Big Data Algorithms for Visualization and Supervised Learning

Nemanja Djuric
Temple University, December 2\textsuperscript{nd}, 2013
Remote sensing

- Djuric, N., Lakesh, K., Vucetic, S., Semi-Supervised Learning for Integration of Aerosol Predictions from Multiple Satellite Instruments, *IJCAI 2013 (Outstanding paper award)*

Large-scale learning

- Djuric, N., Grbovic, M., Vucetic, S., Distributed Confidence-Weighted Classification on MapReduce, *IEEE BigData 2013*

Memory-constrained online learning

- Djuric, N., Vucetic, S., Random Kernel Perceptron on ATTiny2313 Microcontroller, *SensorKDD @ KDD 2010*
Traffic state estimation and prediction


Label ranking

- Grbovic, M., Djuric, N., Vucetic S., Supervised Clustering of Label Ranking Data, *SDM* 2012 (*Best of SDM*)
Bioinformatics


- Lan, L., Djuric, N., Guo, Y., Vucetic, S., MS-kNN: Protein Function Prediction by Integrating Multiple Data Sources, *BMC Bioinformatics* 2012 (*Top performing team*)

Unsupervised object matching


Large data visualization

- Djuric, N., Vucetic S., Efficient Visualization of Large-scale Data Tables through Reordering and Entropy Minimization, *ICDM* 2013
Introduction

- Big Data!
  - Big Data is pervasive; data sets with millions of examples and features are now a rule rather than an exception
  - Crowdsourcing, remote sensing, social networks, etc.

- Globally-recognized, strategic importance of Big Data
  - Focus of major internet companies
  - “Big Data Research and Development Initiative” by the US government
“Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization.”

Many challenges to machine learning and data mining researchers

- Standard tools and frameworks are not capable of addressing new tasks
- Even linear time and space complexity may no longer be tractable
Outline of the presentation

- Large data visualization
  - EM-ordering and TSP-means

- Large-scale learning
  - Adaptive Multi-hyperplane Machines
  - BudgetedSVM: A C++ toolbox for large-scale learning
  - Distributed confidence-weighted learning on MapReduce

- Unsupervised object matching
  - Convex Kernelized Sorting

- Combination of experts
  - Semi-supervised aggregation of noisy experts
  - Using Gaussian CRF to improve aerosol retrieval and traffic estimation
Data visualization

- Immediate feedback that can lead to faster knowledge discovery
  - Intuitive way of interacting with unknown data

- Long history of visualization tools, characterized by slow progress in recent years
  - New visualization approaches are required in order to tackle modern large-scale problems

- Our task: Visualizing large data matrices
How to visualize a data matrix?

- Classical approaches
  - Pie and bar charts, histograms

- Parallel coordinates

Images:
- Classical approaches: Pie and bar charts, histograms
- Parallel coordinates

Years:
- 1883
- 1897
- 2005
How to visualize a data matrix?

- Classical approaches
  - Heat maps

- Idea: Data reordering
  - Reorder matrix so similar rows/columns are grouped together
The proposed method: Example

- Waveform benchmark data set
- Please click here: EM-ordering example
Data reordering can be considered from the viewpoint of data compression
- Reorder the data so that it is maximally compressible
- Assume data set $D = \{x_i, i = 1, \ldots, n\}$ is given, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$ are $m$-dimensional examples

**Differential Predictive Coding (DPC)**
- Use local context to code the value of $x_i$

$$D = \{x_i, i = 1, \ldots, n\} \rightarrow D_{DPC} = \{x_1, \varepsilon_2, \ldots, \varepsilon_n\}$$

where $\varepsilon_i = (x_i - x_{i-1}), i = 2, \ldots, n$

* Djuric, N., Vucetic, S., Efficient Visualization of Large-scale Data Tables through Reordering and Entropy Minimization, ICDM 2013*
**EM-ordering**

- Entropy of the prediction errors used to estimate compressibility

\[
H(\varepsilon) = \frac{n}{2} (m \cdot \log(2\pi) + \sum_{j=1}^{m} \log(\sigma_j(\varepsilon))) + 0.5 \sum_{i=2}^{n} \sum_{j=1}^{m} \frac{(x_{\pi(i),j} - x_{\pi(i-1),j})^2}{\sigma_j^2}
\]

- The optimization problem becomes

\[
(\pi^*, \{\sigma_1^*, ..., \sigma_m^*\}) = \arg \min_{\pi, \{\sigma_1, ..., \sigma_m\}} H(\varepsilon)
\]

- The EM-ordering algorithm

  1. Fix variance of prediction errors, then minimize the overall distance between neighbors in the ordering (equivalent to TSP)
  2. Fix ordering, then find variance of the prediction errors
Super-quadratic time complexity of the best TSP solvers is prohibitive on large data.

We propose an $O(n \log(n))$ method, called TSP-means.

After creating $2^l$-tree through recursive runs of $k$-means, solve TSP defined on each node while traversing the tree breath-first.
Real-world applications

- Minneapolis traffic data set

Original data set

Reordered data set

Locations of the sensors
Classification is one of the fundamental machine learning tasks.

Big data brings new challenges:
- What to do when faced with data sets with millions data points and/or features?

Traditional non-linear classifiers such as kernel SVM are inefficient in this setting, unlike linear models:
- Can we combine accuracy of kernel SVMs with efficiency of linear models?
Assume a data set $D = \{(x_n, y_n), n = 1, \ldots, N\}$, where $x_n$ is a feature vector, and $y_n$ is one of $M$ class labels.

Multi-class SVM (Crammer et al., JMLR 2001)
- Model is parameterized by $M$ weight vectors $w_i$
- Prediction is given as

$$f(x) = \arg \max_{i \in Y} g(i, x), \text{ where } g(i, x) = w_i^T x$$

- By concatenating all weights $w_i$, we can write

$$W = [w_1 \ w_2 \ \ldots \ w_M]$$
Multi-class SVM training

We find the weights by minimizing the following problem

\[ \frac{\lambda}{2} \| \mathbf{W} \|_F^2 + \frac{1}{N} \sum_{n=1}^{N} l(\mathbf{W}; (x_n, y_n)) \]

where \( l(\mathbf{W}; (x_n, y_n)) = \max \left( 0, 1 + \max_{i \in \mathcal{Y} \setminus y_n} g(i, x_n) - g(y_n, x_n) \right) \)

This model was extended (Aiolli et al., JMLR 2005) by assigning a fixed number of weights to each class

\[ g(i, \mathbf{x}) = \max_j \mathbf{w}_{i,j}^T \mathbf{x}, \text{ with } \mathbf{W} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,b_1} & w_{2,1} & \cdots & w_{2,b_2} & \cdots & w_{M,1} & \cdots & w_{M,b_M} \end{bmatrix} \]
The resulting Multi-hyperplane Machine (MM) loss function is non-convex, and the authors propose to solve the modified loss function

\[
l_{cvx}(W; (x_n, y_n); z_n) = \max \left( 0, 1 + \max_{i \in y \setminus y_n} g(i, x_n) - w^T_{y_n, z_n} x_n \right)
\]

where \( z_n \) is a preset index of a true-class weight.

1. We fix a single true-class hyperplane to each training data point at the beginning of a training epoch.
2. After the whole data set is seen, compute and fix new assignments \( z_n \) and repeat the optimization.
Adaptive Multi-hyperplane Machines

- We proposed Adaptive MM (AMM), which adaptively learns an appropriate number of weights for each class.

- In a nutshell
  - Assign to each class an *infinite* number of zero-weights.
  - Use Stochastic Gradient Descent (SGD) to solve the MM optimization problem.
  - During training, the algorithm finds an appropriate number of weights suitable for the problem complexity.
  - Provided theoretical guarantees of convergence and generalization.

Adaptive MM

- At the \( t \)th training iteration, minimize the instantaneous objective function

\[
P^{(t)}(W|z) \equiv \frac{\lambda}{2} ||W||^2 + l_{cvx}(W; (x_t, y_t); z_t)
\]

- SGD optimization
  - If data point misclassified, true-class weight pushed towards the point and winning other-class weight pushed away

- Points are reassigned to weights after each epoch is completed \( \Rightarrow \) zero weights get assigned to points and will be updated to non-zero in the next epoch (adaptability)
Adaptive MM

- **Theorem 1**
  
  If we denote by $\mathbf{W}^*$ the optimal MM solution, it holds

  \[
  \frac{1}{T} \sum_{t=1}^{T} P(t) (\mathbf{W}^{(t)} | \mathbf{z}) - \frac{1}{T} \sum_{t=1}^{T} P(t) (\mathbf{W}^* | \mathbf{z}) \leq \frac{8(\ln(T) + 1)}{\lambda T}
  \]

- **Theorem 2**
  
  Assume we are able to correctly classify an IID-sampled training set of size $N$, then we can upper bound generalization error with probability greater than $1 - \delta$ as

  \[
  \frac{130}{N} \left( \| \mathbf{W} \|^2 B \log(4eN) \log(4N) + \log\left( \frac{2(2N)^K}{\delta} \right) \right)
  \]

  \[B = \sum_{i=1}^{M} b_i + 1 + b_{\text{max}}^2 - b_{\text{max}} - b_{\text{min}}, \quad K = \frac{1}{2} \sum_{i=1}^{M} b_i \sum_{j \neq i}^{M} b_j, \quad b_{\text{min}} = \min_{i=1, \ldots, M}\{b_i\} \quad \text{and} \quad b_{\text{max}} = \max_{i=1, \ldots, M}\{b_i\}\]
Extensions: Online group lasso

- Enforce group-level sparsity of the weight matrix $W$
  - Online truncated gradient (Langford et al., JMLR 2009)

- Let $w_g$ be a vector representing a $g^{th}$ group of elements from the weight matrix $W$, with $g = 1, ..., G$, then

$$w_g \leftarrow \text{sparsify}(w_g, \eta \gamma) = \begin{cases} w^g - \eta \gamma \frac{w^g}{\|w^g\|} & \|w^g\| > \eta \gamma \\ 0 & \|w^g\| \leq \eta \gamma \end{cases}$$

- Theoretical guarantees of convergence
- Analogous to online variant of group lasso
- Straightforward implementation
Extensions: Growing AMM

- SGD fails to learn good weights for highly non-linear patterns
  - Idea: “Clone” existing non-zero weights when misclassification

- Results on $4 \times 4$ checkerboard data set

- Robust to noise, significantly outperforms AMM
LibSVM, LibLinear, and Vowpal Wabbit are popular software packages for classification.

- LibSVM implements state-of-the-art, yet inefficient kernel SVMs.
- LibLinear and Vowpal Wabbit implement very efficient linear classifiers, however the performance is acceptable only on nearly-linearly-separable data sets.

Is there an easy-to-use software out there that allows efficient, non-linear learning in large-scale setting?
We developed a software package that combines accuracy of kernel SVMs with efficiency of linear SVMs.

- Implements 3 large-scale, multi-class, non-linear classifiers in C++ (AMM, LLSVM, BSGD classifiers).
- Budgeted, accurate, non-linear models trained within minutes on data sets with millions of examples/features.
- Highly-optimized routines and data structures.
- Provides an API for handling large-scale data.

BudgetedSVM

- Comparison with the state-of-the-art classifiers

<table>
<thead>
<tr>
<th>Data set</th>
<th>Pegasos</th>
<th>AMM</th>
<th>LLSVM</th>
<th>BSGD</th>
<th>RBF-SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>err.</td>
<td>time</td>
<td>err.</td>
<td>B</td>
<td>time</td>
</tr>
<tr>
<td>webspam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N = 280,000</td>
<td>7.94</td>
<td>0.5s</td>
<td>4.74</td>
<td>9</td>
<td>3s</td>
</tr>
<tr>
<td>M = 254</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rcv1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N = 677,399</td>
<td>2.73</td>
<td>1.5s</td>
<td>2.39</td>
<td>19</td>
<td>9s</td>
</tr>
<tr>
<td>M = 47,236</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mnist8m-bin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N = 8,000,000</td>
<td>22.71</td>
<td>1.1m</td>
<td>3.16</td>
<td>18</td>
<td>4m</td>
</tr>
<tr>
<td>M = 784</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Available from SourceForge.net
- More than 300 downloads since summer, try it out!
The task is to train a linear binary classifier to separate training data points.

Online Confidence-Weighted (CW) classifier, in addition to the prediction margin, outputs our confidence in the prediction for the incoming test data point.

- Assumes a multivariate Gaussian over linear classifiers.
- Given a trained CW model, this induces a Gaussian distribution over the prediction margin for a new point \((x_t, y_t)\):

\[
\hat{y} \sim \mathcal{N}(y_t(\mu^T x_t), x_t^T \Sigma x_t)
\]
Online training algorithm is derived having in mind the following constraints (Dredze et al., ICML 2008; Crammer et al., NIPS 2009)

- New parameter estimates should be close to those from the previous iteration
- Margin for a new training point should be maximized, while uncertainty minimized

Solve the following optimization problem (AROW):

\[
(\mu_{t+1}, \Sigma_{t+1}) = \arg \min_{\mu, \Sigma} D_{KL}(\mathcal{N}(\mu, \Sigma) || \mathcal{N}(\mu_t, \Sigma_t)) + \\
\lambda_1 \left( \max(0, 1 - y_t \mu^T x_t) \right)^2 + \lambda_2 (x_t^T \Sigma x_t)
\]
CW classification on MapReduce

- Train a single AROW classifier on each of $M$ mappers, aggregate them on reducer.
- On reducer, we minimize the following objective function:

$$
\mathcal{L} = \mathbb{E}_{\mathcal{N}(\mu, \Sigma)}[D_{KL}^{S}(\mathcal{N}(\mu_*, \Sigma_*) \parallel \mathcal{N}(\mu, \Sigma))]
$$

or its empirical estimate:

$$
\mathcal{L} = \sum_{m=1}^{M} \mathbb{P}(\mathcal{N}(\mu_m, \Sigma_m)) \ D_{KL}^{S}(\mathcal{N}(\mu_*, \Sigma_*) \parallel \mathcal{N}(\mu_m, \Sigma_m))
$$

- We can obtain closed-form updates for mean vector and covariance matrix.

* Djuric, N., Grbovic, M., Vucetic, S. Distributed Confidence-Weighted Classification on MapReduce, IEEE BigData 2013
Finding a derivative of the objective function with respect to mean and covariance matrix, we obtain

\[ \mu_\ast = \left( \sum_{m=1}^{M} \left( \mathbb{P}(\mathcal{N}(\mu_m, \Sigma_m)) (\Sigma_{\ast}^{-1} + \Sigma_m^{-1}) \right) \right)^{-1} \left( \sum_{m=1}^{M} \left( \mathbb{P}(\mathcal{N}(\mu_m, \Sigma_m)) (\Sigma_{\ast}^{-1} + \Sigma_m^{-1}) \right) \mu_m \right) \]

\[ \Sigma_\ast \left( \sum_{m=1}^{M} \mathbb{P}(\mathcal{N}(\mu_m, \Sigma_m)) \Sigma_m^{-1} \right) \Sigma_\ast = \sum_{m=1}^{M} \mathbb{P}(\mathcal{N}(\mu_m, \Sigma_m)) \left( \Sigma_m + (\mu_\ast - \mu_m)(\mu_\ast - \mu_m)^T \right) \]

The second equation is an algebraic Riccati equation of the form \( XAX = B \), solved as

\[ X = U^{-0.5} B^{0.5} (U^T)^{-0.5}, \text{ with } A = U^T U \]
Results

- Real-world, industrial-size **Ad Latency** data set
  - 1.3 billion data examples, 21 measured features

- Online advertising domain
  - Improve online experience through timely delivery of relevant ads to the users
  - Can we detect if the ad will be late before it is served?

- Representation
  - **user features** (browser type, device type, ISP, location, connection speed, etc.)
  - **ad features** (ad type, ad size, ad dimensions, etc.)
  - **vendor features** (where is the ad served from, hardware used, etc.)
Results

- We compared AROW-MR to non-distributed AROW, as well as to the state-of-the-art Vowpall Wabbit (VW).
- Increased no. of mappers to evaluate effects of parallelization.

**Table 1. Increasing number of mappers**

<table>
<thead>
<tr>
<th># mappers</th>
<th># reducers</th>
<th>Avg. map time</th>
<th>Reduce time</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>408h</td>
<td>n/a</td>
<td>0.8442</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>30.5h</td>
<td>1 min</td>
<td>0.8552</td>
</tr>
<tr>
<td>500</td>
<td>1</td>
<td>34 min</td>
<td>4 min</td>
<td>0.8577</td>
</tr>
<tr>
<td>1,000</td>
<td>1</td>
<td>17.5 min</td>
<td>7 min</td>
<td>0.8662</td>
</tr>
<tr>
<td>10,000</td>
<td>1</td>
<td>2 min</td>
<td>1h</td>
<td>0.8621</td>
</tr>
</tbody>
</table>

**Table 2. Performance of VW**

<table>
<thead>
<tr>
<th># mappers</th>
<th># reducers</th>
<th>Avg. map time</th>
<th>Reduce time</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>7h</td>
<td>n/a</td>
<td>0.8506</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>1h</td>
<td>n/a</td>
<td>0.8508</td>
</tr>
<tr>
<td>500</td>
<td>0</td>
<td>8 min</td>
<td>n/a</td>
<td>0.8501</td>
</tr>
<tr>
<td>1,000</td>
<td>0</td>
<td>6 min</td>
<td>n/a</td>
<td>0.8498</td>
</tr>
</tbody>
</table>

- AROW-MR decreased training time from 17 days to 25 minutes, with further accuracy gains!
- Outperformed linear VW classifier with comparable training times.
Object matching

- Given two sets $X = \{x_i, i = 1, .., m\}$ and $Y = \{y_i, i = 1, .., m\}$, find one-to-one matching between “similar” objects
  - Problem appears in many areas (bioinformatics, computer vision, natural language processing, ...)
  - Closely related to transfer learning

- Match objects from two different domains, but without an option to compare them
Kernelized Sorting

- Hilbert-Schmidt Independence Criterion (HSIC) measures dependency between sets $X$ and $Y$, empirically estimated as

$$\Delta^2 = m^{-2} \text{trace}(K \cdot L)$$

- If the sets are independent, then HSIC is minimal and equal to 0

- Hence, matching problem is defined as ($\pi$ is a permutation matrix: $\pi_{ij} = 1$ if $x_i$ and $y_j$ match, 0 otherwise)

$$\maximize_{\pi \in \Pi_m} \text{trace}(K \cdot \pi^T \cdot L \cdot \pi)$$
Reformulate KS problem as follows:

- Given two $m \times m$ matrices $K$ and $L$, value of $\text{trace}(KL)$ is maximized if rows of $K$ and columns of $L$ are permuted such that rows of $K$ and corresponding columns of $L$ are identical up to a constant multiplier.

To obtain convex problem, we allow $\pi$ to be doubly-stochastic, and KS becomes:

$$\max_{\pi \in \Pi_m} \text{trace}(K^{\top}L^{\top}) - (L \cdot \pi)^{\top} \cdot (K \cdot \pi)^{\top}$$

CKS

KS

* Djuric, N., Grbovic, M., Vucetic, S., Convex Kernelized Sorting, AAAI 2012
Results

- Match English documents to documents in other languages
  - Baseline matches the documents simply by length
  - Average no. of correct matches after 5 runs reported below (best result given in parentheses)

- CKS consistently outperforms the competing state-of-the-art

<table>
<thead>
<tr>
<th>Language</th>
<th>Corpus size</th>
<th>Baseline</th>
<th>KS</th>
<th>KS-p</th>
<th>LSOM</th>
<th>CKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danish</td>
<td>387</td>
<td>39</td>
<td>261 (318)</td>
<td>258 (273)</td>
<td>159 (173)</td>
<td>379</td>
</tr>
<tr>
<td>Dutch</td>
<td>387</td>
<td>50</td>
<td>266 (371)</td>
<td>237 (317)</td>
<td>146 (375)</td>
<td>383</td>
</tr>
<tr>
<td>Finnish</td>
<td>308</td>
<td>54</td>
<td>19 (32)</td>
<td>22 (38)</td>
<td>10 (10)</td>
<td>114</td>
</tr>
<tr>
<td>French</td>
<td>356</td>
<td>64</td>
<td>319 (356)</td>
<td>320 (334)</td>
<td>354 (354)</td>
<td>356</td>
</tr>
<tr>
<td>German</td>
<td>356</td>
<td>50</td>
<td>282 (344)</td>
<td>258 (283)</td>
<td>338 (350)</td>
<td>356</td>
</tr>
<tr>
<td>Italian</td>
<td>387</td>
<td>49</td>
<td>341 (382)</td>
<td>349 (353)</td>
<td>378 (381)</td>
<td>385</td>
</tr>
<tr>
<td>Portuguese</td>
<td>356</td>
<td>46</td>
<td>308 (354)</td>
<td>326 (343)</td>
<td>342 (356)</td>
<td>356</td>
</tr>
<tr>
<td>Spanish</td>
<td>387</td>
<td>48</td>
<td>342 (365)</td>
<td>351 (364)</td>
<td>386 (387)</td>
<td>387</td>
</tr>
<tr>
<td>Swedish</td>
<td>337</td>
<td>76</td>
<td>20 (39)</td>
<td>20 (33)</td>
<td>5 (5)</td>
<td>97</td>
</tr>
</tbody>
</table>
Aerosols are small particles suspended in the atmosphere, originating from natural and man-made sources.

Estimation of global aerosol distribution is one of the biggest challenges in climate research:
- Negative effect on public health
- Profound effect on Earth’s radiation budget

Standard measure of distribution is Aerosol Optical Depth (AOD):
- Ground-based sensors (AERONET network of instruments)
- Satellite-based sensors aboard Terra, Aqua, Aura, Calipso, SeaStar, and other Earth-observing satellites
Measuring aerosol distribution

- Ground-based sensors (Sun photometers)
  - High accuracy of AOD estimates
  - High cost of installment and maintenance

- Satellite-based sensors
  - Lower accuracy of AOD estimation
  - Global daily coverage
Problem setting

- **OBJECTIVE:** find an optimal combination of available satellite measurements, using scarce AERONET measurements as a ground-truth AOD during training

- We are given training data set consisting of targets $y_i$ (AERONET) and of estimates of $y_i$ by $K$ different experts (satellites), with $N_u$ unlabeled and $N_l$ labeled data points

- Considered approaches
  - Semi-supervised combination of experts
  - Gaussian Conditional Random Fields
Combination of experts: Related work

- Bates and Granger, 1969; Granger and Ramanathan, 1984
  - Supervised method, no missing data allowed
- Raykar et al., 2009; Ristovski et al., 2010
  - Unsupervised methods, no missing data allowed
  - Experts assumed independent
- The proposed semi-supervised method presents a significant generalization of the two approaches
  - Allows missing data, correlated experts, and finds different data-generating regimes
Methodology

- Data points sampled IID, and target follows normal distribution,
  \[ y_i \sim \text{Norm}(\mu_y, \sigma_y^2) \]

- Denote by \( \hat{y}_i \) a \( K \)-dimensional vector of expert predictions, sampled from multivariate Gaussian,
  \[ \hat{y}_i \mid y_i \sim \text{Norm}(y_i \mathbf{1}, \Sigma) \]

- Training task is to find the parameters
  \[ \Theta = \{\Sigma, \mu_y, \sigma_y^2\} \]

* Djuric, N., Kansakar, L., Vucetic, S., Semi-Supervised Learning for Integration of Aerosol Predictions from Multiple Satellite Instruments, IJCAI 2013 (outstanding paper award)
Once the training is completed, aggregated prediction can be found as a mean of the posterior distribution (assuming no missing experts)

\[ y_i \mid \hat{y}_i \sim \text{Norm}(\bar{y}_i, (1^T \Sigma^{-1} 1)^{-1}) \]

where the mean can be computed as follows,

\[ \bar{y}_i = \frac{\hat{y}_i^T \Sigma^{-1} 1}{1^T \Sigma^{-1} 1}, \quad \text{with} \quad \hat{y}_i = [\hat{y}^T, \mu_y]^T \quad \text{and} \quad \Sigma' = \begin{pmatrix} \Sigma & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \]
Training

- Learning by maximizing likelihood of the training data
- We write probability of the training data as follows
  \[
P(D | \Theta) = P(D_u | \Theta) P(D_l | \Theta)
\]
- The probability of unlabeled data set is equal to
  \[
P(D_u | \Theta) = \prod_{i=1}^{N_u} P(\hat{y}_i | \Theta) = \prod_{i=1}^{N_u} P(\hat{y}_i | y, \Theta) P(y | \Theta) dy
  \]
  \[
  = \prod_{i=1}^{N_u} \left( \sqrt{\frac{|\Sigma'|^{-1}}{(2\pi)^{K-1}1^T \Sigma'^{-1} 1}} \right) \exp\left(-\frac{1}{2} (\hat{y}'_i - \bar{y}_i 1)^T \Sigma'^{-1} (\hat{y}'_i - \bar{y}_i 1) \right)
  \]
Further, the probability of labeled data can be written as

$$P(D_l | \Theta) = \prod_{i=N_u+1}^{N} P(\hat{y}_i | y_i, \Theta) = \prod_{i=N_u+1}^{N} \frac{1}{(2\pi)^{K/2} |\Sigma|^{0.5}} \exp(-0.5(\hat{y}_i - y_i 1^T)\Sigma^{-1}(\hat{y}_i - y_i 1))$$

To simplify the equations, we assume that $$\sigma_y^2 \rightarrow \infty$$, which amounts to an uninformative prior over the target variable.

After finding derivative of the data log-likelihood with respect to $$\Sigma^{-1}$$, we obtain an iterative update equation,

$$\Sigma = \frac{1}{N}((\hat{Y}_l - y_l 1^T)(\hat{Y}_l - y_l 1^T)^T + \hat{Y}_u^T\hat{Y}_u + \frac{N_u 11^T}{1^T\Sigma^{-1}1} + \sum_{i=1}^{N_u} (\bar{y}_i 11^T - \bar{y}_i (1\hat{y}_i^T + \hat{y}_i 1^T)))$$

Bates and Granger, 1969

Ristovski et al., 2010
Further advantages

- Straightforward to account for missing experts due to the assumption of Gaussianity
- Incorporation of domain knowledge through a Wishart prior over the precision matrix
- We derive an approach for partitioning data into several regimes, where expert predictions within each regime are sampled from a different multivariate Gaussian
  - Learned using the EM algorithm
Results

- We used 5 years of aerosol data from 33 AERONET US locations, and predictions from 5 experts (MISR, Terra MODIS, Aqua MODIS, OMI, SeaWiFS)
- Training data set with 6,913 labeled examples (roughly 200 examples per site)
- 58% of satellite predictions missing
Results

- Evaluating usefulness of partitioning
  - From each site we randomly sampled 100 points, and assumed that 50 are labeled and 50 unlabeled

<table>
<thead>
<tr>
<th>Method</th>
<th># clusters</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 sites, supervised</td>
<td>2</td>
<td>0.0795</td>
</tr>
<tr>
<td>2 sites, semi-super.</td>
<td>2</td>
<td>0.0752</td>
</tr>
<tr>
<td>4 sites, supervised</td>
<td>2</td>
<td>0.0728</td>
</tr>
<tr>
<td>4 sites, semi-super.</td>
<td>2</td>
<td>0.0704</td>
</tr>
<tr>
<td>6 sites, supervised</td>
<td>2</td>
<td>0.0694</td>
</tr>
<tr>
<td>6 sites, semi-super.</td>
<td>2</td>
<td>0.0688</td>
</tr>
</tbody>
</table>
Results

- Evaluating usefulness of unlabeled data
  - Randomly selected 2, 4, and 6 sites and took 100 points from each as labeled data; then, we selected 100 points from each remaining site and treated them as unlabeled
  - Simulates large areas where just few AERONET sites are available
  - Unlabeled data helpful, although benefit decreased when larger amounts of labeled data points were available

<table>
<thead>
<tr>
<th>Method</th>
<th># clusters</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Averaging</td>
<td>—</td>
<td>0.0818</td>
</tr>
<tr>
<td>All sites, semi-super.</td>
<td>1</td>
<td>0.0677</td>
</tr>
<tr>
<td>All sites, semi-super.</td>
<td>2</td>
<td>0.0648</td>
</tr>
</tbody>
</table>
Structured Learning with GCRF

- Conditional Random Field (CRF)
  - Given input variables \( x \) (that include measurements, location, time, and other useful data), probability of output variable \( y \) modeled as

  \[
  P(y|x) = \frac{1}{Z(x, \alpha, \beta)} \exp \left( - \sum_{i=1}^{N} A(\alpha, y_i, x) - \sum_{i \sim j} I(\beta, y_i, y_j, x) \right)
  \]

  \[
  Z(x, \alpha, \beta) = \int_{y} \exp \left( - \sum_{i=1}^{N} A(\alpha, y_i, x) - \sum_{i \sim j} I(\beta, y_i, y_j, x) \right) dy
  \]

  - \( A \) - association, \( I \) - interaction potential, \( \alpha \) and \( \beta \) are to be learned

- If we choose \( A \) and \( I \) as quadratic functions of the outputs \( y \), the model corresponds to a multivariate Gaussian

  \[
  A(\alpha, y_i, x) = \sum_{m=1}^{M} \alpha_m \delta_i^m \left( y_i - \theta_m(x_i) \right)^2
  \]

  \[
  I(\beta, y_i, y_j, x) = \sum_{l=1}^{L} \beta_l \delta_{ij} \cdot (y_i - y_j)^2
  \]
We considered 5 satellite instruments:

- 2 instruments have overpass time in the morning (Terra MODIS and MISR, 10:30am local time)
- 3 instruments have a time of overpass in the afternoon (Aqua MODIS, OMI, and SeaWiFS, 1:30pm local time)

We assume high correlation between AOD at 10:30am and 1:30pm on the same day, as well as between AOD values at the same time between two consecutive days.
We estimate AOD at 10:30am and 1:30pm for each location.

The corresponding graphical model.
GCRF Model II

- Model I assumes the same $\alpha$ parameters for each of the instruments regardless of the availability of other instruments.
- However, if we know that one of the measurements is missing, we should be careful about the available ones as well.

$$A(\alpha, y_i, x) = \delta_i^{100} \alpha_1^{100} (y_i - \theta_i^{qua})^2 + \delta_i^{010} \alpha_2^{010} (y_i - \theta_i^{omi})^2 + \delta_i^{001} \alpha_3^{001} (y_i - \theta_i^{sw})^2 + \delta_i^{110} \left( \alpha_1^{110} (y_i - \theta_i^{qua})^2 + \alpha_2^{110} (y_i - \theta_i^{omi})^2 \right) + \delta_i^{101} \left( \alpha_1^{101} (y_i - \theta_i^{qua})^2 + \alpha_3^{101} (y_i - \theta_i^{sw})^2 \right) + \delta_i^{011} \left( \alpha_2^{011} (y_i - \theta_i^{omi})^2 + \alpha_3^{011} (y_i - \theta_i^{sw})^2 \right) + \delta_i^{111} \left( \alpha_1^{111} (y_i - \theta_i^{qua})^2 + \alpha_2^{111} (y_i - \theta_i^{omi})^2 + \alpha_3^{111} (y_i - \theta_i^{sw})^2 \right) + \delta_i^{10} \alpha_4^{10} (y_i - \theta_i^{terra})^2 + \delta_i^{01} \alpha_5^{01} (y_i - \theta_i^{mier})^2 + \delta_i^{11} \left( \alpha_4^{11} (y_i - \theta_i^{terra})^2 + \alpha_5^{11} (y_i - \theta_i^{mier})^2 \right),$$
## Results

- Results given in terms of RMSE
- We give results for various models, used parameters, and availability patterns

<table>
<thead>
<tr>
<th>Instrument</th>
<th>No. of points</th>
<th>Individual sensors</th>
<th>Without $\beta$ parameters</th>
<th>Model II with $\beta$ parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Model I</td>
<td>Model II</td>
</tr>
<tr>
<td>MODIS Aqua</td>
<td>7,246</td>
<td>0.0872</td>
<td>0.0857</td>
<td>0.0840</td>
</tr>
<tr>
<td>OMI</td>
<td>10,142</td>
<td>0.2390</td>
<td>0.2058</td>
<td>0.2053</td>
</tr>
<tr>
<td>SeaWiFS</td>
<td>2,205</td>
<td>0.0739</td>
<td>0.0676</td>
<td>0.0658</td>
</tr>
<tr>
<td>MODIS Aqua alone</td>
<td>2,102</td>
<td>0.0889</td>
<td>0.0889</td>
<td>0.0889</td>
</tr>
<tr>
<td>OMI alone</td>
<td>4,528</td>
<td>0.2934</td>
<td>0.2934</td>
<td>0.2934</td>
</tr>
<tr>
<td>SeaWiFS alone</td>
<td>237</td>
<td>0.0800</td>
<td>0.0800</td>
<td>0.0800</td>
</tr>
<tr>
<td>MODIS + OMI</td>
<td>3,868</td>
<td>0.0893</td>
<td>0.2123</td>
<td>0.0918</td>
</tr>
<tr>
<td>MODIS + SeaWiFS</td>
<td>222</td>
<td>0.0982</td>
<td>0.0747</td>
<td>0.0694</td>
</tr>
<tr>
<td>OMI + SeaWiFS</td>
<td>692</td>
<td>0.1011</td>
<td>0.0835</td>
<td>0.0817</td>
</tr>
<tr>
<td>MODIS + OMI + SeaWiFS</td>
<td>1,054</td>
<td>0.0717</td>
<td>0.0715</td>
<td>0.0650</td>
</tr>
<tr>
<td>MODIS Terra</td>
<td>8,725</td>
<td>0.0905</td>
<td>0.0847</td>
<td>0.0841</td>
</tr>
<tr>
<td>MISR</td>
<td>2,165</td>
<td>0.0652</td>
<td>0.0684</td>
<td>0.0656</td>
</tr>
<tr>
<td>MODIS Terra alone</td>
<td>7,552</td>
<td>0.0863</td>
<td>0.0863</td>
<td>0.0863</td>
</tr>
<tr>
<td>MISR alone</td>
<td>992</td>
<td>0.0618</td>
<td>0.0618</td>
<td>0.0618</td>
</tr>
<tr>
<td>MODIS + MISR</td>
<td>1,173</td>
<td>0.1142</td>
<td>0.0680</td>
<td>0.0736</td>
</tr>
<tr>
<td>All labeled</td>
<td>44,445</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>All labeled with any satellite</td>
<td>22,420</td>
<td>-</td>
<td>0.1516</td>
<td>0.1512</td>
</tr>
<tr>
<td>All labeled without satellites</td>
<td>20,025</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Results

- Learned parameters of Model II provide insight in the quality of instruments
- Within-day interaction much stronger than day-to-day interaction
- OMI assigned very low $\alpha$, except when all satellites are available (issue with OMI filter?)

<table>
<thead>
<tr>
<th>10:30am</th>
<th>Count</th>
<th>$\alpha_{modis}$</th>
<th>$\alpha_{misl}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODIS + MISR</td>
<td>2,334</td>
<td>–</td>
<td>183.03</td>
</tr>
<tr>
<td>MODIS + MISR</td>
<td>17,714</td>
<td>67.02</td>
<td>–</td>
</tr>
<tr>
<td>MODIS + MISR</td>
<td>2,572</td>
<td>0.03</td>
<td>100.48</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1:30pm</th>
<th>Count</th>
<th>$\alpha_{modis}$</th>
<th>$\alpha_{omi}$</th>
<th>$\alpha_{sw}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODIS + OMI + SeaWiFS</td>
<td>570</td>
<td>–</td>
<td>–</td>
<td>87.40</td>
</tr>
<tr>
<td>MODIS + OMI + SeaWiFS</td>
<td>11,309</td>
<td>–</td>
<td>1.15</td>
<td>–</td>
</tr>
<tr>
<td>MODIS + OMI + SeaWiFS</td>
<td>5,589</td>
<td>97.96</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MODIS + OMI + SeaWiFS</td>
<td>1,363</td>
<td>–</td>
<td>4.42</td>
<td>57.91</td>
</tr>
<tr>
<td>MODIS + OMI + SeaWiFS</td>
<td>461</td>
<td>16.39</td>
<td>–</td>
<td>80.47</td>
</tr>
<tr>
<td>MODIS + OMI + SeaWiFS</td>
<td>8,487</td>
<td>34.30</td>
<td>0.01</td>
<td>–</td>
</tr>
<tr>
<td>MODIS + OMI + SeaWiFS</td>
<td>2,173</td>
<td>20.22</td>
<td>82.47</td>
<td>109.78</td>
</tr>
</tbody>
</table>

$\beta_1$, $\beta_2$

<table>
<thead>
<tr>
<th>Count</th>
<th>$\alpha_{modis}$</th>
<th>$\alpha_{omi}$</th>
<th>$\alpha_{sw}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>315.47</td>
<td>35.45</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
GCRF for traffic speed forecasting

- **Problem setting**
  - Predict travel speeds on I-35W highway, Minneapolis, MN, from April to July, 2003
  - Up to 1h ahead, in 10-min increments, across 11 consecutive sensor stations

* Djuric, N., Radosavljevic, V., Coric, V., Vucetic, S., Travel Speed Forecasting using Continuous Conditional Random Fields, TRR: Journal of the Transportation Research Board 2011
Results

- Compared with linear regression
- Baseline predictors
  - Downstream sensor speed
  - Upstream sensor speed
  - Historical average
  - Current speed

<table>
<thead>
<tr>
<th>TABLE 1</th>
<th>Prediction MAE Errors of Various Predictors Aggregated Across Time Horizons</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Horizon (min)</td>
</tr>
<tr>
<td>Linear Regression Models</td>
<td></td>
</tr>
<tr>
<td>LR Model 2 (four baselines)</td>
<td>5.961</td>
</tr>
<tr>
<td>Baseline Predictors Incorporated into CCRF Models</td>
<td></td>
</tr>
<tr>
<td>CCRF Models, in Increasing Level of Complexity</td>
<td></td>
</tr>
<tr>
<td>CCRF Model 1 (two baselines)</td>
<td>6.198</td>
</tr>
<tr>
<td>CCRF Model 2 (four baselines)</td>
<td>5.929</td>
</tr>
<tr>
<td>CCRF Model 3 (regime switching)</td>
<td>5.922</td>
</tr>
</tbody>
</table>
Ongoing / future work

- Semi-supervised combination of experts
  - Parameterized priors?
  - Structured output, non-IID data?
  - GMRF priors?

- Data visualization
  - Developing software for visual exploration
  - Distributed implementation?
  - Binary features, user-provided constraints on orderings?

- Many-to-many object matching
  - Scaling up CKS? Regularization? Semi-supervised?
Ongoing / future work

- Large-scale learning
  - Completing characterization and implementation of group lasso for AMM
  - Extending more state-of-the-art methods to large-scale domain (GCRF training using MapReduce, GraphLab, GraphChi?)
  - AMM-rank, evaluation of large-scale label ranking method
  - Dirichlet process?

- Structured learning
  - Application of GCRF to aerosol estimation (sunglint, clouds?)
  - Speeding up GCRF, further assumptions on the structure
Inadequacy of standard machine learning tools in large-scale setting is apparent
- Novel methods are necessary to address plethora of Big Data problems

Large-scale learning
- Efficient, non-linear AMM classifiers
- Highly-optimized BudgetedSVM C++ toolbox
- Confidence-weighted classification on MapReduce

Data visualization
- Fast, efficient knowledge discovery

Semi-supervised combination of experts
- Accounts for unlabeled data, missing data, correlations
- Useful in many areas of machine learning

Structured learning in remote sensing and traffic estimation
Thank you!

- Questions?