim, M., and Pavlovic, V. 2009. Discriminative Learning for Dynamic State Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(10):1847–1861.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Kolmogorov, V. 2006. Convergent tree-reweighted message passing for energy minimization. *IEEE transactions on pattern analysis and machine intelligence* 28(10):1568–83.

Krahenbuhl, P., and Koltun, V. 2011. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. *Neural Information Processing Systems*.

Lafferty, J, M. A., and Pereira, F. 2001. Conditional random fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings International Conference on Machine Learning*.

Neville, J.; Gallagher, B.; Eliassi-Rad, T.; and Wang, T. 2012. Correcting evaluation bias of relational classifiers with network cross validation. *Knowledge and Information Systems* 30:31–55.

Paris, S., and Durand, F. 2009. A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach. *International Journal of Computer Vision* 81(1):24–52.

Payet, N., and Todorovic, S. 2010. (RF)ˆ2 - Random Forest Random Field. In *Neural Information Processing Systems (NIPS)*, volume 1, 1–9.

Qin, T.; Liu, T.; Zhang, X.; Wang, D.; and Li, H. 2008. Global Ranking Using Continuous Conditional Random Fields. *Neural Information Processing Systems*.

Radosavljevic, V.; Vucetic, S.; and Obradovic, Z. 2010. Continuous conditional random fields for regression in remote sensing. In *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, 809–814. Amsterdam, The Netherlands, The Netherlands: IOS Press.

Remer, L. A., and Kaufman, Y. 2005. The modis aerosol algorithm, products and validation. *Journal of the Atmospheric Sciences* 62:947–973.

Roth, S., and Black, M. 2005. Fields of Experts: A Framework for Learning Image Priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, 860–867.

Roth, S., and Black, M. J. 2009. Fields of experts. *Int. J. Comput. Vision* 82(2):205–229.

Salmon, J., and Strozecki, Y. 2012. Patch reprojections for non-local methods. *Signal Processing* 92(2):477 – 489.

Schelten, K., and Roth, S. 2012. Mean field for continuous high-order mrfs. In Pinz, A.; Pock, T.; Bischof, H.; and Leberl, F., eds., *DAGM/OAGM Symposium*, volume 7476 of *Lecture Notes in Computer Science*, 52–61. Springer.

Solberg, A H S, T. T., and Jain, A. K. 1996. A Markov Random Field Model for Classification of Multisource Satellite Imagery. *IEEE Transactions on Geoscience and Remote Sensing* 34(1):100–113.

Tappen, M F, L. C.; Adelson, E. H.; and Freeman, W. T. 2007. Learning Gaussian Conditional Random Fields for Low-Level Vision. *IEEE Conference on Computer Vision and Pattern Recognition*.

Tasdizen, T. 2009. Principal neighborhood dictionaries for nonlocal means image denoising. *Image Processing, IEEE Transactions on* 18(12):2649 –2660.

Toyoda, T., and Hasegawa, O. 2008. Random Field Model for Integration of Local Information and Global Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(8):1483–1489.

Van De Ville, D., and Kocher, M. 2011. Nonlocal means with dimensionality reduction and sure-based parameter selection. *Image Processing, IEEE Transactions on* 20(9):2683 –2690.

Vineet, V.; Warrell, J.; and Torr, P. 2012. Filter-based Mean-Field Inference for Random Fields with Higher-Order Terms and Product Label-Spaces. In *European Conference on Computer Vision (ECCV)*.

Vishwanathan, S. V. N.; Schraudolph, N. N.; Schmidt, M. W.; and Murphy, K. P. 2006. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, 969–976. New York, NY, USA: ACM.

Xin, X.; King, I.; Deng, H.; and Lyu, M. R. 2009. A social recommendation framework based on multi-scale continuous conditional random fields. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, 1247–1256. New York, NY, USA: ACM.

Zhang, Y., and Chen, T. 2012. Efficient inference for fully-connected CRFs with stationarity. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 582–589. IEEE.