

2008/2/12 T-FaNT2@University of Tokyo

Dualized L_1 -regularized Log-linear Models and Its Application in NLP

Daisuke Okanohara

D1 Tsujii Lab

University of Tokyo

Abstract

- A dual representation of L_1 -regularized Log-linear Model (L_1 -LL)
 - A simple and efficient SMO optimization method
- An efficient method to extract effective combination of features on a dual representation

- **Background**
- Dual L_1 -regularized log-linear model
 - Derivation
 - SMO algorithm
 - Recovery of primal parameters
 - Extraction of feature combination
- Experiments
- Conclusion

Background

- Log-linear models (LL) are widely used for many NLP tasks
 - Syntactic Parsing, Document Classification, Sequential Labeling
 - Note: LL includes many models, such as maximum entropy models, conditional random fields, logistic regressions
- L_1 -norm regularization: obtain sparse feature set
 - # candidate features in NLP is generally several millions.
 - sparse feature set = interpretable, fast, space-efficient

Background (cont.)

- Extraction of feature combination is still important for linear classifiers
 - $\Phi_{k12}(x,y) = \Phi_{k1}(x,y) \Phi_{k2}(x,y)$
 - Feature engineering is done by hand
 - require deep knowledge about tasks
 - Although kernel methods can employ feature combinations implicitly, they are very heavy and black box
 - In particular in LL (Kernel Logistic Regression), all training data become the support vectors

Log-linear Model (LLM)

$$p(y | x; \mathbf{w}) = \frac{1}{Z(\mathbf{w}, x)} \exp(\langle \mathbf{w}, \phi(x, y) \rangle)$$

$$\log p(y | x; \mathbf{w}) = \underbrace{\langle \mathbf{w}, \phi(x, y) \rangle}_{\text{Linear}} - \log Z(\mathbf{w}, x)$$

$$Z(\mathbf{w}, x) = \sum_{y \in Y'} \exp(\langle \mathbf{w}, \phi(x, y') \rangle)$$

- $\Phi(x, y)$: Feature vector defined by input \mathbf{x} , and output y
- \mathbf{w} : Weight vector, each weight corresponds to each feature
- Many models belong to this model
(e.g. Maximum Entropy Model, Logistic Regression)

Maximum Likelihood Estimation (MLE)

$$\mathbf{w}_{MLE} = \arg \max_{\mathbf{w}} \sum_i \log p(y_i | x_i; \mathbf{w})$$

- Estimate \mathbf{w} by using training data $\{x_i, y_i\}$ ($i=1\dots n$)
- Will overfit to the training data if the training data is insufficient
 - E.g. w_i will diverge to infinity if $\Phi_i(x,y)$ fires on the training data only.

Regularized MLE

$$\mathbf{w}_{MAP} = \arg \max_{\mathbf{w}} \sum_i \log p(y_i | x_i; \mathbf{w}) - \underline{C \cdot r(\mathbf{w})}$$

- Add regularization term to prevent overfit
- Regularization term $r(\mathbf{w}) \in \mathbb{R}^+$
 - $C > 0$: Tradeoff parameter
 - C is small : emphasis on likelihood (Overfit)
 - C is large : emphasis on regularization (Underfit)

Regularized MLE (cont.)

- L_2 regularized LL (L_2 -LL)

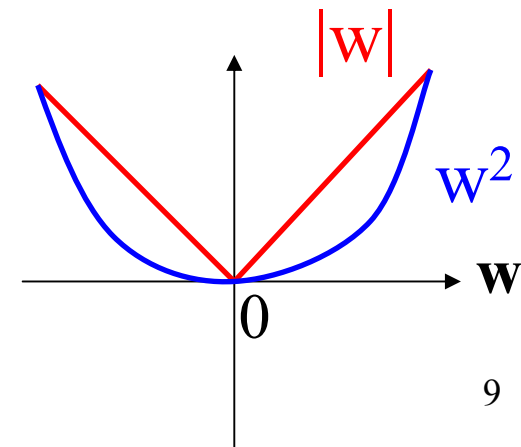
$$r(\mathbf{w}) = \sum_{i=1..m} w_i^2$$

- Maximum a posterior estimation by using Gaussian prior on \mathbf{w}

- L_1 regularized LL (L_1 -LL)

$$r(\mathbf{w}) = \sum_{i=1..m} |w_i|$$

- Maximum a posterior estimation by using Laplace prior on \mathbf{w}
- We will use this regularization



The characteristics of L_1 regularization

- **Sparse feature set**: Many weights are exactly 0
 - **Active feature** : a feature whose have non-zero weight
 - Since the gradient of the L_1 term is always constant, many weights will be **pushed away** to 0
 - In L_2 , all weights are not 0 because the regularization term will rapidly decrease as it approaches to 0
 - If there are many irrelevant features, L_1 achieves higher performance than that of L_2 [A. Ng. ICML 04]

The characteristics of L_1 regularization (cont.)

- In NLP tasks, while the performances of L_1 -LL and L_2 -LL are almost identical, # active features in L_1 -LL is about 1/10 of that of L_2 -LL [J. Gao+ ACL 07]

The optimization of L_1 -LL

$$\mathbf{w}_{MAP} = \arg \max_{\mathbf{w}} \sum_i \log p(y_i | x_i; \mathbf{w}) - \underbrace{\sum_j |w_j|}_{\text{L}_1 \text{ penalty}}$$

- Since it includes non-differentiable term (absolute function), we cannot use gradient based optimization
- Recent work handle this problem by fixing the orthant of parameters at one update
 - Orthant-Wise Limited-memory Quasi-Newton [G. Andrew+, ICML 07]

- Background
- **Dual L_1 -regularized log-linear model**
 - **Derivation**
 - **SMO algorithm**
 - **Recovery of primal parameters**
 - **Extraction of feature combination**
- Experiments
- Conclusion

Proposal

- Convert optimization L_1 -LL into **dual**
 - Parameters are corresponding to each training data
 - Use perceptron-style update
 - Many previous studies focus only on L_2 -regularization (c.f. SVM, L_2 -LL [Collins 07])
- Generalized Lagrange method
 - Use sub-gradient for absolute term

Dualized L_1 -LL (L_1 -DLL)

$$\alpha^* = \arg \min_{\alpha} \sum_i \sum_y \alpha_{i,y} \log \alpha_{i,y}$$

$$|v_k(\alpha)| \leq 1 \quad (k = 1, \dots, m)$$

$$\alpha \in \Delta^n$$

m: # features

n simplex
distributions

- $\alpha_{i,y}$: i -th training example whose label is y
 - $\{\alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,|Y|}\}$ is a simplex distribution for each i , (positive, and sum to 1)
- $v_k(\alpha)$: Gradient of k -th feature

Gradient information

$$\mathbf{v}^* = \mathbf{v}(\boldsymbol{\alpha}^*)$$

$$\mathbf{v}_i^* = \begin{cases} 1 & (\mathbf{w}_i^* \geq 0) \\ -1 & (\mathbf{w}_i^* \leq 0) \\ [-1, 1] & (\mathbf{w}_i^* = 0) \end{cases}$$

$$\mathbf{v}(\boldsymbol{\alpha}) = \frac{1}{C} \sum_{i=1}^n \sum_{y \in \mathcal{Y}} \alpha_{i,y} \psi_{i,y}$$

$$\psi_{i,y} = \phi(x_i, y_i) - \phi(x_i, y)$$

- A feature is active (its weight is not **0**) only if its absolute gradient value is **1**
- Gradient information is represented as a linear combination of training examples

Derivation (1/2)

$$\xi_{i,y} = -\langle \mathbf{w}, \psi_{i,y} \rangle$$

Margin

Define auxiliary variable and construct the Lagrange function

$$L(\mathbf{w}, \xi, \alpha) = -\sum_i \log \left(\sum_y \exp(\xi_{i,y}) \right)$$

$$-C|\mathbf{w}| + \sum_{i=1}^n \sum_{y \in \mathcal{Y}} \alpha_{i,y} (\xi_{i,y} + \langle \mathbf{w}, \psi_{i,y} \rangle)$$

Subtract an incorrect feature vector from correct feature vector

where $\psi_{i,y} = \phi(x_i, y_i) - \phi(x_i, y)$

- Then, check generalized KKT condition

Derivation (2/2)

Take sub-differential of L with regard to \mathbf{w}

$$\frac{\partial L}{\partial \mathbf{w}} \ni 0 \quad \mathbf{v}(\boldsymbol{\alpha}) = \frac{1}{C} \sum_{i=1}^n \sum_{y \in \mathcal{Y}} \alpha_{i,y} \psi_{i,y}$$

\mathbf{v} is the subgradient of $|\mathbf{w}|$ with regard to \mathbf{w}

By differential of L with regard to $\xi_{i,y}$ and taking the point where 0

$$\alpha_{i,y} = \frac{\exp(\xi_{i,y})}{\sum_{y'} \exp(\xi_{i,y'})}$$

$\alpha_{i,y}$ is the distribution over the candidates of each training data ¹⁸

Example (1/4)

- # features : 5, # labels : 2, # examples : 4

	feature vector	label	
- e ₁	(2, 1, 4, -2, -2)	0	} examples
- e ₂	(-3, 3, 1, -1, -1)	1	
- e ₃	(3, 1, 2, -3, 0)	1	
- e ₄	(-3, -2, 0, 0, -1)	2	

Example (2/4)

- # features : 5, # labels : 2, # examples : 4

		Dual parameters			
- e_1	=	(2, 1, 4, -2, -2)	a_{11}	a_{12}	a_{13}
- e_2	=	(-3, 3, 1, -1, -1)	a_{21}	a_{22}	a_{23}
- e_3	=	(3, 1, 2, -3, 0)	a_{31}	a_{32}	a_{33}
- e_4	=	(-3, -2, 0, 0, -1)	a_{41}	a_{42}	a_{43}
w	=	(w_1 , w_2 , w_3 , w_4 , w_5)	1	2	3

Primal parameters

Example (3/4)

Dual parameters

$$\begin{aligned} - e_1 &= (2, 1, 4, -2, -2) \quad 1, 0, 0 \\ - e_2 &= (-3, 3, 1, -1, -1) \quad 0, 1, 0 \\ - e_3 &= (3, 1, 2, -3, 0) \quad 0, 1, 0 \\ - e_4 &= (-3, -2, 0, 0, -1) \quad 0, 0, 1 \end{aligned}$$

At the beginning, parameters are initialized as 1 for correct labels, and 0 for other labels

Example (4/4)

Dual parameters

- e_1	=	(2, 1, 4, -2, -2)	0.8, 0.1, 0.1
- e_2	=	(-3, 3, 1, -1, -1)	0.2, 0.7, 0.1
- e_3	=	(3, 1, 2, -3, 0)	0.2, 0.6, 0.2
- e_4	=	(-3, -2, 0, 0, -1)	0.1, 0.2, 0.7

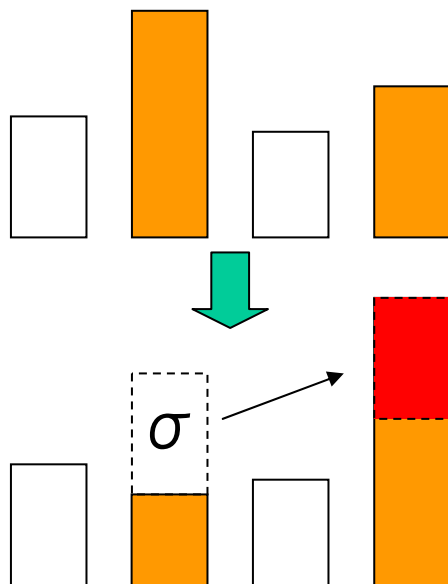
After the optimization, dual parameters are distributed in each label (kinds of smoothing)

Optimization by SMO

(Sequential Minimal Optimization)

- For each training examples, pick up two dual variables $\alpha_{i,y1}$, $\alpha_{i,y2}$, and decide the update width σ , and update as

$$\alpha_{i,y1} = \alpha_{i,y1} + \sigma, \quad \alpha_{i,y2} = \alpha_{i,y2} - \sigma$$
 - The optimal update width can be calculated in closed form
- The \mathbf{v} update is similar to the perceptron update



(1) Choose two variable which violates KKT condition ($\alpha_{i,y1}$, $\alpha_{i,y2}$)

(2) Calculate the update width σ , then update as $\alpha_{i,y1} + \sigma$, $\alpha_{i,y2} - \sigma$

Optimization by SMO (cont.)

- Condition (I) $0 \leq \alpha_i \leq 1$
 $\max(-\alpha_{i,y1}, \alpha_{i,y2} - 1) \leq \delta \leq \min(1 - \alpha_{i,y1}, \alpha_{i,y2})$
- Condition (II) $|v_i(\alpha)| \leq 1$ for $k=1 \dots m$ $\sigma_k = \psi_{i,y1,k} - \psi_{i,y2,k}$
 $C(-1 - v_k(\alpha))/\sigma_k \leq \delta \leq C(1 - v_k(\alpha))/\sigma_k \quad (\sigma_k > 0)$
 $C(1 - v_k(\alpha))/\sigma_k \leq \delta \leq C(-1 - v_k(\alpha))/\sigma_k \quad (\sigma_k < 0)$
- Minimize $\alpha_{i,y1} \log \alpha_{i,y1} + \alpha_{i,y2} \log \alpha_{i,y2}$ in (I) (II)
 - The optimal update width is the closest value to $\sigma = (\alpha_{i,y1} + \alpha_{i,y2})/2$ in the region (I) (II)

Input: training data x_i , label pair y_1, y_2

$$\delta_l = \max(-\alpha_{i,y_1}, \alpha_{i,y_2} - 1)$$

$$\delta_h = \min(1 - \alpha_{i,y_1}, \alpha_{i,y_2})$$

for $k = 0$ **to** m **do**

$$\sigma_k := \psi_{i,y_1,k} - \psi_{i,y_2,k}$$

if $\sigma_k = 0$ **then**

continue

end if

$$x = C(-1 - v_k(\alpha))/\sigma_k$$

$$y = C(1 - v_k(\alpha))/\sigma_k$$

if $\sigma_k < 0$ **then**

swap(x, y)

end if

$$\delta_l := \max(\delta_l, x)$$

$$\delta_h := \min(\delta_h, y)$$

end for

$$\delta_m := (\alpha_{i,y_1} - \alpha_{i,y_2})/2$$

if $\delta_l \leq \delta_m \leq \delta_h$ **then**

$$\delta = \delta_m$$

else if $\delta_m < \delta_l$ **then**

$$\delta = \delta_l$$

else

$$\delta = \delta_h$$

end if

Output: δ

Condition (I) $0 \leq \alpha_i \leq 1$

Condition (II) $|v_i(\alpha)| \leq 1$

Find the optimal update width
in conditions (I) (II)

The computational cost for each example is $O(m')$
where m' is the average number of fired features

Recovery of primal parameters (\mathbf{w})

- In L_1 , primal parameters \mathbf{w}^* cannot be represented by dual parameters \mathbf{a}^* in closed form
 - In L_2 , \mathbf{w}^* can be represented as a linear combination of training examples c.f. Representer Theorem [Sch'olkopf et al., 01]
- In L_1 -LL, the following holds
$$A\mathbf{w}^* = \boldsymbol{\xi}^*$$
 - where \mathbf{A} is the matrix such that each row is $\boldsymbol{\psi}_{i,y}$ and $\xi_{i,y}^* = \log(a_{i,y}^*/a_{i,j}^*)$
- By solving this, we obtain the optimal \mathbf{w}^*
- Or, re-train primal problem by initializing parameters by using the gradient information

Extraction of feature combinations

- Feature combinations are important for linear classifiers
 - Although kernel trick can employ it implicitly, it is heavy, and black box
- Given an original feature set $\{k_1 \dots k_m\}$, we want to find effective feature combinations
 - E.g. : $\Phi_{k_{12}}(x,y) = \Phi_{k_1}(x,y) \Phi_{k_2}(x,y)$
- In dual representation, we can filter out non-active features efficiently
 - If $|v_k(\mathbf{a})| < 1$, the optimization problem is not changed even if this feature is included
 - We can calculate $|v_k(\mathbf{a})|$ by using current dual parameters

Extraction of feature combinations (contd.)

$$\phi_k(x_i, y) = \phi_{k1}(x_i, y)\phi_{k2}(x_i, y)$$

$$Cv_k(\alpha) = \sum_{i=1}^n \phi_{k1}(x_i, y_i)\phi_{k2}(x_i, y_i) - \sum_{i=1}^n \sum_{y \in \mathcal{Y}} \alpha_{i,y} \phi_{k1}(x_i, y)\phi_{k2}(x_i, y)$$

- Check whether $|\mathbf{v}_k(\mathbf{a})|=1$ or not
- $\mathbf{v}_k(\mathbf{a})$ can be calculated for new feature
 - The value is calculated by using co-occurrence sampling technique [Ping Li+ 2007]

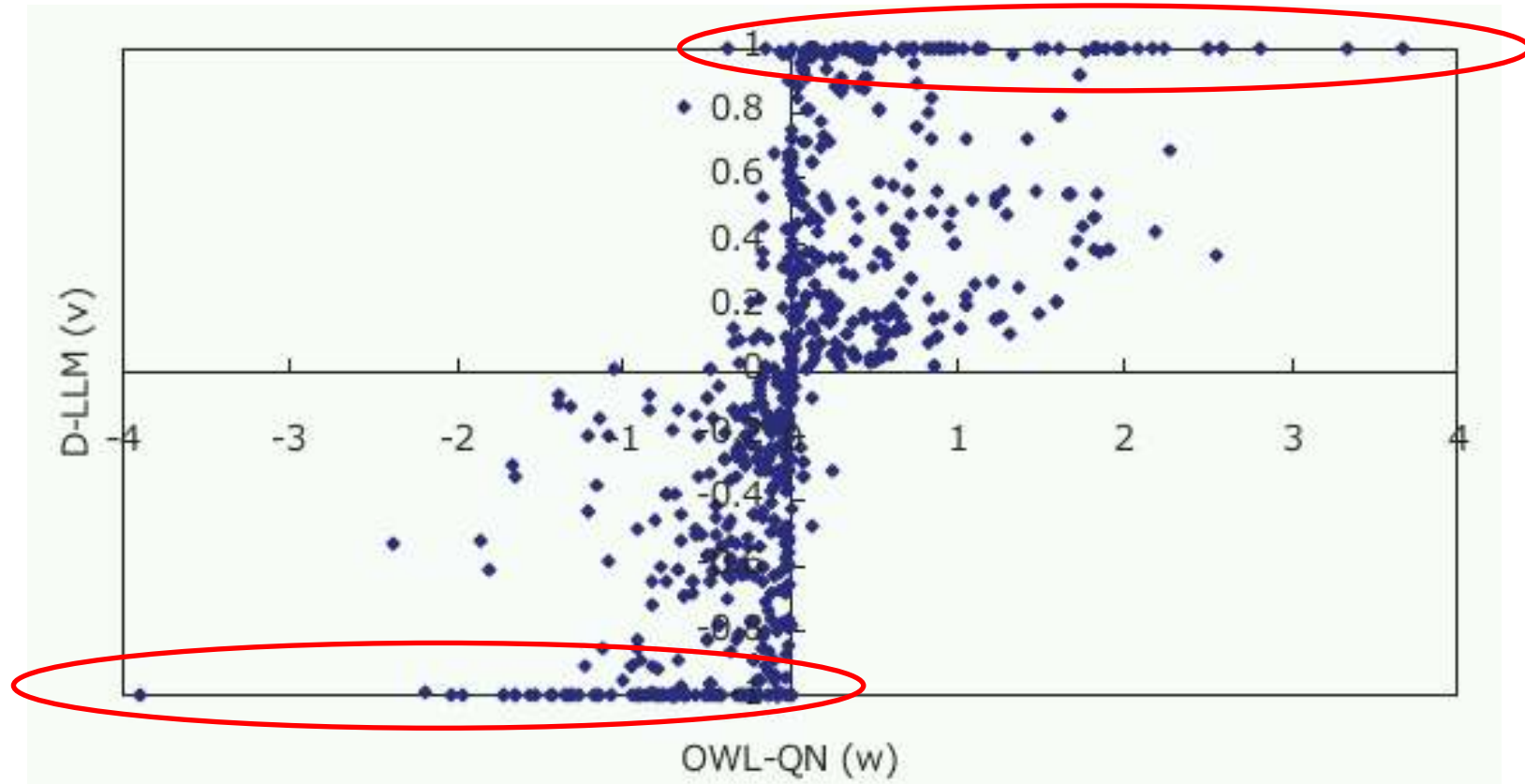
- Background
- Dual L_1 -regularized log-linear model
 - Derivation
 - SMO algorithm
 - Recovery of primal parameters
 - Extraction of feature combination
- **Experiments**
- Conclusion

Experiment (I)

Japanese dependency task

- Data
 - Given two word, estimate whether there is dependency ($y=1$) or not ($y=0$)
 - # examples 179118
 - # features 400530
- Used same dependency algorithm, and features as in [Sassano 01]
- Compared our method (L_1 -DLL) with OWL-QN [G. Andrew+, ICML 07]
 - In our method, we just got the gradient information
- The recovery of \mathbf{w} by using the linear system is a feature work

Experiment (I) Result



- # of active features is 808 in OWL-QN, 691 in $L_1=DLL$ ($|v|>0.9$)
- Same sign 92%, Same active features 60.6%

Discussion

- While # active features is similar, the active feature set in L_1 -DLL and OWL-QN is different
 - In L_1 -LL, the optimal parameters are not unique, but the convex set in general.
 - e.g. : If two features f_1 f_2 fires on identical training set only, then the object function with regard to these two features are always same if w_1+w_2 are same
- The SMO optimization chooses relatively smaller active set because it greedy updates parameters

Experiment (II)

Document Classification

- Data
 - Tech-TC-300 data set [Davidov, et. al. 04]
 - 295 binary classification tasks
 - Used the same feature set and values as distributed without any filtering. (Bag-of-words)
- Compared our method (L_1 -DLL) with OWL-QN [G. Andrew+, ICML 07]
- After obtaining the gradient information in L_1 -DLL, we filter out non-active features and then re-train the model by OWL-QN.

Experiment (II)

Result

	F_2	ACTIVE FEATURES
OWL-QN	0.894	25.58
L_1 -DLL	0.870	18.86 / 9.81

Average # features is 25389

- ACTIVE FEATURES shows the average # of active features
 - In the row L_1 -DLL, left shows the features that have $|v_k| \geq 1$ and, right shows the features which are active after primal optimization
- The F_2 scores in both methods are almost the same
- However, active feature sets in both methods are different.

Case study

Document classification

L₁-DLL

OWL-QN

Task: Exp_2592_3431

features: 20089

examples: 151

Negative:

Business/Business

Services/Consulting/Medical and Life Sciences

Positive: Arts/Performing

Arts/ Dance/Ballroom

-0.105	and
-0.053	regulatory
-0.026	die
0.677	dance
0.053	loan

-0.021	die
-0.076	and
-0.052	gif
-0.003	jpg
-0.051	services
-0.003	consulting
-0.008	training
0	the
0.754	dance
0.069	loan
0.058	alexander

L₁-DLL and OWL-QN have similar result.
However, the active features are different

Case study (contd.)

Feature combination

Task: Exp_2592_3431

features: 20089

examples: 151

- # combination features that have $|v_k| \geq 1$, is 1778
- F_2 is almost the same as in original feature set ($F_2=0.871$ vs 0.870)
- This may be because the feature value was not normalized
- Or, this task is easy for linear classifiers

-0.116	gif	
-0.004	die	nchen
-0.002	die	der
-0.007	regulatory	
-0.004	the	and
-0.02	and	with
0.11	and	dance
0	gif	arobull1
0.001	new	loan
0.001	loans	loan
0.001	auto	loan
0.013	dance	
0.287	the	dance

- Background
- Dual L_1 -regularized log-linear model
 - Derivation
 - SMO algorithm
 - Recovery of primal parameters
 - Extraction of feature combination
- Experiments
- **Conclusion**

Conclusion

- Dual representation of L_1 -LL
 - Dual parameters are simplex distribution over each training examples
 - The gradient information has active information
- Simple and efficient SMO algorithm
 - Perceptron-style update
- Efficient extraction algorithm for feature combination

Conclusion (cont.)

- Since the presentation time is limited, I do not present the followings
 - Dual of L_1 -regularized max-margin
 - Very similar object functions and update algorithm
 - Extension to Structured output
 - Conditional Random Fields
 - Max-margin Markov Network
 - Integrating out the hyper parameter
[G. Cawley, et. al NIPS 06]

Future work

- Robust and efficient recovery of \mathbf{w}
- Efficient extraction of complex features
 - Beyond the simple combination features, Decision Tree, Kernel-inspired features
- Representer Theorem in L_1 version
 - In L_1 regularization, # active features is equal to or smaller than # training example [G. Ratsch 01]
 - c.f. In L_2 \mathbf{w}^* is a linear combination of training examples

Thanks !