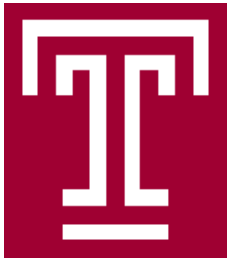
Big Data Algorithms for Visualization and Supervised Learning



Nemanja Djuric

Temple University, December 2nd, 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.***, Lan, L., Vucetic, S., Wang, Z., BudgetedSVM: A Toolbox for Scalable SVM Approximations, ***JMLR 2013***
  - ***Djuric, N.***, Grbovic, M., Vucetic, S., Distributed Confidence-Weighted Classification on MapReduce, ***IEEE BigData 2013***
  - Wang, Z., ***Djuric, N.***, Crammer, K., Vucetic, S., Trading Representability for Scalability: Adaptive Multi-Hyperplane Machine for Nonlinear Classification, ***KDD 2011***
- Memory-constrained online learning
  - ***Djuric, N.***, Vucetic, S., Random Kernel Perceptron on ATTiny2313 Microcontroller, ***SensorKDD @ KDD 2010***

- Traffic state estimation and prediction
  - Coric, V., ***Djuric, N.***, Vucetic, S., Traffic State Estimation from Aggregated Measurements using Signal Reconstruction Techniques, ***TRR: Journal of the Transp. Research Board 2012***
  - ***Djuric, N.***, Radosavljevic, V., Coric, V., Vucetic, S., Travel Speed Forecasting using Continuous Conditional Random Fields, ***TRR: Journal of the Transportation Research Board 2011***
- Label ranking
  - Grbovic*, M., ***Djuric*, N.***, Vucetic S., Multi-prototype Label Ranking with Novel Pairwise-to-Total-Rank Aggregation, ***IJCAI 2013***
  - Grbovic, M., ***Djuric, N.***, Vucetic S., Supervised Clustering of Label Ranking Data using Label Preference Information, ***MLJ 2013***
  - Grbovic, M., ***Djuric, N.***, Vucetic S., Supervised Clustering of Label Ranking Data, ***SDM 2012*** **(*Best of SDM*)**

- Bioinformatics
  - Radivojac, P., Clark, W. T., ..., Toppo, S., Lan, L., ***Djuric, N.***, Guo, Y., Vucetic, S., Bairoch, A., Linial, M., Babbitt, P. C., *et al.*, A Large-scale Evaluation of Computational Protein Function Prediction, ***Nature Methods 2013***
  - Lan, L., ***Djuric, N.***, Guo, Y., Vucetic, S., MS-*k*NN: Protein Function Prediction by Integrating Multiple Data Sources, ***BMC Bioinformatics 2012*** **(*Top performing team*)**
- Unsupervised object matching
  - ***Djuric, N.***, Grbovic, M., Vucetic S., Convex Kernelized Sorting, **AAAI 2012**
- 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

Introduction

- "*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

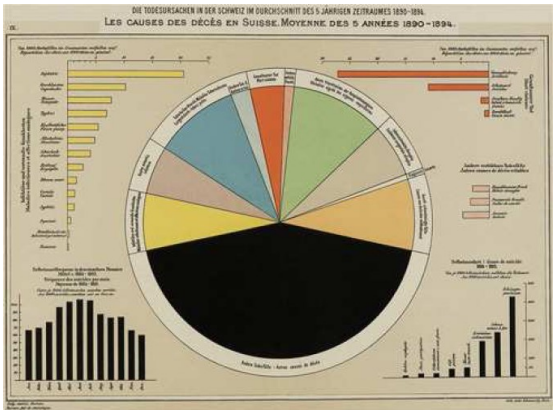
- Large data visualization
  - EM-ordering and TSP-means
- Large-scale learning
  - Adaptive Multi-hyperplane Machines
  - BudgetedSVM: A C++ toolbox for large-scale learning
  - Distributed confidence-weighted learning on MapReduce
- Unsupervised object matching
  - Convex Kernelized Sorting
- Combination 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

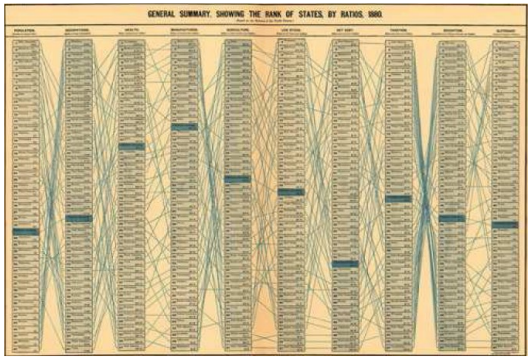
How to visualize a data matrix?

- Classical approaches
  - Pie and bar charts, histograms

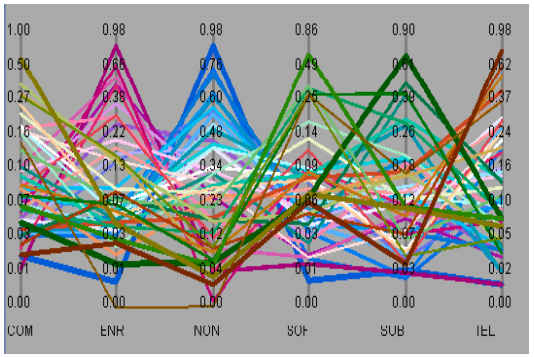


1897.

  - Parallel coordinates



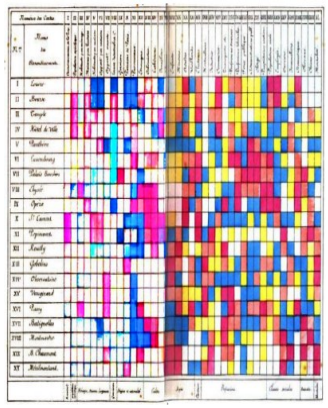
1883.



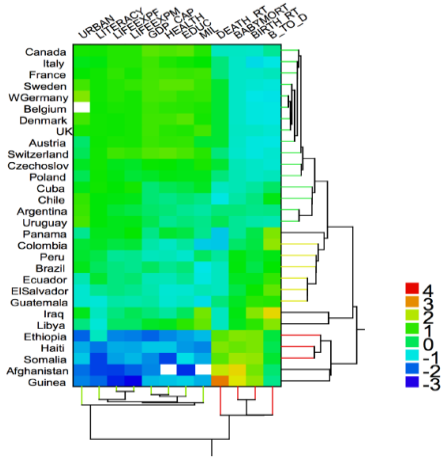
2005.

How to visualize a data matrix?

- Classical approaches
  - Heat maps

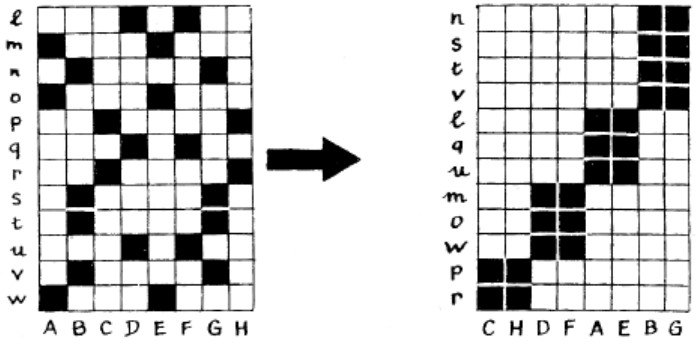


1873.



1995.

- Idea: Data reordering
  - Reorder matrix so similar rows/columns are grouped together



1963.

The proposed method: Example

- Waveform benchmark data set
- Please click here: EM-ordering example

EM-ordering

- Data reordering can be considered from the viewpoint of data compression
  - Reorder the data so that it is maximally compressible
  - Assume data set $D = \{\mathbf{x}_i, i = 1, .., n\}$ is given, where $\mathbf{x}_i = [x_{i1}, x_{i2}, .., x_{im}]$ are m-dimensional examples
- Differential Predictive Coding (DPC)
  - Use local context to code the value of $\mathbf{x}_i$

$$D = \{\mathbf{x}_i, i = 1,..., n\} \rightarrow D_{DPC} = \{\mathbf{x}_1, \varepsilon_2, ..., \varepsilon_n\}$$
$$\text{where } \varepsilon_i = (\mathbf{x}_i - \mathbf{x}_{i-1}), i = 2,..., n$$

* 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

$$H(\varepsilon) = \frac{n}{2}(m \cdot \log(2\pi) + \sum_{j=1}^{m} \log(\sigma_j(\varepsilon))) + 0.5 \sum_{i=2}^{n} \sum_{j=1}^{m} \frac{(\mathbf{x}_{\pi(i),j} - \mathbf{x}_{\pi(i-1),j})^2}{\sigma_j^2}$$

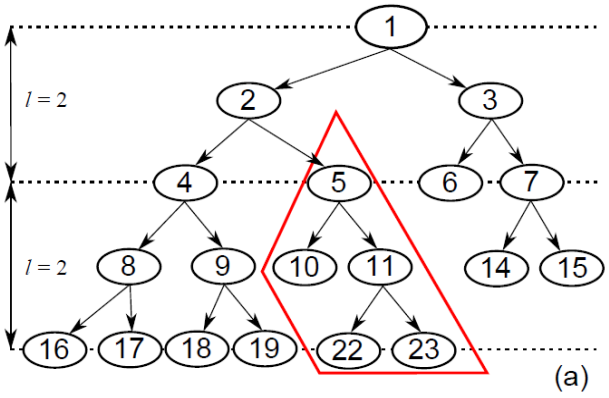
- The optimization problem becomes

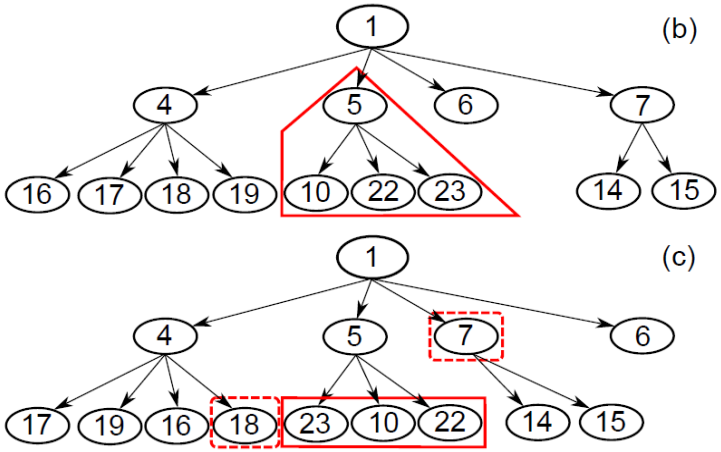
$$(\pi^*, \{\sigma_1^*, ..., \sigma_m^*\}) = \arg \min_{\pi, \{\sigma_1, ..., \sigma_m\}} H(\varepsilon)$$

- 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

TSP-solver

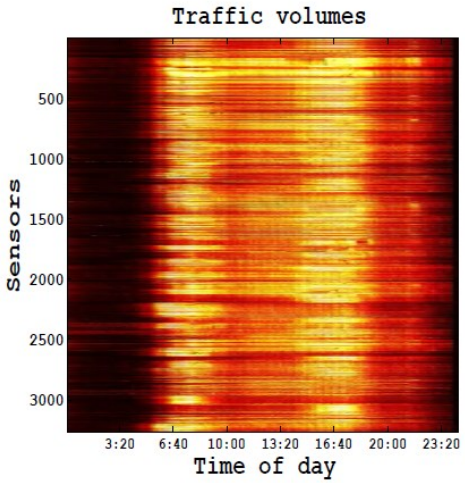
- Super-quadratic time complexity of the best TSP solvers is prohibitive on large data
- We propose an $O(n \log(n))$ method, called TSP-means
  - After creating 2^l-tree through recursive runs of *k*-means, solve TSP defined on each node while traversing the tree breath-first



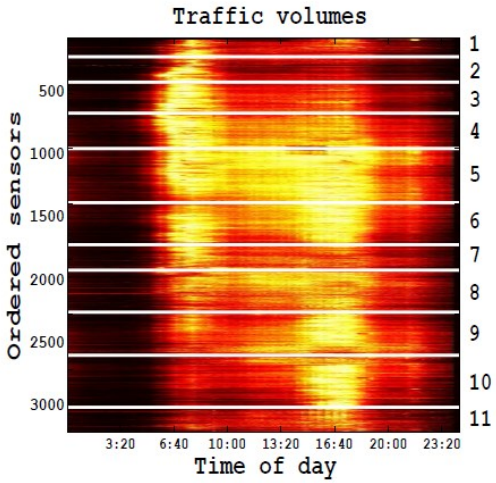


Real-world applications

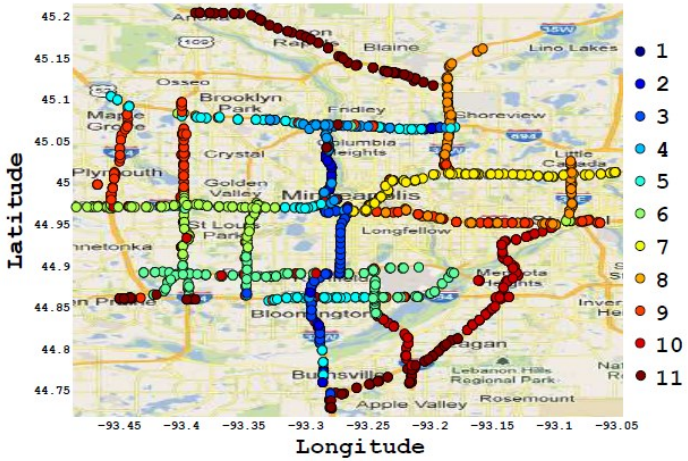
- Minneapolis traffic data set



Original data set



Reordered data set



Locations of the sensors

Large-scale learning

- 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?

Large-scale learning: Multi-class SVM

- Assume a data set $D = \{(\mathbf{x}_n, y_n), n = 1, \ldots, N\}$, where $\mathbf{x}_n$ 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 $\mathbf{w}_i$
  - Prediction is given as

$$f(\mathbf{x}) = \arg\max_{i \in \mathcal{Y}} g(i, \mathbf{x}), \text{ where } g(i, \mathbf{x}) = \mathbf{w}_i^{\mathrm{T}} \mathbf{x}$$

  - By concatenating all weights $\mathbf{w}_i$, we can write

$$\mathbf{W} = [\mathbf{w}_1 \ \mathbf{w}_2 \ \ldots \ \mathbf{w}_M]$$

Large-scale learning: Multi-class SVM

- Multi-class SVM training
  - We find the weights by minimizing the following problem

$$\frac{\lambda}{2}||\mathbf{W}||_F^2 + \frac{1}{N}\sum_{n=1}^{N} l(\mathbf{W};(\mathbf{x}_n, y_n))$$

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

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

$$g(i, \mathbf{x}) = \max_j \mathbf{w}_{i,j}^{\mathrm{T}} \mathbf{x}, \text{ with } \mathbf{W} = \left[\mathbf{w}_{1,1} \ldots \mathbf{w}_{1,b_1} | \mathbf{w}_{2,1} \ldots \mathbf{w}_{2,b_2} | \ldots | \mathbf{w}_{M,1} \ldots \mathbf{w}_{M,b_M}\right]$$

Large-scale learning: Multi-class SVM

- The resulting Multi-hyperplane Machine (MM) loss function is non-convex, and the authors propose to solve the modified loss function

$$l_{cvx}(\mathbf{W}; (\mathbf{x}_n, y_n); z_n) = \max\left(0, 1 + \max_{i \in \mathcal{Y} \setminus y_n} g(i, \mathbf{x}_n) - \mathbf{w}_{y_n, z_n}^{\mathrm{T}} \mathbf{x}_n\right)$$

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

* Wang, Z., Djuric, N., Crammer, K., Vucetic, S. *Trading Representability for Scalability: Adaptive Multi-Hyperplane Machine for Nonlinear Classification*, **KDD 2011**

Adaptive MM

- At the t^{th} training iteration, minimize the instantaneous objective function

$$P^{(t)}(\mathbf{W}|\mathbf{z}) \equiv \frac{\lambda}{2}||\mathbf{W}||^2 + l_{cvx}(\mathbf{W}; (\mathbf{x}_t, y_t); z_t)$$

- 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 ➔ 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

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

$$B = \sum_{i=1}^{M} b_i + 1 + b_{\max}^2 - b_{\max} - b_{\min},\ K = \frac{1}{2}\sum_{i=1}^{M} b_i \sum_{j\neq i}^{M} b_j, b_{\min} = \min_{i=1,\ldots,M}\{b_i\}\ \textit{and}\ b_{\max} = \max_{i=1,\ldots,M}\{b_i\}$$

Extensions: Online group lasso

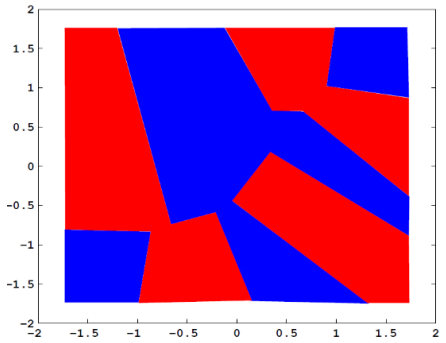
- Enforce group-level sparsity of the weight matrix $\mathbf{W}$
  - Online truncated gradient (Langford et al., JMLR 2009)
- Let $\mathbf{w}_g$ be a vector representing a g^{th} group of elements from the weight matrix $\mathbf{W}$, with $g = 1, \ldots, G$, then

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

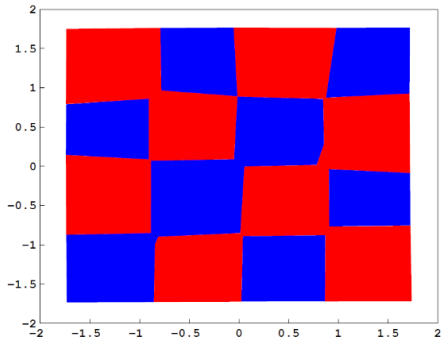
  - Theoretical guarantees of convergence
  - Analogous to online variant of group lasso
  - Straightforward implementation

Extensions: Growing AMM

- SGD fails to learn good weights for highly non-linear patterns
  - Idea: "Clone" existing non-zero weights when misclassification
- Results on 4×4 checkerboard data set



(a) AMM (27.95% err. r.)



(b) GAMM (7.38% err. r.)

- Robust to noise, significantly outperforms AMM

BudgetedSVM

- 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?

BudgetedSVM

- 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

* Djuric, N., Lan, L., Vucetic, S, Wang, Z. *BudgetedSVM: A Toolbox for Scalable SVM Approximations*, **JMLR 2013**

BudgetedSVM

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

	Pegasos		AMM			LLSVM			BSGD			RBF-SVM	
Data set	err.	time	err.	B	time	err.	B	time	err.	B	time	err.	time
webspam						3.46	5*e*2	2.5m	2.04	5*e*2	2.0m		
$N = 280{,}000$	7.94	0.5s	4.74	9	3s	2.60	1*e*3	6.1m	1.72	1*e*3	3.9m	0.77	4.0h
$M = 254$						1.99	3*e*3	0.5h	1.49	3*e*3	0.2h	(#SV: 26,447)	
rcv1						4.97	5*e*2	0.2h	8.70	5*e*2	5.1m		
$N = 677{,}399$	2.73	1.5s	2.39	19	9s	4.23	1*e*3	0.5h	8.64	1*e*3	10.5m	2.17	20.2h
$M = 47{,}236$						3.05	3*e*3	2.2h	6.87	3*e*3	0.8h	(#SV: 50,641)	
mnist8m-bin						6.84	5*e*2	1.6h	2.23	5*e*2	2.3h		
$N = 8{,}000{,}000$	22.71	1.1m	3.16	18	4m	4.59	1*e*3	3.8h	1.92	1*e*3	4.9h	0.43	N/A
$M = 784$						2.59	3*e*3	15h	1.62	3*e*3	16.1h		

- Available from *SourceForge.net*
- More than 300 downloads since summer, try it out!

Confidence-Weighted classification

- 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 $(\mathbf{x}_t, y_t)$

$$\hat{y} \sim \mathcal{N}\left(y_t(\boldsymbol{\mu}^{\mathrm{T}}\mathbf{x}_t), \mathbf{x}_t^{\mathrm{T}}\Sigma\mathbf{x}_t\right)$$

CW classification

- 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):

$$(\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1}) = \arg\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} D_{KL}\big(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \| \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\big) + \lambda_1 \big(\max(0, 1 - y_t \boldsymbol{\mu}^{\mathrm{T}} \mathbf{x}_t)\big)^2 + \lambda_2 (\mathbf{x}_t^{\mathrm{T}} \Sigma \mathbf{x}_t)$$

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

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

or its empirical estimate

$$\mathcal{L} = \sum_{m=1}^{M} \mathbb{P}(\mathcal{N}(\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m))\ D_{KL}^{S}(\mathcal{N}(\boldsymbol{\mu}_*,\boldsymbol{\Sigma}_*)\|\mathcal{N}(\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m))$$

- 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**

CW training on MapReduce

- Finding a derivative of the objective function with respect to mean and covariance matrix, we obtain

$$\boldsymbol{\mu}_* = \Big(\sum_{m=1}^{M} \big(\mathbb{P}(\mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m))\ (\boldsymbol{\Sigma}_*^{-1} + \boldsymbol{\Sigma}_m^{-1}) \big) \Big)^{-1} \Big(\sum_{m=1}^{M} \big(\mathbb{P}(\mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m))\ (\boldsymbol{\Sigma}_*^{-1} + \boldsymbol{\Sigma}_m^{-1}) \big) \boldsymbol{\mu}_m \Big)$$

$$\boldsymbol{\Sigma}_* \Big(\sum_{m=1}^{M} \mathbb{P}(\mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m))\ \boldsymbol{\Sigma}_m^{-1} \Big) \boldsymbol{\Sigma}_* = \sum_{m=1}^{M} \mathbb{P}(\mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m))\ \big(\boldsymbol{\Sigma}_m + (\boldsymbol{\mu}_* - \boldsymbol{\mu}_m)(\boldsymbol{\mu}_* - \boldsymbol{\mu}_m)^{\mathrm{T}} \big)$$

- The second equation is an algebraic Riccati equation of the form $\mathbf{XAX}=\mathbf{B}$, solved as

$$\mathbf{X} = \mathbf{U}^{-0.5} \mathbf{B}^{0.5} (\mathbf{U}^{\mathrm{T}})^{-0.5}, \text{ with } \mathbf{A} = \mathbf{U}^{\mathrm{T}} \mathbf{U}$$

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

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

Table 2. Performance of VW

# mappers	# reducers	Avg. map time	Reduce time	AUC
1	0	7h	n/a	0.8506
100	0	1h	n/a	0.8508
500	0	8 min	n/a	0.8501
1,000	0	6 min	n/a	0.8498

- 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

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

- If the sets are independent, then HSIC is minimal and equal to 0
- Hence, matching problem is defined as (π is a permutation matrix: $\pi_{ij} = 1$ if x_i and y_j match, 0 otherwise)

$$\underset{\pi \in \Pi_m}{\text{maximize}}\ trace(K \cdot \pi^{\mathrm{T}} \cdot L \cdot \pi)$$

Convex Kernelized Sorting

- Reformulate KS problem as follows:
  - Given two $m \times m$ matrices K and L, value of $trace(K\,L)$ 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 π to be doubly-stochastic, and KS becomes:

$$\underset{\pi \in \Pi_m}{\text{maximize}}\ trace(K \cdot \pi^{\mathrm{T}} \cdot L \cdot \pi) \qquad \underset{\pi \in \Pi_m}{\text{minimize}}\ \| K \cdot \pi^{\mathrm{T}} - (L \cdot \pi)^{\mathrm{T}} \|_F^2$$

KS CKS

* 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

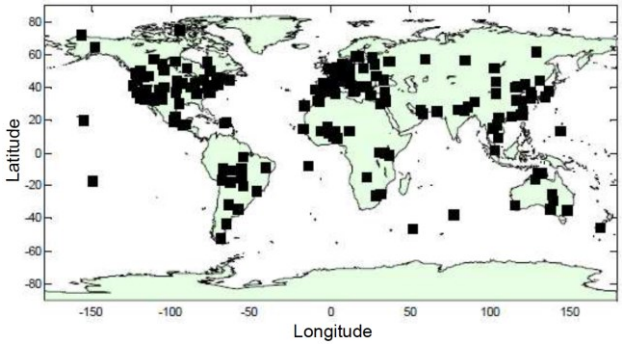
Language	Corpus size	Baseline	KS	KS-p	LSOM	CKS
Danish	387	39	261 (318)	258 (273)	159 (173)	**379**
Dutch	387	50	266 (371)	237 (317)	146 (375)	**383**
Finnish	308	54	19 (32)	22 (38)	10 (10)	**114**
French	356	64	319 (356)	320 (334)	354 (354)	**356**
German	356	50	282 (344)	258 (283)	338 (350)	**356**
Italian	387	49	341 (382)	349 (353)	378 (381)	**385**
Portuguese	356	46	308 (354)	326 (343)	342 (356)	**356**
Spanish	387	48	342 (365)	351 (364)	386 (387)	**387**
Swedish	337	76	20 (39)	20 (33)	5 (5)	**97**

Remote sensing - Aerosols

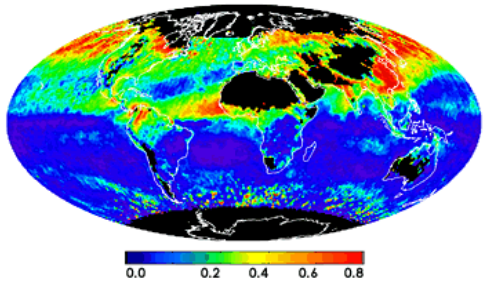
- Aerosols are small particles suspended in the atmosphere, originating from natural and man-made sources
- Estimation of global aerosol distribution is one of the biggest challenges in climate research
  - Negative effect on public health
  - Profound effect on Earth's radiation budget
- Standard measure 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,

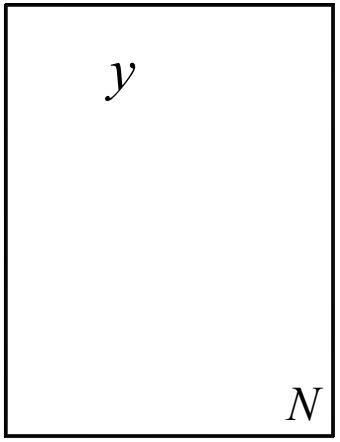
$$y_i \sim Norm(\mu_y, \sigma_y^2)$$

- Denote by $\hat{\mathbf{y}}_i$ a K-dimensional vector of expert predictions, sampled from multivariate Gaussian,

$$\hat{\mathbf{y}}_i \mid y_i \sim Norm(y_i \mathbf{1}, \Sigma)$$

- 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)**

Inference

- Once the training is completed, aggregated prediction can be found as a mean of the posterior distribution (assuming no missing experts)

$$y_i \mid \hat{\mathbf{y}}_i \sim Norm(\bar{y}_i, (\mathbf{1}^{\mathrm{T}} \Sigma'^{-1} \mathbf{1})^{-1})$$

where the mean can be computed as follows,

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

Training

- Learning by maximizing likelihood of the training data
- We write probability of the training data as follows

$$P(D \mid \Theta) = P(D_u \mid \Theta) \prod P(D_l \mid \Theta)$$

- The probability of unlabeled data set is equal to

$$P(D_u \mid \Theta) = \prod_{i=1}^{N_u} P(\hat{\mathbf{y}}_i \mid \Theta) = \prod_{i=1}^{N_u} \sum_{y} P(\hat{\mathbf{y}}_i \mid y, \Theta) P(y \mid \Theta) dy$$

$$= \prod_{i=1}^{N_u} \left(\sqrt{\frac{|\Sigma'|^{-1}}{(2\pi)^{K-1} \mathbf{1}^{\mathrm{T}} \Sigma'^{-1} \mathbf{1}}} \right) \exp\left(-\frac{1}{2} (\hat{\mathbf{y}}'_i - \bar{y}_i \mathbf{1})^{\mathrm{T}} \Sigma'^{-1} (\hat{\mathbf{y}}'_i - \bar{y}_i \mathbf{1}) \right)$$

Training

- Further, the probability of labeled data can be written as

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

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

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

$$\Sigma = \frac{1}{N}\underbrace{((\hat{\mathbf{Y}}_l - \mathbf{y}_l\mathbf{1}^{\mathrm{T}})^{\mathrm{T}}(\hat{\mathbf{Y}}_l - \mathbf{y}_l\mathbf{1}^{\mathrm{T}})}_{\text{Bates and Granger, 1969}} + \underbrace{\hat{\mathbf{Y}}_u^{\mathrm{T}}\hat{\mathbf{Y}}_u + \frac{N_u\mathbf{1}\mathbf{1}^{\mathrm{T}}}{\mathbf{1}^{\mathrm{T}}\Sigma^{-1}\mathbf{1}} + \sum_{i=1}^{N_u}(\bar{y}_i^2\mathbf{1}\mathbf{1}^{\mathrm{T}} - \bar{y}_i(\mathbf{1}\hat{\mathbf{y}}_i^{\mathrm{T}} + \hat{\mathbf{y}}_i\mathbf{1}^{\mathrm{T}}))}_{\text{Ristovski et al., 2010}}$$

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

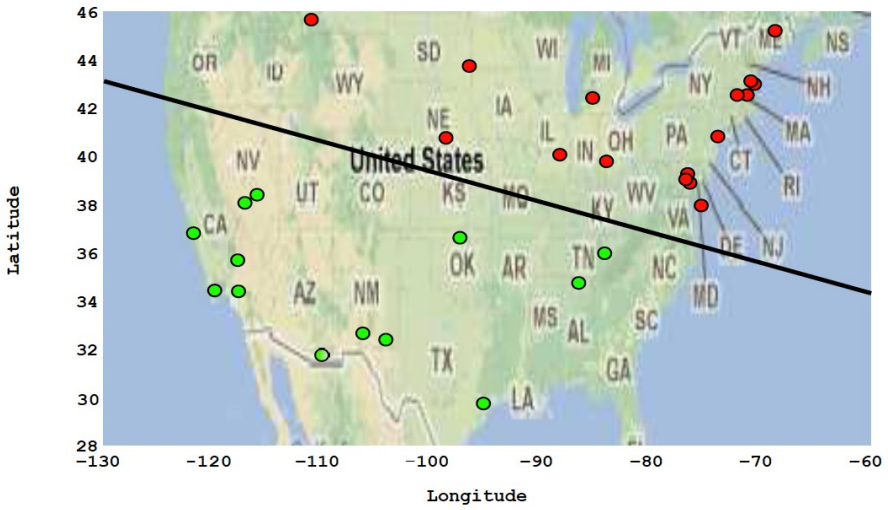
Results

- 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

Method	# clusters	RMSE
2 sites, supervised	2	0.0795
2 sites, semi-super.	2	0.0752
4 sites, supervised	2	0.0728
4 sites, semi-super.	2	0.0704
6 sites, supervised	2	0.0694
6 sites, semi-super.	2	0.0688



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

Method	# clusters	RMSE
Averaging	–	0.0818
All sites, semi-super.	1	0.0677
All sites, semi-super.	2	0.0648

- 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

Structured Learning with GCRF

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

$$\mathbb{P}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta})} \exp\Big(-\sum_{i=1}^{N} A(\boldsymbol{\alpha}, y_i, \mathbf{x}) - \sum_{i\sim j} I(\boldsymbol{\beta}, y_i, y_j, \mathbf{x})\Big)$$

$$Z(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\beta}) = \int_{\mathbf{y}} \exp\big(-\sum_{i=1}^{N} A(\boldsymbol{\alpha}, y_i, \mathbf{x}) - \sum_{i\sim j} I(\boldsymbol{\beta}, y_i, y_j, \mathbf{x})\big) d\mathbf{y}$$

  - A - association, I - interaction potential, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are to be learned

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

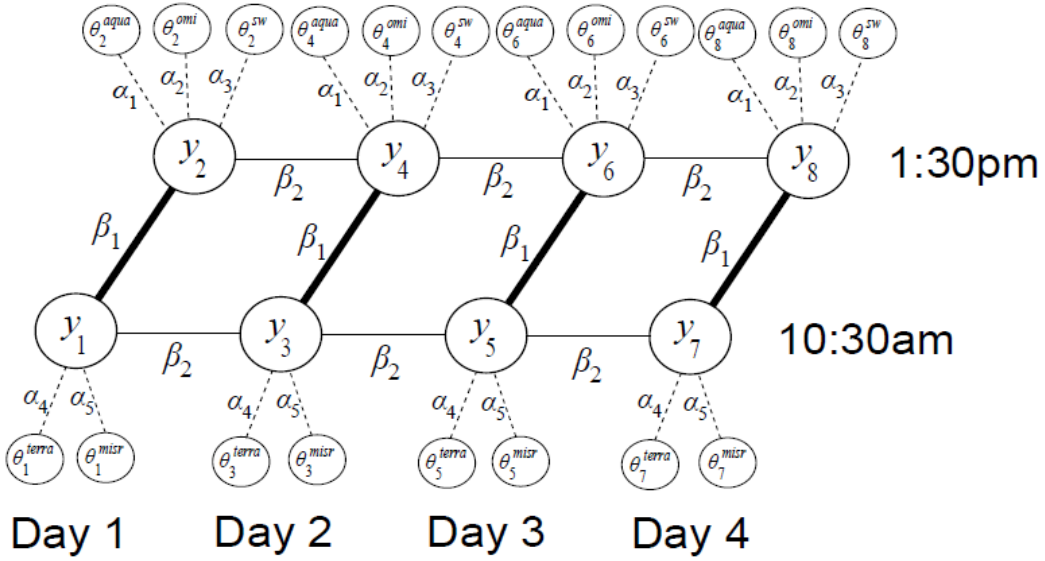
$$A(\boldsymbol{\alpha}, y_i, \mathbf{x}) = \sum_{m=1}^{M} \alpha_m \delta_i^m \big(y_i - \theta_m(\mathbf{x}_i)\big)^2 \qquad I(\boldsymbol{\beta}, y_i, y_j, \mathbf{x}) = \sum_{l=1}^{L} \beta_l \delta_{ij}^l \cdot (y_i - y_j)^2$$

GCRF for AOD estimation

- 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

GCRF Model I

- We estimate AOD at 10:30am and 1:30pm for each location
- The corresponding graphical model



GCRF Model II

- Model I assumes the same $\boldsymbol{\alpha}$ 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

$$
\begin{aligned}
A(\boldsymbol{\alpha}, y_i, \mathbf{x}) = {} & \delta_i^{100}\alpha_1^{100}(y_i - \theta_i^{aqua})^2 \\
& + \delta_i^{010}\alpha_2^{010}(y_i - \theta_i^{omi})^2 + \delta_i^{001}\alpha_3^{001}(y_i - \theta_i^{sw})^2 \\
& + \delta_i^{110}\left(\alpha_1^{110}(y_i - \theta_i^{aqua})^2 + \alpha_2^{110}(y_i - \theta_i^{omi})^2\right) \\
& + \delta_i^{101}\left(\alpha_1^{101}(y_i - \theta_i^{aqua})^2 + \alpha_3^{101}(y_i - \theta_i^{sw})^2\right) \\
& + \delta_i^{011}\left(\alpha_2^{011}(y_i - \theta_i^{omi})^2 + \alpha_3^{011}(y_i - \theta_i^{sw})^2\right) \\
& + \delta_i^{111}\left(\alpha_1^{111}(y_i - \theta_i^{aqua})^2 + \alpha_2^{111}(y_i - \theta_i^{omi})^2\right. \\
& \left. + \alpha_3^{111}(y_i - \theta_i^{sw})^2\right) \\
& + \delta_i^{10}\alpha_4^{10}(y_i - \theta_i^{terra})^2 + \delta_i^{01}\alpha_5^{01}(y_i - \theta_i^{misr})^2 \\
& + \delta_i^{11}\left(\alpha_4^{11}(y_i - \theta_i^{terra})^2 + \alpha_5^{11}(y_i - \theta_i^{misr})^2\right),
\end{aligned}
$$

Results

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

Instrument	No. of points	Individual sensors	Without β parameters		Model II with β parameters		
			Model I	Model II	Only β_1	Only β_2	All β
MODIS Aqua	7,246	0.0872	0.0857	0.0840	0.0822	0.0825	**0.0756**
OMI	10,142	0.2390	0.2058	0.2053	0.1930	0.1482	**0.0934**
SeaWiFS	2,205	0.0739	0.0676	0.0658	0.0630	0.0642	**0.0607**
MODIS Aqua alone	2,102	0.0889	0.0889	0.0889	0.0850	0.0773	**0.0741**
OMI alone	4,528	0.2934	0.2934	0.2934	0.2744	0.2010	**0.1114**
SeaWiFS alone	237	0.0800	0.0800	0.0800	0.0771	0.0784	**0.0753**
MODIS + OMI	3,868	0.0893 | 0.2123	0.0918	0.0894	0.0886	0.0925	**0.0827**
MODIS + SeaWiFS	222	0.0982 | 0.0747	0.0694	0.0717	0.0665	0.0655	**0.0634**
OMI + SeaWiFS	692	0.1011 | 0.0835	0.0817	0.0799	0.0758	0.0760	**0.0691**
MODIS + OMI + SeaWiFS	1,054	0.0717 | 0.0715 | 0.0650	0.0523	0.0485	**0.0475**	0.0506	0.0495
MODIS Terra	8,725	0.0905	0.0847	0.0841	0.0826	0.0819	**0.0800**
MISR	2,165	**0.0652**	0.0684	0.0656	0.0664	0.0735	0.0725
MODIS Terra alone	7,552	0.0863	0.0863	0.0863	0.0845	0.0822	**0.0806**
MISR alone	992	**0.0618**	0.0618	0.0618	0.0636	0.0657	0.0677
MODIS + MISR	1,173	0.1142 | **0.0680**	0.0736	0.0687	0.0686	0.0795	0.0763
All labeled	44,445	–	–	–	0.1634	0.1328	**0.1080**
All labeled with any satellite	22,420	–	0.1516	0.1512	0.1430	0.1158	**0.0843**
All labeled without satellites	20,025	–	–	–	0.1817	0.1481	**0.1271**

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 α, except when all satellites are available (issue with OMI filter?)

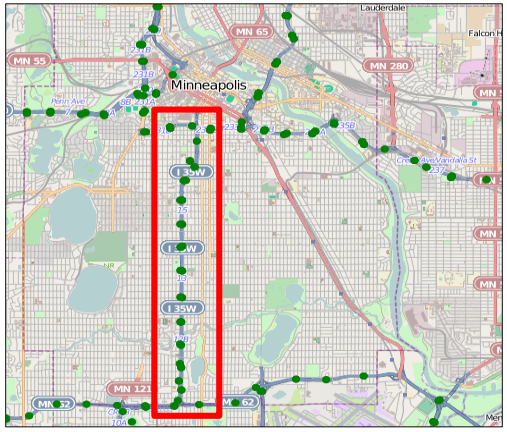
10:30*am*	*Count*	α_{modis}	α_{misr}
MODIS + MISR	2,334	–	183.03
MODIS + MISR	17,714	67.02	–
MODIS + MISR	2,572	0.03	100.48

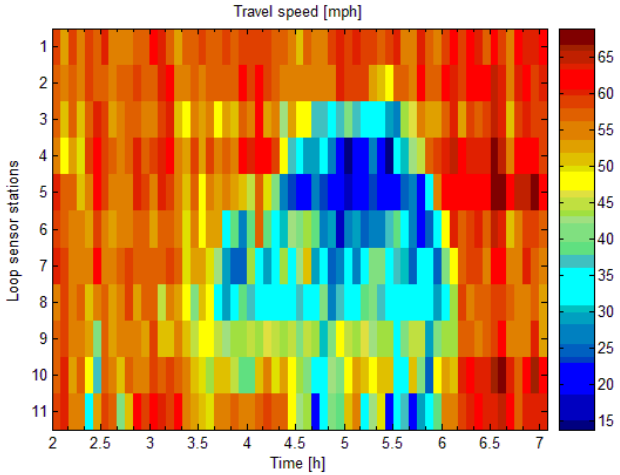
β_1	β_2
315.47	35.45

1:30*pm*	*Count*	α_{modis}	α_{omi}	α_{sw}
MODIS + OMI + SeaWiFS	570	–	–	87.40
MODIS + OMI + SeaWiFS	11,309	–	1.15	–
MODIS + OMI + SeaWiFS	5,589	97.96	–	–
MODIS + OMI + SeaWiFS	1,363	–	4.42	57.91
MODIS + OMI + SeaWiFS	461	16.39	–	80.47
MODIS + OMI + SeaWiFS	8,487	34.30	0.01	–
MODIS + OMI + SeaWiFS	2,173	20.22	82.47	109.78

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 1 Prediction MAE Errors of Various Predictors Aggregated Across Time Horizons

	Horizon (min)						
	+10	+20	+30	+40	+50	+60	Total
Linear Regression Models							
LR Model 1 (two baselines)	6.096	7.634	8.755	9.821	10.605	11.136	9.008
LR Model 2 (four baselines)	5.961	7.547	8.724	9.807	10.625	11.154	8.969
Baseline Predictors Incorporated into CCRF Models							
Random walk	6.130	7.667	8.789	10.033	11.072	11.947	9.273
Historical median	13.090	13.183	13.153	12.994	12.737	12.419	12.931
Current speed of upstream sensor	7.568	8.746	9.789	10.980	11.995	12.849	10.321
Current speed of downstream sensor	7.311	8.627	9.543	10.582	11.505	12.277	9.974
CCRF Models, in Increasing Level of Complexity							
CCRF Model 1 (two baselines)	6.198	7.703	8.752	9.778	10.597	11.239	9.004
CCRF Model 2 (four baselines)	5.929	7.325	8.333	9.384	10.290	11.061	8.720
CCRF Model 3 (regime switching)	5.922	7.327	8.329	9.363	10.214	10.891	8.675
CCRF Model 4 (with correlations)	5.920	7.308	8.314	9.352	10.213	10.905	8.669

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

Conclusions

- 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?