Supplemental Material

A Optimization of Z and E through Alternating Direction Method of Multipliers

The alternating direction method of multipliers (ADMM) has been used in areas such as machine learning, data mining and image processing in recent years (Boyd et al., 2011). This optimization approach aims to find the optimum value for optimization problems which follow the form given in Equation 12, with more than one variable that are linearly related:

$$\begin{aligned} \min_{x,y} \quad & f(x) + g(y) \\ \text{s.t.} \quad & Ax + By = c, \end{aligned} \tag{12}$$

where $f(.)$ and $g(.)$ are convex functions, and A, B and c are constant matrices. ADMM considers the augmented Lagrangian form as $L(x, y, \lambda) = f(x) + g(y) + \lambda^T(Ax + By - c) + \frac{\sigma}{2}\|Ax+By-c\|^2$. λ is a variable in the dual space, and $\sigma > 0$ is a penalty parameter to set the importance of penalizing the constraint. At the t-th iteration, ADMM alternates between minimizing the Langrangian function $L(x, y, \lambda)$ with respect to the main variables x and y (denoted x_t and y_t), as given in Equation 13:

$$\begin{aligned} x_t &= \min_x \quad L(x,\, y_{t-1},\, \lambda_{t-1}); \\ y_t &= \min_y \quad L(x_t,\, y,\, \lambda_{t-1}). \end{aligned} \tag{13}$$

Then, λ is updated by means of gradient ascent in the dual space, such as in Equation 14:

$$\lambda_t = \lambda_{t-1} + \sigma(Ax_t + By_t - c). \tag{14}$$

The standard ADMM algorithm has a convergence rate of $O(1/T)$, where T is the number of iterations (He and Yuan, 2012). In past years, the authors of (Beck and Teboulle, 2009) and (Nesterov, 2004) proposed methods to accelerate gradient descent based methods leading to similar accelerated approaches for ADMM, which results in a faster $O(1/T^2)$ convergence rate.

The ADMM method is an appropriate choice to find the optimum value for Z and E in Equation 8. Therefore, we can write the Langrangian form of the problem as in Equation 15:

$$\begin{aligned} & L(Z, \lambda) \\ &= \|Z\|_1 + \alpha\|E\|_{2,1} + \lambda(Y - (Y_sZ + E)) \\ &\quad + \frac{\sigma}{2}\|Y - (Y_sZ + E)\|_F^2 \end{aligned} \tag{15}$$

The λ parameter is the Lagrangian multiplier updated by means of gradient ascent. The penalty parameter σ is a positive parameter, which according to fine grained validation tests, we increase by multiplying in $\rho = 1.1$ in order to penalize the error more for this constraint.

$$\begin{aligned} Z_{t+\frac{1}{2}} &= ((Y_s^TY_s)^{-1}(Y_s^T(Y - E_t) + \frac{\lambda}{\sigma}Y_s^T)) \\ Z_{t+1} &= \operatorname*{argmin}_Z \left(\frac{1}{\sigma}\|Z\|_1 + \frac{1}{2}\|Z - Z_{t+\frac{1}{2}}\|^2\right) \\ E_{t+\frac{1}{2}} &= (Y - Y_sZ_t) + \frac{\lambda}{\sigma} \\ E_{t+1} &= \operatorname*{argmin}_E \left(\alpha\|E\|_{2,1} + \frac{1}{2}\|E - E_{t+\frac{1}{2}}\|^2\right) \end{aligned} \tag{16}$$

Algorithm 4 Optimization of affinity matrix Z.

Input: label matrix Y.
Initialization: $\lambda = \mathbf{0}$, $\rho = 1.1$, $\epsilon = 10^{-6}$.
Repeat:

1: Update Z, E;
2: Update Lagrangian multiplier λ;
3: Update penalty parameter: $\sigma = \rho\sigma$;

Until convergence condition $\|Y-(YZ+E)\|_\infty < \epsilon$
Output: Z.

B Proof of Theorems

Proof of Theorem 4:

Proof. $\Delta_{cut}(k|S) = \sum_{i\in V\setminus S} w_{i,k} - \sum_{i\in S} w_{k,i}$. This function is monotone for $|S| \ll |V|$. For $R \subseteq S$, $\sum_{i\in V\setminus S} w_{i,k} \leq \sum_{i\in V\setminus R} w_{i,k}$, and $-\sum_{i\in S} w_{k,i} < -\sum_{i\in R} w_{k,i}$. Then $\Delta_{cut}(k|S) \leq \Delta_{cut}(k|R)$. □

Proof of Theorem 5:

Proof. Since the first term is a submodular term, we only need to prove that the second penalty term is a submodular term. $\Delta_{pen}(k|S) = -\lambda\sum_{i\in S}(w_{i,k} + w_{k,i})$. Therefore, if $R \subseteq S$, $\Delta_{pen}(k|S) \leq \Delta_{pen}(k|R)$. The penalized max-cut function is a submodular function and monotone for non-large values of λ. □

	Proposed	PD-sparse	LEML	CPLST	CS	ML-CSSP
Bibtex						
nDCG@1	**64.56**±0.79	61.29±0.65	62.54±0.52	62.38±0.63	58.87±0.61	44.98±1.15
nDCG@3	**60.11**±0.53	55.83±0.57	58.22±0.42	57.63±0.56	52.19±0.56	44.67±1.01
nDCG@5	**62.18**±0.49	57.35±0.49	60.53±0.38	59.71±0.42	53.25±0.54	47.97±0.98
Delicious						
nDCG@1	*65.13*±0.39	51.82±1.40	**65.67**±0.73	*65.31*±0.88	61.36±0.38	63.04±1.29
nDCG@3	60.51±0.39	46.00±1.12	**61.77**±0.50	61.16±0.45	57.66±0.34	57.91±1.15
nDCG@5	57.12±0.35	42.02±1.01	**58.47**±0.47	57.80±0.49	54.44±0.32	53.36±0.94
Mediamill						
nDCG@1	**84.25**±0.27	81.86±4.08	*84.01*±0.31	83.35±0.33	83.82±5.92	78.95±0.23
nDCG@3	**75.33**±0.26	70.21±2.37	*75.23*±0.25	74.21±0.24	*75.29*±4.99	68.97±0.28
nDCG@5	**72.03**±0.21	63.71±1.73	*71.96*±0.18	70.55±0.17	*71.92*±4.03	62.88±0.26
Eurlex						
nDCG@1	**81.04**±0.80	76.43±1.04	63.40±1.58	72.28±0.99	58.52±1.06	62.09±2.12
nDCG@3	**71.29**±0.86	64.31±0.72	53.56±1.47	61.64±1.02	48.67±0.75	51.63±1.31
nDCG@5	**65.64**±0.84	58.78±0.70	48.47±1.24	55.92±0.97	40.79±0.65	47.11±1.10
Wiki10-31k						
nDCG@1	**86.05**	82.14	73.47	-	-	-
nDCG@3	**79.11**	72.63	64.92	-	-	-
nDCG@5	**72.26**	64.33	58.69	-	-	-

Table 5: nDCG@k on the small-scale datasets with k=100. Best in **bold** and not significantly different to best at p=0.05 in *italics*

C nDCG Results

The most well-known and frequently used measures for the large-scale multi-label learning problem are the precision-at-k and the normalized discounted cumulative gain-at-k (nDCG-at-k), which represent the accuracy over the highly ranked predictions. Precision-at-k results are reported in the main text, nDCG results are reported here.

The normalized discounted cumulative gain-at-k (nDCG-at-k), which represent the accuracy over the highly ranked predictions, is shown in Tables 5 and 6. Precision-at-k results are reported in the main text, nDCG results are reported here.

$$\text{P@}k := \frac{1}{k} \sum_{l \in \text{rank}_k(\hat{\mathbf{y}})} \mathbf{y}_l. \tag{17}$$

$$\text{DCG@}k := \sum_{l \in \text{rank}_k(\hat{\mathbf{y}})} \frac{\mathbf{y}_l}{\log(l+1)}. \tag{18}$$

$$\text{nDCG@}k := \frac{\text{DCG@}k}{\sum_{l=1}^{\min(k, \|\mathbf{y}\|_0)} \frac{1}{\log(l+1)}}. \tag{19}$$

	Proposed	SLEEC	FastXML
Bibtex			
nDCG@1	*64.56*	**65.08**	63.42
nDCG@3	*60.11*	**60.47**	59.51
nDCG@5	*62.18*	**62.64**	61.70
Delicious			
nDCG@1	65.13	67.59	**69.61**
nDCG@3	60.51	62.87	**65.47**
nDCG@5	57.12	59.28	**61.90**
Mediamill			
nDCG@1	84.25	**87.82**	84.22
nDCG@3	75.33	**81.50**	75.41
nDCG@5	72.03	**79.22**	72.37
Eurlex			
nDCG@1	**81.04**	79.26	71.36
nDCG@3	**71.29**	68.13	62.87
nDCG@5	**65.64**	61.60	58.06
Wiki10-31k			
nDCG@1	**86.05**	85.88	84.31
nDCG@3	**79.11**	72.98	75.35
nDCG@5	**72.26**	62.70	63.36

Table 6: nDCG@k on the ensemble-based nonlinear models. Best in **bold** and not significantly different to best in *italics*.

	f_{pen}	f_{score}	$f_{pen} + \alpha f_{score}$	**+Outliers**
Bibtex				
nDCG@1	60.98	63.27	63.29	64.55
nDCG@3	54.15	57.16	57.49	60.11
nDCG@5	56.45	58.86	59.66	62.18
Mediamill				
nDCG@1	81.12	81.83	84.25	84.25
nDCG@3	71.03	73.76	75.12	75.33
nDCG@5	68.65	70.40	71.79	72.03
Delicious				
nDCG@1	62.71	62.71	64.33	65.14
nDCG@3	58.31	58.31	59.71	60.54
nDCG@5	55.04	55.04	56.16	57.15
Eurlex				
nDCG@1	56.60	3.84	56.60	81.04
nDCG@3	42.07	3.27	42.07	71.29
nDCG@5	36.61	3.27	36.61	65.64

Table 7: Ablation Study