GAUSSIAN CONDITIONAL RANDOM FIELDS FOR REGRESSION IN REMOTE SENSING

A Dissertation
Submitted
to the Temple University Graduate Board

In Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

By
Vladan Radosavljevic
January, 2012

Examining Committee Members:

Dr Zoran Obradovic, Advisory Chair, Computer and Information Science
Dr Slobodan Vucetic, Computer and Information Science
Dr Longin Jan Latecki, Computer and Information Science
Dr Jeremy Mennis, External Member, Department of Geography & Urban Studies

ABSTRACT

In recent years many remote sensing instruments of various properties have been employed in an attempt to better characterize important geophysical phenomena. Satellite instruments provide an exceptional opportunity for global long-term observations of the land, the biosphere, the atmosphere, and the oceans. The collected data are used for estimation and better understanding of geophysical parameters such as land cover type, atmospheric properties, or ocean temperature. Achieving accurate estimations of such parameters is an important requirement for development of models able to predict global climate changes. One of the most challenging climate research problems is estimation of global composition, load, and variability of aerosols, small airborne particles that reflect and absorb incoming solar radiation.

The existing algorithm for aerosol prediction from satellite observations is deterministic and manually tuned by domain scientist. In contrast to domain-driven method, we show that aerosol prediction is achievable by completely data-driven approaches. These statistical methods consist of learning of nonlinear regression models to predict aerosol load using the satellite observations as inputs. Measurements from unevenly distributed ground-based sites over the world are used as proxy to ground-truth outputs. Although statistical methods achieve better accuracy than deterministic method this setup is appropriate when data are independently and identically distributed (IID). The IID assumption is often violated in remote sensing where data exhibit temporal, spatial, or spatio-temporal dependencies. In such cases, the traditional supervised learning approaches could result in a model with degraded accuracy.

Conditional random fields (CRF) are widely used for predicting output variables that have some internal structure. Most of the CRF research has been done on structured classification where the outputs are discrete. We propose a CRF model for continuous outputs that uses multiple unstructured predictors to form its features and at the same time exploits structure among outputs. By constraining the feature functions to quadratic functions of outputs, we show that the CRF model can be conveniently represented in a Gaussian canonical form. The appeal of proposed Gaussian Conditional Random Fields (GCRF) model is in its conceptual simplicity and computational efficiency of learning and inference through use of sparse matrix computations. Experimental results provide strong evidence that the GCRF achieves better accuracy than non-structured models. We improve the representational power of the GCRF model by 1) introducing the adaptive feature function that can learn nonlinear relationships between inputs and outputs and 2) allowing the weights of feature functions to be dependent on inputs. The GCRF is also readily applicable to other regression applications where there is a need for knowledge integration, data fusion, and exploitation of correlation among output variables.

ACKNOWLEDGMENTS

Type your acknowledgments text here, double spaced.

Type your dedication here.
You will need to adjust the number of blank
lines above depending on the length
of your dedication.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Motivation

Understanding processes of the Earth’s atmosphere, land, and ocean is of greatest importance to human society. In recent years many remote sensing instruments of various properties have been employed in an attempt to better characterize important geophysical phenomena. The collected remote sensed data are used for estimation and better understanding of global dynamics of geophysical parameters such as land cover type, humidity, clouds, aerosols, greenhouse gases, or ocean temperature. Achieving accurate estimations of such parameters is an important requirement for development of models able to predict global climate changes. The report by the Intergovernmental Panel on Climate Change (IPPC) [1] points out that the estimation of global composition, load, and variability of aerosols is one of the largest unknowns in climate change studies.

Aerosols are small airborne particles that reflect and absorb incoming solar radiation. Numerous ground and satellite based instruments are employed for monitoring aerosols whose observations could differ in spatial, temporal and spectral resolution, coverage, and quality. Aerosols are characterized by Aerosol Optical Depth (AOD), a quantity which represents total attenuation of radiation from the top of the atmosphere down to the surface. The nontrivial process of obtaining aerosol estimates from the instrument's raw observations is called the AOD prediction. The nature of the aerosol remote sensing creates significant methodological challenges for data mining. This research focuses on AOD prediction, although the developed methods are also readily applicable to a larger class of remote sensing problems.

Chapter 2 contains background information related to remote sensing of aerosols. To clarify challenges in AOD prediction, Section 2.1 and Section 2.2 describe the aerosol remote sensing instruments and the physics of AOD prediction. Section 2.3 introduces deterministic satellite-based AOD prediction that relies on domain knowledge. This section also points out potential flaws of deterministic predictor. Section 2.4 describes the data fusion from different sensors. Finally, Section 2.5 defines multiple evaluation measures used for assessment of prediction quality.

1.2. Main contributions

In contrast to domain-driven methods, in Chapter 3 we demonstrate that accurate AOD prediction is achievable by a completely data-driven approach [2-8]. This statistical method consists of training a nonlinear regression model using the satellite observations as inputs and ground based AOD as outputs. Section 3.2 shows how to construct AOD predictor that works well over a range of accuracy measures defined in Section 2.5. We propose an approach that builds an ensemble of neural networks, each trained with slightly different measure. The outputs of the ensemble are then used as inputs to a meta-level neural network that produces the actual AOD predictions. Results show that the proposed ensemble works well over all considered accuracy measures and at the same time is more accurate than the deterministic algorithm. In Section 3.3 we study a budget-cut scenario that requires a reduction in a number of ground-based sensors. We develop an iterative method that removes sensors one by one from locations where AOD can be predicted most accurately using training data from the remaining sites. Extensive experiments provide strong evidence that sensors selected using the proposed algorithm are more informative than the competing approaches that select sensors at random or that select sensors based on spatial diversity. In Section 3.4 we propose a method that automatically discovers homogeneous spatio-temporal partitions through the competition of regression models. We show that spatio-temporal partitioning followed by building specialized predictors results in increased prediction accuracy when compared to learning a single predictor on all the data and to learning specialized predictors on the data partitions used in deterministic algorithm.

Although statistical methods from Chapter 3 achieve better accuracy than deterministic method this setup is appropriate when examples are independently and identically distributed (IID). The IID assumption is often violated in remote sensed data, where examples exhibit sequential, temporal, spatial, or spatio-temporal dependencies. In such cases, the traditional supervised learning approaches could result in a model with degraded prediction accuracy. Conditional random fields (CRF) are widely used for predicting output variables that have some internal structure. Most of the CRF research has been done on structured classification where the outputs are discrete. In Chapter 4 we propose a CRF probabilistic model for structured regression that uses multiple unstructured predictors to form its features and exploits structure among outputs. By constraining the feature functions as quadratic functions of outputs, we show in Section

4.3 that the model can be conveniently represented in a Gaussian canonical form [9]. In Section 4.5 we improve the representational power of the resulting Gaussian CRF model by 1) introducing the adaptive feature function that can learn nonlinear relationships between inputs and outputs and 2) allowing the weights of feature functions to be dependent on inputs. The appeal of proposed model is in conceptual simplicity and computational efficiency of learning and inference through use of sparse matrix computations. Results presented in Section 4.4 and Section 4.5 provide strong evidence that the proposed GCRF model achieves better accuracy than non-structured models. The proposed method is also readily applicable to other regression applications where there is a need for knowledge integration, data fusion, and exploitation of correlation among output variables.

CHAPTER 2

REMOTE SENSING OF AEROSOLS

2.1. Remote sensing

Remote sensing is defined as the acquisition of information about an object without being in physical contact with it [10]. A typical source of remote sensing data is electromagnetic radiation which is emitted or reflected from the observed object. Information about our environment could be obtained by imaging the Sun's electromagnetic radiation that propagates through the atmosphere. Remote sensing of AOD relies on the concept that solar radiation is modified by aerosols as it travels through the atmosphere.

2.2. Ground and Satellite Based Instruments

There are two major types of instruments that collect aerosol related data: 1) satellite instruments, such as MODIS and MISR [11], POLDER [12], TOMS [13], SeWiFS [14], AVHRR/2 [15], and CALIPSO [16]; and 2) ground-based instruments, represented by AERONET [17]. The MODerate resolution Imaging Spectrometer (MODIS) is an instrument aboard NASA Terra and Aqua satellites that are part of the NASA's coordinated fleet of satellites often referred to as Earth Observation System's (EOS) [18]. Terra's orbit is descending (southward) the equator observing the location

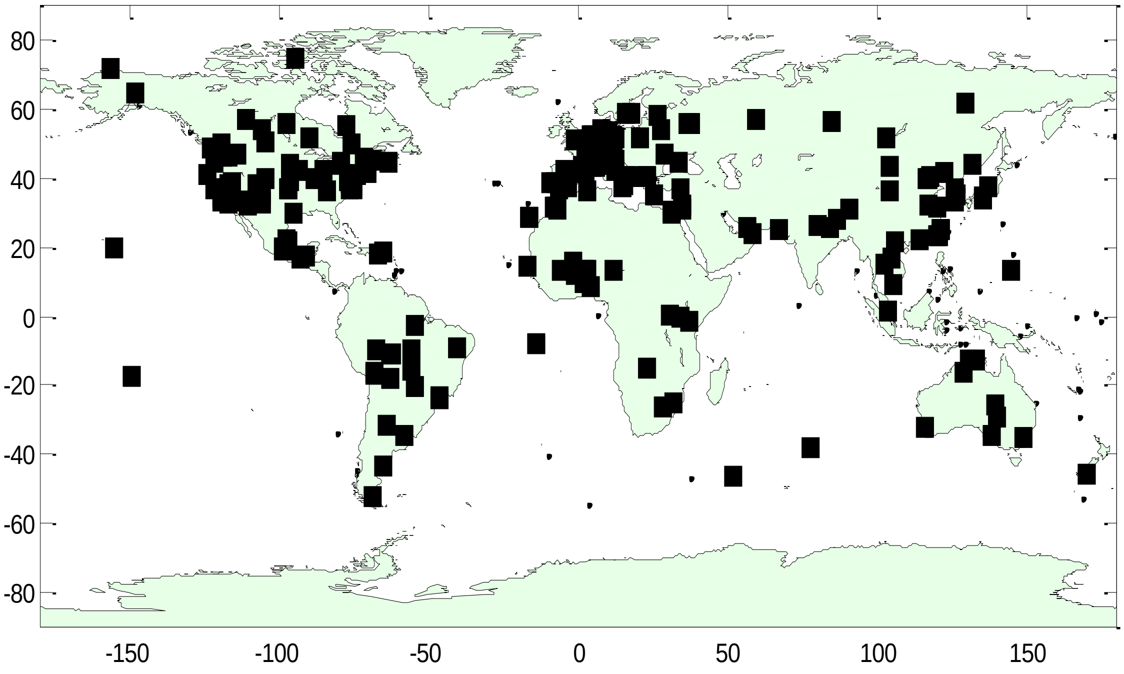


Figure 2.1. Global distribution of AERONET sites.

around 10:30 AM local time. Aqua’s orbit is ascending (northward) over the equator observing the location around 1:30 PM local time. MODIS measures radiation in the spectrum region from 0.41 μm to 14.235 μm [18]. The AErosol RObotic NETwork (AERONET) [17] is a federation of more than 200 operational sun/sky radiometers located at various places over the globe. Figure 2.1 shows that AERONET instruments are relatively dense over industrialized regions, while they are sparse elsewhere; oceans are severely underrepresented, but this is not a problem because aerosol predictions over oceans are of much higher quality than over land. Satellite instruments provide global coverage with high spatial resolution, have relatively low temporal resolution, and allow for moderately accurate AOD predictions. Ground-based instruments have limited spatial

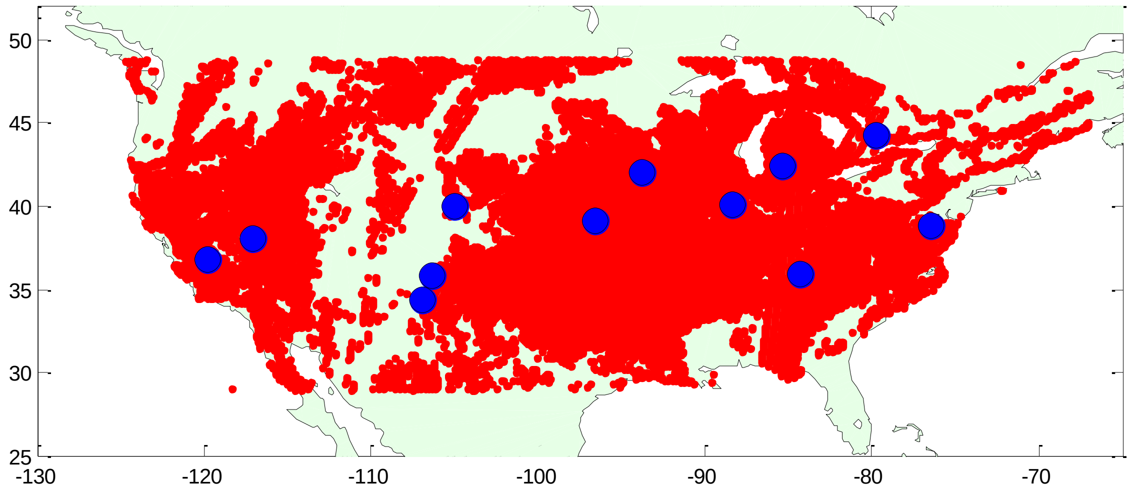


Figure 2.2. Coverage map of satellite and ground based instruments. Red dots – satellite instrument MODIS; blue circles – ground based instrument AERONET.

coverage, provide relatively large temporal resolution (few measurements per hour), but allow for highly accurate predictions. As a result, AERONET predictions are typically treated as ground-truth and used to validate quality of satellite-based predictions.

To illustrate the difference in coverage and resolution between ground-based and satellite-based instruments, Figure 2.2 shows AOD predictions of MODIS aboard Terra satellite and AERONET over the continental USA on September 9th, 2005. AOD predictions from 12 active AERONET sites are shown as blue circles. Terra orbit shifts in space each day and the cycle is repeated every 16 days. In Figure 2.2 Terra satellite passed over the US three times. Red dots correspond to AOD predictions from MODIS who observes the Earth in tracks (swaths) 2,330 km wide. With its large swath MODIS observes every location daily. Each red pixel is of size 10 × 10 km^2. However, there are

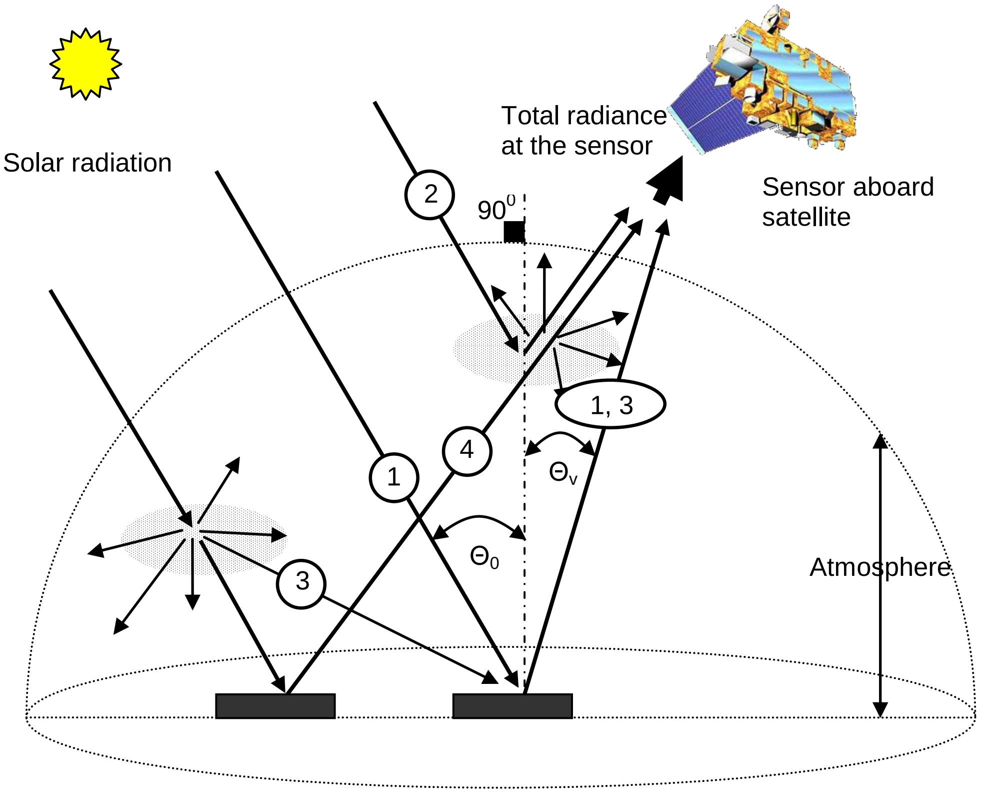


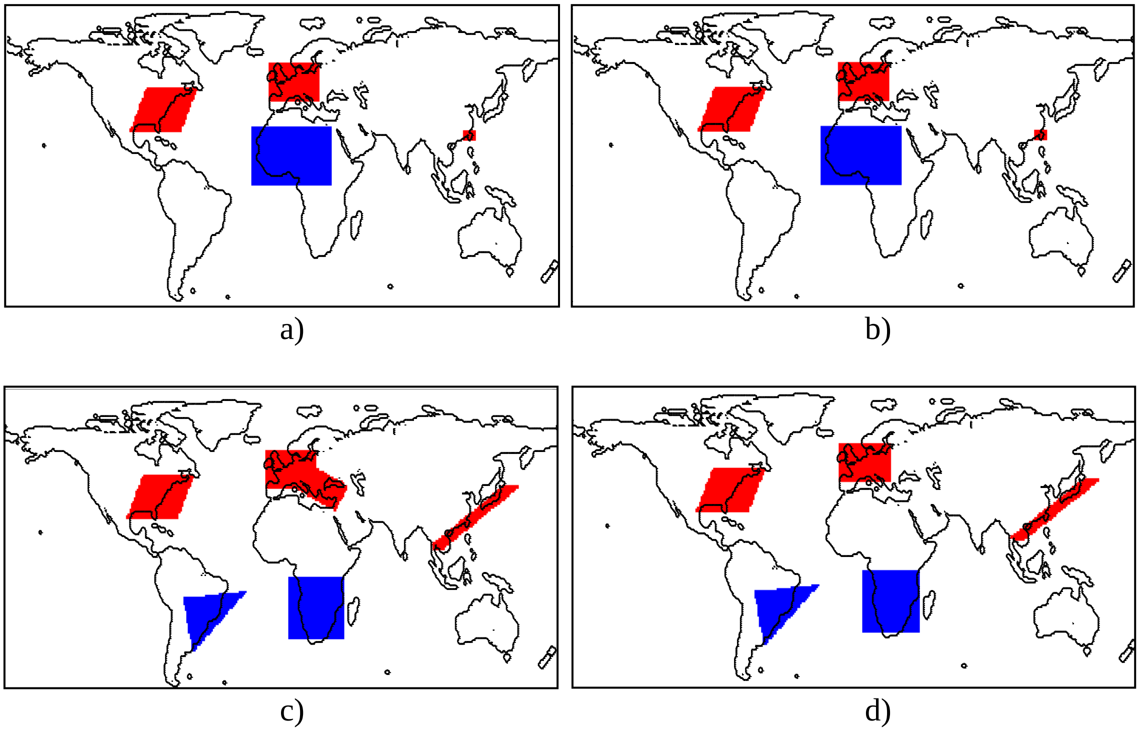
Figure 2.3. Physics of remote sensing of aerosols.

many holes within each red track with missing predictions caused by presence of clouds or unfavorable topography or land cover.

Figure 2.2 describes a typical satellite instrument which measures the incoming Sun’s electromagnetic radiation. As it can be seen, the solar radiation interacts with the atmosphere, the Earth’s surface, and again with the atmosphere along its path to the satellite instrument. To predict AOD using satellite observations, one needs to accurately determine the exact amount of radiance reflected from the atmosphere (Path 2 in Figure 2.3) as it conveys information directly related to AOD. In this case, radiance reflected from the surface (Paths 1, 3 and 4 in Figure 2.3) is considered as noise. The radiance observed by the instrument depends on properties of atmosphere as well as surface. Therefore, predicting the AOD is a highly non-linear and noisy problem. Accuracy of satellite based AOD prediction is one of the major limiting factors influencing climate change studies [11].

2.3. Domain-driven AOD Predictors

Most operational aerosol prediction algorithms are constructed as inverse operators of high-dimensional non-linear functions derived from forward-simulation models according to the domain knowledge of aerosol physical properties. Operational algorithms used to predict AOD from MODIS observations are based on matching the atmospheric component of the observed reflected radiation to the simulated values stored in lookup tables. The atmospheric component is obtained by removing the effect of the surface and is dominantly influenced by aerosol optical properties. Since aerosol properties and abundance change through time and over space, using a single model would not be able to fully describe the aerosol optical properties over a global scale. Recently developed operational AOD prediction algorithm, called C005, utilizes domain knowledge for spatio-temporal partitioning of the Earth. Figure 2.4 represents spatial partitioning of the globe over the four seasons [18]. For each spatio-temporal partition, C005 consults the look-up table constructed by forward simulations of the physical model of aerosol optical properties. For each component aerosol, the corresponding radiative properties are computed using wavelength, illumination, and view geometry information. The results are recorded in a look-up table. By using a modified linear mixing theory, the



a) b)

c) d)

Figure 2.4. Domain based global spatio-temporal partitioning of dominant “fine” aerosol properties. Blue – absorbing; red – non absorbing; white – neutral. a) December, January, February; b) March, April, May; c) June, July, August; d) September, October, November.

radiative properties of a mixture are calculated during the prediction process. These simulated data are then compared to actual observations for the appropriate scene type (land or ocean). According to a set of goodness-of-fit criteria based on the domain knowledge, the matched aerosol model in the look-up table is used for AOD computation. The simplified flow of C005 algorithm is summarized in Table 2.1 [18].

Table 2.1. Domain based C005 AOD prediction algorithm.

1. **Lookup tables:** one for “coarse” and three for “fine” aerosols (absorbing, non-absorbing and neutral). Tables contain the amount of multispectral radiances R_a reflected from the atmosphere that satellite would observe. R_a at 0.44, 0.66 and 2.1μm were obtained by forward-simulations on seven discrete values of AOD (0, 0.25, 0.5, 1, 2, 3, 5), nine solar zenith angles, sixteen sensor zenith and relative azimuth angles.

2. **Inputs:** multispectral satellite observations R_{sat} , view angles, elevation, spatio-temporal coordinates.

3. **AOD prediction**

3.1. Determine fine aerosol model based on spatio-temporal coordinates.

3.2. Interpolate lookup tables to observed geometry (eliminates dependence on geometry, R_a(AOD) becomes function of AOD)

3.3. For each discrete AOD find the amount of surface reflection at 2.1μm (S and T are parameters obtained during forward simulation)

$$R_{s2.1}(AOD) = (R_{a2.1}(AOD) - R_{sat2.1}) / (S_{2.1} \cdot (R_{a2.1}(AOD) - R_{sat2.1}) - T_{2.1})$$

3.4. Determine surface reflectance at 0.44 and 0.660μm (coefficients C depend on observation)

$$R_{s0.66}(AOD) = C_1 \cdot R_{s2.1}(AOD) + C_2$$

$$R_{s0.44}(AOD) = C_3 \cdot R_{s0.66}(AOD) + C_4$$

3.5. For each discrete AOD an both models (fine and coarse) add surface reflectance to reflectance R_a. at 0.44, 0.66 and 2.1μm

$$R(AOD) = R_a(AOD) + T \cdot R_s(AOD) / (1-S \cdot R_s(AOD))$$

3.6. For each discrete fine/(fine+coarse) ratio η = -0.1, 0, 0.1, …1.1

$$R_{total}(AOD) = \eta \cdot R_{fine}(AOD) + (1 - \eta) \cdot R_{coarse}(AOD)$$

3.7. Interpolate R_{sat} at 0.44 μm and find AOD (R - linear interpolation between reflectance; AOD - logarithmic interpolation between optical depths). Interpolations of R_{sat} for each η give thirteen AOD(η).

3.8. Interpolate AOD(η) and find $R_{est0.66}(\eta)$ at 0. 66 μm. Difference between $R_{est0.66}(\eta)$ and $R_{sat0.66}$ represents fitting error.

4. **OUTPUT:** AOD that corresponds to minimal fitting error.

Potential drawbacks of deterministic prediction methods include 1) high computational cost due to inversion of nonlinear forward models; 2) slow development due to manual construction of the postulated physical models; 3) difficulties in describing complex radiance-aerosol relationships in all realistic scenarios; and 4) inaccuracies that are due both to the instrument limitations and imperfections in the prediction algorithms.

2.4. Data Fusion

Given a data set that consists of satellite observations and AERONET AOD measurements, a regression model can be trained to use satellite observations as inputs and predict the labels which are AERONET AODs. For that reason, satellite observations need to be collocated and merged with AERONET measurements. In this study we consider data from MODIS, an instrument aboard NASA's Terra and Aqua satellites.

MODIS has high spatial resolution (pixel is as small as 250 × 250 m^2) and achieves global coverage daily. On the other hand, AERONET sites, situated at fixed geographical locations, acquire data at intervals of 15 min on average. This gives rise to the need for both spatial and temporal data fusion (Figure 2.5). The fusion method involves aggregating MODIS pixels into blocks of size 50 × 50 km^2 and spatially collocating them with an AERONET site. The MODIS observations are said to be temporally collocated with the corresponding AERONET AOD predictions if there is a valid AERONET AOD prediction within 30 minutes of the satellite overpass. The data collocated in this way can be obtained from the official MODIS website of NASA [19].

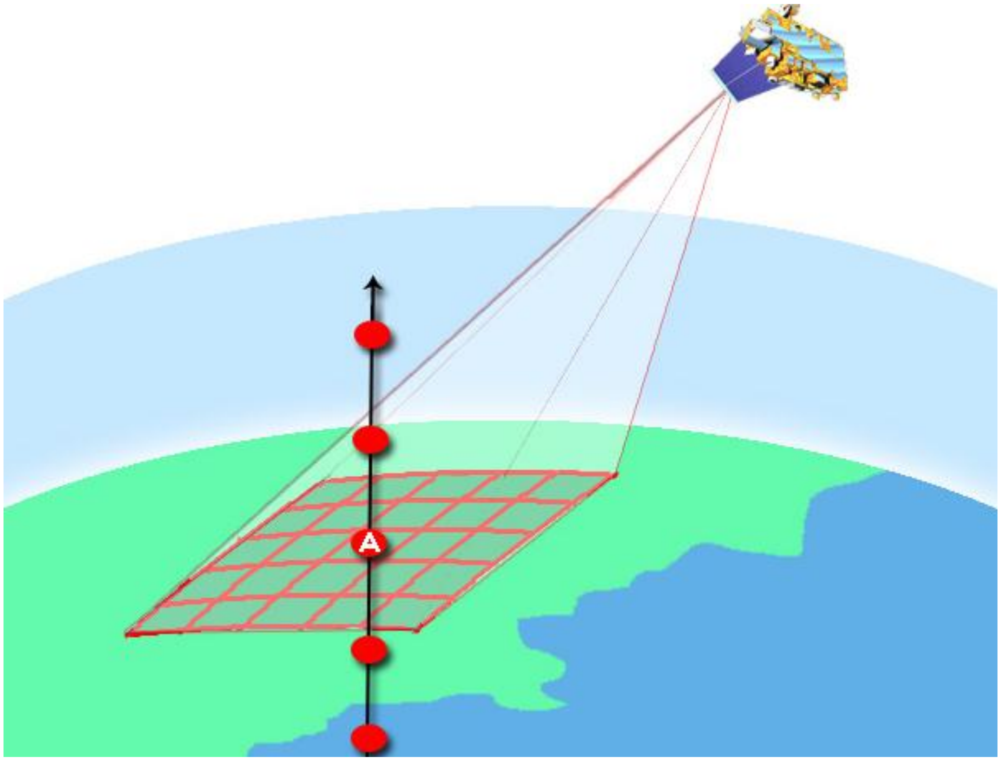


Figure 2.5. Spatio-temporal collocation of MODIS and AERONET data. A is an AERONET site with AOD predicted within a short time before and after the satellite overpass (circle dots). The square regions are MODIS observations in a proximity of site A at the satellite overpass time.

We extracted satellite-based observations, by consulting inputs to the MODIS operational prediction algorithm. The radiances at four wavelengths were taken from the MODIS range 0.44–2.1 μm, as these are sufficient to describe aerosol properties. We used average and standard deviation of radiances within 50 × 50 km^2 as inputs. We also collected solar and sensor angles and surface elevation.

Deterministic algorithm predicts AOD at 0.55 μm. Since AERONET instruments do not provide AOD value at that wavelength, we performed linear interpolation in the log scale of AERONET AOD at 0.44 μm and 0.66 μm to get AOD at 0.550 μm [18].

2.5. Evaluation Measures

To demonstrate the need for multiple evaluation measures let us analyze the accuracy of currently operational NASA's MODIS prediction algorithm C005 [18]. A scatter plot of C005 AOD prediction vs. ground based AOD prediction in period of three years from 2005 to 2007 over whole globe is presented in Figure 2.6. Solid line represents the perfect prediction, while dashed lines represent boundaries of an area within which predictions are acceptable to domain scientists. Large absolute errors are more tolerable when predicting large AOD than when predicting small AOD. Therefore, a fraction of data points inside the bounded area (*FRAC*) is a suitable accuracy measure. Mean squared error (*MSE*) measure is also used for AOD prediction, but it is not as informative because 1) prediction error increases with AOD, 2) distribution of AOD is skewed towards small values, and 3) there are many outliers. In addition to *FRAC* and *MSE*, domain scientists are also interested in the relative squared error that considers larger absolute errors more tolerable when predicting large AOD than when predicting small AOD.

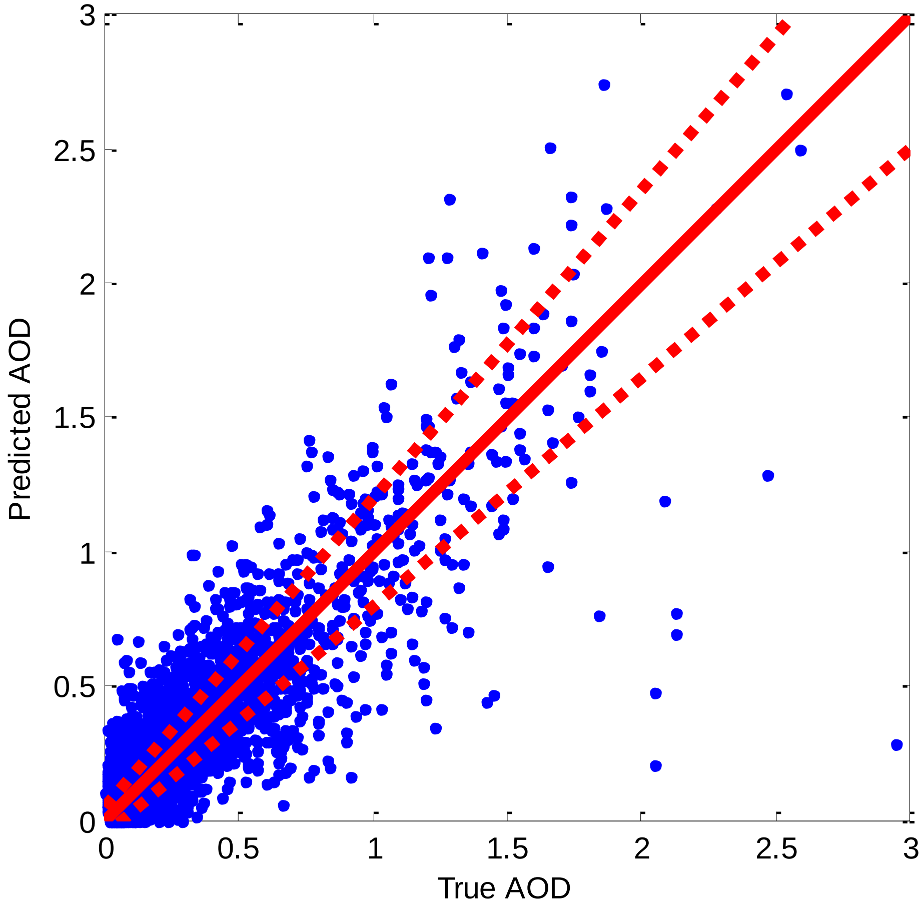


Figure 2.6. Scatter plot of predicted vs. true AOD. Solid line – ideal predictions. Dashed lines - boundaries of acceptable predictions.

Given vector **t** = [t_1, t_2, …, t_N]^T of N true output values (i.e. true AOD values) and vector **y** = [y_1, y_2, …, y_N]^T of the corresponding predictions, the standard mean squared error (*MSE*) is defined as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - t_i)^2 \; . \tag{2.1}$$

A closely related to *MSE* is root mean square error (*RMSE*) defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - t_i)^2} \tag{2.2}$$

RMSE is in the same units as the data while *MSE* is in square units of data. *RMSE* may give better understanding of errors than *MSE*.

The coefficient of determination (R^2) is defined as

$$R^2 = 1 - \left(\sum_{i=1}^{N}(y_i - t_i)^2\right) / \left(\sum_{i=1}^{N}(\bar{t} - t_i)^2\right), \tag{2.3}$$

where $\bar{t}$ represents the mean value of vector **t**. R^2 value describes fraction of the variance that the predictor successfully explains. The highest R^2 is 1, while R^2 of the model that simply predicts the output variable mean is 0. R^2 of some poor predictors can even be negative.

Another related measure, which is insensitive to the correctable bias, is correlation coefficient (*CORR*)

$$CORR = \left(\sum_{i=1}^{N}(y_i - \bar{y})(t_i - \bar{t})\right) / \left(\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{N}(t_i - \bar{t})^2}\right), \tag{2.4}$$

where $\bar{y}$ represents the mean of vector **y**.

We also consider several domain-specific measures. Geoscientists showed both theoretically and empirically that, taking into consideration the physical constraints of remote sensing of aerosols, the desired absolute AOD prediction error should be between 0.05 and 0.1 for small AOD and that it could increase to 15-20% × AOD for large AOD [18]. Thus, the AOD prediction is considered successful if the absolute error is

$$|y_i - t_i| \le 0.05 + 0.15 t_i \,. \tag{2.5}$$

We may now define the fraction of successful predictions (*FRAC*) as

Table 2.2. C005 Vs AERONET prediction accuracy.

Model	R^2	*CORR*	R_r^2	*FRAC*
C005	0.70	0.87	0.28	64.8%

$$FRAC = \frac{I}{N} \times 100\%, \tag{2.6}$$

where *I* is the number of predictions that satisfy relation (2.5).

Domain specific relative squared error (*RSE*) is defined as

$$RSE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - t_i}{0.05 + 0.15t_i}\right)^2 . \tag{2.7}$$

RSE values less than 1 indicate that AOD predictions are satisfactory. The closer *RSE* is to 0 the better performance of a predictor is. A related measure of accuracy is relative coefficient of determination (R_r^2) defined as

$$R_r^2 = 1 - \left(\sum_{i=1}^{N}\left(\frac{y_i - t_i}{0.05 + 0.15t_i}\right)^2\right) / \left(\sum_{i=1}^{N}\left(\frac{\bar{t}_r - t_i}{0.05 + 0.15t_i}\right)^2\right), \tag{2.8}$$

where $\bar{t}_r = \sum w_i t_i / \sum w_i$, $w_i = (0.05 + 0.15t_i)^{-2}$, represents the weighted mean of vector **t**. R_r^2 is derived according to general definition of coefficient of determination [20]. The highest R_r^2 is 1, while R_r^2 of the model that predicts the output weighted mean is 0.

Let us analyze the accuracy of the operational AOD prediction algorithm called C005 whose scatter plot is presented in Figure 2.6. The values of four different accuracies are shown in Table 2.2. C005 has an excellent performance based on *CORR*. However, R^2 tells us that there is a significant portion of variance which C005 was unable to explain.

Furthermore, domain specific R_r^2 accuracy is small which indicates lower than desired accuracy. Finally, *FRAC* measure shows that more than 35% of predictions are of insufficient accuracy.

CHAPTER 3

UNSTRUCTURED DATA-DRIVEN PREDICTORS

3.1. Background

A data mining approach for regression is based on learning relationships between inputs and the output variable. In the standard regression setting we are given a data set with N training examples, $D = \{(\mathbf{x}_i, y_i), i = 1 \ldots N\}$, where $\mathbf{x}_i \in \mathbf{X} \subset R^M$ is an M dimensional vector of inputs and $y \in R$ is a real-valued output variable. The objective of regression is to learn a non-linear mapping f from training data D that predicts the output variable y as accurately as possible given an input vector $\mathbf{x}$. Typically the following data-generating model is assumed

$$y = f(\mathbf{x}) + \varepsilon, \varepsilon \sim \mathrm{N}(0, \sigma^2), \tag{3.1}$$

where ε is Gaussian additive noise with constant variance σ^2. In AOD prediction application inputs $\mathbf{x}$ are multivariate observations collected from a satellite instrument spatio-temporally collocated with the corresponding ground-based AERONET AOD values y.

Neural networks are often a regression model of choice in data-driven prediction of atmospheric properties [21], [22]. Neural networks have been trained to predict AERONET AOD over continental US [23] and whole globe [3], [5] using inputs derived from satellite data. Comparing to the domain-based AOD predictions, neural network AOD predictions were significantly more accurate.

In following sections we present methods for solving specific problems in AOD prediction application.

3.2. Prediction across Multiple Accuracy Measures

3.2.1. Introduction

Ideally, one would like to have a predictor that provides good accuracy with respect to multiple accuracy measures. The complication is that predictors which perform well on one measure may not perform well on other measures. An important challenge is to train a predictor where the objective is not optimal performance on a single measure, but robust performance across several measures.

To construct a model that is accurate with respect to *FRAC*, *MSE* and relative squared error measures defined in Section 2.5, we propose to train an ensemble of neural networks, each with a different relative error measure, and to combine their predictions. We explored different methods for combining ensemble predictions: 1) average of ensemble outputs, 2) a neural network which takes ensemble outputs and provides final AOD prediction and 3) weighted average of ensemble outputs according to a gating neural network that approximates probability of large AOD. Proposed predictors were compared to neural network models optimized for a single accuracy measure as well as to the operational MODIS AOD prediction algorithm C005.

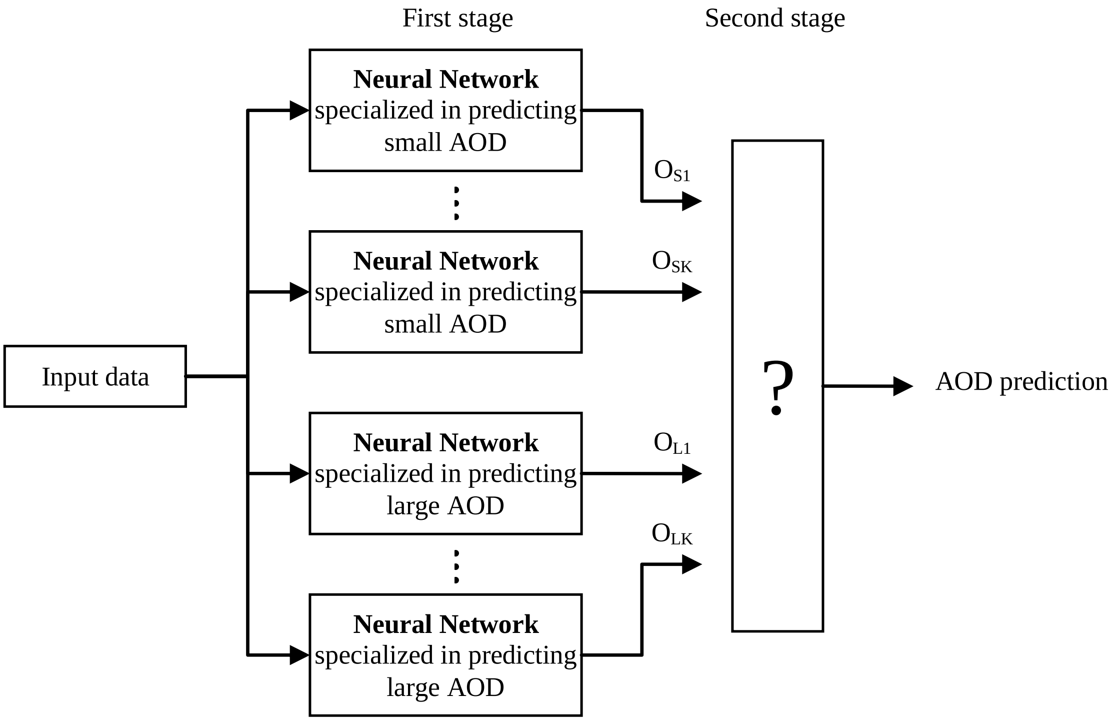

First stage
Second stage
Neural Network specialized in predicting small AOD
Neural Network specialized in predicting small AOD
Neural Network specialized in predicting large AOD
Neural Network specialized in predicting large AOD
Input data
O_{S1}
O_{SK}
O_{L1}
O_{LK}
?
AOD prediction

Figure 3.1. Architecture of the proposed two-stage ensemble for AOD prediction.

3.2.2. Adaptive Cost Function

Neural networks are typically trained by minimizing *MSE*. This kind of cost function treats all errors equally regardless of the output value. Earth scientists prefer small *relative errors* rather than small *absolute errors*. Hence, *MSE* function is not the most appropriate cost function for this application. As a more general choice we introduce a function defined as the *relative error* ($REL_{a,b}$) between predicted and ground truth AOD

$$REL_{a,b} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - t_i}{a + bt_i}\right)^2 , \tag{3.2}$$

where a and b are positive user defined parameters. Here, the level of penalization of prediction errors can be controlled by tuning parameters a and b. Note that $REL_{1,0}$ is equivalent to *MSE* while $REL_{0.05,0.15}$ is equal to *RSE*.

We employ $REL_{a,b}$ as a cost function for training neural networks. When a is small, bt_i is dominant and so the emphasis is on reducing the error of predicting small AOD. On the other hand, when a is large, errors for small and large AOD have similar importance. Sensitivity of a neural network optimization to t_i also depends on parameter b – for large b the network becomes more sensitive to the errors made when predicting small AOD.

3.2.3. Ensembles with Adaptive Cost Functions

Minimization of $REL_{a,b}$ cost function with a = 0.05 and b = 0.15 directly leads to the optimization of domain specific measures mentioned in the Section 2.5 - maximization of *FRAC* and minimization of *RSE*. However, a neural network trained in this way would have decreased *MSE* accuracy. We are interested in construction of a model that is accurate with respect to all accuracy measures.

$REL_{0.05,0.15}$-optimized neural networks will be more accurate when AOD is small, while *MSE*-optimized networks will work better when AOD is large. However, the problem arises because it is not known in advance whether the AOD value is small or large. If we used the model which has the ability to decide whether AOD value is large or small, the accurate prediction of medium level AOD values would still be the problem. More specifically, such a model would either overestimate or underestimate AOD depending on whether it was "classified" as large or small, respectively. To solve this problem we propose a two-stage approach:

1. Constructing an ensemble of *2K* neural networks among which *K* are specialized in predicting small AOD while the remaining *K* are specialized in predicting large AOD. This is achieved by using different values of parameters *a* and *b*. Since the distribution of AOD is skewed to the small AOD, by design all component networks are trained to penalize errors at small AOD. However, intensity of this penalization varies per component network.
2. Combining outputs of the component networks to obtain an integrated AOD prediction.

The architecture of the proposed system is illustrated in Figure 3.1. All first-stage component networks are trained using the same dataset. O_{S1}, O_{S2},…, O_{SK} correspond to networks specialized for smaller AOD, while O_{L1}, O_{L2},…, O_{LK} correspond to networks specialized for larger AOD. Those outputs are integrated at the second stage using one of the following methods.

3.2.3.1. Integration by Averaging

Here, the final AOD prediction is obtained as a simple average of O_S and O_L neural networks. We will refer to this approach as AVERAGE.

3.2.3.2. Integration by a Meta Neural Network

Here, predictions of O_S and O_L neural networks are used to train a second-stage meta neural network. The meta neural network is optimized to minimize $REL_{0.05,0.15}$. We will refer to this two stage structure as META.

3.2.3.3. Integration by a Gating Neural Network

In the GATING approach, the first-stage networks are linearly combined according to the weights assigned by a gating network. A gating neural network is built as a binary classifier that predicts whether AOD is small or large. If the gating network O_G predicts large AOD, (i.e. O_G is close to 1) larger weights are assigned to the O_L neural networks specialized for predicting large AOD. On the other hand, O_G close to 0 gives larger weights to O_S networks. Finally, O_G near 0.5 means that large and small AOD are equally likely, and weights of O_S and O_L are equal. To avoid bias sum of all weights is set to 1. The final AOD prediction is then computed as

$$y = \sum_{i=1}^{K} \left(\frac{o_G}{K} o_{Li} + \frac{1 - o_G}{K} o_{Si} \right), \tag{3.3}$$

where O_{Li} and O_{Si} are outputs of first-stage networks.

To train the gating network we assign large and small labels to AOD values. Domain knowledge suggests that AOD values that are less than 0.15 should be considered small [18]. To prevent problems related with training on imbalanced data, instead of using 0.15, we use median AOD value as the *threshold*. AOD values larger than threshold are considered as large while the remaining ones are considered as small.

3.2.4. Experimental Results and Discussion

We used collocated data in years 2005-2007. Spatial-temporal cross-validation was applied in all experiments. First, we split AERONET locations into 5 subsets S_i, i = 1…5, and create five data sets D_i, $i = 1…5$, each with data points from one of the AERONET subsets. Then, we split each D_i into D_i^{56} containing data from 2005 and 2006 and D_i^7 containing data from 2007. We reserved one of D_i^{56} datasets for testing and merged data from the remaining 4 datasets D_j^{56}, $j \neq i$, for training. The trained predictor was tested on 3 datasets

(TEST1) D_i^{56} – data collected during 2005 and 2006 at the locations unobserved during training;

(TEST2) $\{D_j^7, j \neq i\}$ – data collected during 2007 at the locations observed during 2005 and 2006;

(TEST3) D_i^7 – data collected during 2007 at the locations unobserved during training.

The procedure was repeated five times, for values $j = 1…5$, and the average accuracy over the 5 runs was reported. It is expected that TEST3 is the most challenging for prediction.

Table 3.1. Satellite-Based Vs. AERONET AOD prediction accuracy on TEST3 (unobserved locations, unobserved time).

Model	***R^2***	***CORR***	***R_r^2***	***FRAC***
C005	0.65	0.86	0.13	63.9%
SingleMSE	0.74	0.87	0.40	66.2%
SingleREL	0.68	0.85	0.55	69.3%
EnsembleMSE	0.76	0.88	0.45	68.9%
EnsembleREL	0.67	0.86	0.56	70.6%
DIFFREG	0.65	0.84	0.07	66.8%
AVERAGE	0.75	0.88	0.54	70.5%
META	0.75	0.87	0.50	69.5%
GATING	**0.76**	**0.88**	**0.53**	**70.9%**

Table 3.2. Gating Vs. AERONET AOD accuracy on different test sets: TEST1 – unobserved locations, observed time; TEST2 – observed locations, unobserved time; TEST3 – unobserved locations, unobserved time.

Test set	***R^2***	***CORR***	***R_r^2***	***FRAC***
TEST1	0.76	0.88	0.55	71.4%
TEST2	0.79	0.89	0.61	73.5%
TEST3	0.76	0.88	0.53	70.9%

3.2.4.1. Benchmark Methods

Deterministic algorithm C005. The primary benchmark for comparison with our predictors was the most recent version of the MODIS operational algorithm C005.

Single neural networks. As a baseline data mining algorithm we used single neural networks trained to predict AERONET AOD from MODIS observations. Two different single neural network models were evaluated. The first network is trained by minimizing a standard *MSE* cost function (SingleMSE), while the second network minimized our novel $REL_{a,b}$ measure (SingleREL). Parameters *a* and *b* were fixed to $a = 0.05$ and $b = 0.15$.

Simple ensembles of neural networks. We also compared the proposed methods to two ensemble algorithms. Each ensemble consisted of 10 neural networks. Outputs of the 10 neural networks were used as inputs to the second-level neural network. In EnsembleMSE approach, neural networks were trained using *MSE* cost function. In EnsembleREL the cost function for all networks was $REL_{0.05,0.15}$.

Ensemble of networks specialized for low and high AOD. In DIFFREG approach $K = 5$ neural networks were trained using a portion of training data with small AOD, while another *K* networks were trained using data with large AOD. To permit smooth transition in input space, overlapping between two training datasets was allowed. Small AOD was defined as AOD < $threshold+\varepsilon$ while large AOD was defined as AOD > $threshold-\varepsilon$, where ε was 0.05. All networks were trained to minimize *MSE* and the two sets of networks were integrated using the gating neural network described in previous section.

3.2.4.2. Results on TEST3

Ensemble neural networks having 13 inputs, 10 neurons in a single hidden layer, and one in the output layer were used in all experiments. Sigmoid activation function was used in hidden neurons while the linear activation function was used for the output neuron.

Average accuracies of the proposed AVERAGE, META, and GATING predictors and of six benchmark algorithms using R^2, *CORR*, R_r^2, and *FRAC* measures are reported in Table 3.1. These results were obtained on the most challenging TEST3 data. We note that averaging of coefficient of determination measure over 5 different cross validation experiments might be misleading since those measures depend on standard deviation of a particular test set. However, variation of R^2 in five sets used in these experiments was negligible and so we decided to also report average R^2. Let us look at the results in more detail.

Operational prediction algorithm C005. C005 accuracies are shown in the first row of Table 3.1. As discussed in Section 2.5, C005 has an excellent performance based on *CORR*, but R^2 accuracy reveals that it was not able to explain a large portion of variance. Also, domain specific R_r^2 and *FRAC* measures indicate that C005 based predictions are of insufficient accuracy.

Single neural networks. SingleMSE and SingleREL accuracies are in rows 2 and 3 in Table 3.1. Both single neural networks were more accurate in predicting AOD than the operational C005 algorithm on all accuracy measures. However, their performance was quite different over individual accuracy measures; SingleMSE was more accurate with respect to R^2, and *CORR*, while SingleREL was a better choice with respect to R_r^2 and *FRAC* measures.

Simple ensembles of neural networks. EnsembleMSE and EnsembleREL accuracies are listed in rows 4 and 5 of Table 3.1. Both predictors outperformed C005 in all accuracy measures. Also, they were more accurate than single neural networks. However, neither ensemble achieved consistently high accuracy on all four measures; EnsembleMSE achieved better accuracy than EnsembleREL with respect to R^2 and *CORR*, while EnsembleREL was better according to R_r^2 and *FRAC* measures.

Ensemble of specialized neural networks. DIFFREG accuracies are listed in row 6 of Table 3.1. This benchmark method was quite unsuccessful, with accuracies below SimpleMSE and just slightly better than C005.

Ensembles with adaptive cost neural networks. In AVEARGE, META, and GATING predictors, five neural networks of the ensemble were specialized for prediction of small AOD. This was achieved by using $REL_{a,b}$ cost function with a = 0.05 and b changing from b = 0.03 to b = 0.15 in the steps of 0.03. Another five neural networks in the ensemble were specialized for prediction of large AOD by using a = 1 and b changing from b = 0.03 to b = 0.15 in the steps of 0.03.

Results for AVERAGE, META and GATING adaptive cost ensembles are presented in last 3 rows of Table 3.1. All 3 predictors were robust across all accuracy measures. GATING ensemble with a second-level gating neural network was slightly more accurate than the alternatives. On standard measures (R^2 and *CORR*) GATING was as good as the most successful benchmark method on these measures (EnsembleMSE) and it had similar accuracy with the best benchmark method EnsembleREL on domain specific measures (*R_r^2* and *FRAC*). This result shows that it is possible to simultaneously achieve high accuracy on disparate measures using a two-level ensemble neural network architecture.

3.2.4.3. Results on TEST1 and TEST2

Accuracies on TEST1 and TEST2 experiments were fully consistent with TEST3 results reported at Table 3.1. Our experiments showed that if a certain method was more accurate than an alternative method on TEST3, it was most often also more accurate on TEST1 and TEST2. In particular, in all three tests GATING method was the most accurate over all four measures. The results of GATING method over three types of tests are compared in Table 3.2.

Experiments over three types of tests showed that all methods were most accurate when tested on data at unobserved time but over previously seen locations (TEST2). Predicting AOD at unseen locations during the same two years (TEST1) was a more challenging objective but not as difficult as predicting AOD at unseen locations and in an unseen year (TEST3). These results suggest that in our data temporal correlation was stronger than spatial correlation, but also that both kinds of correlation could be exploited to improve quality of AOD predictions.

3.3. Reduction of Ground Based Sensor Sites

3.3.1. Introduction

Ground based stations are often located without a rigorous statistical design. Decisions are typically based on practical circumstances (e.g. overrepresentation in urban regions and industrialized nations) and according to domain experts' assumptions about the importance of specific sites. Furthermore, the total number of sensor sites depends on financial constraints. Costs related to equipment, location, and the availability of trained staff often dictate the number of sites and their global distribution. As shown in Figure 2.1, AERONET sites are not uniformly distributed over the globe. The highest density is within the U.S. and Europe. On the other hand, continental Asia, Africa, and Australia are poorly covered. Given these circumstances, the aims are to evaluate performance of the current design of AERONET sensor network and to apply data mining techniques to assist in future modifications of the sensor network.

A specific scenario considered assumes that there is a pending budget cut for maintenance of the existing AERONET sites. The objective is to shut down a fraction of the AERONET sites while making sure that the utility of the remaining sites is as high as possible. We make a simplifying assumption that operational costs for each AERONET site around the globe are equal.

Common to most selection techniques originating from the spatial statistics is a tendency to overlook the time dimension of data collected by the sensor network. For the problem of selecting a subset of data collection sites, we consider series of observations and propose to optimize AERONET sensor selection based on the concept of prediction accuracy. The intuition behind our proposal is straightforward. Each AERONET site provides a time series which can be used in training a regression model to predict future AOD. Sites that can be removed are those whose observations are best predicted by the model built on data from the remaining sites. The performance of the proposed approach is compared with the random site selection and with the classical selection principle of selecting spatially dispersed sites.

3.3.2. Determining an Appropriate Output Transformation

The assumption of constant variance is a basic requirement in constructing a traditional model defined in Section 3.1. In many cases there is strong reason to suspect that the error variance is not constant. Variance stabilizing transformations of output variable are often useful in these cases [24]. The strength of transformation depends on its curvature. Square root and logarithmic transformations are popular in practice. In square root transformation, a regression model that predicts $Z = \sqrt{Y}$ is trained and the prediction is provided as $\hat{Y} = \hat{Z}^2$ while in logarithmic transformation $Z = \log(Y)$ and $\hat{Y} = \exp(\hat{Z})$. Square transformation is considered as a relatively mild [24] comparing to the logarithmic and is often applied when variance of residuals increases linearly with predicted variable. In the experimental section we compared both of them with the standard approach that does not transform the output variable.

3.3.3. Selection of Informative Sensor Sites

Let us assume that a mission objective is to close down a fraction (33% or 66% in our experiments) of AERONET sites as to reduce ground-based data collection costs. Given such a budget cut situation, question of our interest is how to select *L* (<*K*) of the currently available *K* AERONET sites such that this subset captures as much information as possible compared to the entire set *S* of AERONET sites. The goodness criterion for a selection is accuracy of a regression model built on labeled data from the retained sites.

Intuitively, it appears that the selection of sites that are spatially dispersed would be a better choice than a random elimination. Such a spatial selection might be aided by domain experts - they would prefer to keep representative sites around the globe that cover a variety of meteorological and environmental conditions. However, regardless of the experts' effort, spatial representatives selected this way may not be optimal with respect to the quality of the resulting regression model *f*.

The sites selected by a domain expert are likely to be spatially diverse. To approximate the decision-making process of domain experts, for benchmarking purposes we use the spatial selection strategy based on spatial distance among sites. In the first step two sites that are closest to each other are determined. One of them whose removal better preserves global coverage is excluded from the set *S*. To decide which one is going to be removed, we are consulting the nearest neighbors of those two sites. The site which has the closer second nearest neighbor is removed. This procedure is iteratively repeated until the desired number of *L* sites is reached.

Our proposed strategy for selection of *L* sites out of *K* is accuracy-based. At the first step, the regression model *f* is trained on the data from the entire set of AERONET sites. At successive steps, every location is taken out and a model is built on data from the remaining sites. By $\hat{Y}$ we denote AOD prediction obtained by a model trained on whole dataset and by $\hat{Y}^{(i)}$ AOD prediction obtained by a model trained on $S \backslash S_i$ sites that exclude examples from site S_i. The intuition is that if AODs from site S_i can be estimated with a model which has not seen that site, then site S_i can be considered as redundant and therefore can be removed. To quantitatively define redundancy, we measure the difference in AOD prediction accuracy between the model trained on the whole dataset and model trained on a dataset without examples from site S_i. The difference in prediction accuracy is measured at data from site S_i as a sum of squared differences in predicted AODs computed over all points from site S_i

$$SSE_i = \sum_t (\hat{y}_i - \hat{y}_t^{(i)})^2 \ . \tag{3.4}$$

A site that is removed is the one with the smallest *SSE* as its AOD is the easiest to estimate given data from the remaining sites. Once a site is removed the proposed procedure is repeated. It continues by comparing the reduced models to the model built on the entire data, where data from the most recently excluded site are used for calculating *SSE* based loss.

3.3.4. Experimental Results and Discussion

Collocated data are distributed over entire globe at 217 AERONET sites during years 2005 and 2006. To assess efficiency of the proposed methods, we performed training on 2005 data and used 2006 data for testing. However, during that time period measurements from AERONET sites were not uniformly distributed, neither temporally or spatially. There were many more points from June to August than from January to May. Also, at some cloudy locations it was not possible to measure AOD and those locations contained a small number of data points. To maintain uniformity of the training dataset, in each training session we randomly selected 30 sites in year 2005 as the initial set *S*. Only 70 randomly chosen observations from each of those AERONET site were retained and remaining ones were removed. Finally, the training set consisted of 2,100 data points distributed over 30 AERONET sites each containing 70 collocated observations. As the test set, we randomly sampled 50 points from each site in 2006. Sites with less than 50 valid observations were excluded. The constructed test set contained 3,500 data points distributed over 70 AERONET sites each having 50 collocated observations. It is worth mentioning that among 70 test sites, 30 were the same as in the training set, while 40 sites were not seen during training. To evaluate the proposed approach, we report R^2 accuracy on the test set.

3.3.4.1. Determining an Appropriate AOD Transformation

To validate the assumption that error variance is not constant, we performed the following experiment. Thirty sites in 2005 were chosen randomly. Three regression models, one with data preprocessed by the square root transformation (NNSQ), one with data preprocessed by the log transformation (NNLG) and the other without the transformation (NN), were trained on the selected dataset and compared on the test set. As a regression model we used a neural network with ten hidden neurons trained to optimize standard *MSE* function.

Table 3.3. R^2 statistics on 2006 data for neural network models without (NN) or with log (NNLG) or square root (NNSQ) transformed output each built on ten different sets of 30 randomly selected sites using 2005 data.

Model	$\boldsymbol{R^2}$				
	Mean	**Std**	**Median**	**Min**	**Max**
NN	0.659	0.086	0.671	0.459	0.742
NNLG	0.664	0.091	0.703	0.444	0.721
NNSQ	0.746	0.042	0.754	0.644	0.789

This procedure was repeated ten times for different sets of 30 randomly selected sites. We report R^2 accuracy achieved on the fixed test set covering the 70 sites during 2006. To estimate sensitivity of constructed models to distribution of the initial 30 sites, we report mean, standard deviation, median and minimum and maximum of R^2 in those ten iterations. The results are presented in Table 3.3. These results provide strong evidence that the neural networks trained to predict AOD squared root (NNSQ) are more accurate than those trained to predict raw AOD (NN) or log-AOD (NNLG). Additionally, the presented results reveal that prediction accuracy is sensitive to the choice of the initial set of *S* sites. Although each time the selected 30 sites were globally distributed covering all parts of the world, in some cases accuracy dropped significantly. A possible explanation could be that some of those sites have noisy data that negatively influence model performance.

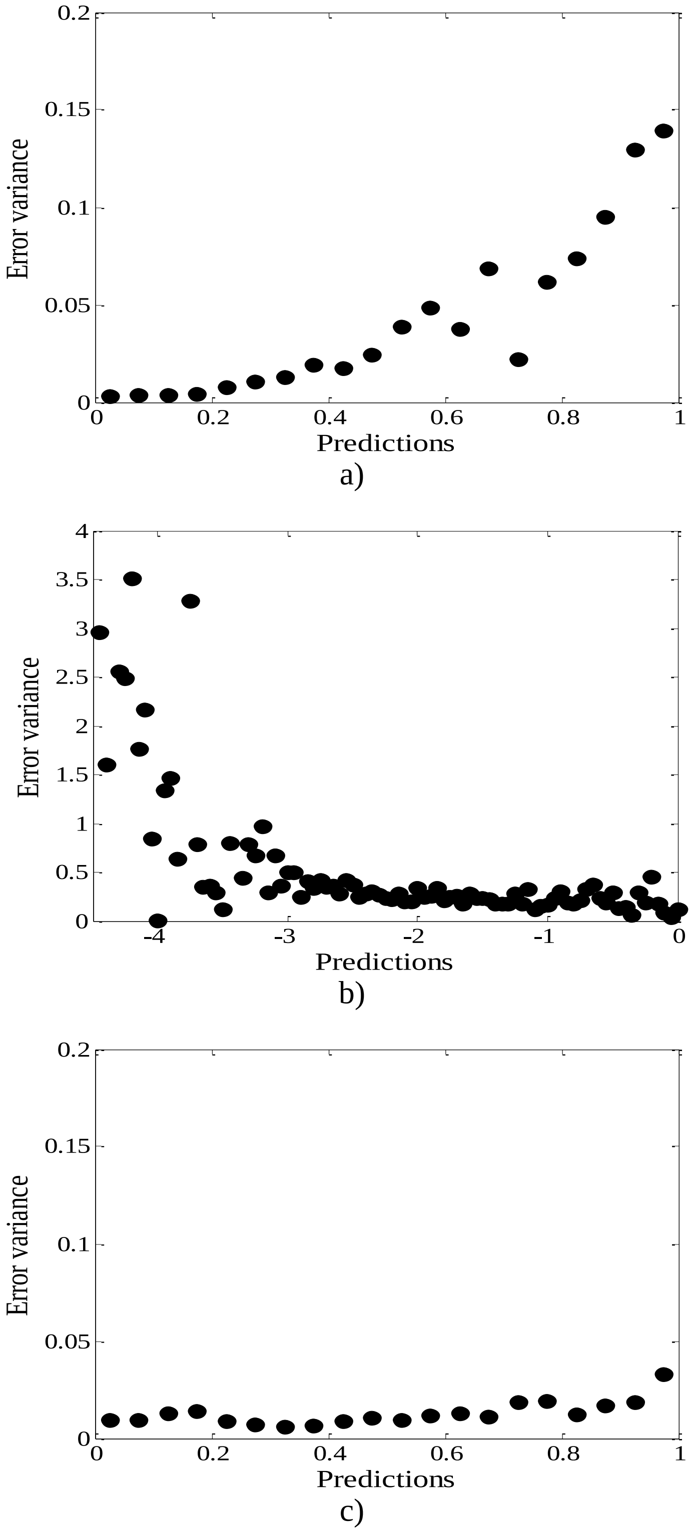


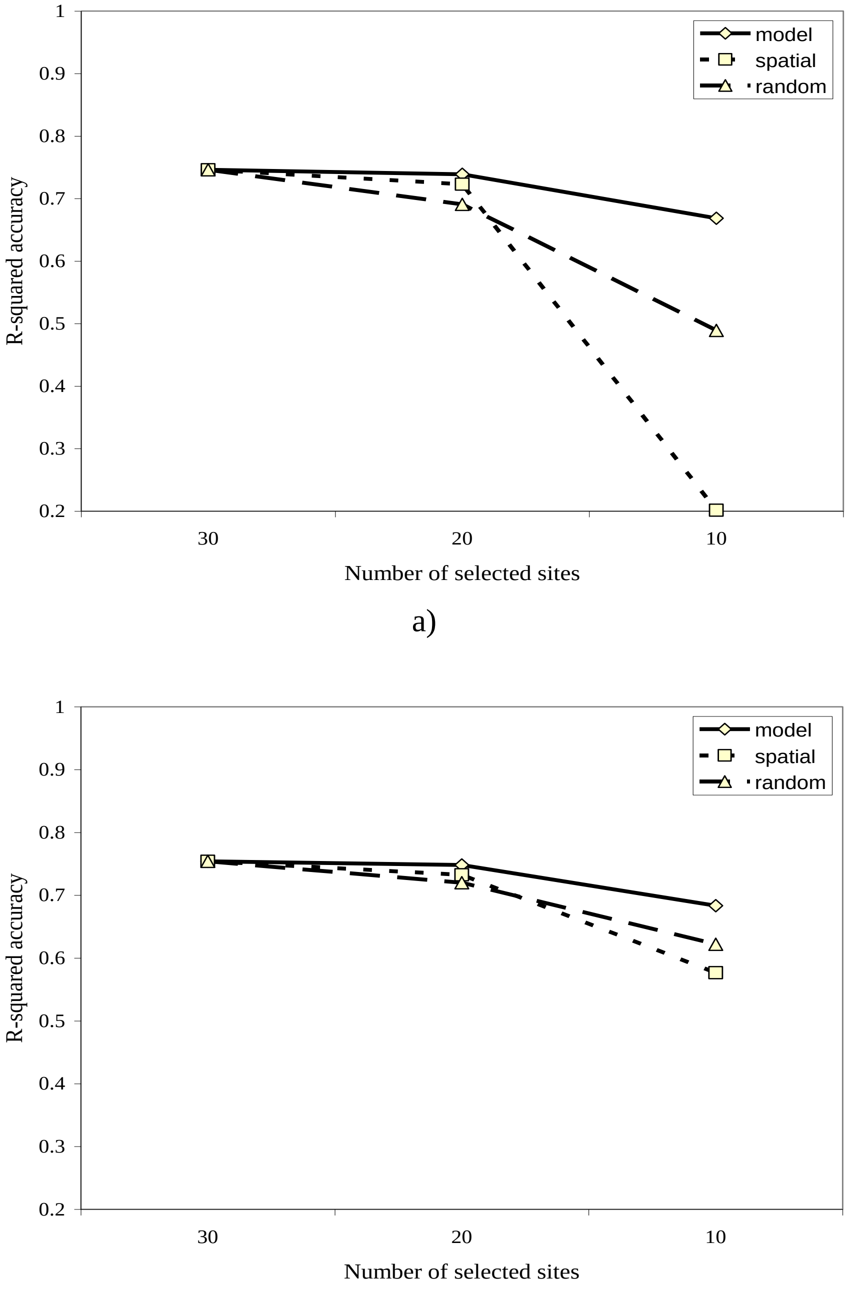
Figure 3.2. Variance stabilizing effect of output transformations. Error variance as a function of a) predictions without transformation b) predictions with log transformation c) predictions with square root transformation.

To better illustrate the effect of the square root transformation, we show variance of prediction errors as a function of predictions in Figure 3.2. As can be seen, if the transformation is not used, the error variance is large when large AODs are predicted. On the other hand, when the strong log-transformation is used, the error variance is large when small AODs are predicted. Finally, when square root transformation is used, error variance is practically constant and does not depend on the value of predicted AOD. Thus, minimizing *MSE* assuming constant variance (as in (3.1)) is justified for the square-root transformed AOD.

3.3.4.2. Selection of Informative Sites

We are considering a scenario when current operational AERONET sites have to be reduced by 33% or 66%. In all experiments, we started from a set of 30 AERONET sites and applied the proposed method and the two alternatives (spatial and random selection) to identify a subset of 20 or 10 AERONET sites to be retained. The NNSQ models were trained on labeled data from 2005. To test the goodness of the identified subset we tested the NNSQ models on 70 sites from 2006 (as described in the beginning of Section 3.3.4.).

The R^2 results averaged over 10 repetitions are presented in Figure 3.3a. We noticed that in some cases R^2 drops significantly when spatial and random selection strategies are used. Therefore, we also report median values of R^2 after 10 repetitions (Figure 3.3b). In our experiments, the proposed accuracy-based selection achieved consistently better results than the alternatives. Also, accuracy of the proposed site reduction method did not change much even after removing 20 of the 30 AERONET



a)

b)

Figure 3.3. R^2 in 10 iterations for different initial sets of 30 AERONET sites a) mean, b) median.

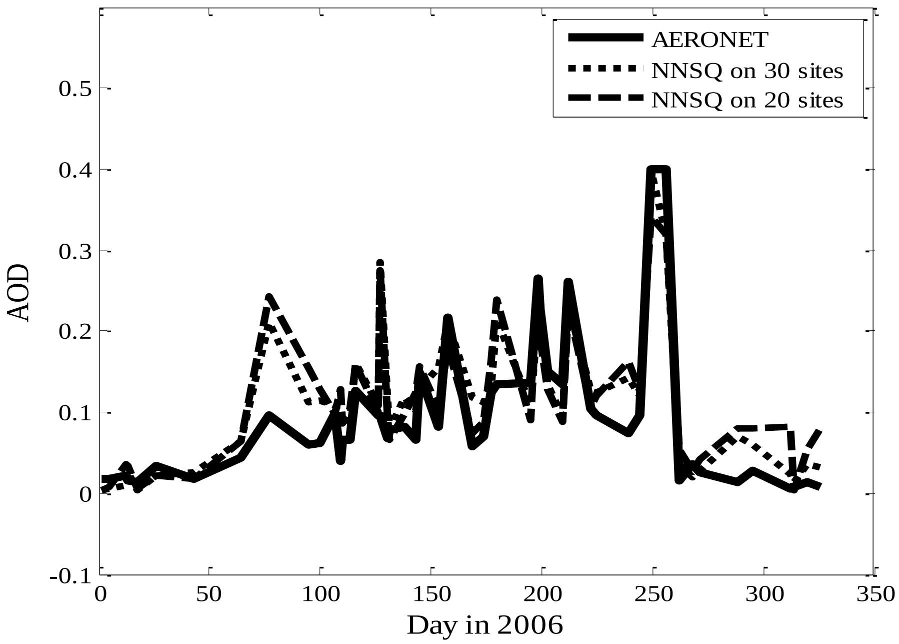
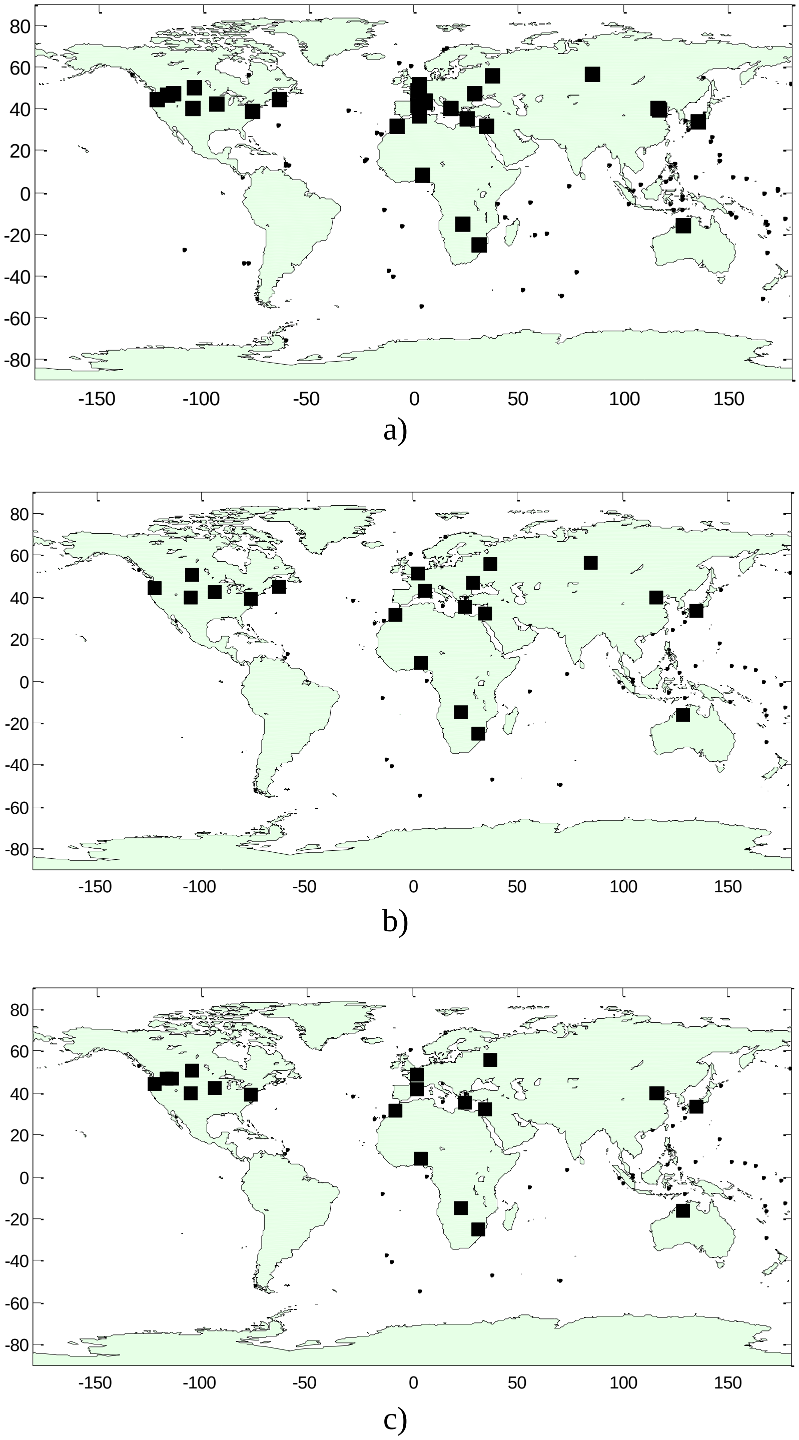


Figure 3.4. AOD predictions at site ‘BSRN_BAO_Boulder’ by NNSQ models trained on entire and reduced dataset.

sites. Interestingly, on average, the spatial selection strategy performed slightly worse than the random selection strategy.

Let us now consider the effect of the proposed sites reduction method on predictions at the site ‘BSRN_BAO_Boulder’ (40ºN, -105ºW). Sensor platform is on the rooftop of the building which is located on the high plains about 15 miles east of Boulder, CO, USA. Surrounding farmers’ fields make satellite AOD retrieval easier in some yearly seasons. Satellite predictions are more accurate over green regions which are often considered as dark [18] and therefore do not have an influence on observed radiation. Time series of AOD predictions at this site for a single placement of 30 training sites are presented in Figure 3.4. NNSQ model trained on a reduced dataset was able to predict ground-truth AOD slightly less accurately than the model trained on data from all 30 sites. In terms of R^2 accuracy, NNSQ trained on a reduced dataset achieved R^2 = 0.64 while NNSQ trained on non-reduced dataset achieved R^2 = 0.72 at the site 'BSRN_BAO_Boulder'. The conclusion is that accuracy-based reduction retains most of the accuracy of the model built on non-reduced dataset.

In Figure 3.5 and Figure 3.6 we illustrate site reduction for one initial placement of 30 AERONET sites. Spatial-based selection of AERONET sites nicely covers whole globe but it is not necessarily optimal for data-driven AOD prediction problem as we already noticed (Figure 3.3). On the other hand, some regions of the world were underrepresented when an accuracy-based principle was applied (Figure 3.6b). The accuracy was retained to a certain extent although no site from East US or from middle Asia was selected.

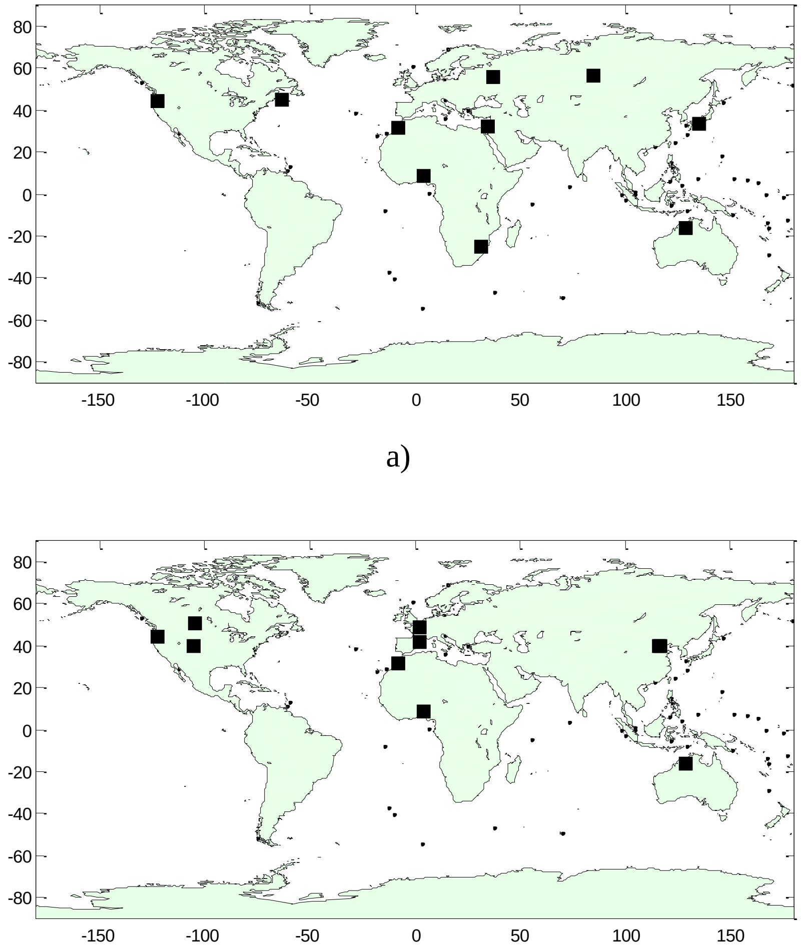


a)

b)

c)

Figure 3.5. a) Initial set of 30 AERONET sites b) spatial-based reduction to 20 sites c) accuracy-based reduction to 20 sites.



a)

b)

Figure 3.6. a) spatial-based reduction to 10 sites b) accuracy-based reduction to 10 sites.

3.4. Spatio-temporal Partitioning for Improving Prediction Accuracy

3.4.1. Introduction

In different spatio-temporal regions relations among the inputs can be different. Therefore, a single global predictor constructed using the entire dataset could be biased toward the dominant distribution while being less accurate on data points that do not follow the dominant distribution. If both space and time are partitioned in such a way that the observed inputs in each spatio-temporal subset have the same distribution, training specialized predictors on the identified subsets can be beneficial. Applying those local predictors on the corresponding spatio-temporal partitions would increase overall prediction accuracy as compared to the accuracy achieved by applying a single predictor on the entire dataset.

When the data generating process changes as a function of time and location, the same values of observed inputs could result in very different output values at various spatial-temporal regions. Therefore, proximity in the input space does not necessarily mean the data points should belong to the same spatio-temporal partition. In this situation, unsupervised clustering algorithms are not suitable for discovering spatio-temporal partitions.

The method for spatio-temporal partitioning explored in this Section is inspired by competition-based algorithm for learning from spatial data generated by a mixture of distributions [25]. In this approach multiple regression models are learned on disjoint spatial partitions followed by repartitioning based on competition between models where a data point is assigned to the model which has the highest prediction accuracy on it. The competition process is iterated as long as it leads to the accuracy gains. The method was successfully applied for discovering homogeneous regions in heterogeneous spatial data. A similar idea was successfully exploited for improving accuracy of nonstationary time series prediction through competition based on time partitioning [26]. The novel challenge addressed in this Section is how to deal with the distributional change that occurs over both spatial and temporal dimension.

The hypothesis we investigated is that, due to the variability of aerosol, AOD predictors specialized to specific spatio-temporal regions should result in the increased AOD prediction accuracy. We argue that the existing domain-based spatio-temporal partitioning used in C005 algorithm is not necessarily the best choice because of the fundamental differences in the nature of data-driven and domain-based AOD prediction approaches. For example, while the domain-based algorithms such as C005 eliminate surface effects from the satellite observations as a preprocessing step, data-driven algorithms use the observations directly as the inputs. The goal is to develop a method that automatically discovers a successful spatial-temporal partitioning for AOD prediction through the competition of regression models as an alternative to the domain based partitioning of space and time.

3.4.2. Statistical Foundation

When the data generating process changes over time and space, prediction accuracy can be significantly improved by learning a number of regression models specialized for certain spatio-temporal partitions as compared to a single (global) model learned on whole dataset. Let us assume that a spatio-temporal dataset is a union of K

disjoint partitions P_i, $i = 1,\ldots,K$, where the number of partitions and their spatio-temporal locations are not known in advance. The data generating process for P_i can be represented as

$$y_{st} = f_i(\mathbf{x}_{st}) + e_{st}, e_{st} \sim N(0,\sigma^2), st \in P_i, \tag{3.5}$$

where f_i is the regression function of partition P_i, $\mathbf{x}_{\mathrm{st}}$ and e_{st} are the input vector and the error term of observation at location s and time t. Domains of the observed inputs at different partitions generally overlap, which means that the same vector $\mathbf{x}$ can produce quite different outputs at different partitions.

Without any prior knowledge about the spatio-temporal partitions, learning a global prediction model over the entire dataset would result in learning the global regression function defined as

$$h^*(\mathbf{x}) = \underset{h(\mathbf{x})}{\arg\min}\, E_{Y|\mathbf{x}}[(Y - h(\mathbf{x})^2], \tag{3.6}$$

for any given $\mathbf{x}$. The *MSE* of the global prediction model h^* on the data from partition P_i, mse_i, can be expressed as [25]

$$\begin{aligned} mse_i &= noise_i + bias_i \\ noise_i &= E_{D_i}[e^2] \\ bias_i &= E_{D_i}[(h^*(\mathbf{x}) - f_i(\mathbf{x}))^2], \end{aligned} \tag{3.7}$$

over the domain D_i that corresponds to the partition P_i. The term $noise_i$ corresponds to an unavoidable error which would be obtained by a local predictor specialized for partition P_i and the term $bias_i$ corresponds to the bias of the global prediction model on the data from partition P_i. If spatio-temporal partitions were already known, the $bias_i$ from the previous equation would be eliminated by learning a local model on each partition.

3.4.3. Competition Based Spatio-temporal Partitioning

As we showed in Section 3.4.2, introducing local prediction models can improve prediction accuracy when data generating process is heterogeneous. We propose a method that discovers the appropriate spatio-temporal partitions.

We first describe an algorithm by ignoring information about location and time of data points. The algorithm relies on the competition among specialized predictors for each point in the dataset D. It starts by randomly dividing the entire dataset into K disjoint subsets D_i, i = 1…K, where K is the number of the specialized prediction models. A specialized predictor M_i is trained on each subset D_i, i = 1…K. The resulting predictors are competing for points from dataset D such that all the points that are best predicted with predictor M_i are assigned to subset D_i. The competition procedure is repeated until convergence. The described competition algorithm is noise-sensitive because it can easily lead to assignment of points near in space and time to different subsets. This is clearly an undesirable behavior and we need a mechanism that prevents this from happening.

Our solution is to group the data into spatio-temporal cells that contain multiple data points close in space and time and to run the competition procedure over the cells instead of the individual points. Similarly to the original algorithm, the cell is assigned to the prediction model that achieves the smallest average prediction error over the data points in the cell. After reassigning all cells, competition procedure is iteratively repeated until there is no improvement in prediction accuracy.

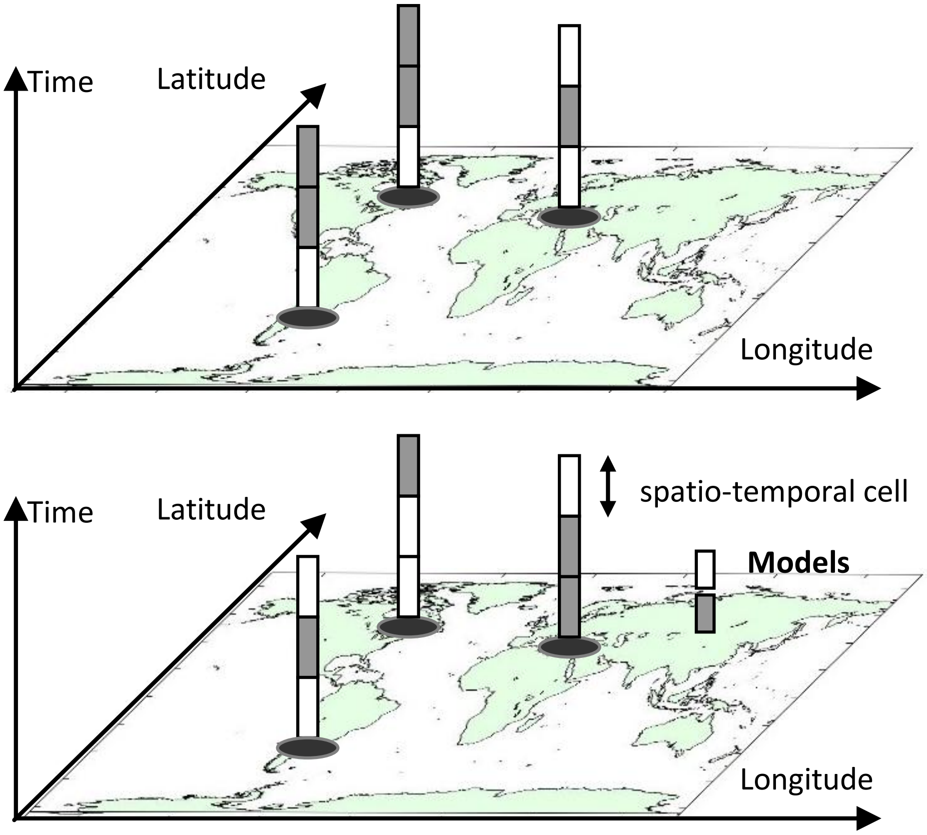


Figure 3.7. The simple example of partitioning procedure. Top picture – three spatial locations, data points are assigned to two models gray and white. Bottom picture – reassigned models.

The choice of the cell size is important because the small cells are sensitive to noise while the large cells could be heterogeneous. In the first case, the partitioning procedure would be unstable and the resulting specialized predictors would be just the artifacts of the procedure. In the second case, the partitioning would be too constrained and would result in highly similar specialized predictors. To achieve the best possible partitioning several choices for time period length should be considered.

Let us demonstrate on a simple example how the proposed method operates. In the Figure 3.7, in the top picture, all spatio-temporal data points from three locations are assigned to one of the two models – ‘gray’ or ‘white’. In our example entire time interval is divided into three parts. Therefore, there are nine spatio-temporal cells. Models are trained and predictions for all data points are obtained. Data points in a cell are assigned to the model achieving better accuracy. The resulting partitioning results in the temporally more homogeneous partitions. The competition procedure iterates until stable solution is found.

3.4.4. Experimental Results and Discussion

The primary benchmark AOD predictor for comparison with our approach is the most recent version of the MODIS operational algorithm, C005, as validation studies show that version C005 is significantly more accurate than the previous version C004 [18]. However, at the time of study, C005 predictions were available only for the first eight months of 2005. In order to give a more complete evaluation of our partitioning algorithm, we compared our algorithm with C004 predictions that were available during the longer period, between April 2003 and November 2005. The C004 data set consisted of 23,903 data points containing MODIS observations, C004 AOD predictions, and collocated AERONET AOD measurements from 129 AERONET sites over the globe.

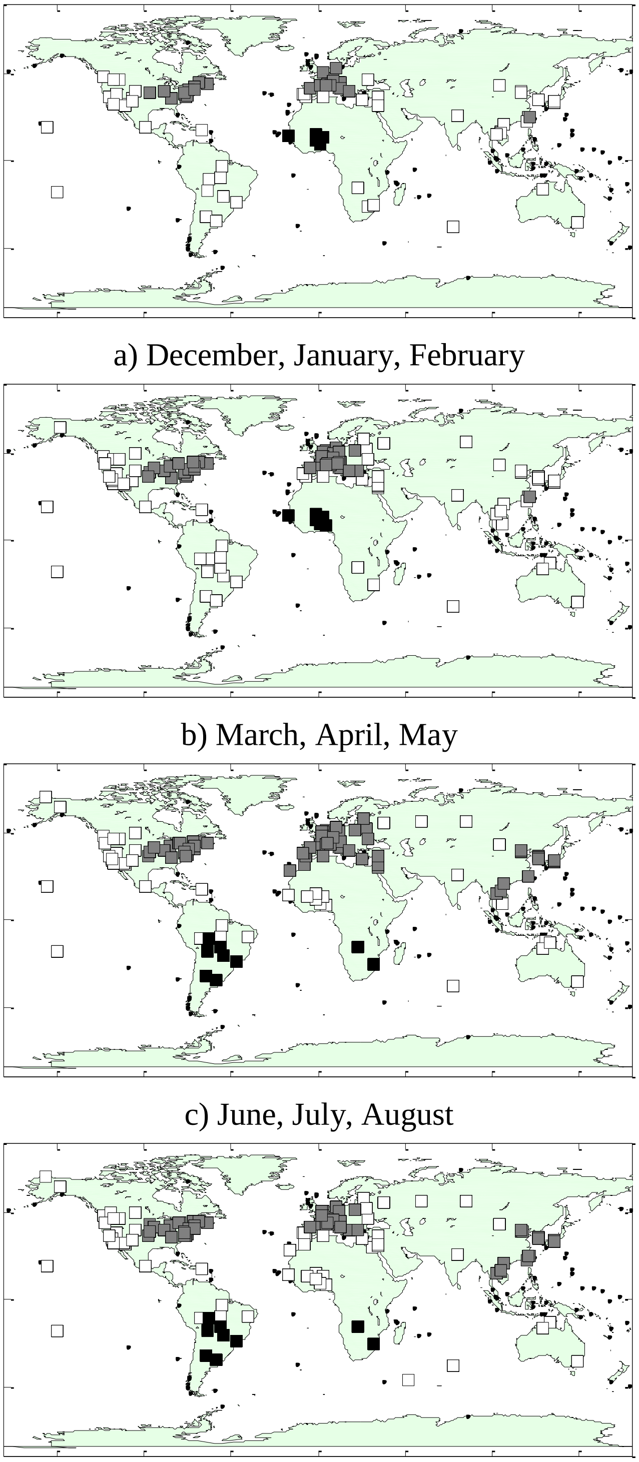
Neural networks were trained on 16,328 collocated data points, collected between April 2003 and November 2004. The remaining 7,575 data points, from December 2004 to November 2005, were used as test set for accuracy estimation. It should be noted that the test data covered the consecutive four seasons that were not seen during training. By dividing dataset in this way, we avoided the problem of memorizing training data, which would have occurred if the training-test split was performed randomly. The memorization would occur because of the temporal correlation in AOD values that can remain significant over periods of up to a few weeks.

Neural networks with one hidden layer and one neuron in the output layer were used throughout all experiments. The sigmoid activation function has been used for all hidden neurons while the linear activation function was used for the output neuron. The neural networks were trained using *MSE* function as a cost function.

3.4.4.1. Experiments on Predictors Specialized for C005 Spatio-temporal Partitions

C005 utilizes domain knowledge for spatio-temporal partitioning of the Earth. For each spatio-temporal partition, C005 consults the look-up table constructed by forward simulations of the physical model of aerosol optical properties.

C005 defines four aerosol models corresponding to prevalent atmospheric conditions over several characteristic spatio-temporal regions of Earth (Figure 2.4) [18]. The partitioning was obtained by studying observations from AERONET ground-based instruments and combining this information with the climatology domain knowledge. One of those four models is invariant through time and can be applied globally while the other three models, used to adjust the global model, depend on the location and time. When defining aerosol models as a function of location and time, the assumption was that



a) December, January, February

b) March, April, May

c) June, July, August

d) September, October, November

Figure 3.8. AERONET sites assigned to the spatio-temporal models of operational C005 AOD prediction algorithm. Three models are represented by white, gray and black colors.

aerosol properties would not change a lot during a three-month season. For each AERONET site and each season, the percentage of data points best described by each of three models was determined. This was used to assign the dominant aerosol type to each AERONET site during each season. The resulting data partitioning used in C005 divides the world into three spatial-temporal regions that differ based on the location and the time of the year as summarized at Figure 2.4 and Figure 3.8.

To test whether spatio-temporal partitions defined in operational C005 algorithm could be used in data-driven AOD prediction approach, we trained neural networks specialized for the three regions presented in Figure 3.8. Each of the three neural networks was trained on data belonging to one of the partitions (white, gray, or black) depicted in Figure 3.8. We used data between April 2003 and November 2004 for training, while the test set was taken between December 2004 and November 2005.

The results are presented in Table 3.4. While the performance achieved was better than the performance of C004 algorithm, considering all accuracy measures, it was surprisingly worse than the performance of a single neural network predictor.

3.4.4.2. Experiments with Predictors Trained on Spatio-temporal Partitions Discovered by Competition

The previous results indicate that domain-based partitioning is not suitable for learning specialized AOD predictors. Instead, we applied the proposed competition method to find spatio-temporal partitions.

To run the procedure, we have to choose an appropriate size of spatio-temporal cells. Each cell is defined as a time interval for a specific AERONET site. We evaluated several cell size choices, empirically. The largest temporal size we considered was T = 12 months as the aerosol concentration is periodic with yearly cycles. In addition, we considered smaller temporal sizes of T = 6, T = 4 and T = 1 months. Cell temporal size is fixed during the competition procedure. The competition starts with K = 2 models. In the first step, the entire dataset is divided randomly into K equal sized subsets. Next, neural network predictors were trained on each of the two subsets. Data from each AERONET site were partitioned into the consecutive, disjoint temporal cells of size T. Given predictions of the competing predictors on all examples within a cell, the cell is assigned to the model achieving the smallest prediction error. The competition iterated until a stable solution was found. The experiment was repeated for a number of different parameter values K and T. Finally, all possible partitioning were evaluated on the independent test set and the best one was chosen as the final solution.

There were several additional issues that had to be addressed. First, we wanted to avoid making evaluations of the competing predictors on the training data. Instead of training competing neural networks on the complete training data set, we applied 4-cross-validation procedure. Data from each month were partitioned into 4 weekly intervals; one week was used for validation, while the remaining three weeks were merged and used for training.

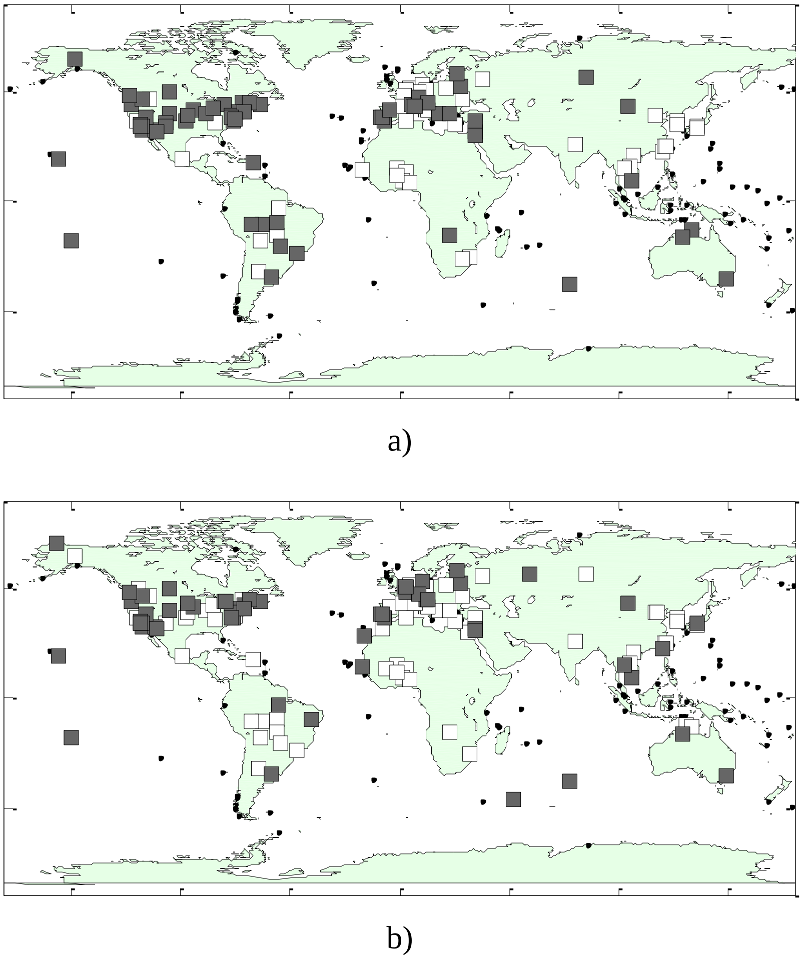
Second, the cost function for training the neural networks in the competition procedure had to be determined. Due to an abundance of outliers in the training data, the standard *MSE* function was not the most appropriate choice because the training procedure would be dominated by the outliers and it would be difficult to find a stable

Table 3.4. Different predictors Vs. AERONET AOD accuracy.

Model	**T (month)**	***MSE***	***R^2***	***CORR***	***RSE***	***FRAC***
C004	-	0.034	0.46	0.79	5.36	52%
NN, Single	-	0.021	0.67	0.82	1.95	60%
NN on C005	-	0.023	0.63	0.80	2.40	61%
NN, 2 models	12	0.017	0.73	0.85	1.80	65%
NN, 2 models	6	**0.015**	**0.75**	**0.87**	**1.55**	**68%**
NN, 2 models	3	0.19	0.70	0.84	1.76	64%
NN, 2 models	1	0.017	0.73	0.85	1.82	65%
NN, 3 models	12	0.19	0.70	0.84	1.83	65%
NN, 3 models	6	0.018	0.72	0.85	1.77	67%
NN, 3 models	3	0.018	0.72	0.85	1.70	66%
NN, 3 models	1	0.022	0.65	0.82	2.23	67%

solution. To overcome this problem we used $REL_{0.05,0.15}$ from (3.2) as a cost function. As we showed this function is less sensitive to the outliers.

Third, prediction models compete for the cells based on the prediction error, which has to be defined. To avoid the possibility that outliers could dominate the competition procedure, we used average $REL_{0.05,0.15}$ error over the cell to determine the winning model. The model that achieves minimal $REL_{0.05,0.15}$ was considered the winner. Finally, the neural network predictors were built on the discovered spatio-temporal partitions. Those networks had to be trained using the standard *MSE* cost function, since networks trained with $REL_{0.05,0.15}$ as a cost function tend to underestimate large AOD values.



a)

b)

Figure 3.9. AERONET spatio-temporal partitions obtained by competition procedure a) two partitions for winter-spring months, b) two partitions for summer-fall months.

The competition procedure was applied on the explained training data from between April 2003 and November 2004. The learned spatio-temporal partitions were evaluated on the test data between December 2004 and November 2005. The results for the different *K* and *T* values are presented in Table 3.4.

Based on the results from Table 3.4, we can conclude that the proposed competition-based spatio-temporal data partitioning approach resulted in large accuracy improvements. The best results were obtained for cell size of six months (T = 6), where one interval covered winter-spring months and another summer-fall months, and for spatio-temporal partitioning that results in two specialized predictors (K = 2).

The resulting spatio-temporal partitions are shown in Figure 3.9. From Figure 3.9, we can see that during the winter-spring months the whole U.S. was assigned to the same partition, while during the summer-fall months some U.S. sites moved to the other partition. Also, AERONET sites in Africa did not change their assignment during the year. It is interesting to mention that average AOD in the 'gray' partition is 0.13 with standard deviation 0.12 while average AOD in white partition is 0.29 with standard variation 0.35. Although it might appear that the competition procedure discovered partitions based on the average AOD values, the standard deviation suggests that the underlying process is more complicated.

CHAPTER 4

STRUCTURED DATA-DRIVEN PREDICTORS

4.1. Introduction

Traditional supervised learning models, like neural networks (NN), are powerful tools for learning non-linear mappings. However, such models mainly focus on the prediction of a single output and could not exploit relationships that exist between multiple outputs. In structured learning, the model learns a mapping f: $\mathbf{X}^N \rightarrow R^N$ to simultaneously predict *all* outputs given *all* input vectors. For example, let us assume that the value of y_i is dependent on that of y_{i-1} and y_{i+1}, as is the case in temporal data. Let us also assume that input $\mathbf{x}_i$ is noisy. A traditional model that uses only information contained in $\mathbf{x}_i$ to predict y_i might predict the value for y_i to be quite different from those of $y_{i\text{-}1}$ and y_{i+1} because it treats them individually. A structured predictor uses dependencies among outputs to take into account that y_i is more likely to have value close to $y_{i\text{-}1}$ and y_{i+1} thus improving final predictions. In structured learning we usually have some prior knowledge about relationships among the outputs y. Mostly, those relationships are application-specific where the dependencies are defined in advance, either by domain knowledge or by assumptions, and represented by statistical models.

4.2. Related Work

There has been a rich body of work in spatial statistics that aims to exploit the correlation in structured data. Previous studies [27], [28] showed that the Spatial AutoRegression model (SAR), a generalization of the linear regression model that accounts for spatial autocorrelation, improves classification and prediction accuracy for many spatial datasets. An extension of the Gaussian Processes (GP) framework to structured outputs was proposed in [29]. In that approach by assuming outputs to be linear combinations of latent functions, GP priors are placed over each of the latent functions. In geostatistics this approach is known as Linear Model of Coregionalization (LMC) [30]. LMC achieves better prediction accuracy over the models that do not account for spatial correlation. Twin Gaussian Processes (TGP) [31] is a recently proposed structured prediction method that captures not only the interdependencies between inputs, as classical GP do, but also the correlations among outputs. TGP was successfully applied for recovery of 3D human pose estimation from an image. However, original TGP is impractical for large size problems as it requires the inversion of covariance matrices. This was overcome by TGPKNN that reduces input set to K nearest neighbors of test set.

Relationships among outputs can be represented by graphical models. The advantage of the graphical models is that one can make use of sparseness in the interactions between outputs and develop efficient learning and inference algorithms. In learning from spatial-temporal data, the Markov Random Fields [32] and the more recently proposed Conditional Random Fields (CRF) [33] are among the most popular graphical models. Originally, CRF were designed for classification of sequential data

[33] and have found many applications in areas such as computer vision [34] and computational biology [35].

CRF for regression is a less explored topic. The Conditional State Space Model (CSSM) [36], an extension of the CRF to a domain with the continuous multivariate outputs, was proposed for regression of sequential data. It is an attractive discriminative alternative to Linear Dynamic Systems (LDS), a generative model also known as the Kalman filter. Unlike LDS, CSSM is an undirected model that makes no independence assumptions between outputs, which results in a more flexible modeling framework. Continuous CRF (CCRF) [37] is a ranking model that takes into account relations among ranks of objects in document retrieval. In [38], a conditional distribution of pixels given a noisy input image is modeled using the weighted quadratic factors obtained by convolving the image with a set of filters. Feature functions in [38] were specifically designed for image de-noising problems and are not readily applicable to regression.

Most CRF models represent linear relationships between attributes and outputs. On the other hand, in many real-world applications this relationship is highly complex and nonlinear and cannot be accurately modeled by a linear function. CRF that models nonlinear relationship between observations and outputs has recently been applied to the problem of image de-noising [38]. Integration of CRF and Neural Networks (CNF) [39-41] has been recently proposed for classification problems to address these limitations by adding a middle layer between attributes and outputs. This layer consists of a number of gate functions, each acting as a hidden neuron, that capture the nonlinear relationships. As a result, such models can be much more expressive than CRF.

4.3. Gaussian Conditional Random Fields

4.3.1. Continuous Conditional Random Fields Model

Conditional Random Fields (CRF) provide probabilistic framework for exploiting complex dependence structure among outputs by directly modeling the conditional distribution $P(\mathbf{y}|\mathbf{x})$. In regression problems, the output y_i is associated with input vectors $\mathbf{x} = (\mathbf{x}_1,\dots\ \mathbf{x}_N)$ by a real-valued function called *association potential* $A(\boldsymbol{\alpha},y_{\mathrm{i}},\mathbf{x})$, where $\boldsymbol{\alpha}$ is K-dimensional set of parameters. The larger the value of A is the more y_{i} is related to $\mathbf{x}$. Usually, A is a combination of functions. We can use as many association functions as we find necessary to model input-output relations in data. In general, A takes as input all input data $\mathbf{x}$ to predict a single output y_{i} meaning that it does not impose any independency relations among inputs $\mathbf{x_i}$.

To model interactions among outputs, a real valued function called *interaction potential* $I(\boldsymbol{\beta},y_{\mathrm{i}},y_{\mathrm{j}},\mathbf{x})$ is used, where $\boldsymbol{\beta}$ is an L dimensional set of parameters. Interaction potential represents the relationship between two outputs and in general can depend on an input $\mathbf{x}$. Different applications can have different interaction potentials. For example, in the AOD prediction problem, interaction potential can be modeled as a correlation between neighboring (in time and space) outputs. The larger the value of the interaction potential, the more related outputs are.

For the defined association and interaction potentials, continuous CRF models a conditional distribution $P(\mathbf{y}|\mathbf{x})$, $\mathbf{y} = (y_1\dots y_N)$, according to the associated graphical structure (an example of the structure is shown in Figure 4.1.)

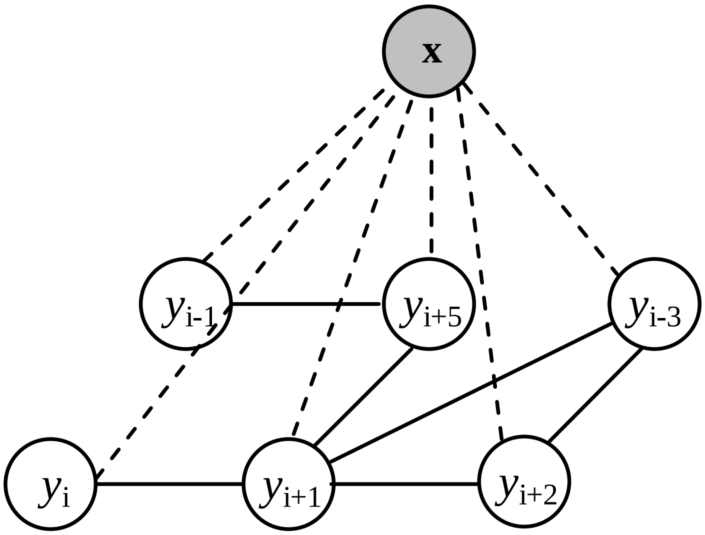


Figure 4.1. Continuous CRF graphical structure. **x**-inputs (observations); *y*-outputs; dashed lines-associations between inputs and outputs; solid lines-interactions between outputs.

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})} \exp(\sum_{i=1}^{N} A(\boldsymbol{\alpha}, y_i, \mathbf{x}) + \sum_{j\sim i} I(\boldsymbol{\beta}, y_i, y_j, \mathbf{x})), \tag{4.1}$$

where *j~i* denotes the connected outputs y_i and y_j (connected with solid line at Figure 4.1) and where *Z*(**x**,**α**,**β**) is normalization function defined as

$$Z(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{y} \exp(\sum_{i=1}^{N} A(\boldsymbol{\alpha}, y_i, \mathbf{x}) + \sum_{j\sim i} I(\boldsymbol{\beta}, y_i, y_j, \mathbf{x}))dy\,. \tag{4.2}$$

The learning task is to choose values of parameters **α** and **β** to maximize the conditional log-likelihood of the set of training examples

$$L(\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum \log P(\mathbf{y} \mid \mathbf{x})$$
$$(\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{\beta}}) = \arg\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} (L(\boldsymbol{\alpha},\boldsymbol{\beta})). \tag{4.3}$$

This can be achieved by applying standard optimization algorithms such as gradient descent. To avoid overfitting, we regularize L(**α**,**β**) by adding $\boldsymbol{\alpha}^2/2$ and $\boldsymbol{\beta}^2/2$ terms to log-likelihood in formula (4.3) that prevents the parameters from becoming too large.

The inference task is to find the outputs **y** for a given set of observations **x** and estimated parameters **α** and **β** such that the conditional probability *P*(**y**|**x**) is maximized

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} (P(\mathbf{y} \mid \mathbf{x})) . \tag{4.4}$$

Learning and inference in models with real valued outputs pose quite different challenges than in the discrete-valued case. The most important difference is that the normalizing function *Z* is an integral instead of the sum. Discrete valued models are always feasible as *Z* is a finite number defined as a sum over finitely many possible values of **y**. On the contrary, to have a feasible model with real valued outputs, *Z* must be integrable. Proving directly that *Z* is integrable might be difficult due to the complexity of association and interaction potentials.

In CRF applications, *A* and *I* could be defined as linear combinations of a set of fixed features in terms of **α** and **β** [33]

$$A(\boldsymbol{\alpha}, y_i, \mathbf{x}) = \sum_{k=1}^{K} \alpha_k f_k(y_i, \mathbf{x})$$
$$I(\boldsymbol{\beta}, y_i, y_j, \mathbf{x}) = \sum_{l=1}^{L} \beta_l g_l(y_i, y_j, \mathbf{x}). \tag{4.5}$$

The use of features to define the model is convenient because it allows us to include arbitrary properties of input-output pairs into the compatibility measure. This way, any potentially relevant feature could be included to the model because parameter estimation automatically determines their actual relevance by feature weighting.

4.3.2. 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 simple constructs to reduce the complexity of learning and inference from CRF. In general, to evaluate $P(\mathbf{y}|\mathbf{x})$ during learning and inference, one would need to use time consuming sampling methods. However, if A and I are defined as quadratic functions of $\mathbf{y}$, we will show that $P(\mathbf{y}|\mathbf{x})$ becomes multivariate Gaussian distribution and that learning and inference can be accomplished in a computationally efficient manner.

In the following, we describe the proposed feature functions that lead to Gaussian CRF. Let us assume we are given K unstructured predictors, $R_k(\mathbf{x})$, k=1,…K, that predict single output y_i taking into account $\mathbf{x}$ (as a special case, only $\mathbf{x}_i$ can be used as $\mathbf{x}$). To model the dependency between the prediction and output, we introduce quadratic feature functions

$$f_k(y_i, \mathbf{x}) = -(y_i - R_k(\mathbf{x}))^2, k = 1,...K\ . \tag{4.6}$$

These feature functions follow the basic principle for association potentials in that their values are large when predictions and outputs are similar. To model the correlation among outputs, we introduce the quadratic feature function

$$g_l(y_i, y_j, \mathbf{x}) = -e_{ij}^{(l)} S_{ij}^{(l)}(\mathbf{x})(y_i - y_j)^2, \ e_{ij}^{(l)} = 1 \text{ if } (i,j) \in G_l, \ e_{ij}^{(l)} = 0, \text{ otherwise}, \qquad (4.7)$$

that imposes that outputs y_i and y_j have similar values if they have an edge in the graph G_l. It should be noted that using multiple graphs G_l can facilitate modeling of different aspects of correlation between outputs (for example, spatial and temporal). $S_{ij}^{(l)}(\mathbf{x})$ function represents similarity between outputs y_i and y_j, that depends on inputs $\mathbf{x}$. The larger $S_{ij}^{(l)}(\mathbf{x})$ is, the more similar the outputs y_i and y_j are.

4.3.3. Gaussian Canonical Form

In this section, we show that $P(\mathbf{y}|\mathbf{x})$ for CRF model (4.1), which uses quadratic feature functions defined in Section 4.3.2, can be represented as a multivariate Gaussian distribution. The resulting CRF model can be written as

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z} \exp\left(-\sum_{i=1}^{N}\sum_{k=1}^{K} \alpha_k (y_i - R_k(\mathbf{x}))^2 - \sum_{i,j}\sum_{l=1}^{L} \beta_l e_{ij}^{(l)} S_{ij}^{(l)}(\mathbf{x})(y_i - y_j)^2\right). \qquad (4.8)$$

The exponent in (4.8), which we denote as E, is a quadratic function in terms of $\mathbf{y}$. Therefore, $P(\mathbf{y}|\mathbf{x})$ can be transformed to a Gaussian form by representing E as

$$E = -\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}) = -\frac{1}{2}\mathbf{y}^{\mathbf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{y} + \mathbf{y}^{\mathbf{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + const\,. \qquad (4.9)$$

To transform $P(\mathbf{y}|\mathbf{x})$ to the Gaussian form, we determine $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ by matching (4.8) and (4.9). We first represent the quadratic terms of $\mathbf{y}$ in the association and interaction potentials as $-\mathbf{y}^T\mathbf{Q}_1\mathbf{y}$ and $-\mathbf{y}^T\mathbf{Q}_2\mathbf{y}$, respectively, and combine them to get

$$\boldsymbol{\Sigma}^{-1} = 2(\mathbf{Q}_1 + \mathbf{Q}_2). \qquad (4.10)$$

By combining the quadratic terms of $\mathbf{y}$ from the association potential, it follows that $\mathbf{Q}_1$ is diagonal matrix with elements

$$Q_{1ij} = \begin{cases} \sum_{k=1}^{K} \alpha_k, i = j \\ \\ 0, i \neq j. \end{cases} \tag{4.11}$$

By repeating this for the interaction potential, it follows that **Q**₂ is symmetric with elements

$$Q_{2ij} = \begin{cases} \sum_{k} \sum_{l=1}^{L} \beta_l e_{ik}^{(l)} S_{ik}^{(l)}(\mathbf{x}), i = j \\ \\ -\sum_{l=1}^{L} \beta_l e_{ij}^{(l)} S_{ij}^{(l)}(\mathbf{x}), i \neq j. \end{cases} \tag{4.12}$$

To get $\boldsymbol{\mu}$**,** we match linear terms in *E* with linear terms in the exponent of (4.8) and obtain

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\mathbf{b}, \tag{4.13}$$

where **b** is vector with elements

$$b_i = 2(\sum_{k=1}^{K} \alpha_k R_k(\mathbf{x})). \tag{4.14}$$

By calculating *Z* using the transformed exponent, it follows

$$Z(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{x}) = (2\pi)^{N/2} |\boldsymbol{\Sigma}|^{1/2} \exp(const). \tag{4.15}$$

Since exp(*const*) terms from *Z* and *P*(**y**|**x**) cancel out, we finally get

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{1}{(2\pi)^{N/2} |\boldsymbol{\Sigma}|^{1/2}} \exp(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu})), \tag{4.16}$$

where $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ are defined in (4.10) and (4.13). Therefore, the resulting conditional distribution is Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. We observe that $\boldsymbol{\Sigma}$ is a function of parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, interaction potential graphs G_l, and similarity functions *S*, while $\boldsymbol{\mu}$ is also a function of inputs **x**. We call the resulting CRF *the Gaussian CRF* (GCRF).

4.3.4. Learning

The learning task is to choose **α** and **β** to maximize the conditional log-likelihood,

$$(\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{\beta}}) = \arg\max_{\boldsymbol{\alpha},\boldsymbol{\beta}}(L(\boldsymbol{\alpha},\boldsymbol{\beta})), \text{where } L(\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum \log P(\mathbf{y}\,|\,\mathbf{x})\,. \qquad (4.17)$$

Models with real valued outputs pose quite different challenges than in the discrete-valued case. Discrete valued models are always feasible because Z is finite and defined as a sum over finitely many possible values of **y**. On the contrary, to have a feasible model with real valued outputs, Z must be integrable. Proving that Z is integrable in general might be difficult due to the complexity of association and interaction potentials. Let us analyze the feasibility condition for GCRF model. In order for the model to be feasible, the precision matrix $\boldsymbol{\Sigma}^{-1}$ has to be positive semi-definite. $\boldsymbol{\Sigma}^{-1}$ is defined as a double sum of $\mathbf{Q}_1$ and $\mathbf{Q}_2$. $\mathbf{Q}_2$ is a symmetric matrix with a property that the absolute value of a diagonal element is equal to the sum of absolute values of non-diagonal elements from the same row

$$|\,Q_{2ii}\,| = \sum_{j\neq i} |\,Q_{2ij}\,|\,. \qquad (4.18)$$

By Gershgorin's circle theorem [42], a symmetric matrix is positive semi-definite if all diagonal elements are non-negative and if matrix is diagonally dominant. Therefore, one way to ensure that GCRF model is feasible is to impose the constraint that all elements of **α** and **β** are greater than 0. In this setting, learning is a constrained optimization problem. To convert it to the unconstrained optimization, we adopt a technique used in [37] that applies the exponential transformation on **α** and **β** parameters to guarantee that they are positive

$$\alpha_k = e^{\mathrm{u}_k}, \text{for } k = 1,\ldots K$$

$$\beta_l = e^{v_l}, \text{for } l = 1,\ldots L, \tag{4.19}$$

where *u* and *v* are real valued parameters. As a result, the new optimization problem becomes unconstrained.

All parameters are learned by the gradient-based optimization. To apply it, we need to find the gradient of the conditional log-likelihood. Let us start from the expression for log*P*

$$\log P = -\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu}) - \frac{1}{2}\log|\boldsymbol{\Sigma}|. \tag{4.20}$$

The first term in (4.20) is a product of three matrices of dimensions [1 × *N*], [*N* × *N*], and [*N* × 1], respectively. Hence, the product is scalar. Knowing that trace *Tr* of a matrix has nice properties such as *Tr*(**ABC**) = *Tr*(**BCA**) = *Tr*(**CAB**), *Tr*(**AB**) = *Tr*(**BA**) for matrices **A**, **B**, and **C**, when these products are defined, and *Tr*(*scalar*) = *scalar,* we can replace the first term with its trace and apply *Tr*(**ABC**) = *Tr*(**BCA**). Therefore,

$$\log P = -\frac{1}{2}Tr(\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})(\mathbf{y}-\boldsymbol{\mu})^T) - \frac{1}{2}\log|\boldsymbol{\Sigma}|. \tag{4.21}$$

The derivative of log|**A**| = *Tr*($\mathbf{A}^{-1}$*d***A**), derivative of trace is trace of derivative, and by treating $\boldsymbol{\Sigma}^{-1}$**(y- μ)**(y- **μ)$^{\mathbf{T}}$** as a product of $\boldsymbol{\Sigma}^{-1}$ and quadratic form **(y- μ)(**y- **μ)$^{\mathbf{T}}$,** we get

$$d\log P = -\frac{1}{2}Tr(d\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})(\mathbf{y}-\boldsymbol{\mu})^T - 2\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})d\boldsymbol{\mu}^T) - \frac{1}{2}Tr(\boldsymbol{\Sigma}^{-1}\cdot d\boldsymbol{\Sigma}). \tag{4.22}$$

Equation (4.22) can be simplified by using the following expressions for the derivative of an inverse matrix

$$d\mathbf{\Sigma} = -\mathbf{\Sigma}\cdot d\mathbf{\Sigma}^{-1}\cdot\mathbf{\Sigma}$$

$$\begin{aligned} d\boldsymbol{\mu}^T &= d(\mathbf{\Sigma}\mathbf{b})^T = (d\mathbf{\Sigma}\cdot\mathbf{b}+\mathbf{\Sigma}\cdot d\mathbf{b})^T \\ &= (\text{-}\mathbf{\Sigma} d\mathbf{\Sigma}^{-1}\mathbf{\Sigma}\cdot\mathbf{b}+\mathbf{\Sigma}\cdot d\mathbf{b})^T \\ &= (\text{-}\mathbf{\Sigma} d\mathbf{\Sigma}^{-1}\boldsymbol{\mu}+\mathbf{\Sigma}\cdot d\mathbf{b})^T \\ &= (d\mathbf{b}-d\mathbf{\Sigma}^{-1}\boldsymbol{\mu})^T\mathbf{\Sigma}^T. \end{aligned} \tag{4.23}$$

By replacing (4.23) in (4.22) and using the fact that $\mathbf{\Sigma^T}$=$\mathbf{\Sigma}$**,** as $\mathbf{\Sigma}$ is symmetric, we obtain

$$\begin{aligned} d\log P = &-\frac{1}{2}Tr(d\mathbf{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})(\mathbf{y}-\boldsymbol{\mu})^T - 2\mathbf{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})(d\mathbf{b}^T-\boldsymbol{\mu}^T d\mathbf{\Sigma}^{-1})\mathbf{\Sigma}) \\ &+\frac{1}{2}Tr(\mathbf{\Sigma}^{-1}\cdot\mathbf{\Sigma}\cdot d\mathbf{\Sigma}^{-1}\cdot\mathbf{\Sigma}). \end{aligned} \tag{4.24}$$

By applying again the algebra of traces, we get

$$d\log P = -\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T d\mathbf{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu}) + (d\mathbf{b}^T-\boldsymbol{\mu}^T d\mathbf{\Sigma}^{-1})(\mathbf{y}-\boldsymbol{\mu}) + \frac{1}{2}Tr(d\mathbf{\Sigma}^{-1}\cdot\mathbf{\Sigma}). \tag{4.25}$$

From (4.25) we can calculate derivatives $\partial\log P/\partial\alpha_k$ and $\partial\log P/\partial\beta_l$. The expression for $\partial\log P/\partial\alpha_k$ is

$$\begin{aligned} \frac{\partial\log P}{\partial\alpha_k} = &-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\frac{\partial\mathbf{\Sigma}^{-1}}{\partial\alpha_k}(\mathbf{y}-\boldsymbol{\mu}) + (\frac{\partial\mathbf{b}^T}{\partial\alpha_k}-\boldsymbol{\mu}^T\frac{\partial\mathbf{\Sigma}^{-1}}{\partial\alpha_k})(\mathbf{y}-\boldsymbol{\mu}) \\ &+\frac{1}{2}Tr(\mathbf{\Sigma}\cdot\frac{\partial\mathbf{\Sigma}^{-1}}{\partial\alpha_k}). \end{aligned} \tag{4.26}$$

To calculate $\partial\log P/\partial\beta_l$, we use $\partial\mathbf{b}/\partial\beta_l$=0 to obtain

$$\frac{\partial\log P}{\partial\beta_l} = -\frac{1}{2}(\mathbf{y}+\boldsymbol{\mu})^T\frac{\partial\mathbf{\Sigma}^{-1}}{\partial\beta_l}(\mathbf{y}-\boldsymbol{\mu}) + \frac{1}{2}Tr(\mathbf{\Sigma}\cdot\frac{\partial\mathbf{\Sigma}^{-1}}{\partial\beta_l}). \tag{4.27}$$

Gradient ascent algorithm cannot be directly applied to a constrained optimization problem [37]. Here we use previously defined exponential transformation on **α** and **β** and then use gradient ascent. Specifically, we maximize log-likelihood with respect to u_k = logα_k and v_l = logβ instead to α_k and β_l. As a result, the new optimization problem becomes unconstrained. Derivatives of log-likelihood function and updates of α's and β in gradient ascent can be computed as

$$u_k = \log \alpha_k, \quad v_l = \log \beta_l$$

$$u_k^{new} = u_k^{old} + \eta \frac{\partial L}{\partial u_k}, \quad \frac{\partial L}{\partial u_k} = \frac{\partial L}{\partial \log \alpha_k} = \alpha_k \frac{\partial L}{\partial \alpha_k} \tag{4.28}$$

$$v_k^{new} = v_k^{old} + \eta \frac{\partial L}{\partial v_l}, \quad \frac{\partial L}{\partial v_l} = \frac{\partial L}{\partial \log \beta_l} = \beta_l \frac{\partial L}{\partial \beta_l},$$

where η is is the learning rate.

The negative log-likelihood is a convex function of parameters **α** and **β** and its optimization leads to globally optimal solution. To show that negative log-likelihood is convex, let us derive negative log-likelihood from (4.8). Negative log-likelihood is equal to

$$-\log P = -E + \log \int_{\mathbf{y}} e^E d\mathbf{y}, \tag{4.29}$$

where E is defined in (4.9). Logarithm of an integral of exponential is convex if E is concave [43]. E is linear function of parameters **α** and **β** which means it can be considered as concave and convex at the same time. Therefore, negative likelihood is convex function with respect to parameters **α** and **β**. Exponential function is bijective so the change of variables to **u** and **v** does not affect convexity.

4.3.5. Inference

In inference, since the model is Gaussian, the prediction will be expected value, which is equal to the mean **μ** of the distribution,

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x}) = \mathbf{\Sigma b} . \qquad (4.30)$$

Vector **μ** is a point estimate that maximizes $P(\mathbf{y}|\mathbf{x})$, while **Σ** is a measure of uncertainty. The simplicity of inference that can be achieved using matrix computations is in stark contrast to a general CRF model defined in (4.1) that usually requires advanced inference approaches such as Markov Chain Monte Carlo or belief propagation. Moreover, by exploiting the sparsity of precision matrix Q, which is inherent to spatio-temporal data, the inference can be performed without the need to calculate **Σ** explicitly which reduces computational time to even linear with the dimensionality of **y** (depends on the level of sparsity).

4.3.6. Computational Complexity and Memory Requirements

If size of the training set is N and gradient ascent lasts T iterations, the straightforward matrix computation results in $O(T{\cdot}N^3)$ time to train the model. The main cost of computation is matrix inversion, since during the gradient-based optimization we need to find **Σ** as an inverse of $\mathbf{\Sigma}^{-1}$. However, this is the worst case performance. Since matrix $\mathbf{\Sigma}^{-1}$ is typically very sparse (it depends on the imposed neighborhood structure), the training time can be decreased to $O(T{\cdot}N^2)$.

Let us assume that $\mathbf{\Sigma}^{-1}$ is sparse. Instead of calculating $\mathbf{A}=\mathbf{\Sigma}{\cdot}\mathbf{d\Sigma}^{-1}$ directly, we can solve the system $\mathbf{\Sigma}^{-1}{\cdot}\mathbf{A}= \mathbf{d\Sigma}^{-1}$. To solve the system, we first convert $\mathbf{\Sigma}^{-1}$ to a banded matrix using the Reverse Cuthill-McKee algorithm [44]. A banded matrix is a sparse matrix with all elements around the main diagonal. Then, we apply the Cholesky transformation to $\mathbf{\Sigma}^{\mathbf{-1}}$ and we get $\mathbf{\Sigma}^{\mathbf{-1}}=\mathbf{LL}^{\mathbf{T}}$, where **L** is banded lower triangular matrix. For each column c_i of $\mathbf{d\Sigma}^{\mathbf{-1}}$ we solve $\mathbf{L}v_i=c_i$ for v_i and then $\mathbf{L}^{\mathbf{T}}a_i=v_i$, for a_i, where a_i is the i-th column of **A**. This process needs to be repeated for each column of **A**. The total computation time depends on the neighborhood structure of the interaction potential in Gaussian CRF. For example, [45] indicates that time scaling is $O(T{\cdot}N^{3/2})$ if the neighborhood is spatial and $O(T{\cdot}N^2)$ if it is spatio-temporal. As we eventually need to calculate the trace of matrix **A**, only element from a_i that corresponds to the main diagonal should be stored. Therefore, memory requirements are proportional to $O(N)$.

4.3.7. An extension of the GCRF by Indicator Functions

Multiple predictors R_k, may have large variability in prediction accuracy depending on the underlying conditions. For example, in aerosol application a certain algorithm might be preferred over specific land surfaces while it might underperform elsewhere. This issue can be addressed by enhancing the feature functions (4.6) and (4.7) as

$$\begin{gathered} f_{km}(y_i,\mathbf{x}) = -\delta_m(\mathbf{x})(y_i - R_k(\mathbf{x}))^2, k = 1,...K \\ g_{lm}(y_i, y_j, \mathbf{x}) = -\delta_m(\mathbf{x})e_{ij}^{(l)}S_{ij}^{(l)}(\mathbf{x})(y_i - y_j)^2, l = 1,...L, \end{gathered} \tag{4.31}$$

where δ_m are *the indicator functions* that have value 1 if some conditions are satisfied and 0 otherwise. The effect of indicator functions on model (4.8) is in replacing α_k with sum $\Sigma_m\delta_m(\mathbf{x})\alpha_{mk}$ and β_l with sum $\Sigma_m\delta_m(\mathbf{x})\beta_{ml}$. By introducing indicator functions we essentially partition the whole data set into smaller subsets. $\boldsymbol{\alpha}$ represents belief in R_k in different subsets, corresponding to different prediction conditions. $\boldsymbol{\beta}$ represents level of correlation in different subsets.

4.3.8. An extension of the GCRF to Handle Missing Observations and Partially Labeled Outputs

By utilizing previously defined indicator functions, Gaussian CRF can be extended to handle missing observations unlike Gaussian process based approaches [46] that deal with missing observations externally either by ignoring such data or by interpolating the missing observations. We introduce a special type of indicator function from (4.31), where δ_k = 1 if all observations needed to apply R_k are available and δ_k = 0 otherwise. In this case, precision matrix **Q** might become singular as we are decreasing the values on main diagonal and **Q** may not be strictly diagonally dominant any more. In order to avoid potential numerical problems with **Q**, we introduce a constant function R_0 that, for example, predicts mean value of **y** on labeled data and is always active (δ_0 is fixed to 1).

We also propose an extension of GCRF if **y** is partially labeled, namely if part of observations is labeled ($\mathbf{y_L}$) and another part is unlabeled ($\mathbf{y_U}$). Having the Gaussian CRF model with joint probability ($\mathbf{y_L}$,$\mathbf{y_U}$) from (4.8), where $\boldsymbol{\mu}=[\boldsymbol{\mu}_\mathbf{L}^T\ \boldsymbol{\mu}_\mathbf{U}^T]^T$ and $\mathbf{Q}=[\mathbf{Q_{LL}}\ \mathbf{Q_{LU}};\ \mathbf{Q_{UL}}\ \mathbf{Q_{UU}}]$, we can calculate the conditional probability of unlabeled data as

$$P(\mathbf{y_U}|\mathbf{y_L},\mathbf{x}) \sim Gaussian(\boldsymbol{\mu}^*,\mathbf{Q}^*)$$

$$\text{where } \boldsymbol{\mu}^* = \boldsymbol{\mu}_\mathbf{U} + \mathbf{Q_{UU}}^{-1}\,\mathbf{Q_{UL}}\,(\mathbf{y_L} - \boldsymbol{\mu}_\mathbf{L}),\ \mathbf{Q}^* = \mathbf{Q_{UU}}. \qquad (4.32)$$

This directly solves the problem of inference. It should be observed that the conditional probability distribution utilizes all the available information for prediction. This approach is closely related to semi-supervised learning on graphs using Gaussian random fields [47]. In [47] a Gaussian random field is built on the top of the partially labeled graph. All available information was utilized by conditioning unobserved variables on observed ones. GCRF approach is different in the sense that it naturally estimates parameters along with utilizing unlabeled data.

4.3.9. An extension of GCRF to Discrete Outputs

Relaxation of discrete outputs to continuous output space may bring many appealing properties [47]. Discrete output models require approximate learning and inference approaches (such as Markov Chain Monte Carlo) on structures which are more complex than one dimensional chain. In contrast to such models, GCRF offers closed form solution for learning and inference on any structure through matrix multiplications. Furthermore, defining correlations directly on discrete outputs may introduce unnecessary noise to the model in many applications like action tracking in social networks [48]. As it is suggested in [48] to avoid introducing unnecessary noise it may be beneficial to define correlations on a latent continuous variable space **z** that follows GCRF.

Without losing generalization we assume that y_i are discrete binary outputs (the approach can be easily extended to multi-label outputs). Then, we assume each y_i is conditionally independent given z_i and we assign each y_i to a continuous latent variable z_i as

$$P(y_i \mid z_i, \mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - z_i)^2}{2\sigma^2}} . \tag{4.33}$$

σ can be learned along with other parameters of the model or set by cross-validation. Continuous **z** follows GCRF

$$P(\mathbf{z} \mid \mathbf{x}) = \frac{1}{(2\pi)^{N/2} \left|\mathbf{Q}^{-1}\right|^{1/2}} \exp(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^{\mathbf{T}} \mathbf{Q}(\mathbf{z} - \boldsymbol{\mu})) . \tag{4.34}$$

The joint conditional probability density function of **y** and **z** can be defined as

$$P(\mathbf{y}, \mathbf{z} \mid \mathbf{x}) = P(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) P(\mathbf{z} \mid \mathbf{x}) = \frac{1}{(2\pi)^N \mid \mathbf{Q}_{\mathbf{joint}}^{-1} \mid^{1/2}} e^{-\frac{1}{2}(\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_{\mathbf{y}}\\ \boldsymbol{\mu}_{\mathbf{z}}\end{bmatrix})^T \mathbf{Q}_{\mathbf{joint}} (\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_{\mathbf{y}}\\ \boldsymbol{\mu}_{\mathbf{z}}\end{bmatrix})} , \tag{4.35}$$

where

$$\mathbf{Q}_{\mathbf{joint}} = \begin{bmatrix} \mathbf{Q}_{\mathbf{yy}} & \mathbf{Q}_{\mathbf{yz}} \\ \mathbf{Q}_{\mathbf{zy}} & \mathbf{Q}_{\mathbf{zz}} \end{bmatrix}$$

$$\mathbf{Q}_{\mathbf{yy}ij} = \begin{cases} \frac{1}{2\sigma^2}, i = j \\ 0, i \neq j \end{cases} , \quad \mathbf{Q}_{\mathbf{yz}ij} = \mathbf{Q}_{\mathbf{zy}ij} = \begin{cases} -\frac{1}{2\sigma^2}, i = j \\ 0, i \neq j \end{cases} , \quad \mathbf{Q}_{\mathbf{zz}} = \mathbf{Q} + \mathbf{Q}_{\mathbf{yy}} , \tag{4.36}$$

and

$$\begin{bmatrix} \boldsymbol{\mu}_{\mathbf{y}} \\ \boldsymbol{\mu}_{\mathbf{z}} \end{bmatrix} = \mathbf{Q}_{\mathbf{joint}}^{-1} \begin{bmatrix} \mathbf{b}_{\mathbf{y}} \\ \mathbf{b}_{\mathbf{z}} \end{bmatrix} , \tag{4.37}$$

where $\mathbf{b}_{\mathbf{y}} = 0$ and $\mathbf{b}_{\mathbf{z}}$ is defined in (4.14).

In order for $P(\mathbf{y},\mathbf{z}|\mathbf{x})$ to be valid we need to prove that $\mathbf{Q}_{\mathbf{joint}}$ is positive definite matrix. We observe that $\mathbf{Q}_{\mathbf{joint}}$ is [2*N* × 2*N*] block matrix. Let $\mathbf{S}_{\mathbf{yy}}$ be the Schur complement of $\mathbf{Q}_{\mathbf{yy}}$ in $\mathbf{Q}_{\mathbf{joint}}$, namely $\mathbf{S}_{\mathbf{yy}} = \mathbf{Q}_{\mathbf{zz}} - \mathbf{Q}_{\mathbf{yz}}^{\mathrm{T}} \mathbf{Q}_{\mathbf{yy}}^{-1} \mathbf{Q}_{\mathbf{yz}}$. $\mathbf{Q}_{\mathbf{joint}}$ is positive definite if and only if $\mathbf{Q}_{\mathbf{yy}}$ and $\mathbf{S}_{\mathbf{yy}}$ are both positive definite [49]. $\mathbf{Q}_{\mathbf{yy}}$ is positive definite as a diagonal matrix with all positive elements on a main diagonal. Knowing that $\mathbf{Q_{yz} = - Q_{yy}}$, and $\mathbf{Q_{zz}} = \mathbf{Q} + \mathbf{Q_{yy}}$ the Schur complement becomes $\mathbf{S_{yy}} = \mathbf{Q} + \mathbf{Q_{yy}} - \mathbf{Q_{yy}Q_{yy}}^{-1}\mathbf{Q_{yy}} = \mathbf{Q} + \mathbf{Q_{yy}} - \mathbf{Q_{yy}} = \mathbf{Q}$ which is positive definite from (4.34). Therefore $\mathbf{Q_{joint}}$ is positive definite.

Parameters of the joint distribution with a latent variable can be learned by the Expectation-Maximization (EM) algorithm [50]. EM algorithm starts with an initial guess of parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and then iteratively updates them by an expectation (E-step) and maximization (M-step) until convergence. In the E-step the EM algorithm finds an estimate of the posterior probability of **z** given **y**, **x** and all parameters. If the current parameter estimates are $\boldsymbol{\alpha}^{\mathbf{old}}$ and $\boldsymbol{\beta}^{\mathbf{old}}$, the E-step can be defined as

$$E(\mathbf{z} \mid \mathbf{y}, \mathbf{x}, \boldsymbol{\alpha}^{\mathbf{old}}, \boldsymbol{\beta}^{\mathbf{old}}) = \boldsymbol{\mu}_{\mathrm{z}} - \mathbf{Q}_{\mathbf{zz}}^{-1}\mathbf{Q}_{\mathbf{zy}}(\mathbf{y} - \boldsymbol{\mu}_{\mathrm{y}}). \tag{4.38}$$

The EM algorithm updates parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in M-step to maximize

$$(\boldsymbol{\alpha}^{\mathbf{new}}, \boldsymbol{\beta}^{\mathbf{new}}) = \arg\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} E_{\mathbf{z}|\mathbf{y}}(\log P(\mathbf{y}, \mathbf{z} \mid \mathbf{x}, \boldsymbol{\alpha}^{\mathbf{old}}, \boldsymbol{\beta}^{\mathbf{old}})). \tag{4.39}$$

The conditional expectation can be expressed as (for the simplicity we will assume that **x**, $\boldsymbol{\alpha}^{\mathbf{old}}$ and $\boldsymbol{\beta}^{\mathbf{old}}$ are known and we will omit them from the equations)

$$\begin{aligned}
&E_{\mathbf{z}|\mathbf{y}}(\log P(\mathbf{y}, \mathbf{z})) = \\
&= E_{\mathbf{z}|\mathbf{y}}(-\frac{1}{2}(\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_{\mathrm{y}}\\ \boldsymbol{\mu}_{\mathrm{z}}\end{bmatrix})^T \mathbf{Q}_{\mathbf{joint}}(\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_{\mathrm{y}}\\ \boldsymbol{\mu}_{\mathrm{z}}\end{bmatrix}) - \frac{1}{2}\log|\mathbf{Q}_{\mathbf{joint}}{}^{-1}| \\
&= E_{\mathbf{z}|\mathbf{y}}(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_{\mathrm{y}})^T \mathbf{Q}_{\mathbf{yy}}(\mathbf{y} - \boldsymbol{\mu}_{\mathrm{y}}) - \frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_{\mathrm{y}})^T \mathbf{Q}_{\mathbf{yz}}(\mathbf{z} - \boldsymbol{\mu}_{\mathrm{z}}) \qquad (4.40) \\
&\quad - \frac{1}{2}(\mathbf{z} - \boldsymbol{\mu}_{\mathrm{z}})^T \mathbf{Q}_{\mathbf{zy}}(\mathbf{y} - \boldsymbol{\mu}_{\mathrm{y}}) - \frac{1}{2}(\mathbf{z} - \boldsymbol{\mu}_{\mathrm{z}})^T \mathbf{Q}_{\mathbf{zz}}(\mathbf{z} - \boldsymbol{\mu}_{\mathrm{z}})) \\
&- \frac{1}{2}\log|\mathbf{Q}_{\mathbf{joint}}{}^{-1}|
\end{aligned}$$

$$= -\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}_{\mathrm{y}})^T \mathbf{Q}_{\mathbf{yy}}(\mathbf{y}-\boldsymbol{\mu}_{\mathrm{y}}) - \frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}})^T \mathbf{Q}_{\mathbf{yz}}(E(\mathbf{z}\,|\,\mathbf{y})-\boldsymbol{\mu}_{\mathbf{z}})$$

$$-\frac{1}{2}(E(\mathbf{z}\,|\,\mathbf{y})-\boldsymbol{\mu}_{\mathbf{z}})^T \mathbf{Q}_{\mathbf{yz}}(\mathbf{y}-\boldsymbol{\mu}_{\mathrm{y}}) - \frac{1}{2}E_{\mathbf{z}|\mathbf{y}}((\mathbf{z}-\boldsymbol{\mu}_{\mathbf{z}})^T \mathbf{Q}_{\mathbf{zz}}(\mathbf{z}-\boldsymbol{\mu}_{\mathbf{z}}))$$

$$-\frac{1}{2}\log|\mathbf{Q}_{\mathbf{joint}}{}^{-\mathbf{1}}|.$$

We have

$$E_{\mathbf{z}|\mathbf{y}}((\mathbf{z}-\boldsymbol{\mu}_{\mathbf{z}})^T \mathbf{Q}_{\mathbf{zz}}(\mathbf{z}-\boldsymbol{\mu}_{\mathbf{z}})) =$$

$$= E_{\mathbf{z}|\mathbf{y}}(\mathbf{z}^T\mathbf{Q}_{\mathbf{zz}}\mathbf{z}) - \boldsymbol{\mu}_{\mathbf{z}}{}^T\mathbf{Q}_{\mathbf{zz}}E(\mathbf{z}\,|\,\mathbf{y}) - E(\mathbf{z}\,|\,\mathbf{y})^T\mathbf{Q}_{\mathbf{zz}}\boldsymbol{\mu}_{\mathbf{z}} + \boldsymbol{\mu}_{\mathbf{z}}{}^T\mathbf{Q}_{\mathbf{zz}}\boldsymbol{\mu}_{\mathbf{z}}. \tag{4.41}$$

Then by using linear algebra apparatus we get

$$\begin{aligned}
E_{\mathbf{z}|\mathbf{y}}(\mathbf{z}^T\mathbf{Q}_{\mathbf{zz}}\mathbf{z}) &= Tr(E_{\mathbf{z}|\mathbf{y}}(\mathbf{z}^T\mathbf{Q}_{\mathbf{zz}}\mathbf{z})) \\
&= E_{\mathbf{z}|\mathbf{y}}(Tr(\mathbf{Q}_{\mathbf{zz}}\mathbf{z}\mathbf{z}^T)) \\
&= Tr(\mathbf{Q}_{\mathbf{zz}}E_{\mathbf{z}|\mathbf{y}}(\mathbf{z}\mathbf{z}^T)) \qquad (4.42)\\
&= Tr(\mathbf{Q}_{\mathbf{zz}}(\mathbf{Q}_{\mathbf{zz}}{}^{-1} + E(\mathbf{z}\,|\,\mathbf{y})E(\mathbf{z}\,|\,\mathbf{y})^T)) \\
&= N + E(\mathbf{z}\,|\,\mathbf{y})\mathbf{Q}_{\mathbf{zz}}E(\mathbf{z}\,|\,\mathbf{y})^T.
\end{aligned}$$

Since N is constant and does not depend on parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ we can omit it. If we replace (4.42) in (4.41) and then (4.41) in (4.40) we will get

$$(\boldsymbol{\alpha}^{\mathbf{new}}, \boldsymbol{\beta}^{\mathbf{new}}) =$$

$$= \arg\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} (-\frac{1}{2}(\begin{bmatrix} \mathbf{y} \\ E(\mathbf{z}\,|\,\mathbf{y}) \end{bmatrix} - \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{y}} \\ \boldsymbol{\mu}_{\mathbf{z}} \end{bmatrix})^T \mathbf{Q}_{\mathbf{joint}}(\begin{bmatrix} \mathbf{y} \\ E(\mathbf{z}\,|\,\mathbf{y}) \end{bmatrix} - \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{y}} \\ \boldsymbol{\mu}_{\mathbf{z}} \end{bmatrix}) \tag{4.43}$$

$$-\frac{1}{2}\log|\mathbf{Q}_{\mathbf{joint}}{}^{\mathbf{-1}}|).$$

(4.43) is identical to (4.20) and we can apply the same gradient ascent technique to find optimal values of **α** and **β**. EM algorithm iterates between (4.38) and (4.43) until convergence.

Inference can be done as in [47], [48]. Since **y** and **z** are unknown, we can first calculate marginal expectation for **z**. To find discrete **y** we find an average value of **z** over positive examples z^1 and an average value of **z** over negative examples z^0. We decide about the **y** as

$$y_i = \begin{cases} 0, |z_i - z^0| < |z_i - z^1| \\ \\ 1, otherwise. \end{cases} \tag{4.44}$$

4.3.10. Related Work Revisited: Spatial Statistics

Given spatial random variable **y** kriging can be represented as multivariate Gaussian distribution *Gaussian*(**μ**,**Σ**) where **Σ** is a covariance matrix that accounts for spatial correlation [51]. When **μ** is constant, we get the ordinary kriging, or Gaussian process (GP) model. In universal kriging **μ** is a linear function of attributes **x**, **μ** = $\boldsymbol{\beta}^{\mathbf{T}}\mathbf{x}$, where **β** is a parameter vector. Learning consists of fitting covariance kernel for **Σ** and learning **β**. In inference universal kriging provide predictions for x_i given a set of neighboring locations where x_j are known. The main difference between the GCRF and universal kriging is that universal kriging parameterizes covariance matrix directly which is usually full while GCRF parameterizes inverse covariance matrix which is usually very sparse. Therefore, GCRF through the linear algebra apparatus permits learning and inference on much larger data than kriging does. Also, during inference partial information from neighboring locations cannot be handled directly by universal kriging. As we showed in Section 4.3.8, GCRF can handle partial observations by using indicator functions.

Markov random field (MRF) and the related conditional autoregressive (CAR) and simultaneous autoregressive (SAR) models [52] are generative approaches that represent posterior through joint probability $P(\mathbf{y}|\mathbf{x}) \sim P(\mathbf{y},\mathbf{x}) = P(\mathbf{x}|\mathbf{y})P(\mathbf{y})$ by modeling prior $P(\mathbf{y})$ and conditional $P(\mathbf{x}|\mathbf{y})$. Correlation between **y** is modeled by $P(\mathbf{y})$ as MRF. For computational tractability, $P(\mathbf{x}|\mathbf{y})$ is often assumed to have a factorized form $\mathrm{P}(\mathbf{x}|\mathbf{y}) = \Pi_\mathrm{i}\mathrm{P}(x_i|y_i)$. MRF has several important limitations when compared to GCRF. The first is limited representational power due to the conditional independence assumption for $P(\mathbf{x}|\mathbf{y})$. GCRF models $P(\mathbf{y}|\mathbf{x})$ directly without any assumptions about inputs **x**. Second, $P(\mathbf{y})$ does not depend on inputs **x**. This means that correlations between outputs are data independent. On the other hand, GCRF models correlations that can vary with input data providing more flexibility than MRF.

4.3.11. Limitations of Proposed Approach and Future Directions

Learning of GCRF parameters brings set of challenges. The simple gradient ascent algorithm that we used to learn parameters of GCRF is acceptable if the number of

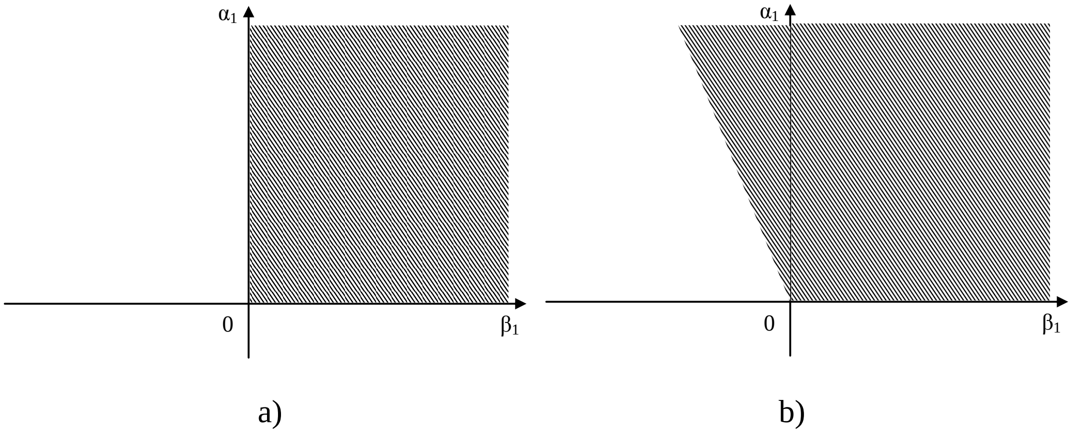


Figure 4.2. Valid parameter space for two dimensional precision matrix a) by diagonal dominance criterion b) by definition of positive definite matrix.

parameters in GCRF is small. In learning of GCRF with large number of parameters standard gradient ascent algorithm will have extremely slow convergence. In future work we will consider the conjugate gradient approach that was successfully applied in many similar problems [53]. Another issue to be addressed is that simultaneously learning of all may lead to serious overfitting. Although we suggested L_2-norm we may need to explore other approaches like L_1-norm regularization [54].

A diagonal dominance criterion, which is satisfied for GCRF if all parameters are larger than 0, is sufficient but not necessary condition for positive definites. It is also know that diagonal dominance criterion is too conservative [45]. Let us examine on a simple example how restrictive diagonal dominance criterion is. We assume model with only two outputs. Two dimensional precision matrix can be parameterized as

$$Q = \begin{bmatrix} \alpha_1 + \beta_1 & -\beta_1 \\ -\beta_1 & \alpha_1 + \beta_1 \end{bmatrix}, \tag{4.45}$$

which is in agreement with parameterization of our GCRF model. Diagonal dominance criterion gives us conditions for positive definites $\alpha_1 + \beta_1 > 0$ and $\beta_1 > 0$. This implies that the parameter search space is restricted to $\alpha_1 > 0$ and $\beta_1 > 0$. On the other hand, sufficient and necessary conditions for positive definites give us $\alpha_1 + \beta_1 > 0$ and $(\alpha_1 + \beta_1)^2 - \beta_1^2 > 0$. Valid parameter spaces are presented in Figure 4.2. Diagonal dominance criterion reduces the search so that negative interactions between outputs are not allowed. This condition may be too strong in some applications (for example social networks). Instead of relying on diagonal dominance criteria we will use convex optimization apparatus to perform searches in broader parameter space [55].

4.4. Gaussian Conditional Random Fields for Regression in Aerosol Prediction

4.4.1. GCRF Model for AOD Prediction

In the following we describe in detail the proposed CRF for regression in remote sensing, using the AOD prediction as the motivating example. As we showed in Chapter 3 given a data set that consists of satellite observations and ground based AOD measurements, a statistical prediction model (SP) can be trained to use satellite observations as attributes and predict the labels which are ground-based AODs. We also showed in Chapter 2 that the deterministic AOD prediction models (DP) are based on solid physical principles and tuned by domain scientists. GCRF model is able to ingrate

SP and DP through the association potential. To model the association potential, i.e. the dependency between the predictions and target AOD, we introduce two feature functions

$$\begin{aligned} f_1(y_i, \mathbf{x_i}) &= -(y_i - SP(\mathbf{x_i}))^2 \\ f_2(y_i, \mathbf{x_i}) &= -(y_i - DP(\mathbf{x_i}))^2, \end{aligned} \tag{4.46}$$

where for a given observation $\mathbf{x}_i$, SP($\mathbf{x}_i$) and DP($\mathbf{x}_i$) are outputs of statistical and deterministic models, respectively. These feature functions follow the basic principle for association potentials (their values are larger for more accurate predictions). Learned parameters $\boldsymbol{\alpha}$ of the linear combination of these features

$$A(\boldsymbol{\alpha}, y_i, \mathbf{x_i}) = -\alpha_1(y_i - SP(\mathbf{x_i}))^2 - \alpha_2(y_i - DP(\mathbf{x_i}))^2, \tag{4.47}$$

provide some insight on how much to trust the SP and DP prediction algorithms. For example, large α_1 places large penalty on mistakes of SP model and is an indicator of large quality of this predictor.

To improve expressiveness of the CRF model we introduce various indicator functions. Here are some examples of possible indicator functions

$$\begin{aligned} \delta_1(\mathbf{x_i}) &= \begin{cases} 1, \text{if } \mathbf{x_i} \text{ belongs to NorthAmerica} \\ 0, \text{otherwise} \end{cases} \\ \delta_2(\mathbf{x_i}) &= \begin{cases} 1, \text{if } \mathbf{x_i} \text{ is a data point of high quality} \\ 0, \text{otherwise.} \end{cases} \end{aligned} \tag{4.48}$$

Association potential now becomes

$$A(\boldsymbol{\alpha}, y_i, \mathbf{x_i}) = \sum_{j=1}^{J} (\alpha_{1j}\delta_j(\mathbf{x_i})(y_i - SP(\mathbf{x_i}))^2 + \alpha_{2j}\delta_j(\mathbf{x_i})(y_i - DP(\mathbf{x_i}))^2), \tag{4.49}$$

where *J* is number of indicator functions. By introducing indicator functions we essentially partition the whole data set into smaller subsets. Learned **α** represents our belief in SP and DP in different subsets, corresponding to different prediction conditions.

To model the interaction potential we introduce feature function

$$g_1(y_i, y_j, \mathbf{x}) = -w_{ij}(y_i - y_j)^2 . \tag{4.50}$$

In AOD prediction problem data are irregularly sampled in both space and time. Weight w_{ij} is positive number representing a measure of spatio-temporal proximity between data points *i* and *j* (closer points are given larger weight). The corresponding interaction potential is

$$I(\beta, y_i, y_j, \mathbf{x}) = -\beta w_{ij}(y_i - y_j)^2 . \tag{4.51}$$

4.4.2. Experimental results and discussion

4.4.2.1. Data

For this experiment we collected MODIS Terra observations collocated with AERONET Level 2.0 points. In addition to average and standard deviation of radiances at four wavelengths in 50 × 50 km^2 blocks, solar and sensor angles, and surface elevation we extracted information about the spatio-temporal location of each data point (time, longitude and latitude) and a quality of observation (QA) assigned to each point provided by domain scientist. There are four levels of qualities from lowest quality QA = 0 to highest quality QA = 3. We collected 28,374 data points distributed over entire globe at 217 AERONET sites during years 2005 and 2006.

4.4.2.2. Evaluation

To assess the efficiency of the proposed methods, we performed training on 2005 data and used 2006 data for testing. Because we trained neural network (NN) on 2005 data and then use its predictions as inputs to CRF, we applied a nested cross-validation. First, we split AERONET locations into 5 subsets and created five data sets D_i, $i = 1,\ldots 5$, each with data points from one of the AERONET subsets in year 2005. We reserved one of D_i datasets for testing and merged data from the remaining 4 datasets D_j, $j \neq i$, for training. The trained NN predictor was tested on D_i. The procedure was repeated five times, for values $j = 1,\ldots 5$. Finally, we get five NN models and NN predictions for all points in training set.

4.4.2.3. Benchmark Methods

Deterministic prediction algorithm C005. The primary benchmark for comparison with our predictors was the most recent version of the MODIS deterministic algorithm C005.

Statistical prediction by neural network. As a baseline statistical algorithm we used a neural network trained to predict AERONET AOD from all MODIS attributes except location information and quality flag. The neural network has a hidden layer with 10 nodes and an output layer with one node. In nested 5-cross-validation experiments we trained 5 neural networks. When tested on 2006 data, we used a single network trained on the whole training set.

4.4.2.4. The GCRF model

Integration of models. We first consider the case when interaction potential does not exist (β = 0). NN and C005 predictions are inputs to CRF. We partitioned the world into five regions: North America, South America, Europe, Africa, and Asia and Australia. Asia and Australia were treated together due to the small number of data points in each of them. Then, we defined five indicator functions. Each function indicates belonging to one of five regions. We determined ten α parameters corresponding to C005 and NN predictions over these regions. Results are presented in Table 4.1. Over all regions GCRF achieved better accuracy than either NN or C005 alone. Values of obtained α parameters suggest that we should trust NN more in the North America (ratio of *α's* is NN:C005 = 24:13 approximately) while in Africa we should trust C005 a little bit more (ratio of *α's* is NN:C005 = 8:9 approximately). Also, GCRF improves domain-based accuracy measure *FRAC* (Table 4.2).

Second, we check how much we should rely on NN and C005 over observations with different qualities. We partitioned data into four subsets having quality flags QA = 0, 1, 2, and 3. We introduced four indicator functions to indicate belonging to each of subsets. We determined eight α parameters corresponding to C005 and NN predictions over these subsets. Results are presented in Table 4.3. For all data qualities GCRF achieved better accuracy than either NN or C005 alone. As expected, error of the deterministic predictor C005 decreases as data quality increases. Values of obtained α parameters also suggest that we should trust NN more for low data quality QA = 0 (ratio of *α's* is NN:C005 = 21:10 approximately) while for high data quality we should equally trust to C005 and NN (ratio of *α's* is NN:C005 = 16:16 approximately). *FRAC* is also improved by GCRF, Table 4.4.

Integration of spatio-temporally correlated models. Here we consider the case when interaction potential does exist ($\beta \neq 0$). NN and C005 predictions are inputs to GCRF. To model interaction potential we need to define weights w_{ij} in (4.51). After analysis of spatial and temporal AOD autocorrelation (Figure 4.3) we decided to define spatial-temporal neighbors as a pair of observations where temporal distance *temporalDist*(*i*,*j*) is less than 60 days and spatial distance *spatialDist*(*i*,*j*) is less than 100 km. As a measure for temporal distance *temporalDist* we used absolute difference between timestamps t_i and t_j *temporalDist*(*i*,*j*) = $|t_i - t_j|$. As a measure for spatial distance *spatialDist* we used haversine formula [56] that gives a distance along great circle between two points specified by longitudes and latitudes on a sphere. We used weighted distance for defining w_{ij}, weights are multiplication of Gaussians

$$w_{ij} = \begin{cases} e^{-\frac{spatialDist(i,j)^2}{2\sigma_s^2}-\frac{temporalDist(i,j)^2}{2\sigma_t^2}}, i \sim j \\ 0, otherwise, \end{cases} \tag{4.52}$$

where $\sigma_s = 50$ and $\sigma_t = 10$ were determined based on spatial and temporal correlation.

Taking into account spatio-temporal correlation and comparing to the GCRF model with ($\beta = 0$) when the world was partitioned into five regions, we get better results globally and over all regions separately except Africa where two models were equally good and Asia&Australia where the latter model was better (Table 4.1). This result suggests that level of spatio-temporal correlation is different in different regions, and each region should have its own β. β was estimated to 0.049, which does not indicate significant correlation, but it is still enough to improve single-output based predictors.

Including spatial-temporal correlation in the model when data were partitioned based on quality also improves final prediction (Table 4.3), β was estimated to 0.06.

Table 4.1. *RMSE* of C005, NN, and NN+GCRF using features defined over five regions, without ($\beta = 0$) and with spatio-temporal correlation ($\beta \neq 0$).

Region	**RMSE**			
	C005	**NN**	**GCRF, *β=0***	**GCRF, *β≠0***
Whole Globe	0.123	0.112±0.001	0.107±0.001	**0.105±0.001**
N. America	0.098	0.085±0.001	0.083±0.001	**0.081±0.001**
S. America	0.140	0.110±0.005	0.104±0.003	**0.098±0.002**
Europe	0.080	0.080±0.001	0.073±0.001	**0.072±0.001**
Africa	0.172	0.154±0.001	0.152±0.001	**0.149±0.001**
Asia & Aus.	0.161	0.156±0.001	**0.145±0.001**	0.148±0.001

Table 4.2. *FRAC* of C005, NN, and NN+GCRF using features defined over five regions, without ($\beta = 0$) and with spatio-temporal correlation ($\beta \neq 0$).

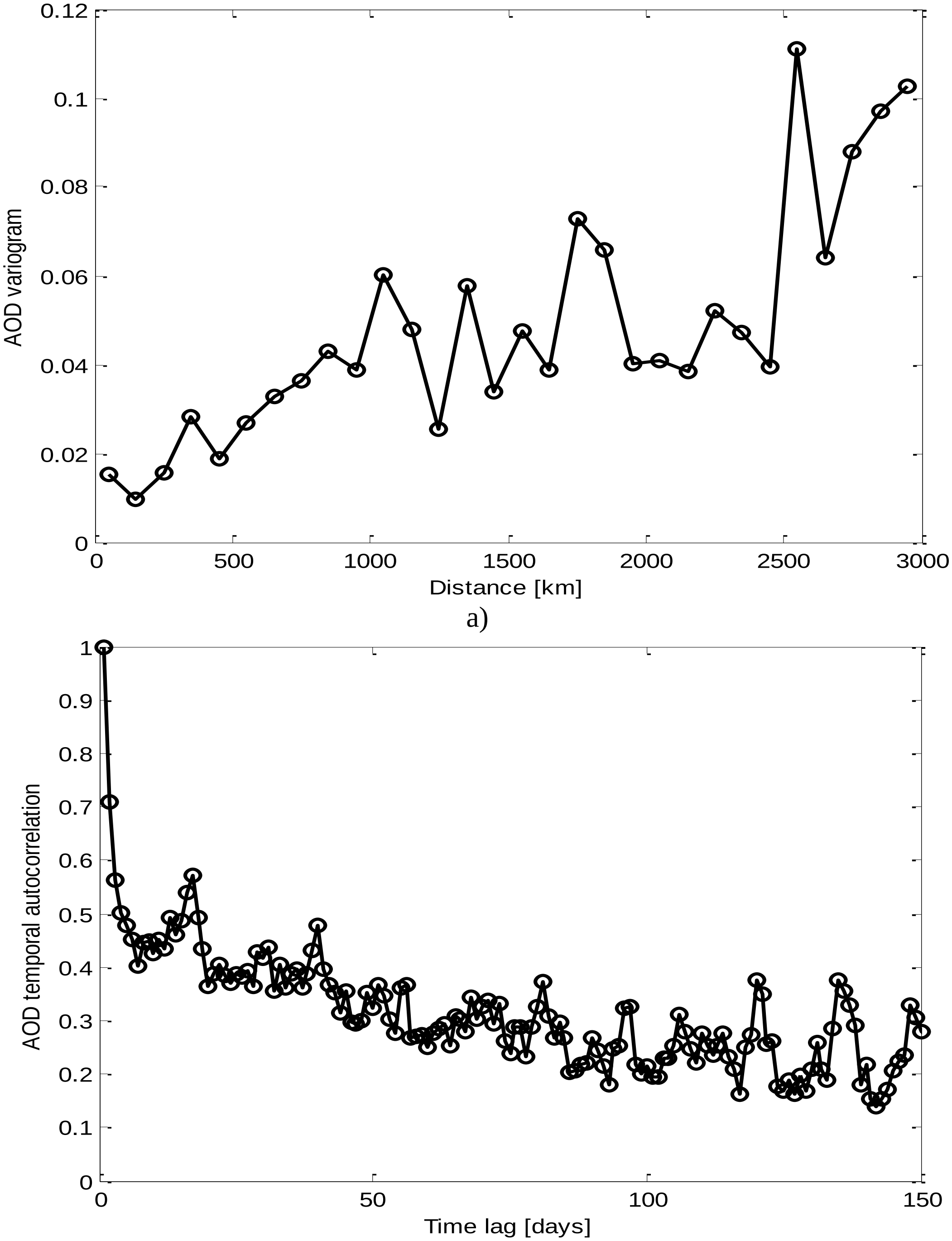
Region	**FRAC**			
	C005	**NN**	**GCRF, *β=0***	**GCRF, *β≠0***
Whole Globe	0.65	0.667±0.002	0.704±0.003	**0.708±0.004**
N. America	0.64	0.667±0.008	0.71±0.01	**0.71±0.01**
S. America	0.55	0.56±0.01	0.58±0.02	**0.60±0.02**
Europe	0.76	0.762±0.005	0.807±0.006	**0.812±0.006**
Africa	0.53	0.560±0.006	0.568±0.007	**0.577±0.006**
Asia & Aus.	0.64	0.66±0.01	**0.71±0.01**	0.70±0.01

Table 4.3. *RMSE* of C005, NN, and NN+GCRF using features defined over four subsets of data of different quality (QA = 0 lowest, QA = 3 highest), without ($\beta = 0$) and with spatio-temporal correlation ($\beta \neq 0$).

Region	**RMSE**			
	C005	**NN**	**GCRF, *β=0***	**GCRF, *β≠0***
Entire set	0.123	0.112±0.001	0.107±0.001	**0.105±0.001**
QA = 0	0.151	0.128±0.002	0.123±0.001	**0.121±0.001**
QA = 1	0.130	0.108±0.001	0.109±0.001	**0.107±0.002**
QA = 2	0.118	0.110±0.002	0.104±0.001	**0.101±0.001**
QA = 3	0.105	0.104±0.002	0.097±0.001	**0.096±0.001**

Table 4.4. *FRAC* of C005, NN, and NN+GCRF using features defined over four subsets of data of different quality (QA = 0 lowest, QA = 3 highest), without ($\beta = 0$) and with spatio-temporal correlation ($\beta \neq 0$).

Data quality	**FRAC**			
	C005	**NN**	**GCRF, *β=0***	**GCRF, *β≠0***
Entire set	0.65	0.667±0.002	0.705±0.004	**0.709±0.004**
QA = 0	0.59	0.60±0.01	**0.64±0.01**	**0.64±0.01**
QA = 1	0.58	0.623±0.006	0.65±0.01	**0.652±0.005**
QA = 2	0.64	0.65±0.01	0.686±0.007	**0.689±0.007**
QA = 3	0.70	0.714±0.007	0.755±0.003	**0.761±0.002**



a)

b)

Figure 4.3. AOD a) spatial variogram; b) temporal autocorrelation.

4.5. Neural Gaussian Conditional Random Fields

In this section we further enhance GCRF model to increase its representational power.

4.5.1. Adaptive Feature Function

The expert predictors R_k in feature functions of the GCRF model are developed externally, before model learning. For example, the neural network used in Section 4.3 was trained by minimizing mean squared error (*MSE*), which is not necessarily an optimal strategy with respect to maximizing the log-likelihood of the Gaussian CRF. Motivated by the recently proposed Conditional Neural Fields [41], we will consider using the adaptive feature function defined as

$$f_a(y_i, \mathbf{x}) = -(y_i - R_a(\mathbf{x}, \mathbf{w}))^2 \tag{4.53}$$

where $R_a(\mathbf{x},\mathbf{w})$ is a function of weights **w** that can be trained simultaneously with other GCRF parameters.

$R_a(\mathbf{x},\mathbf{w})$ will be trained directly with the goal of maximizing the log-likelihood such that it complements the existing predictors R_k. Let us assume that predictor $R_a(\mathbf{x},\mathbf{w})$ is a feed-forward neural network. Training of the network, by keeping other CRF parameters constant, can be done by maximizing the log-likelihood of GCRF. To apply a gradient-based method for learning weights **w**, we need to find the gradient of the conditional log-likelihood $\partial \log R/\partial R_a$ and use it in back-propagation algorithm to learn **w**.

4.5.2. Parameters **α** and **β** as Functions

As defined in (4.5), Gaussian CRF assigns weights **α** and **β** to the feature functions. Considering that feature functions for the association potential are defined as squared errors of unstructured predictors, the role of weights **α** is to measure their prediction uncertainty. Since it is likely that the quality of different predictors changes with **x**, we enhance GCRF such that parameters α_k and β_l are replaced with the uncertainty functions $\alpha_k(\boldsymbol{\theta}_\mathbf{k}, \mathbf{x})$ and $\beta_l(\boldsymbol{\psi}_\mathbf{l}, \mathbf{x})$, where $\boldsymbol{\theta}_\mathbf{k}$ and $\boldsymbol{\psi}_\mathbf{l}$ are the parameters. We allow using feed-forward neural networks for the uncertainty functions.

$\alpha_k(\boldsymbol{\theta}_\mathbf{k}, \mathbf{x})$ models the varying degree of importance of predictor R_k over different conditions. Similarly, $\beta_l(\boldsymbol{\psi}_\mathbf{l}, \mathbf{x})$ models varying importance of correlation between outputs. As a result, **Σ** from becomes dependent on inputs thus allowing for error heteroscedasticity (error depends on **x**). Conditional distribution of the enhanced GCRF is still Gaussian as in (4.16).

Since both adaptive feature and uncertainty functions are assumed to be feed-forward neural networks, we call the resulting model the *Neural GCRF*.

4.5.3. Learning and Inference in Neural GCRF

For the Neural GCRF (NGCRF) we find parameters (**θ**, **ψ**, **w**) by maximizing the log-likelihood. To ensure feasibility of the model, we apply an exponential transformation on **α** and **β** parameters as

$$\alpha_k = e^{\mathrm{u}_k(\boldsymbol{\theta}_k, \mathbf{x})}, \text{for } k = 1,...K \qquad \alpha_a = e^{u_a(\theta_a, \mathbf{x})} \qquad \beta_l = e^{v_l(\boldsymbol{\psi}_l, \mathbf{x})}, \tag{4.54}$$

Table 4.5. Pseudocode for NGCRF learning.

```
1. Learn ($\boldsymbol{\theta}_\mathbf{k}$, $\boldsymbol{\psi}_\mathbf{l}$) $k$ = 1,…$K$, $l$ = 1,…$L$ not taking into account $R_a$
2. Initialize $\boldsymbol{\theta}_a$
3. Repeat until convergence
    3.1. Learn predictor $R_a$ using (4.43)
    3.2. Apply gradient-based optimization to learn $\boldsymbol{\theta}$
```

where u and v are differentiable functions with respect to parameters $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. To apply the gradient-based method for learning, we need to find the gradient of the conditional log-likelihood. The derivatives of log$P$ with respect to $\boldsymbol{\theta}$, $\boldsymbol{\psi}$, and $\mathbf{w}$ are

$$\begin{aligned} \frac{\partial \log P}{\partial \theta_k} &= \frac{\partial \log P}{\partial \alpha_k}\frac{\partial \alpha_k}{\partial u_k}\frac{\partial u_k}{\partial \theta_k} \\ \frac{\partial \log P}{\partial \psi_l} &= \frac{\partial \log P}{\partial \beta_l}\frac{\partial \beta_l}{\partial v_l}\frac{\partial v_l}{\partial \psi_l} \quad . \\ \frac{\partial \log P}{\partial w} &= \frac{\partial \log P}{\partial R_a}\frac{\partial R_a}{\partial w} \end{aligned} \tag{4.55}$$

Terms ∂log$P/\partial\alpha_k$ and ∂log$P/\partial\beta_l$ are defined in (4.26) and (4.27). As $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are functions of $\mathbf{x}$, these derivatives are diagonal matrices. From (4.55), it follows $\partial\alpha_k/\partial u_k=\alpha_k$ and $\partial\beta_l/\partial v_l=\beta_l$. Terms $\partial u_k/\partial\theta_k$ and $\partial v_l/\partial\psi_l$ depend on the chosen functions u_k and v_l.

The gradient of log$P$ with respect to $\mathbf{w}$ depends on the functional form of R_a. Since $\boldsymbol{\Sigma}^{-1}$ does not depend on R_a, ∂log$P/\partial R_a$ becomes

$$\frac{\partial \log P}{\partial R_a} = 2\boldsymbol{\alpha}_a^T(\mathbf{y}-\boldsymbol{\mu})\,. \tag{4.56}$$

We observe that an update for the adaptive predictor R_a is proportional to the difference between true output and the mean of NGCRF distribution. This means that R_a will be updated only if NGCRF is not able to predict the output correctly and R_a will be updated more aggressively when the error is larger. This justifies our hypothesis that R_a will work as a complement of the existing unstructured predictors.

We propose the iterative procedure in Table 4.5 for learning model parameters according to update equations derived earlier in this section.

To avoid overfitting, which is a common problem for maximum likelihood optimization, we add regularization terms for $\boldsymbol{\alpha}$, $\boldsymbol{\theta}$, β, $\boldsymbol{\psi}$ to the log-likelihood. In this way, we penalize large outputs of $\boldsymbol{\alpha}$ and β as well as large weights $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$.

Since the NGCRF model is Gaussian, inference is identical to GCRF.

4.5.4. Experimental Results and Discussion

4.5.4.1. Data

For this experiment we the same data as in Section 4.3. In addition to average and standard deviation of radiances at four wavelengths in 50 × 50 km^2 blocks, solar and sensor angles, and surface elevation we extracted information about the spatio-temporal location of each data point (time, longitude and latitude) and a quality of observation (QA) assigned to each point provided by domain scientist. The QA flag has four possible values, from the lowest quality QA = 0 to the highest quality QA = 3. Our data set had 28,374 observations collected during 2005 and 2006 at 217 AERONET sites over the globe.

4.5.4.2. Benchmark Methods

Deterministic prediction algorithm (C005). The primary benchmark for comparison with our GCRF predictors was the most recent version of the MODIS operational algorithm C005.

Statistical prediction by a neural network (NN). As a baseline statistical algorithm we used a neural network model trained to predict AOD from MODIS observations excluding information about location and QA flag. The neural network was trained to minimize the mean squared error of the AOD prediction. It has been shown previously that neural network achieves higher accuracy than C005. The neural network had a hidden layer with 10 nodes and an output layer with one node. In the nested 5-cross-validation experiments we trained 5 neural networks on 2005 data. When tested on 2006 data, we used a single network trained on the entire training set.

Structured prediction by GCRF. We use the GCRF model defined in Section 4.3 Association potential utilizes C005 and NN. C005 and NN are the deterministic prediction algorithm and neural network predictor defined above. In addition, we partitioned data into four subsets corresponding to quality flags QA = 0, 1, 2, and 3. For unstructured predictors *C005* and *NN*, we created feature functions over these subsets by enhancing GCRF with the indicator functions that have value 1 if some condition QA is satisfied and 0 otherwise. Spatial-temporal neighbors defined as a pair of observations within certain spatio-temporal are used to define the interaction potential.

4.5.4.3. The Neural GCRF Model for Aerosol Prediction

In addition to the feature functions of GCRF explained in Section 4.3, Neural GCRF used an adaptive feature function with prediction model (R_a), being a neural network with 10 hidden nodes. Its weight α_a followed the definition in (4.57).

We also used functions instead of parameters $\boldsymbol{\alpha}$. Instead of defining manual partitions of the dataset using the QA flag, we used all observations as inputs to the $\boldsymbol{\alpha}$ functions. We defined $\boldsymbol{\alpha}$ as an exponential function of linear combinations of inputs. To incorporate the potential bias, one input was vector with all ones.

$$\alpha_k(\theta, x^i) = e^{\sum \theta_l x_l^i}, \tag{4.57}$$

where $\mathbf{x}_1^{\mathrm{i}}$ was a vector with all ones, $\mathbf{x}_{2,3,4,5}^{\mathrm{i}}$ were quality flags.

To model spatio-temporal correlation, we used spatial and temporal distance between *i* and *j* as two inputs for the β function. Similar to (4.57), we defined β as

$$\beta(\psi, x^{i,j}) = e^{\sum \psi_l x_l^{i,j}}, \tag{4.58}$$

where $\mathbf{x}_1^{\mathrm{i,j}}$ was a vector with all ones, $\mathbf{x}_2^{\mathrm{i,j}}$ represented spatial distance between *i* and *j* and $\mathbf{x}_3^{\mathrm{i,j}}$ represented their temporal proximity.

4.5.4.4. Results

RMSE error of four models is presented in Table 4.6, where smaller numbers mean more accurate predictions. *FRAC* accuracy of these four models is also shown in Table 4.6 where larger numbers correspond to better predictions. We can see that in our experiments NN was more accurate than operational C005 algorithm. GCRF showed an improvement in accuracy over both NN and C005, by taking advantage of combination of

Table 4.6. *RMSE* and *FRAC* of C005, NN, GCRF and NGCRF on data with four quality flags Vs AERONET prediction accuracy.

	C005	NN	GCRF	NGCRF
RMSE	0.123	0.112	0.105	**0.102**
FRAC	0.65	0.68	0.71	**0.74**

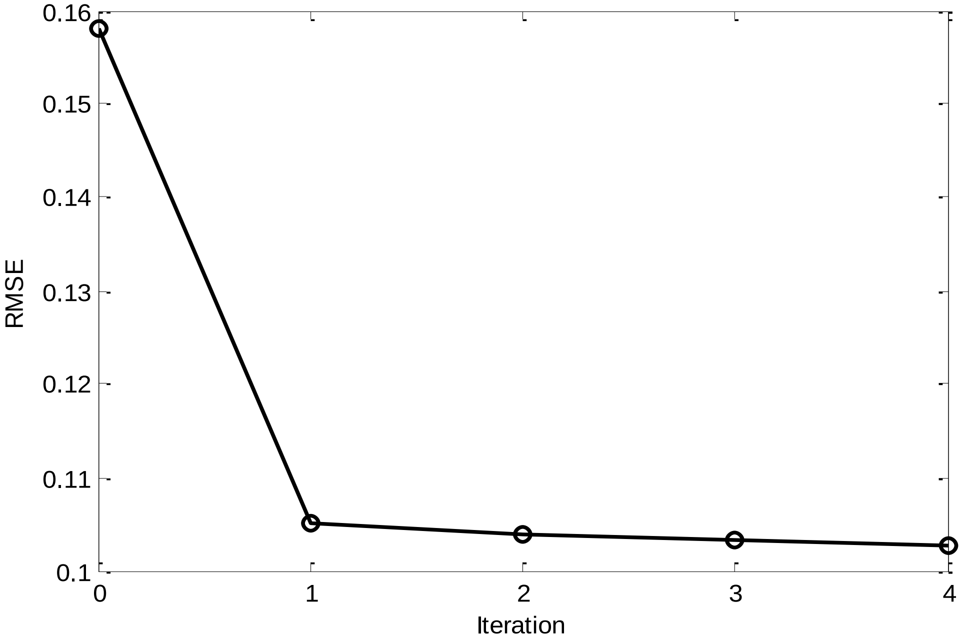


Figure 4.4. NGCRF convergence

models and spatio-temporal correlation in data. Finally, Neural GCRF achieved even better accuracy by utilizing nonlinear weights, an adaptive statistical model, and learning instead of assuming the level of correlation between neighboring observations. Convergence of NGCRF was achieved after a small number of iterations as illustrated in Figure 4.3 where it is shown how the *RMSE* accuracy changed on training data during iterative learning over 4 iterations.

CONCLUSION

Structured learning, as a fairly new research area in machine learning, had great success in classification, but its application on regression problems has not been explored sufficiently. In this paper, we proposed a Gaussian CRF model that is able to combine the outputs of unstructured regression models, such as pre-trained neural networks or other available predictors, and exploit the correlation between output variables. The proposed neural GCRF extends the GCRF model, by training an additional neural network to further improve accuracy by maximizing the log likelihood of the GCRF model and by introducing uncertainty functions that can account for changing quality of the baseline predictors as a function of inputs.

The proposed method was applied to a challenging remote sensing problem of predicting aerosols from satellite-based observations. The obtained results provide strong evidence that the GCRF and Neural GCRF can be successfully applied to the remote sensing problem where a small improvement of prediction quality could be very beneficial to many geophysical studies that rely on AOD predictions. The proposed method is also readily applicable to other regression applications where there is a need for knowledge integration, data fusion, and exploitation of correlation among output variables.

BIBLIOGRAPHY

[1] G. Pitari et al., “Aerosols, their Direct and Indirect Effects,” in *Climate Change 2001: The Scientific Basis. Contribution of Working Group I to the Third Assessment Report of the Intergovernmental Panel on Climate Change*, vol. 5, J. T. Houghton et al., Eds. Cambridge University Press Cambridge, UK, 2001, pp. 289-348.

[2] V. Radosavljevic, S. Vucetic, and Z. Obradovic, “Aerosol Optical Depth Retrieval by Neural Networks Ensemble with Adaptive Cost Function,” in *Proceedings of the 10th Int’l Conf. Engineering Applications of Neural Networks*, 2007, pp. 266-275.

[3] V. Radosavljevic, S. Vucetic, and Z. Obradovic, “Spatio-Temporal Partitioning for Improving Aerosol Prediction Accuracy,” in *Proceedings of the Eighth SIAM International Conference on Data Mining*, 2008, pp. 609-620.

[4] V. Radosavljevic, S. Vucetic, and Z. Obradovic, “Reduction of ground-based sensor sites for spatio-temporal analysis of aerosols,” in *Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data - SensorKDD’09*, 2009, pp. 71-78.

[5] V. Radosavljevic, S. Vucetic, and Z. Obradovic, “A Data-Mining Technique for Aerosol Retrieval Across Multiple Accuracy Measures,” *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 2, pp. 411-415, Apr. 2010.

[6] Z. Wang, V. Radosavljevic, B. Han, Z. Obradovic, and S. Vucetic, “Aerosol Optical Depth Prediction from Satellite Observations by Multiple Instance Regression,” in *Proceedings of the Eighth SIAM International Conference on Data Mining*, 2008, pp. 165-176.

[7] D. Das, V. Radosavljevic, S. Vucetic, and Z. Obradovic, “Reducing Need for Collocated Ground and Satellite based Observations in Statistical Aerosol Optical Depth Estimation,” in *Proceedings of the IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, 2008, p. II-879-II-882.

[8] G. Jun, J. Ghosh, V. Radosavljevic, and Z. Obradovic, “Predicting Ground-based Aerosol Optical Depth with Satellite Images Via Gaussian Processes,” in

Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, 2010, pp. 370-375.

[9] V. Radosavljevic, S. Vucetic, and Z. Obradovic, "Continuous Conditional Random Fields for Regression in Remote Sensing," in *Proceeding of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, 2010, pp. 809-814.

[10] C. Elachi and J. Van Zyl, *Introduction to the Physics and Techniques of Remote Sensing*. Wiley Interscience, 2006.

[11] Y. J. Kaufman, D. Tanré, L. A. Remer, E. F. Vermote, A. Chu, and B. N. Holben, "Operational remote sensing of tropospheric aerosol over land from EOS moderate resolution imaging spectroradiometer," *Journal of Geophysical Research*, vol. 102, no. 14, pp. 17051-17067, 1997.

[12] P. Y. Deschamps et al., "The POLDER mission: instrument characteristics and scientific objectives," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 3, pp. 598-615, 1994.

[13] D. F. Heath, A. J. Krueger, H. A. Roeder, and B. D. Henderson, "The Solar Backscatter Ultraviolet and Total Ozone Mapping Spectrometer /SBUV/TOMS/ for Nimbus G," *Optical Engineering*, vol. 14, no. 4, pp. 323-331, 1980.

[14] H. R. Gordon and M. Wang, "Retrieval of water-leaving radiance and aerosol optical thickness over the oceans with SeaWiFS: a preliminary algorithm," *Applied Optics*, vol. 33, no. 3, pp. 443-52, 1994.

[15] D. T. Llewellyn-Jones, P. J. Minnett, R. W. Saunders, and A. M. Zavody, "Satellite multi-channel infra-red measurements of sea surface temperature of the N E Atlantic Ocean using AVHRR/2," *Q J R Meteorol Soc*, vol. 110, pp. 613-631, 1984.

[16] D. M. Winker et al., "Overview of the CALIPSO Mission and CALIOP Data Processing Algorithms," *Journal of Atmospheric and Oceanic Technology*, vol. 26, no. 11, pp. 2310-2323, 2009.

[17] B. N. Holben et al., "AERONET—A Federated Instrument Network and Data Archive for Aerosol Characterization," *Remote Sensing of Environment*, vol. 66, no. 1, pp. 1-16, 1998.

[18] L. A. Remer, D. Tanré, and Y. J. Kaufman, "Algorithm for Remote Sensing of Tropospheric Aerosol from MODIS : Collection 005 : Revision 2," 2009. [Online]. Available: http://modis-

atmos.gsfc.nasa.gov/_docs/ATBD_MOD04_C005_rev2.pdf. [Accessed: 11-Nov-2011].

[19] "Official MODIS website." [Online]. Available: http://modis.gsfc.nasa.gov/. [Accessed: 11-Nov-2011].

[20] N. J. D. Nagelkerke, "A note on a general definition of the coefficient of determination," *Biometrika*, vol. 78, no. 3, pp. 691-692, 1991.

[21] F. Aires, C. Prigent, W. B. Rossow, and M. Rothstein, "A new neural network approach including first guess for retrieval of atmospheric water vapor , cloud liquid water path , surface temperature , and emissivities over land," *Journal of Geophysical Research*, vol. 106, no. 14, pp. 14887-14907, 2001.

[22] M. D. Müller, "Ozone profile retrieval from Global Ozone Monitoring Experiment (GOME) data using a neural network approach (Neural Network Ozone Retrieval System (NNORSY))," *Journal of Geophysical Research*, vol. 108, no. 16, 2003.

[23] B. Han, S. Vucetic, A. Braverman, and Z. Obradovic, "A statistical complement to deterministic algorithms for the retrieval of aerosol optical thickness from radiance data," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 7, pp. 787-795, Oct. 2006.

[24] G. E. P. Box and D. R. Cox, "An Analysis of Transformations," *Journal of the Royal Statistical Society Series B Methodological*, vol. 26, no. 2, pp. 211-252, 1964.

[25] S. Vucetic and Z. Obradovic, "Discovering Homogeneous Regions in Spatial Data through Competition," in *Proceedings of the Seventeenth International Conference Machine Learning ICML 00*, 2000, pp. 1095-1102.

[26] S. Vucetic, K. Tomsovic, and Z. Obradovic, "Discovering price-load relationships in California's electricity market," *IEEE Transactions on Power Systems*, vol. 16, no. 2, pp. 280-286, May 2001.

[27] L. Anselin, *Spatial Econometrics: Methods and Models*. Kluwer Academic, 1988, p. 289.

[28] S. Shekhar, P. R. Schrater, R. R. Vatsavai, and S. Chawla, "Spatial contextual classification and prediction models for mining geospatial data," *IEEE Transactions on Multimedia*, vol. 4, no. 2, pp. 174-188, 2002.

[29] M. Seeger, Y. Teh, and M. Jordan, "Semiparametric Latent Factor Models," in *Workshop on Artificial Intelligence and Statistics*, 2005, vol. 10, pp. 333-340.

[30] M. Goulard and M. Voltz, "Linear coregionalization model: Tools for estimation and choice of cross-variogram matrix," *Mathematical Geology*, vol. 24, no. 3, pp. 269-286, 1992.

[31] L. Bo and C. Sminchisescu, "Twin Gaussian Processes for Structured Prediction," *International Journal of Computer Vision*, vol. 87, no. 1-2, pp. 28-52, 2009.

[32] A. H. S. Solberg, T. Taxt, and A. K. Jain, "A Markov random field model for classification of multisource satellite imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 1, pp. 100-113, 1996.

[33] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proceedings of the 18th International Conference on Machine Learning*, 2001, vol. 18, pp. 282-289.

[34] S. Kumar and M. Hebert, "Discriminative Random Fields," *International Journal of Computer Vision*, vol. 68, no. 2, pp. 179-201, 2006.

[35] Y. Liu, J. Carbonell, J. Klein-Seetharaman, and V. Gopalakrishnan, "Comparison of probabilistic combination methods for protein secondary structure prediction," *Bioinformatics*, vol. 20, no. 17, pp. 3099-3107, 2004.

[36] M. Kim and V. Pavlovic, "Discriminative learning for dynamic state prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1847-1861, 2009.

[37] T. Qin, T.-Y. Liu, X.-D. Zhang, D.-S. Wang, and H. Li, "Global Ranking Using Continuous Conditional Random Fields," in *Proceedings of NIPS'08*, 2008, vol. 21, pp. 1281-1288.

[38] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman, "Learning Gaussian Conditional Random Fields for Low-Level Vision," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, vol. C, no. 14, pp. 1-8, 2007.

[39] T.-minh-tri Do and T. Artieres, "Neural conditional random fields," in *Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, vol. 9, pp. 177-184.

[40] F. Zhao, J. Peng, and J. Xu, "Fragment-free approach to protein folding using conditional neural fields," *Bioinformatics*, vol. 26, no. 12, p. i310-i317, 2010.

[41] J. Peng, L. Bo, and J. Xu, "Conditional Neural Fields," in *Advances in Neural Information Processing Systems NIPS'09*, 2009, vol. 9, pp. 1-9.

[42] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 10, no. 8. The Johns Hopkins University Press, 1996, p. 48.

[43] L. Paninski, “Log-concavity results on Gaussian process methods for supervised and unsupervised learning,” in *Advances in Neural Information Processing Systems NIPS’05*, 2005, vol. 17, pp. 1025-1032.

[44] E. Cuthill and J. McKee, “Reducing the bandwidth of sparse symmetric matrices,” *Proceedings of the 1969 24th national conference*, vol. 1972, no. 2, pp. 157-172, 1969.

[45] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, vol. 48, no. 1. Chapman & Hall/CRC, 2005, p. 263 p.

[46] N. Cressie and G. Johannesson, “Fixed rank kriging for very large spatial data sets,” *Journal Royal Statistical Society Series B Statistical Methodology*, vol. 70, no. 1, pp. 209-226, 2008.

[47] X. Zhu, Z. Ghahramani, and J. Lafferty, “Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions,” *Machine Learning*, vol. 20, no. 2, p. 912, 2003.

[48] C. Tan, J. Tang, J. Sun, Q. Lin, and F. Wang, “Social action tracking via noise tolerant time-varying factor graphs,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD’10*, 2010, p. 1049.

[49] R. W. Cottle, “Manifestations of the Schur complement,” *Linear Algebra and its Applications*, vol. 8, no. 1, pp. 189-211, 1974.

[50] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood estimation from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Association*, vol. 39, no. 1, pp. 1-38, 1977.

[51] N. A. C. Cressie, *Statistics for spatial data*, vol. 2. Wiley, 1993, p. 900.

[52] J. Besag, “Spatial interaction and the statistical analysis of lattice systems,” *Journal of the Royal Statistical Society Series B Methodological*, vol. 36, no. 2, pp. 192-236, 1974.

[53] C. E. Rasmussen and C. Williams, “Gaussian Processes for Machine Learning,” *International Journal of Neural Systems*, vol. 14, no. 2, pp. 69-106, 2006.

[54] C. Sutton and A. McCallum, “1 An Introduction to Conditional Random Fields for Relational Learning,” *Introduction to statistical relational learning*, no. x, p. 93, 2007.

[55] J. Songsiri, J. Dahl, and L. Vandenberghe, “Graphical models of autoregressive processes,” in *Convex Optimization in Signal Processing and Communications*, Cambridge University Press, 2009, pp. 89-116.

[56] R. W. Sinnott, “Virtues of the Haversine,” *Sky and Telescope*, vol. 68, no. 2, p. 159, 1984.