10-708: Probabilistic Graphical Models, Spring 2015

23 : Max-margin Learning of Graphical Models

Lecturer: Eric P. Xing *Scribes: Lili Gao, (Adams) Wei Yu, Xun Zheng*

1 Classical Predictive Models

In most of the binary classification tasks, we adopt the following model:

- Input and output space: $\mathcal{X} \triangleq \mathbb{R}^{M_x}, \mathcal{Y} \triangleq \{-1, +1\}$
- Predictive function $h(x) : y^* = h(x) \triangleq \arg\max_{y \in \mathcal{Y}} F(x, y; \mathbf{w})$
- Example of the function $F(x, y; \mathbf{w})$:

$$F(x, y; \mathbf{w}) = g(\mathbf{w}^T f(x, y))$$

- The parameter learning is the following optimization problem:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w} \in \mathcal{W}} \ell(x, y; \mathbf{w}) + \lambda R(\mathbf{w})$$

  where $\ell(\cdot)$ is a convex loss function, such as ℓ_2-loss and $R(\cdot)$ is a regularizer such as ℓ_1 norm.

There are two typical settings that fall in this category of framework:

1. Logistic Regression:
   - It is essentially a Maximum likelihood estimation:

$$\max_{\mathbf{w}} \quad \mathcal{L}(\mathcal{D} : \mathbf{w}) \triangleq \sum_{i=1}^{N} \log p(y^i | x^i; \mathbf{w}) + \mathcal{N}(\mathbf{w})$$

   - It corresponds to the a Log loss with ℓ_2 regularizer.

$$\ell(x, y; \mathbf{w}) \triangleq \ln \sum_{y' \in \mathcal{Y}} \exp\{\mathbf{w}^T f(x, y')\} - \mathbf{w}^T f(x, y)$$

   - The advantages are three-fold: i) It has full probabilistic interpretation. ii) It enables straightforward Bayesian or direct regularization. iii) It has hidden structures or generative hierarchy.
2. Support vector machine:
   - It is the max-margin learning with the following formulation:

$$\begin{aligned} \min_{\mathbf{w}, \xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \forall i, \forall y' = y^i : \mathbf{w}^T \Delta f_i(y') \geq 1 - \xi_i \end{aligned}$$

- It corresponds to a hinge loss of with ℓ_2 regularizer:

$$\ell(x, y; \mathbf{w}) \triangleq \max_{y' \in \mathcal{Y}} \mathbf{w}^T f(x, y') - \mathbf{w}^T f(x, y') + \ell'(y', y)$$

- The advantages are three-fold: i) Its dual solution is usually sparse, which corresponds to sparseness of the support vectors. ii) It enables the kernel tricks by replacing the inner products with kernels functions. iii) It usually has strong empirical results.

2 Structured Prediction Problem

The problems stated in the previous section are unstructured prediction problems, whose input is usually a single data point with a corresponding label.

On the other hand, there is another type of problem, i.e. the structured prediction, in which the input is usually a sequence and the output is the corresponding labels sequence. There are several examples of such kind of problems:

- The first example is in the Part of speech (POS) tagging problem, the input is a sentence, such as "Do you want sugar in it?" and the corresponding label sequence is the POS tagging, such as "verb pron verb noun prep pron".
- Another example lies in the Optical character recognition (OCR) problem (lhs of Figure. 1), where the input is a sequence of hand-written objects and the output is their corresponding characters.
- The third example is in image segmentation, in which one wants to jointly segment and annotate images (see rhs of Figure. 1).

There two key problems related to structured prediction: 1) Given structure (feature), how to learn the model parameter θ. 2) How to learn sparse, interpretable and predictive structures/features.

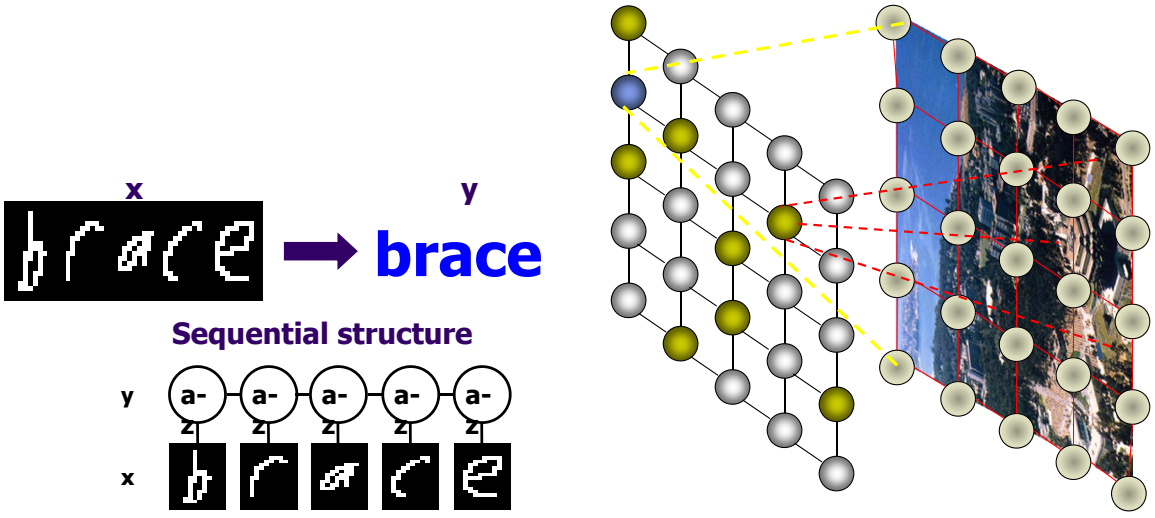

Figure 1: Examples of structured prediction: lhs is OCR and rhs is image segmentation.

3 Structured Prediction Graphical Models

3.1 Classical Examples

We are ready to present several structured prediction graphical models. Note that now the input and output spaces are respectively $\mathcal{X} \triangleq \mathbb{R}_{X_1} \times ...\mathbb{R}_{X_K}$ and $\mathcal{Y} \triangleq \mathbb{R}_{Y_1} \times ...\mathbb{R}_{Y_K}$.

There are two typical Structured Prediction Graphical Models:

1. Conditional Random Fields
   - Based on Logistic Regression Loss
   - It is a maximum likelihood estimation (point-estimate) with the following loss function.

$$\mathcal{L}(\mathcal{D}; \mathbf{w}) \triangleq \log \sum_{y'} \exp(\mathbf{w}^T f(x, y')) - \mathbf{w}^T f(x, y)$$

2. Max-margin Markov Network (M^3Ns)
   - Based on a Hinge Loss
   - It is a Max-margin learning (point-estimate) of the following loss function

$$\mathcal{L}(\mathcal{D}; \mathbf{w}) =\triangleq \log \max_{y'} \mathbf{w}^T f(x, y') - \mathbf{w}^T f(x, y) + \ell(y', y)$$

In both models above, the Markov properties are encoded in the feature functions $f(x, y)$.

3.2 Challenges

There are several challenges associating with the these problems:

- How to learn a sparse interpretable prediction model?
- How to leverage the prior which encodes the information of structures?
- How to learn the latent structures and variables?
- How to deal with the time series without stationarity?
- How to scale it up to large problems?

3.3 Formulation

The structured prediction problem could be modeled as follows:

$$h(x) = \arg \max_{y \in \mathcal{Y}(x)} s(x, y)$$

where $s(x, y)$ is a scoring function and $\mathcal{Y}(x)$ is the feasible outputs. Most of the time, we assume the score function has the following form:

$$s(x, y) = \mathbf{w}^T f(x, y) = \sum_p \mathbf{w}^T f(x_p, y_p)$$

which is essentially a linear combination of the features, and index p represents a part in the structure.

4 Large Margin Estimation

In the large margin estimation problem, we actually want the following: Given training example (x, y^*), we would like to have

$$\arg\max_y \mathbf{w}^T f(x, y) = y^*$$

and

$$\mathbf{w}^T f(x, y^*) > \mathbf{w}^T f(x, y), \quad \forall y \neq y^*$$

or

$$\mathbf{w}^T f(x, y^*) > \mathbf{w}^T f(x, y) + \gamma \ell(y^*, y), \quad \forall y$$

where γ is the maximum margin and

$$\ell(y^*, y) = \sum_i I(y_i^* \neq y_i)$$

is the number of mistakes in y.

Recall that from SVM, Maximizing margin γ is to minimizing the square of the ℓ_2 norm of the weight vector $\mathbf{w}$. Then we have the following new objective function:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & \mathbf{w}^T f(x_i, y_i) > \mathbf{w}^T f(x_i, y_i') + \ell(y_i, y_i'), \quad \forall i, y_i' \in \mathcal{Y}_i \end{aligned}$$

However, the number of constraints in this formulation is exponential to the size of the structure. Can we reduce the number of constraints? Consider the OCR example. Among all possible label combinations, *e.g.*, "aaaaa", "aaaab", etc, some of them are wrong but close (*e.g.*, "brare"), whereas most of them are far from the truth (*e.g.*, "zzzzz"). Then a natural idea would be to consider the most violated constraints only:

$$\min_{\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{1}$$

$$\text{s.t.} \quad \mathbf{w}^\top f(x, y^*) \geq \max_{y \neq y^*} \mathbf{w}^\top f(x, y) + \ell(y^*, y). \tag{2}$$

This formulation reduces the number of constraints from exponential to polynomial.

However, the resulting constraint is not easy to deal with: it is discrete due to the max function over all possible labels. Good news is this can be reparameterized into a continuous problem. The key step is to use binary indicators $\mathbf{z}$ to represent the label codebook $\mathbf{y}$. Then the maximization can be rewritten in terms of indicator variables:

$$\max_{\mathbf{z}} \sum_{j,m} z_j(m) \left[\mathbf{w}^\top f_{\text{node}}(x_j, m) + \ell_j(m)\right] + \sum_{jk,m,m} z_{jk}(m, n) \left[\mathbf{w}^\top f_{\text{edge}}(x_{jk}, m, n) + \ell_{jk}(m, n)\right] \tag{3}$$

$$\text{s.t.} \quad \sum_m z_j(m) = 1, \quad \sum_n z_{jk}(m, n) = z_j(m). \tag{4}$$

Notice that the constraints are similar to the ones in LBP (singleton marginals and pairwise marginals should be consistent).

In a more compact form,

$$\max_{A\mathbf{z}=b} (F^\top \mathbf{w} + \ell)^\top \mathbf{z} \equiv q^\top \mathbf{z}. \tag{5}$$

Making use of duality, we have

$$\max_{\mathbf{z}\geq 0, A\mathbf{z}=b} q^\top \mathbf{z} = \min_{A^\top \mu \geq q} b^\top \mu. \tag{6}$$

Therefore the original problem now becomes

$$\min_{\mathbf{w},\mu} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{7}$$

$$\text{s.t.} \quad \mathbf{w}^\top f(x, y^*) \geq b^\top \mu \tag{8}$$

$$A^\top \mu \geq q, \tag{9}$$

which is a standard QP.

5 MED Markov Nets

Introduced by Jaakkola et al, maximum entropy discrimination (MED) aims to generalized max-margin principal to a Bayesian setting. Instead of learning a point-estimate of $\mathbf{w}$, it learns a distribution of $\mathbf{w}$ by finding a proximal point from the prior that satisfies the constraints:

$$\min_{q(\Theta)} \quad \mathrm{KL}\left[p(\Theta)\|p_0(\Theta)\right] \tag{10}$$

$$\text{s.t.} \quad \int p(\Theta)\left[y_i F(x;\mathbf{w}) - \xi_i\right] \mathrm{d}\Theta \geq 0, \forall i \tag{11}$$

where $\Theta = \{\mathbf{w}, \boldsymbol{\xi}\}$. At test time, similar to most Bayesian settings, uncertainties are integrated out, leaving only the predictive distribution:

$$\widehat{y} = \text{sign} \int p(\mathbf{w}) F(x;\mathbf{w}) \,\mathrm{d}\mathbf{w} \tag{12}$$

Applying this idea to M^3N, we have Structured MED:

$$\min_{p(\mathbf{w}),\boldsymbol{\xi}} \quad \mathrm{KL}\left[p(\mathbf{w})\|p_0(\mathbf{w})\right] + U(\boldsymbol{\xi}) \tag{13}$$

$$\text{s.t.} \quad p(\mathbf{w}) \in \mathcal{F}, \xi_i \geq 0, \forall i, \tag{14}$$

$$\mathcal{F} = \left\{ p(\mathbf{w}) : \int p(\mathbf{w})\left[\Delta F_i(y;\mathbf{w}) - \Delta\ell_i(y)\right] \mathrm{d}\mathbf{w} \geq -\xi_i, \forall i, \forall y \neq y^i \right\}. \tag{15}$$

Similarly at test time,

$$h(x; p(\mathbf{w})) = \operatorname*{argmax}_{y\in\mathcal{Y}(x)} \int p(\mathbf{w}) F(x, y; \mathbf{w}) \,\mathrm{d}\mathbf{w}. \tag{16}$$

Using duality theory, the solution to the SMED has the form

$$p(\mathbf{w}) = \frac{1}{Z} p_0(\mathbf{w}) \exp\left\{ \sum_{i,y} \alpha_i(y)\left[\Delta F_i(y;\mathbf{w}) - \Delta\ell_i(y)\right] \right\}, \tag{17}$$

where the Z is the normalization constant and $\alpha_i(y)$ are the solutions of the following dual problem:

$$\max_{\alpha} \quad -\log Z - U^*(\alpha) \tag{18}$$

$$\text{s.t.} \quad \alpha_i(y) \geq 0, \forall i, \forall y, \tag{19}$$

and $U^*(\alpha) = \sup_\xi \sum_{i,y} \alpha_i(y)\xi_i - U(\xi)$ is the Fenchel conjugate of $U(\alpha)$.

6 Gaussian MaxEnDNet

Assume that $F(x,y;w) = w^T f(x,y)$, $U(\xi) = C\sum_i \xi_i$, and standard Gaussian prior $p_0(w) = \mathcal{N}(w|0,I)$, then we can get the posterior distribution of w as

$$p(w) = \mathcal{N}(w|\mu_w, I), \text{ where } \mu_w = \sum_{i,y} \alpha_i(y)\Delta f_i(y)$$

The weights α_i are computed in the same way as in the structured SVM M^3N through dual optimization

$$\max_{\alpha} \sum_{i,y} \alpha_i(y)\Delta l_i(y) - \frac{1}{2}\left|\left|\sum_{i,y}\alpha_i(y)\Delta f_i(y)\right|\right|^2$$

$$s.t. \sum_y \alpha_i(y) = C;\ \alpha_i(y) \geq 0, \forall i, \forall y$$

The predictive rule becomes

$$h_1(x) = \arg\max_{y\in\mathcal{Y}(x)} w^T f(x,y)$$

Therefore, the Gaussian MaxEnDNet is a probabilistic version of M^3N that computes that posterior probability of the weights w, and it admits all the merits of max-margin learning, while it is a more general framework that can take forms of M^3N while also provides the advantage of probabilistic models.

MaxEnDNEt has at least three advantages, which includes the PAC-Bayesian prediction guarantee offered by an averaging Model, the ability to introducing useful biases through entropy regularization, and integrating generative and discriminative principles.

7 Laplace MaxEnDNet (LapMEDN)

The Laplace prior is specified as

$$p_0(w) = \prod_{k=1}^{K} \frac{\sqrt{\lambda}}{2} e^{-\sqrt{\lambda}|w_k|} = \left(\frac{\sqrt{\lambda}}{2}\right)^K e^{-\sqrt{\lambda}||w||}$$

The nature of this prior has a regularization effect over the weights w, resulting in an l_1 shrunk version of M^3N. The parameter λ controls the values of the coefficients. The model becomes more regularized as λ increases. This can be seen through the posterior mean of w under Laplace prior

$$\forall k, \langle w_k \rangle_p = \frac{2\eta_k}{\lambda - \eta_k^2}$$

where η is a linear combination of the support vectors

$$\eta = \sum_{\alpha} \alpha_i(y)\Delta f_i(y)$$

while Gaussian MaxEnDNet and the regular M^3N does not have such shrinkage.

the corresponding primal optimization problem is

$$\min_{\mu,\xi} \sqrt{\lambda}\sum_{k=1}^{K}\left(\sqrt{\mu_k^2 + \frac{1}{\lambda}} - \frac{1}{\sqrt{\lambda}}\log\frac{\sqrt{\lambda\mu_k^2+1}+1}{2}\right) + C\sum_{i=1}^{N}\xi_i$$

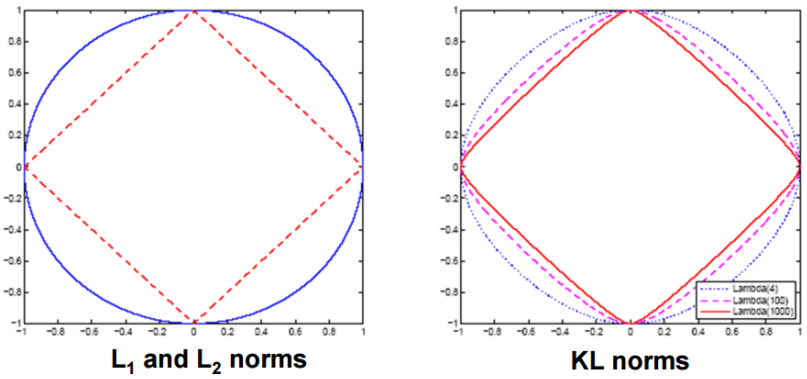

Figure 2: The KL norm on the right allows for a smooth transition between the l_1 and l_2 norms (left) determined by λ

$$\text{s.t. } \mu^T \Delta f_i(y) \geq \Delta l_i(y) - \xi_i, \xi_i \geq 0, \forall i, \forall y \neq y^i$$

The KL norm $||\mu||_{KL} \equiv \sum_{k=1}^{K} \left(\sqrt{\mu_k^2 + \frac{1}{\lambda}} - \frac{1}{\sqrt{\lambda}} \log \frac{\sqrt{\lambda \mu_k^2 + 1} + 1}{2} \right)$ imposed by this model is parametrized by λ and can be adjusted to represent any intermediate step between an equivalent l_1 norm and and l_2 norm shown in Figure 2.

The LapMEDN is a model that is primal sparse due to the Laplace shrinkage effect. In addition, this model is also dual sparse as the M^3N due to being a maximum-margin Markov network. This means that the posterior is decided by a controlled number of support vectors, an aspect that efficiently selects only the most important features. The dual sparsity of M^3N can be seen by recalling its dual problem

$$\max_{\alpha} \sum_{i,y} \alpha_i(y) \Delta l_i(y) - \frac{1}{2} \eta^T \eta$$

$$\text{s.t. } \forall i, \forall y : \sum_{y} \alpha_i(y) = C, \alpha_i(y) \geq 0, \text{ where } \eta = \sum_{i,y} \alpha_i(y) \Delta f_i(y)$$

The exact primal or dual function of LapMEDN is hard to optimize and thus we can use the hierarchical representation of Laplace prior to get

$$KL(p||p_0) = -H(p) - \left\langle \log \int p(w|\tau) p(\tau|\lambda) d\tau \right\rangle_p$$

$$\leq -H(p) - \left\langle \int q(\tau) \frac{\log p(w|\tau) p(\tau|\lambda)}{q(\tau)} d\tau \right\rangle_p \equiv \mathcal{L}(p(w), q(\tau))$$

and optimize the upper bound

$$\min_{p(w) \in \mathcal{F}_1, q(\tau), \xi} \mathcal{L}(p(w), q(\tau)) + U(\xi)$$

This optimization problem is easier because we can take a alternating minimization procedure to get nice optimization problems:

First, we keep $q(\tau)$ fixed and effective prior is normal $\forall k$, $p_0(w_k|\tau_k) = \mathcal{N}\left(w_k|0, \left\langle \frac{1}{\tau_k} \right\rangle_{q(\tau)}^{-1}\right)$; second, we keep $p(w)$ fixed and get closed form solution of $q(\tau)$ and its expectation $\left\langle \frac{1}{\tau_k} \right\rangle_q = \sqrt{\frac{\lambda}{\langle w_k \rangle_p}}$.

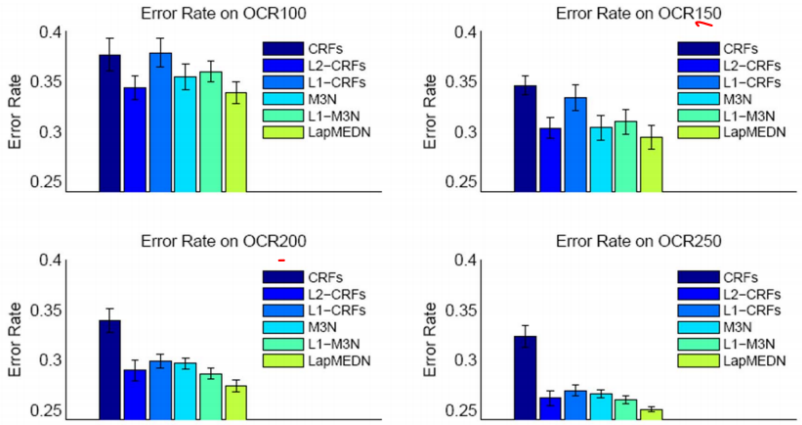

Figure 3: Error rate of different classification algorithms on four different randomly constructed Optical Character Recognition (OCR) problems.

Figure 3 shows the comparison in accuracy of the LapMEDN in the problem of Optical Character Recognition (OCR). The algorithm is compared to ordinary and regularized (in l_1 and l_2 norm) versions of Conditional Random Fields (CRF) and M^3N. This comparison shows that LapMEDN achieves the smallest error rate of all the four scenarios.

Figure 4 shows the actual features used for generating the model and the features selected by each of the algorithms. We can see the improved response of LapMEDN over its competing algorithms through its ability to efficiently assigns weights of zero to irrelevant features while maintaining a dominant performance.

Figure 5 shows the sensitivity to regularization constants of different algorithms. L_1-CRF are much more sensitive to regularization constants than others, and LapMEDN is the most stable.

Figure 6 shows the relationship between different margin-based learning paradigms.

8 Elements of Learning

- Here are some important elements to consider before you start:
  - Task:
    - Embedding? Classification? Clustering? Topic extraction? ...
- Data and other info:
  - Input and output (e.g., continuous, binary, counts, ...)
  - Supervised or unsupervised, of a blend of everything?
  - Prior knowledge? Bias?
- Models and paradigms:
  - BN? MRF? Regression? SVM?
  - Bayesian/Frequents ? Parametric/Nonparametric?
- Objective/Loss function:

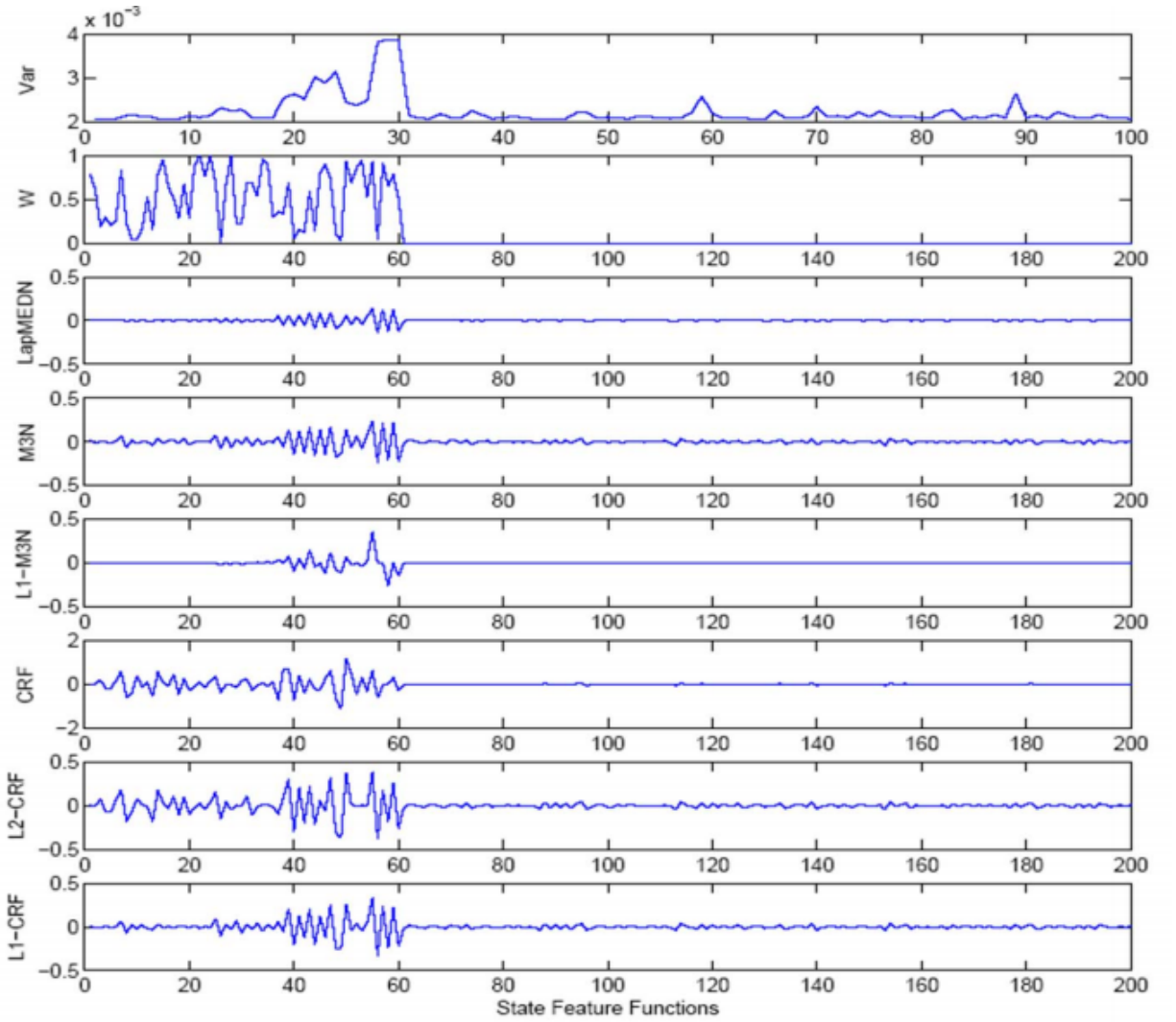

Figure 4: Feature selection in the OCR problem. Var and W at the top two panels are the parameters of the model used to generate the data. The other panels show the features selected by each of the competing algorithms.

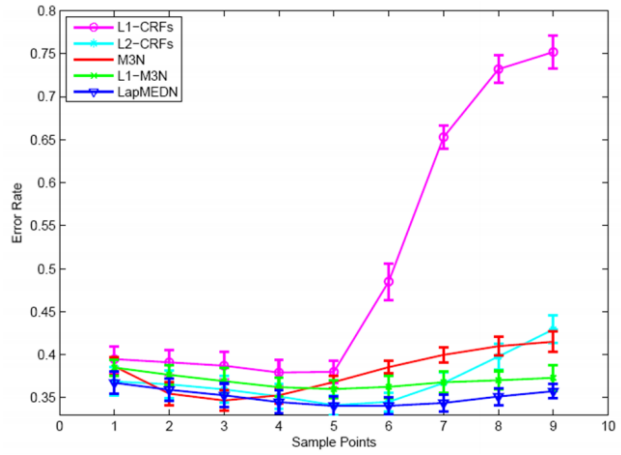

Figure 5: Sensitivity to regularization constants of different algorithms. L_1-CRF are much more sensitive to regularization constants than others, and LapMEDN is the most stable.

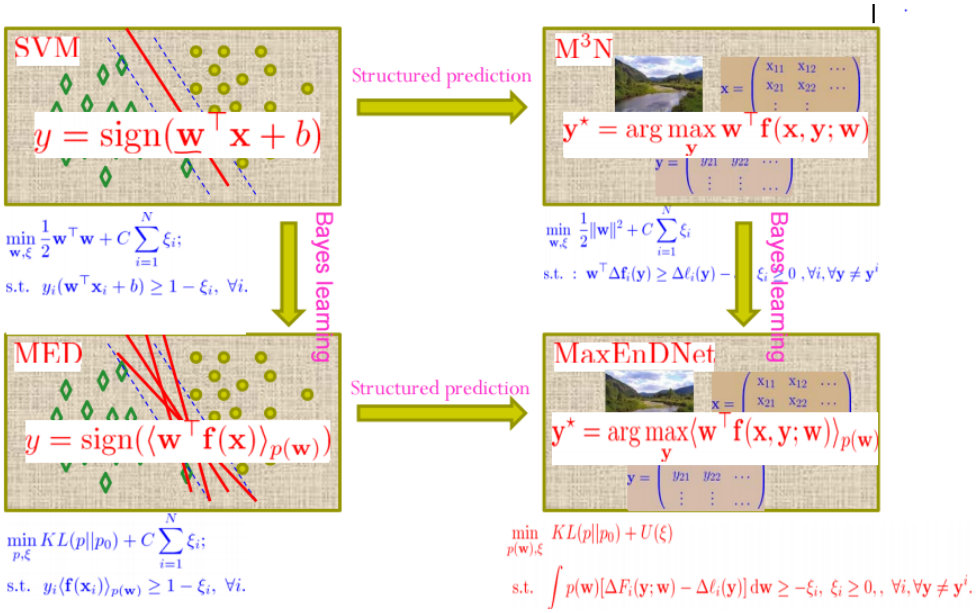

Figure 6: Relationship between different margin-based learning paradigms.

  - MLE? MCLE? Max margin?
  - Log loss, hinge loss, square loss? ...

- Tractability and exactness trade off:
  - Exact inference? MCMC? Variational? Gradient? Greedy search?
  - Online? Batch? Distributed?

- Evaluation:
  - Visualization? Human interpretability? Perperlexity? Predictive accuracy?

- It is better to consider one element at a time!