Paired-Dual Learning for Fast Training of Latent Variable Hinge-Loss MRFs: Appendices

A. Probabilistic Soft Logic

In this supplement, we describe the models used in our experiments using probabilistic soft logic (PSL) (Bach et al., 2015), a language for defining hinge-loss potential templates. PSL's variables are logical atoms, and its rules use logical operators such as conjunction and implication to define dependencies between these variables. All variables are continuous in the $[0, 1]$ interval. Conjunction of Boolean variables $X \wedge Y$ are generalized to continuous variables using the hinge function $\max\{X + Y - 1, 0\}$, which is known as the *Lukasiewicz t-norm*. Disjunction $X \vee Y$ is relaxed to $\min\{X + Y, 1\}$, and negation $\neg X$ is relaxed to $1 - X$. To define a model, PSL rules are grounded out with all possible substitutions for logical terms. The groundings define hinge-loss potentials that share the same weight, and whose values are the ground rule's *distance to satisfaction*, e.g., for $X \Rightarrow Y$ the distance to satisfaction is $\max\{X - Y, 0\}$. In PSL, rules consist of a conjunction of literals in the body and a disjunction of literals in the head of the rule. The continuous interpretation of the rule becomes a hinge-loss function for the rule's distance to satisfaction. Finally, each rule is annotated with a non-negative weight, which is the parameter shared across all potentials templated by that rule.

B. Discovering Latent Groups in Social Media

Data set The data set from Bach et al. (2013a) is roughly 4.275M tweets collected from about 1.350M Twitter users via a query that focuses on South American users. The tweets were collected from Oct. 6 to Oct. 8, 2012, a 48-hour window around the Venezuelan presidential election on Oct. 7. The two major candidates were Hugo Chávez, the incumbent, and Henrique Capriles. Chávez won with 55% of the vote.

The goal is to learn a model that relates language usage and social interactions to latent group membership. We first identify 20 users as *top users* based on being the most retweeted or, in the case of the state-owned television network's account, being of particular interest. Recorded Twitter interactions form the features in this model. We identify all other users that either retweeted or mentioned at least one of the top users and used at least one hashtag in a tweet that was not a mention or a retweet of a top user. Filtering by these criteria, the set contains 1,678 *regular users* (i.e., users that are not top users).

We organize the variables in our model using PSL predicates. Whether each regular user tweeted a hashtag is represented with the PSL predicate USEDHASHTAG. Tweets that mention or retweet a top user are not counted, since they are too closely related to the target interaction. For example, if User 1 tweets the hashtag `#hayuncamino` then USEDHASHTAG$(1, \texttt{\#hayuncamino})$ has an observed truth value of 1.0. The PSL predicate REGULARUSERLINK represents whether a regular user retweeted or mentioned *any* user in the full data set that is not a top user, regardless of whether that mentioned or retweeted user is a regular user. Whether a regular user retweeted or mentioned a top user is represented with the PSL predicate TOPUSERLINK. Finally, the latent group membership of each regular user is represented with the PSL predicate INGROUP.

Latent group model We construct a HL-MRF model for predicting interactions of regular users with top users via latent group membership. We treat atoms with the USEDHASHTAG or REGULARUSERLINK predicate as the set of conditioning variables $\boldsymbol{x}$, atoms with the TOPUSERLINK predicate as the set of target variables $\boldsymbol{y}$, and atoms with the INGROUP predicate as the set of latent variables $\boldsymbol{z}$.

When defining our model, let H be the set of hashtags used by at least 15 different regular users ($|H| = 33$), let T be the set of top users ($|T| = 20$), and let $\mathcal{G} = \{g_0, g_1\}$ be the set of latent groups.

We first include rules that relate hashtag usage to group membership. For each hashtag in H and each latent group, we include a rule of the form

$$w_{h,g} : \text{USEDHASHTAG}(U, h) \rightarrow \text{INGROUP}(U, g) \quad \forall h \in H, \forall g \in \mathcal{G}$$

so that there is a different rule weight governing how strongly each commonly used hashtag is associated with each latent group. Second, we include a rule associating social interactions with group commonality:

$$w_{\text{social}} : \text{REGULARUSERLINK}(U_1, U_3) \wedge \text{REGULARUSERLINK}(U_2, U_3) \wedge U_1 \neq U_2 \wedge \text{INGROUP}(U_1, G) \rightarrow \text{INGROUP}(U_2, G).$$

This rule encodes the intuition that regular users who interact with the same people on Twitter are more likely to belong to the same latent group. Adding this rule leverages one the advantages of general log-linear models with latent variables: the ability to easily include dependencies among latent variables. Third, we include rules of the form

$$w_{g,t} : \textsc{InGroup}(U, g) \rightarrow \textsc{TopUserLink}(U, t) \quad \forall g \in \mathcal{G}, \forall t \in T$$

for each latent group and each top user so that there is a parameter governing how strongly each latent group tends to interact with each top user. Last, we constrain the InGroup atoms for each regular user to sum to 1.0, making InGroup a mixed-membership assignment.

We specify initial parameters $\boldsymbol{w}$ by initializing $w_{h,g}$ to 2.0 for all hashtags and groups, w_{social} to 2.0, and $w_{g,t}$ to 5.0 for all top users and groups, *except* two hashtags and two top users which we assign as seeds. We initially associate the top user `hayuncamino` (Henrique Capriles's campaign account) and the hashtag for Capriles's campaign slogan `#hayuncamino` with Group 0 by initializing the parameters associating them with Group 0 to 10.0 and those associating them with Group 1 to 0.0. We initially associate the top user `chavezcandanga` (Hugo Chávez's account) and the hashtag for Chávez's campaign slogan `#elmundoconchávez` with Group 1 in the same way.

For entropy surrogates we add the following rules, all with fixed weights of 10.0:

$$w_{\text{entropy}} : \neg\textsc{InGroup}(U, g) \quad \forall g \in \mathcal{G},$$
$$w_{\text{entropy}} : \neg\textsc{TopUserLink}(U_1, U_2).$$

The full results for all ten folds are presented in Figures 1, 2, 3, and 4.

C. Modeling Latent User Features in Trust Networks

We build our model based on that of Huang et al. (2013), which encodes rules consistent with *triadic closure* in social networks. Instead, we include rules for all possible configurations of directed triads, including those that do not imply balanced behavior, so the learning algorithm can attribute weight to any configuration if it helps optimize its objective. Removing symmetries, there are 12 distinct logical formulas. Four are for a cyclic structure:

$$w^1_{\text{cyc}} : \textsc{Trusts}(A, B) \wedge \textsc{Trusts}(B, C) \rightarrow \textsc{Trusts}(C, A),$$
$$w^2_{\text{cyc}} : \textsc{Trusts}(A, B) \wedge \neg\textsc{Trusts}(B, C) \rightarrow \textsc{Trusts}(C, A),$$
$$w^3_{\text{cyc}} : \neg\textsc{Trusts}(A, B) \wedge \neg\textsc{Trusts}(B, C) \rightarrow \textsc{Trusts}(C, A),$$
$$w^4_{\text{cyc}} : \neg\textsc{Trusts}(A, B) \wedge \neg\textsc{Trusts}(B, C) \rightarrow \textsc{Trusts}(C, A).$$

And eight are for a non-cyclic "v" structure:

$$w^1_{\text{v}} : \textsc{Trusts}(A, B) \wedge \textsc{Trusts}(B, C) \rightarrow \textsc{Trusts}(C, B),$$
$$w^2_{\text{v}} : \textsc{Trusts}(A, B) \wedge \neg\textsc{Trusts}(B, C) \rightarrow \neg\textsc{Trusts}(C, B),$$
$$w^3_{\text{v}} : \neg\textsc{Trusts}(A, B) \wedge \textsc{Trusts}(B, C) \rightarrow \neg\textsc{Trusts}(C, B),$$
$$w^4_{\text{v}} : \neg\textsc{Trusts}(A, B) \wedge \neg\textsc{Trusts}(B, C) \rightarrow \textsc{Trusts}(C, B),$$
$$w^5_{\text{v}} : \textsc{Trusts}(A, B) \wedge \textsc{Trusts}(B, C) \rightarrow \neg\textsc{Trusts}(C, B),$$
$$w^6_{\text{v}} : \textsc{Trusts}(A, B) \wedge \neg\textsc{Trusts}(B, C) \rightarrow \textsc{Trusts}(C, B),$$
$$w^7_{\text{v}} : \neg\textsc{Trusts}(A, B) \wedge \textsc{Trusts}(B, C) \rightarrow \textsc{Trusts}(C, B),$$
$$w^8_{\text{v}} : \neg\textsc{Trusts}(A, B) \wedge \neg\textsc{Trusts}(B, C) \rightarrow \neg\textsc{Trusts}(C, B).$$

We also include pairwise interactions:

$$w^+_{\text{pair}} : \textsc{Trusts}(A, B) \rightarrow \textsc{Trusts}(B, A),$$
$$w^-_{\text{pair}} : \neg\textsc{Trusts}(A, B) \rightarrow \neg\textsc{Trusts}(B, A).$$

To add latent variable reasoning, we add predicates Trusting and Trustworthy that take a single actor as input. The rules

$$w^1_{\text{latent}} : \textsc{Trusting}(A) \rightarrow \textsc{Trusts}(A, B),$$
$$w^2_{\text{latent}} : \textsc{Trustworthy}(B) \rightarrow \textsc{Trusts}(A, B),$$
$$w^3_{\text{latent}} : \textsc{Trusting}(A) \wedge \textsc{Trustworthy}(B) \rightarrow \textsc{Trusts}(A, B)$$

infer trust from these latent predicates, and the rules

$$w^4_{\text{latent}} : \textsc{Trusts}(A, B) \rightarrow \textsc{Trusting}(A),$$
$$w^5_{\text{latent}} : \textsc{Trusts}(A, B) \rightarrow \textsc{Trustworthy}(B)$$

infer the latent values from other trust predictions and observations. All rules are initialized to weights of 1.0. Note that in this problem the structure of the social network is observed, so these rules are grounded for $\textsc{Trusts}(A, B)$ atoms where A and B are observed to know each other. For entropy surrogates, we use the following rules, all with fixed weights of 10.0:

$$w_{\text{entropy}} : \textsc{Trusts}(A, B),$$
$$w_{\text{entropy}} : \neg\textsc{Trusts}(A, B),$$
$$w_{\text{entropy}} : \textsc{Trusting}(A, B),$$
$$w_{\text{entropy}} : \neg\textsc{Trusting}(A, B),$$
$$w_{\text{entropy}} : \textsc{Trustworthy}(A, B),$$
$$w_{\text{entropy}} : \neg\textsc{Trustworthy}(A, B).$$

The full results for all eight folds are presented in Figures 5, 6, and 7.

D. Image Reconstruction

The latent HL-MRF model we use for image reconstruction reasons over variables representing the brightness of pixel values BRIGHT, a binary, thresholded brightness of observed pixels (i.e., an indicator of whether have intensity greater than 0.5) BINARY, and a set of six latent states LATSTATE. The intuition behind the model is that the observed pixel intensities and the thresholded intensities provide evidence about which latent states are active for a particular image, and these latent states imply patterns in the output pixels. For each latent state S_k, and each pixel P_{ij}, whether observed or not, we include the rules

$$w^{++}_{\text{bright}}((i,j),k) : \textsc{LatState}(I, S_k) \rightarrow \textsc{Bright}(I, P_{ij})$$
$$w^{+-}_{\text{bright}}((i,j),k) : \textsc{LatState}(I, S_k) \rightarrow \neg\textsc{Bright}(I, P_{ij})$$
$$w^{-+}_{\text{bright}}((i,j),k) : \neg\textsc{LatState}(I, S_k) \rightarrow \textsc{Bright}(I, P_{ij})$$
$$w^{--}_{\text{bright}}((i,j),k) : \neg\textsc{LatState}(I, S_k) \rightarrow \neg\textsc{Bright}(I, P_{ij}).$$

For observed pixels, we encode analogous rules for thresholded pixel intensities to provide more information to the model:

$$w^{++}_{\text{binary}}((i,j),k) : \textsc{LatState}(I, S_k) \rightarrow \textsc{Binary}(I, P_{ij})$$
$$w^{+-}_{\text{binary}}((i,j),k) : \textsc{LatState}(I, S_k) \rightarrow \neg\textsc{Binary}(I, P_{ij})$$
$$w^{-+}_{\text{binary}}((i,j),k) : \neg\textsc{LatState}(I, S_k) \rightarrow \textsc{Binary}(I, P_{ij})$$
$$w^{--}_{\text{binary}}((i,j),k) : \neg\textsc{LatState}(I, S_k) \rightarrow \neg\textsc{Binary}(I, P_{ij}).$$

For every pair of latent states S_i and S_j, we include rules to encode their tendency or aversion to co-occur:

$$w^{+}_{\text{state}}(i,j) : \textsc{LatState}(I, S_i) \rightarrow \textsc{LatState}(I, S_j)$$
$$w^{-}_{\text{state}}(i,j) : \textsc{LatState}(I, S_i) \rightarrow \neg\textsc{LatState}(I, S_j)$$

Finally, we use fixed-weight priors on the free variables:

$$1.0 : \textsc{LatState}(I, S)$$
$$1.0 : \neg\textsc{LatState}(I, S)$$
$$1.0 : \textsc{Bright}(I, P)$$
$$1.0 : \neg\textsc{Bright}(I, P)$$

which serve as surrogate entropies.

We initialize weights using a heuristic to fit the latent states to individual training images. We first compute the average pixel intensities among all training images, then for each latent state S_k, we randomly choose a seed image. We set the positively correlated binary pixel rule weights w^{++}_{binary} and $w_{\text{binary}^{--}}$ to 1.0 if the seed image pixel intensity is higher than the average, and the negatively correlated binary pixel rules w^{+-}_{binary} and w^{-+}_{binary} to 1.0 if the seed image pixel is dimmer than the average. This scheme makes the initial model assign corresponding latent features to images that share bright and dark pixel locations with the seed images. Starting with this initialization, which includes no information about the unthresholded pixel intensities, the learning algorithms fit the models to also predict pixel intensity. Figure 8 shows details of the learned model and example reconstructions.

E. Learner Settings

During learning, the regularization parameter λ is 0.01, and the ADMM parameter η is 1.0. These parameters were selected with some light tuning on development sets. The differences among the performances of the learners were not sensitive to changes. For EM, during each M step, we fit the parameters by taking ten subgradient steps, using the MPE state of $P(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}; \boldsymbol{w})$ to estimate $\mathbb{E}\left[\phi(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}; w)\right]$ in the maximum likelihood gradient, as is standard for supervised learning for HL-MRFs (Bach et al., 2013b).

F. Convergence of ADMM

We determine whether $L_{\boldsymbol{w}}(\boldsymbol{v}, \boldsymbol{\alpha}, \bar{\boldsymbol{v}})$ or $L'_{\boldsymbol{w}}(\boldsymbol{v}', \boldsymbol{\alpha}', \bar{\boldsymbol{v}}')$ has converged by examining the primal and dual residuals at iteration t:

$$\|r^t\| := \|\mathbf{c}(\boldsymbol{v}^t, \bar{\boldsymbol{v}}^t)\|$$

and

$$\|s^t\| \equiv \eta \left(\sum_{i=1}^{n} \mathcal{K}_i \left(\bar{v}_i^t - \bar{v}_i^{t-1} \right)^2 \right)^{1/2}$$

where n is the number of components of $\bar{\boldsymbol{v}}$ and $\mathcal{K}_i$ is the number of local copies of the consensus variable $\bar{v}_i$ (Boyd et al., 2011).

We use the convergence criteria suggested by Boyd et al. (2011):

$$\begin{aligned} \|r^t\| &\leq \epsilon^{\text{abs}} \sqrt{\sum_{i=1}^{n} \mathcal{K}_i} \\ &\quad + \epsilon^{\text{rel}} \max \left\{ \|\boldsymbol{v}^t\|, \left(\sum_{i=1}^{n} \mathcal{K}_i \left(\bar{v}_i^t \right)^2 \right)^{1/2} \right\} \\ \|s^t\| &\leq \epsilon^{\text{abs}} \sqrt{\sum_{i=1}^{n} \mathcal{K}_i} + \epsilon^{\text{rel}} \|\boldsymbol{\alpha}^t\| \end{aligned}$$

where $\epsilon^{\text{abs}}, \epsilon^{\text{rel}} > 0$ are parameters and, again, n is the number of components of $\bar{\boldsymbol{v}}$ and $\mathcal{K}_i$ is the number of local copies of the consensus variable $\bar{v}_i$. We set ϵ^{abs} to 10^{-6} and ϵ^{rel} to 10^{-4}, which Boyd et al. (2011) suggest as a reasonable choice.

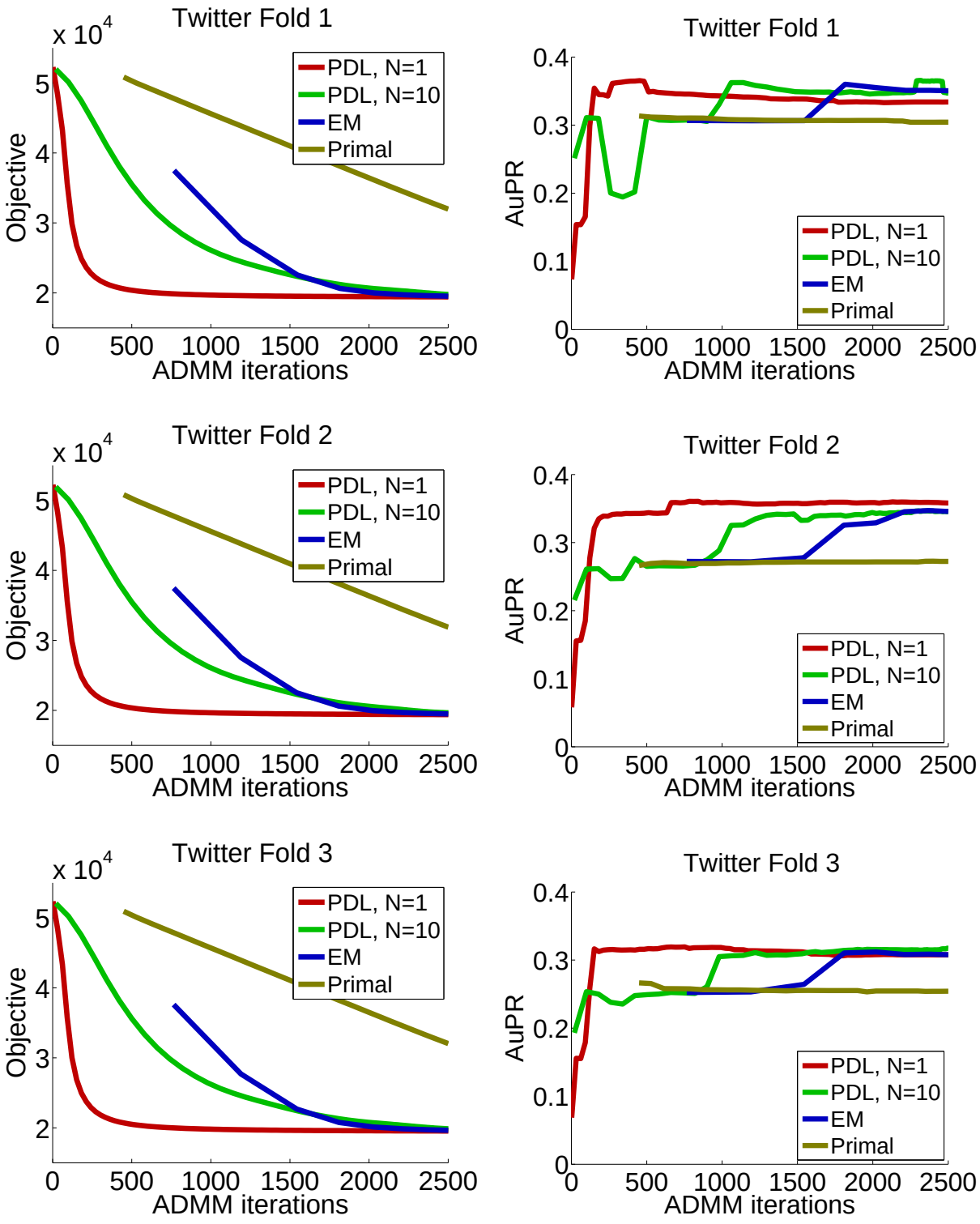

Figure 1. Results for interaction prediction on Twitter data set, folds 1-3.

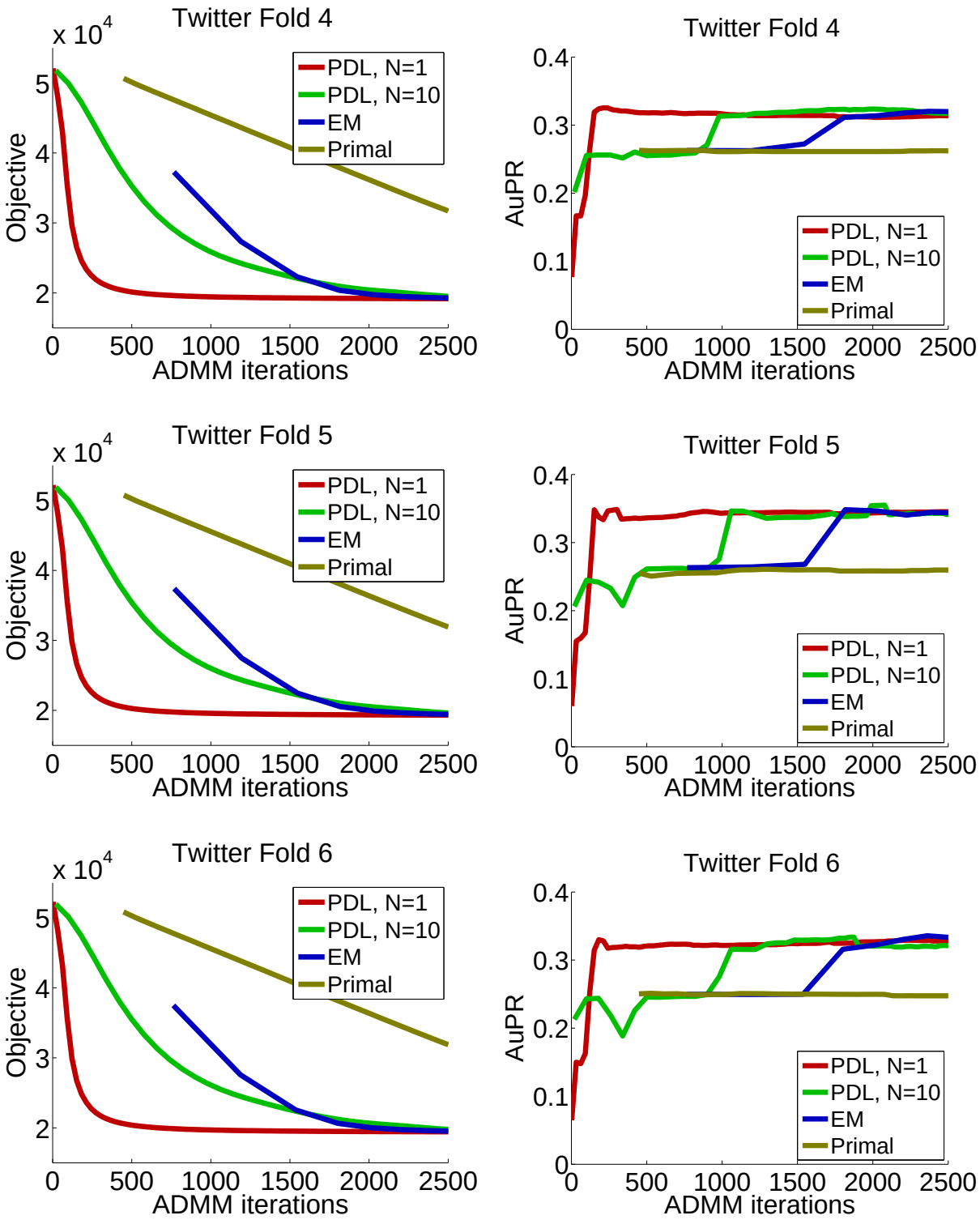

Figure 2. Results for interaction prediction on Twitter data set, folds 4-6.

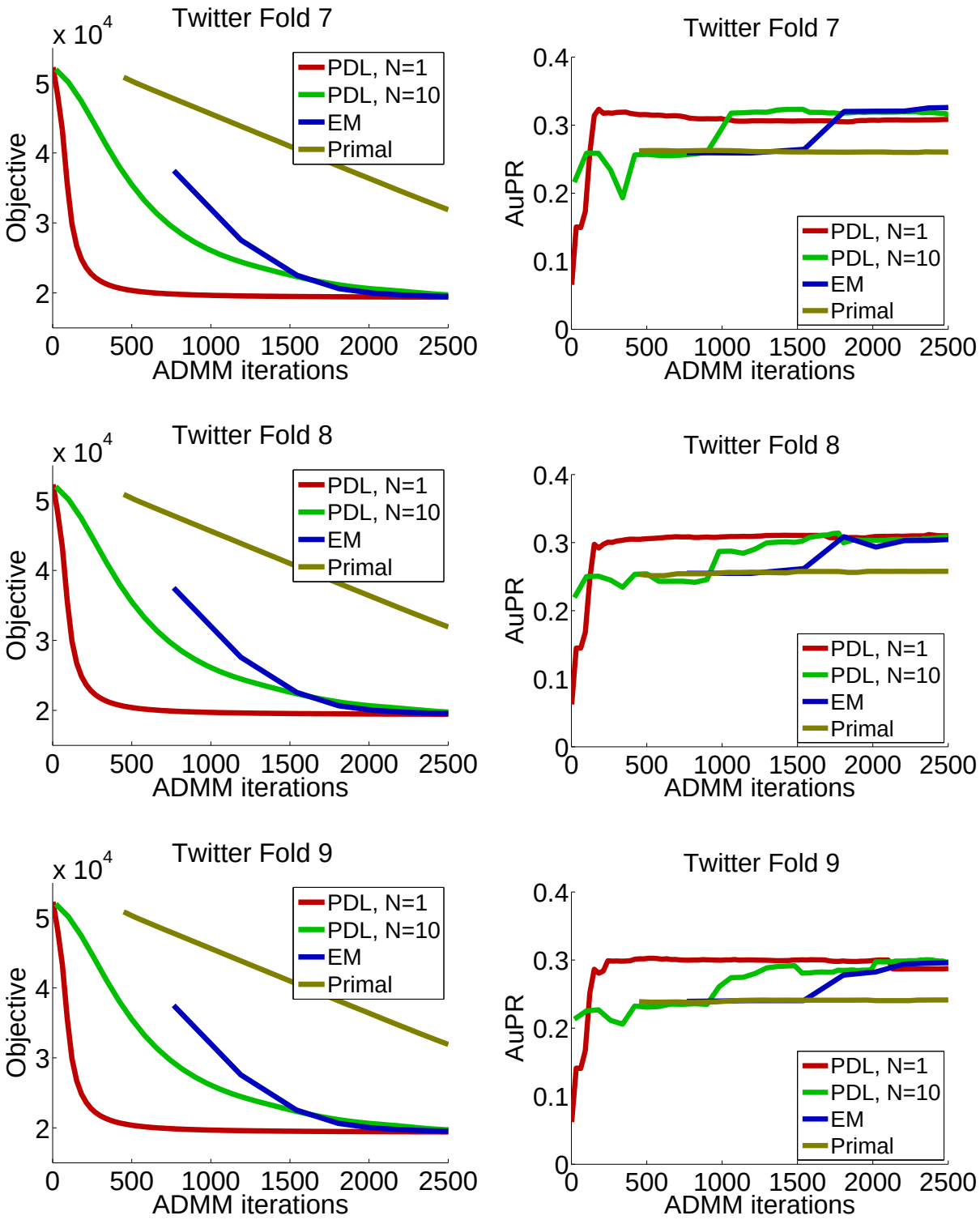

Figure 3. Results for interaction prediction on Twitter data set, folds 7-9.

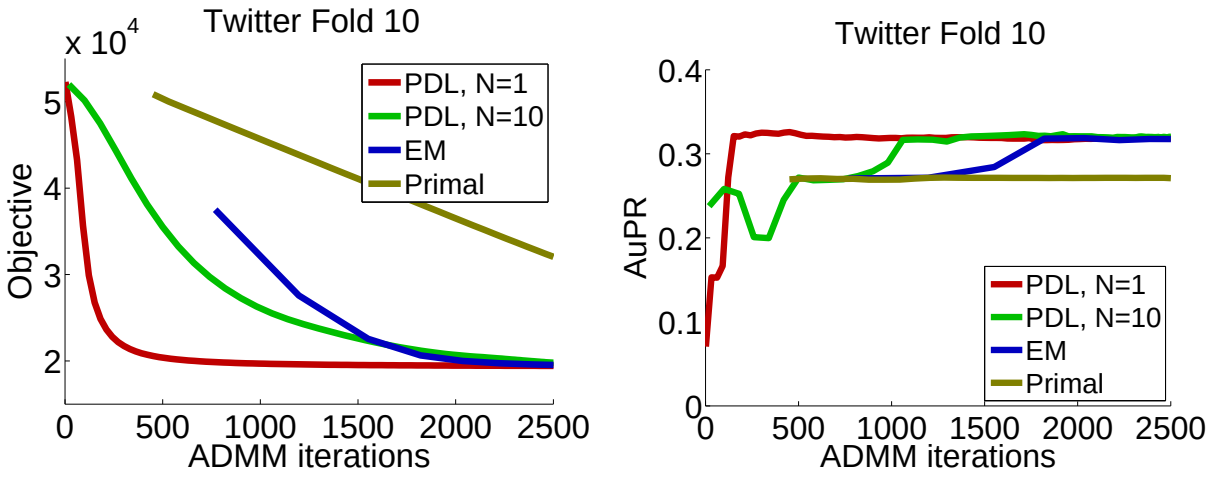

Figure 4. Results for interaction prediction on Twitter data set, fold 10.

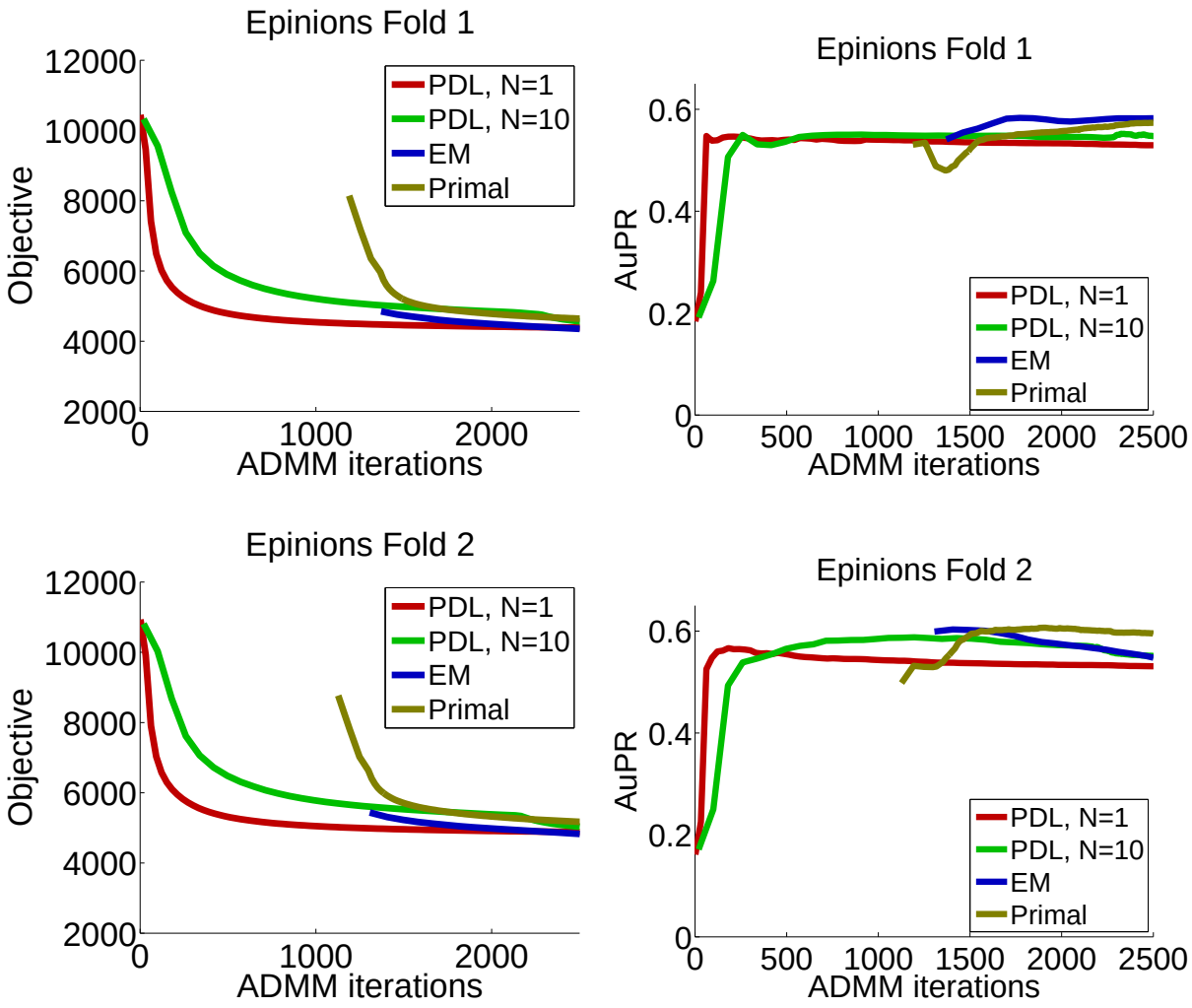

Figure 5. Results for social-trust prediction on Epinions data set, folds 1-2.

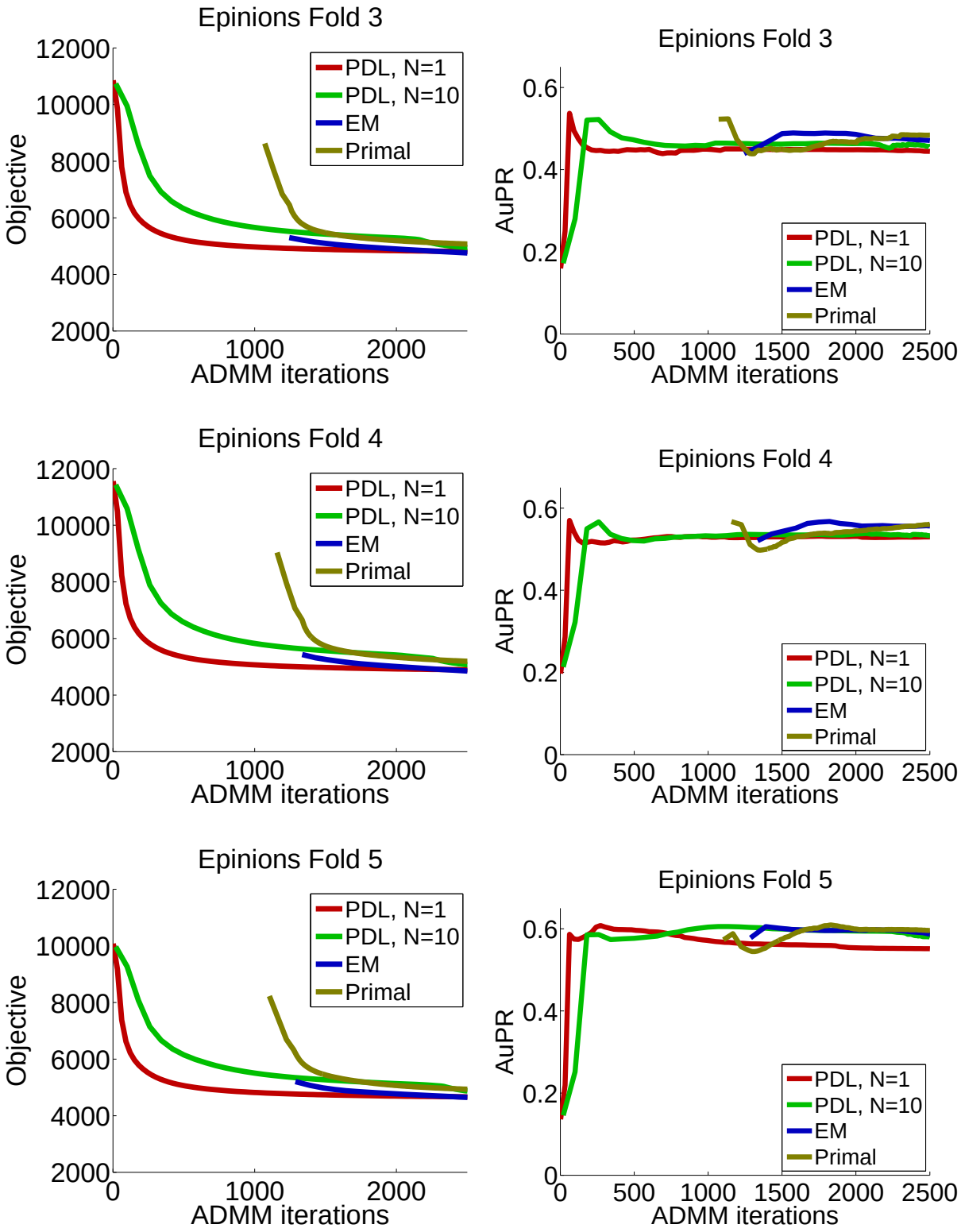

Figure 6. Results for social-trust prediction on Epinions data set, folds 3-5.

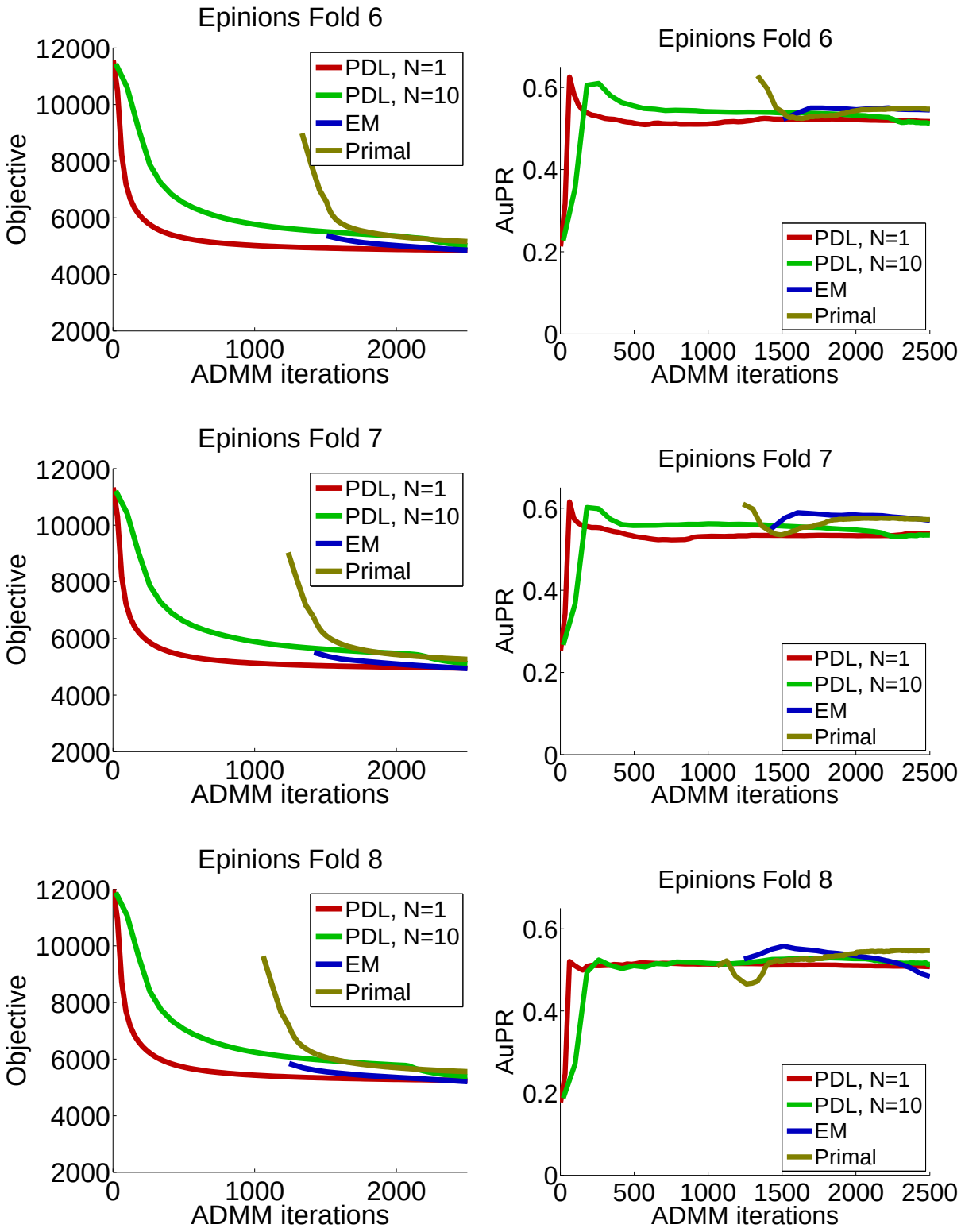

Figure 7. Results for social-trust prediction on Epinions data set, folds 6-8.

(a) Bottom-half model

Example reconstructions

Figure 8. Visual representation of learned face models and outputs. In (a), we visualize the six latent states learned by the model. The images plot the quad root (to enhance contrast at low values) of learned weights for the six latent states. The top row depicts the weights of potentials preferring bright pixels and the bottom row depicts the weights of potentials preferring dim pixel intensities. In (b), we compare the reconstructions of bottom-half faces. The left column is the original, and the middle and right are the latent and flat HL-MRF, respectively.

References

Bach, S. H., Huang, B., and Getoor, L. Learning latent groups with hinge-loss Markov random fields. In *ICML Workshop on Inferning: Interactions between Inference and Learning*, 2013a.

Bach, S. H., Huang, B., London, B., and Getoor, L. Hinge-loss Markov random fields: Convex inference for structured prediction. In *Uncertainty in Artificial Intelligence*, 2013b.

Bach, S. H., Broecheler, M., Huang, B., and Getoor, L. Hinge-loss Markov random fields and probabilistic soft logic. arXiv:1505.04406 [cs.LG], 2015.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 2011.

Huang, B., Kimmig, A., Getoor, L., and Golbeck, J. A flexible framework for probabilistic models of social trust. In *Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction*, 2013.