School of Computer Science
Carnegie Mellon

SAILING LAB
Laboratory for Statistical Artificial InteLligence & INtegrative Genomics

Grafting-Light: *Fast*, *Incremental* Feature Selection and Structure Learning of *MRFs*

Jun Zhu, Ni Lao, and Eric P. Xing

junzhu@cs.cmu.edu

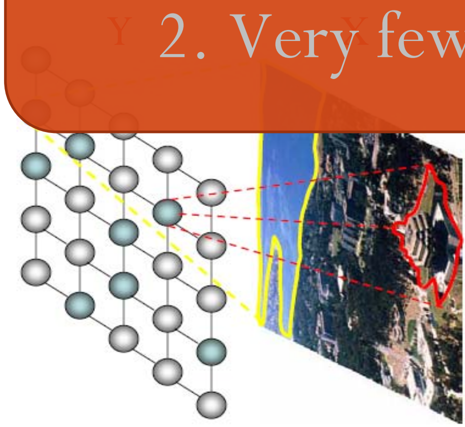
School of Computer Science, Carnegie Mellon University

(*Conditional*) Markov Random Fields

- Undirected GMs with sound theoretical foundation (probability + graph theory).
- Have been widely applied in many application domains:
  - Natural language processing [Sha & Pereria, 2003; Smith, 2008], Social network [Shi et al., 2009], Web mining [Zhu et al. 2006], Bioinformati[...] Computer vis[...], 2005], etc.
- Consider Conditional MRFs (CRFs) because of their superior performance [Lafferty et al., 2001]

An Ideal Algorithm for MRFs

1. Perform inference on a very sparse graph
2. Very few gradient calculation to converge



$$p(y|x) = \frac{1}{Z(x)} \exp\left\{ \sum_{c \in C} w^\top f(x; y_c) \right\}$$

MCLE (Max-Conditional Likelihood Estimation)

$$\min_w L(w) \triangleq -\sum_n \log p(y_n|x_n)$$

Gradient Computation is a Key & Difficult Step:

$$\frac{\partial L(w)}{\partial w_k} = \sum_{n;c} E_{p(y_c|x_n)}[f_k(x_n; y_c)] - \sum_{n;c} f_k(x_n; y_{n;c})$$

Expensive Subroutine (Infer marginal prob)

Hard on dense graphs; denser means more difficult!
Approximation: Loopy BP, Variational/MCMC.

Two Problems – FS & SL

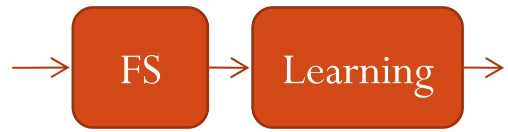
- Conditional MRFs (CRFs) can use arbitrary features
  - **E.g.**, in NP-chunking, the total number of features is *>3,000,000* (*N*-gram word and *N*-gram POS tags) [Sha & Pereria, 2003]
  - ***Feature Selection (FS)***: selecting a subset of features
    - **E.g.**, in NP-chunking, *99.9%* features can be discarded with *<1%* performance decrease in F1 score
    - FS in general is good for generalization and model interpretation

- Hand-crafting MRFs become less applicable as the variety and scale of problems increase
  - **E.g.**, in computer vision, it's hard to specify a structure among many patches (regions) in a pre-segmented image
  - ***Structure Learning (SL)***: learning the structures of MRFs
    - SL can automatically discover inherent structures underlying complex data

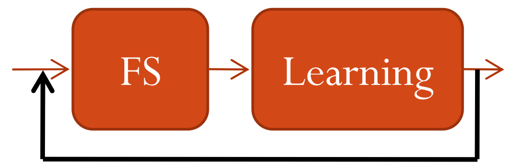
P1: Feature Selection (FS)

- FS in general:
  - Selecting an optimal set of features in NP-hard [Weston et al., 2003]

- Approximate approaches:
  - Filter methods [Kira & Rendell, 1992] (**Separate**)
    - Based on feature ranking (individual predictive power);
    - A pre-processing step and independent of prediction models (optimal under very strict assumptions!) [Guyon & Elisseeff, 2003]



  - Wrapper methods [Kohavi & John, 1997] (**Half-integrated**)
    - Use learning machine as a black box to score subsets of variables according to their predictive power
    - Can waste of resources to do many re-training!



  - Embedded methods (**Integrated**)
    - Perform FS during the process of training; Usually specific to given learning machines
    - Data efficient and Can avoid many re-training!

FS via L1-norm Regularized Opt.

- Solving a hybrid optimization problem:

$$\min_{M \in H} L(M) + \lambda \Omega(M)$$

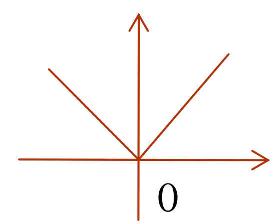
Goodness of fit: e.g., training error

Measure of model complexity: e.g., # of non-zero features

- In CRFs, we consider:
  - *M* is represented with natural parameters **w**

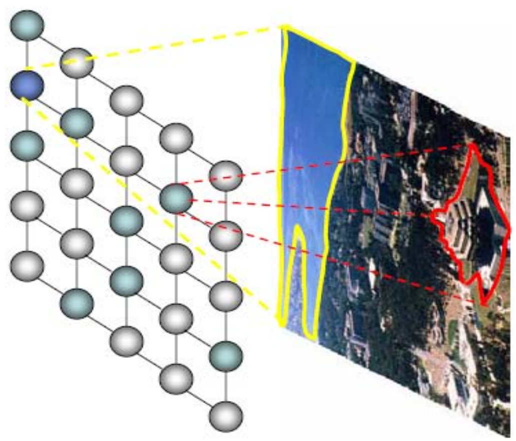
$$\min_{\mathbf{w} \in R^d} L(\mathbf{w}) + \lambda \Omega(\mathbf{w})$$

  - L(**w**) is the convex and 2nd-order differentiable log-loss
  - Ω(**w**) is the L1-norm, which is convex but singular at origin!

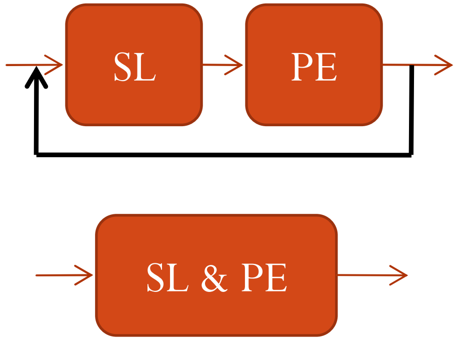


P2: Structure Learning (SL) of MRFs

- How is the graph structure constructed?

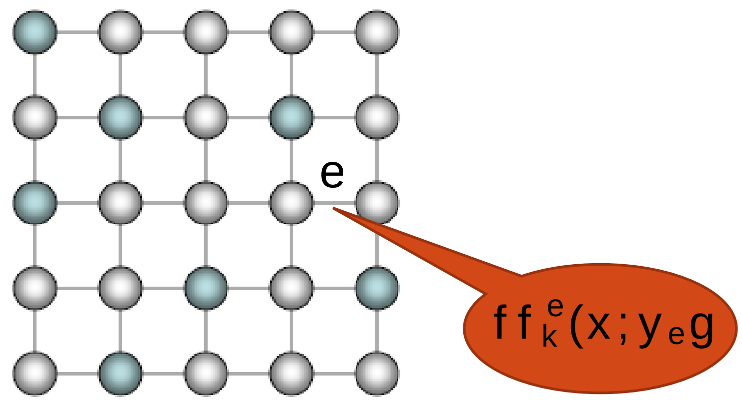


- Approximate Approaches:
  - Local heuristic search guided by a scoring function towards improving an objective function, e.g., marginal likelihood [Parise & Welling, 2006]
    - Need parameter estimation at each step
  - SL as solving an L1-regularized MCLE problem [Lee et al., 2006; Wainwright et al., 2006]
    - Joint parameter estimation and structure learning



SL via L1-norm Regularized Opt.

- Each possible edge ***e*** is associated with a set of feature functions $\{f_k^e(x, y_e)\}$



- Perform feature selection by solving L1-regularized MCLE
- If the weights of $\{f_k^e(x, y_e)\}$ are zero, the edge ***e*** doesn't exist

$$\min_{w \in R^d} L(w) + \lambda \|w\|$$

- ***Consider all features together will result in a complete graph!***

Solving the L1-regularized Opt. in MRFs

$$\min_{w \in R^d} \mathcal{L}(w) \triangleq L(w) + \lambda \|w\|$$

An Ideal Algorithm for MRFs

1. Perform inference on a very sparse graph
2. Very few gradient calculation to converge

- Batch Methods (all features considered together):
  - Many examples:
    - Quasi-Newton gradient descent methods (OWL-QN) [Andrew & Gao, 2007]
    - Gradient descent + L1-ball projection [Duchi et al., 2008]
    - Stochastic gradient descent [Vishvanathan et al., 2006; Tsuruoka et al., 2009]
    - Gauss-Seidel co-ordinate descent [Shevade & Keerthi, 2003]
  - Can scale up to millions of features, e.g., OWL-QN
  - Not applicable for structure learning
    - Inference on complete graphs can be extremely slow and inaccurate!

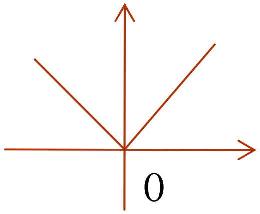
- Incremental Methods:
  - Start from simple (sparse) model, iteratively add new features
  - Example: Grafting [Perkins et al., 2003]

Grafting-Light

Fast, Incremental Algorithm

Grafting-Light

$$\min_{w \in R^d} L(w) \triangleq L(w) + \lambda \|w\|$$



- ***Two-step iterative procedure***
  - One-step orthant-wise gradient descent over working set ***S***

$$\diamond L(w) = \begin{cases} \partial_k L(w) + \lambda\, \mathrm{sgn}(w_k); & w_k \neq 0 \\ \partial_k L(w) + \lambda; & w_k = 0, \partial_k L(w) < -\lambda \\ \partial_k L(w) - \lambda; & w_k = 0, \partial_k L(w) > \lambda \\ 0; & w_k = 0, |\partial_k L(w)| \leq \lambda \end{cases}$$

  - Select top ***M*** features from the set *G* and add them to ***S***

$$G = \{ f_k : f_k \in U, \text{ and } |\partial_k L(w)| > \lambda \}$$

- $L(w)$ ***is differentiable at one orthant***
  - Choose an orthant into which $\diamond L(w^t)$ leads

$$\forall k, \ \xi_k = \begin{cases} \mathrm{sgn}(w_k); & w_k \neq 0 \\ \mathrm{sgn}(-\diamond L(w)); & w_k = 0 \end{cases}$$

  - Choose a step-size with backtracking line search

$$d^t = \pi(H_t p^t; \xi)$$

  - Update model weights

$$w^{t+1} = \pi(w^t + \alpha d^t; \xi)$$

$$\forall k, \ \pi_k(\mu; v) = \begin{cases} \mu_k; & \mathrm{sgn}(\mu_k) = \mathrm{sgn}(v_k) \\ 0; & \text{otherwise} \end{cases}$$

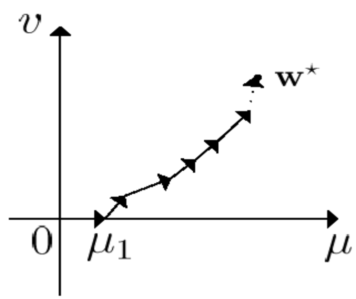
- ***M*** is the ***Select Unit***
- Choose from inactive features that violate the optimal conditions

$$\forall k, \ \begin{cases} \partial_k L(w) + \lambda\, \mathrm{sgn}(w_k) = 0; & w_k \neq 0 \\ |\partial_k L(w)| \leq \lambda; & \text{otherwise} \end{cases}$$

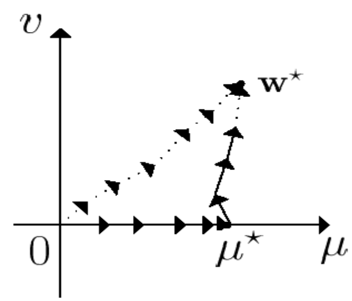
Grafting-Light

- **Thrm**: when $L(\mathrm{w})$ is convex, bounded below, and continuously differentiable, Grafting-Light converges to the global optimum.

- Connections to existing algorithms:
  - A lazy version of the incremental Grafting (***converge faster!***)



Grafting-Light



Grafting

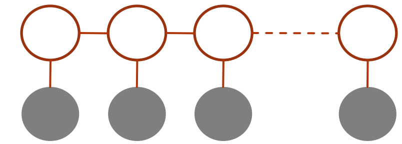
  - An incremental version of the batch OWL-QN [Andrew & Gao, 2007] (***suitable for learning structures of MRFs***)

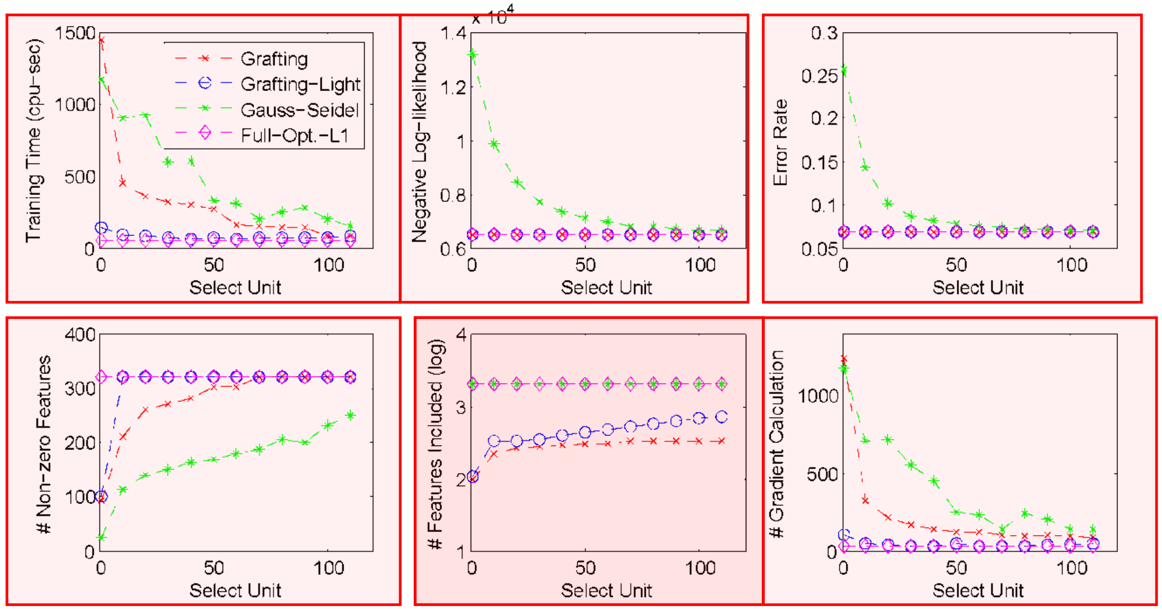
Experimental Results

- Tasks:
  - Synthetic data on sequence labeling
  - NP Chunking on CoNLL-2000 data
  - Structure learning of MRFs on OCR characters
- Algorithms to compare:
  - ***Incremental*** Grafting [Perkins et al., 2003]
  - ***Batch*** quasi-Newton method [Andrew & Gao, 2007] (Full-L1-Opt.)
  - ***Batch*** co-ordinate Gauss-Seidel [Shevade & Keerthi, 2003]
- Implementation
  - Standard PC with Intel 2.00 GHz processor
  - C++ programming language

Synthetic Sequence Labeling

- **# Features:** **2000 state features + 4 pairwise dependency features**
- **Linear-Chain CRFs:** **Gradients and Objective can be exactly computed**

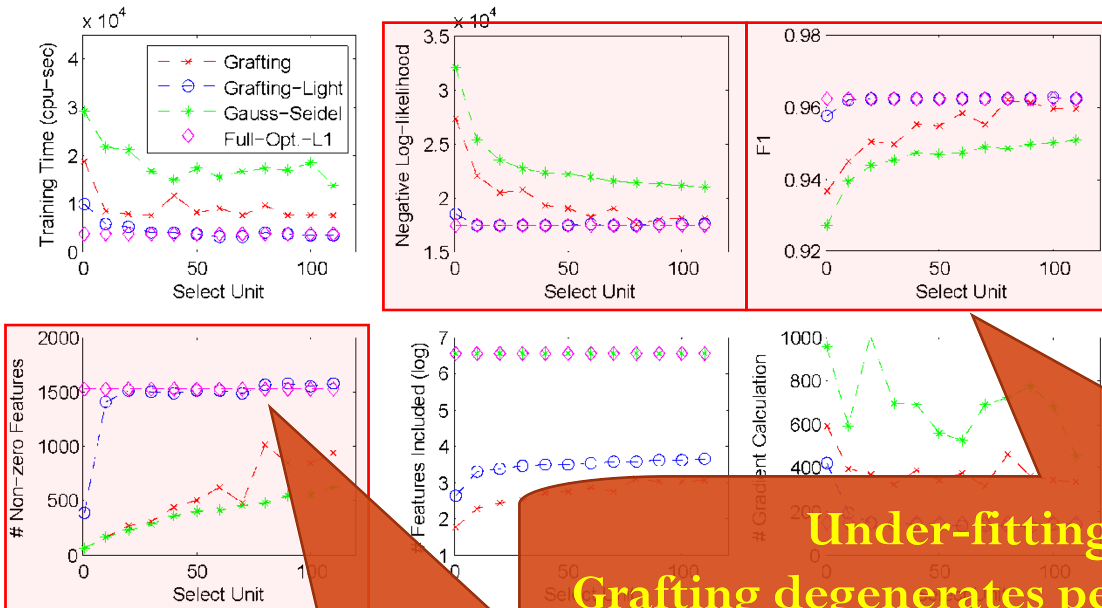




- Grafting-L performs as good as optimal Full-Opt-L1 (exact gradient and all info used! Expected to be fasted!)
- Grafting-L is much more efficient than greedy Grafting and co-ordinate Gauss-Seidel (fewer number of gradient computation).
- During training, Grafting-L may include redundant features, but these can be effectively removed when converge!
- Greedy Grafting and Gauss-Seidel can under-fit the data, i.e., selecting fewer number of features.

NP-Chunking on CoNLL-2000

- **# Features:** > 3M (e.g., unigram, bigram word pairs and POS tag pairs, etc.) [Sha & Pereria, 2003]
- **Linear-Chain CRFs:** Gradients and Objective function can be exactly computed by using message-passing



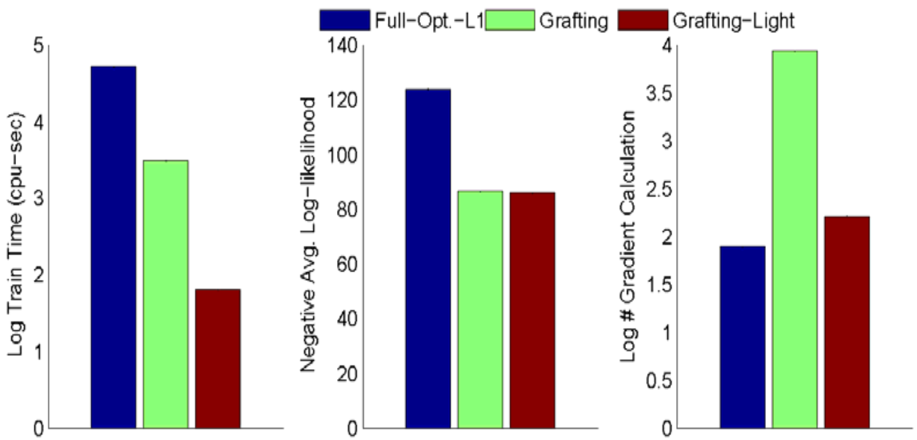
Under-fitting: Grafting degenerates performance!

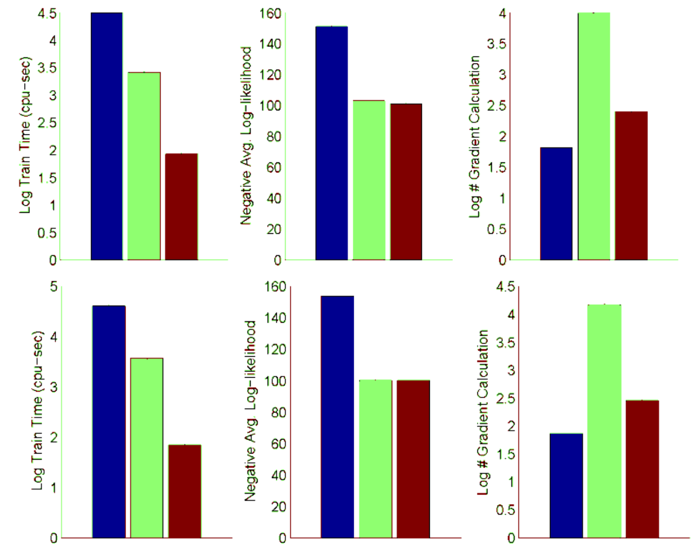
99.9% features can be discarded!

- Grafting-L perfo... Full-Opt-L1,(exact gradient computation!)
- Grafting-L is much more efficient than greedy Grafting and co-ordinate Gauss-Seidel (fewer number of gradient computation).
- During training, Grafting-L may include redundant features, but these can be effectively removed when converge!
- Greedy Grafting can under-fit the data, i.e., selecting fewer number of features and *degenerate the performance*

Structure Learning of MRFs

- Performance of different methods on different OCR characters, e.g., S, I, G:
  - 20 x 20 images; Total features: **>80,000**

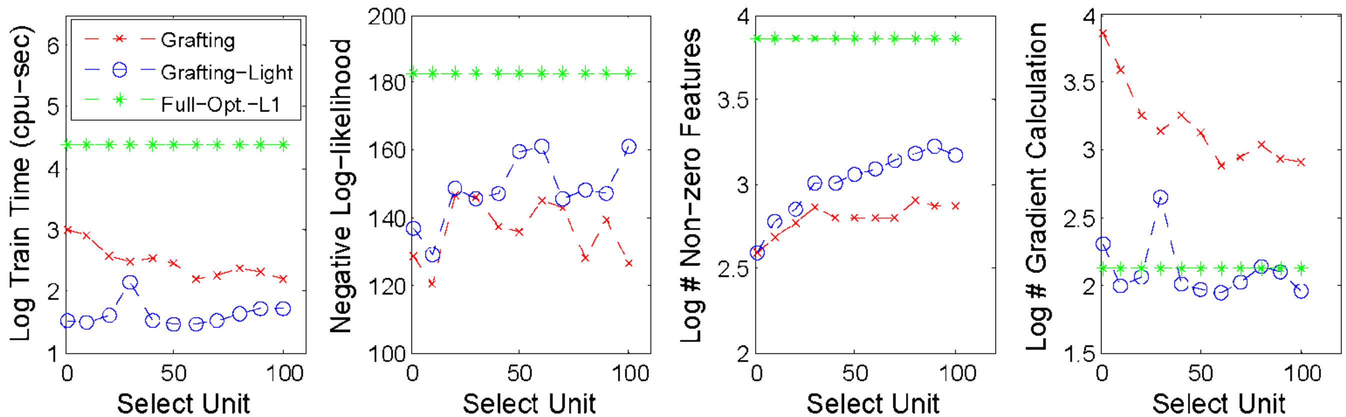




- Grafting-Light is consistently more efficient than Grafting and Full-Opt.-L1
  - Greedy Grafting needs much more number of gradient computation
  - Gradient computation in Full-Opt.-L1 is expensive due to the difficult inference on complete graph
- Incremental methods consistently more efficient and accurate than batch methods
  - Full-Opt.-L1 do expensive inference on complete graphs and gradients can be very inaccurate!

Structure Learning of MRFs

- Performance change against Select-Unit (# features selected at each iteration)



- Grafting-Light is consistently more efficient than Grafting and Full-Opt.-L1
  - Greedy Grafting needs much more number of gradient computation
  - Gradient computation in Full-Opt.-L1 is expensive due to the difficult inference on complete graph
- Incremental
  - Full-Opt.-L1 do expensive inference on complete graphs and gradients can be very inaccurate!

The batch Full-Opt.-L1 doesn't achieve sparse structures because of inaccurate gradients!

Structure Learning of MRFs

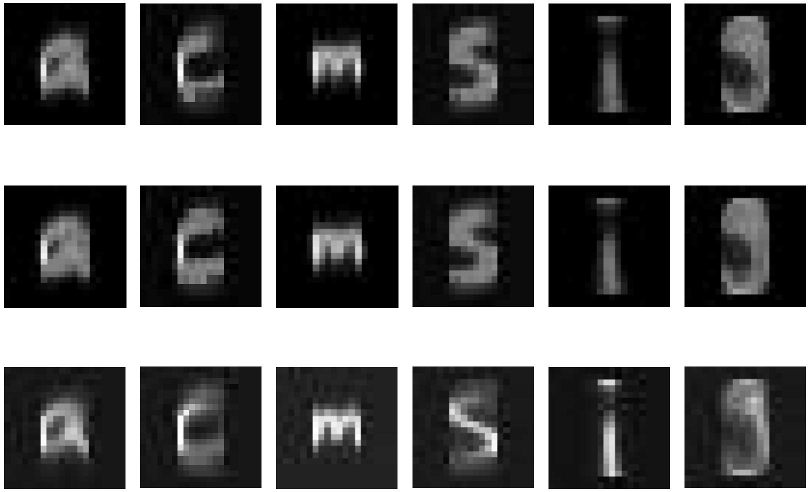
- Average image produced from the learned model by different algorithms

"ACMSIG"

Grafting-Light:

Grafting:

Full-Opt.L1:



- The batch Full-Opt.-L1 produces blurry images because of ***inaccurate gradient computation*** on complete graphs (Non-sparse results!)

Conclusions & Future Work

- Conclusions:
  - We present Grafting-Light: a fast, incremental algorithm for solving the L1-regularized MLE for FS and SL of MRFs
  - We show that:
    - Incremental methods are better than batch methods for feature selection and structure learning of MRFs
    - Message-passing on complete graphs can lead to inaccurate gradients or marginals, which are not good for feature selection or structure learning
    - Grafting-Light is more efficient than the greedy Grafting algorithm
- Future Work:
  - Convergence rate and time complexity analysis
  - Apply to solve non-convex problems, e.g., learning structures of MRFs with latent variables
  - Regularization path analysis and comparison with more existing methods, e.g., stochastic gradient descent, etc.

Thank you!

Poster ID: 11