Predicting YouTube Views and Ratings using Conditional Random Fields

Helen Jiang
Stanford University
helenjiang@stanford.edu

1 Introduction

Since its launch in 2007, YouTube has been a fast-growing network, with over 4 billion videos viewed a day[1]. This social network continues to grow as more videos (nodes) get added every day. In this project, we explore different models for incorporating graph structure on related videos to predict ratings and view counts of YouTube videos. we propose and test a probabilistic model for how a video's attributes compare to the attributes of other videos in its "related" section. Our model is based on conditional random fields from image segmentation literature and aims to capture the intuition that related videos are more likely to have similar ratings and view counts.

Although our work is specific to the related videos network on YouTube, we hope that the ideas in this paper can generalize to other prediction tasks over networks where attributes of neighboring nodes are related, but there is no outright clustering in these attributes.

2 Related Work

Our general question is inspired by the use of conditional random fields in image segmentation for computer vision. A common approach in computer vision (Silberman et al., 2012; Nathan Silberman and Fergus, 2012) for producing a detailed scene segmentation is to first apply a classifier on local patches of the image, and then tune the global set of labels in the image by optimizing an objective which penalizes neighboring pixels for having different labels. This approach has been shown to increase the accuracy of the entire image labeling by a few percentage points, because neighboring pixels are likely to have the same label. We explore whether applying this same model to related videos will allow us to more accurately predict ratings and view counts of YouTube videos. We expect that our results can give us insight on the structure of YouTube ratings across related videos.

We are interested in comparing our computer vision-inspired approach with feature representations of the graphical structure. Semi-supervised methods such as node2vec (Grover and Leskovec, 2016) and DeepWalk (Perozzi et al., 2014) learn a feature representation of the nodes in the graph by optimizing an objective that preserves information about the neighborhood of the nodes. In (Grover and Leskovec, 2016), it is shown that node2vec and DeepWalk can be used to predict labels for a set of nodes in the graph, given supervision (labels) for a smaller subset of the graph. We are interested in whether these feature embedding methods will also be able to predict YouTube ratings and view counts, which could have higher variance across re-

[1] http://www.personal.ceu.hu/tex/footnote.htm

lated videos.

In a social network context, the problem of prediction based on information from similar nodes has been tackled in the literature on recommendation systems. In (He and Chu, 2010), He and Chu investigate the problem of using social network information to predict a user's interest in a product based on information about a user's friends. They model interactions between users in a similar way that CRFs from computer vision model interactions between neighboring pixels; however, the difference here is that they have full access to past ratings. In our setting, the ratings of neighbors are not known.

On the other end of the supervision spectrum, there are unsupervised techniques for clustering such as modularity maximization (Chen et al., 2014). Chen, et. al explore heuristics for modularity maximization, which is an unsupervised methods for extracting clusters from a graph. While similar optimization techniques apply to our setting (simulated annealing, for example), we have the benefit of a small training set to train local classifiers which are then used in the CRF. We also note that intuitively, we should not expect videos to cluster based on ratings and view counts the same way they would cluster based on video category. Thus, our objective is different from these clustering approaches.

3 Method

3.1 Dataset

We use the YouTube dataset provided to us by (Cheng et al., 2008) for our project. This dataset includes 749361 crawled YouTube videos before February 2007. However, since some of the videos were only partially crawled, we prune out these videos, ending up with 730110 total videos used in our dataset.

The method with which (Cheng et al., 2008) crawl the dataset induces a graph structure: after crawling a video, they add up to 20 related videos to the crawl queue, thus performing a breadth-first search from the starting video. This allows us to construct an undirected graph where an edge exists between videos if one video is in the related section of another. Figure 1 shows the resulting degree distribution on this graph on a log-log scale. We see that the degrees do not entirely follow a power-law distribution because there are fewer low-degree videos, resulting from the fact that most videos will have at least 20 related videos.

For each of these videos, we are given:

- an 11-digit unique string as the video ID
- the video uploader's username
- video age and category
- video length and number of views (at the time of the crawl)
- video rating and number of ratings.
- number of comments
- up to 20 video ID's that are related to this video

Because the dataset we are using was created in 2007, ratings on YouTube were on a scale of 0 to 5. In each video, we are given the average rating.

In our dataset, we found that roughly 15.6% of the videos had a very low rating of less than 1. 2.8% had a rating between 1 and 2, 3.8% between 2 and 3, 10.9% between 3 and 4, and 67.0% with a high rating of greater than 4. This means there is a very uneven distribution between the ratings, and because we want the slots to be as even as possible, we categorize ratings into the following buckets, which we have found makes the ratings more or less equal:

- rating < 2.8

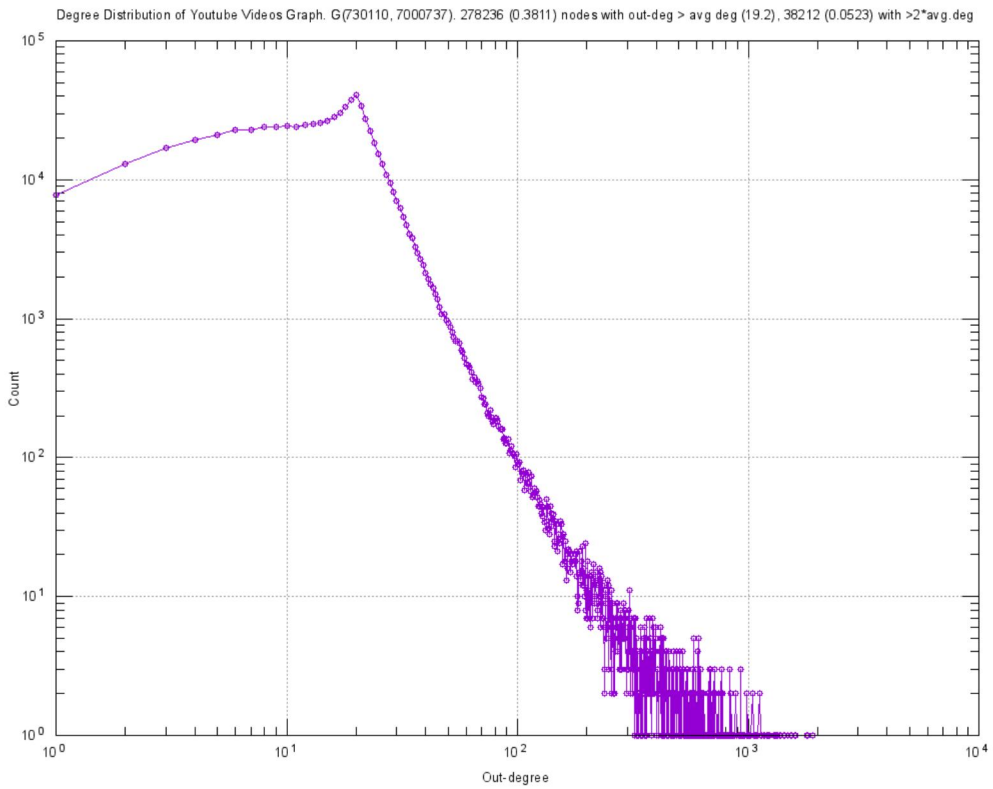


Figure 1: Degree distribution for the crawled YouTube videos.

- $2.8 \leq \text{rating} < 4.5$
- $4.5 \leq \text{rating} < 4.95$
- $4.95 \leq \text{rating}$

For our prediction tasks, we attempt to predict which of the four ratings categories a given video falls under. We label these categories 0 through 3. We also discretize view counts similarly, dividing videos into two classes based on whether number of views is less than 1200, labeling these categories 0 through 1. We find that this splits our dataset into two classes of roughly equal size. In Figure 2, we show the precise proportions in each class.

0	1	2	3
0.215	0.270	0.267	0.248

(a) Proportion of nodes in each ratings category.

0	1
0.491	0.510

(b) Proportion of nodes in each views category.

Figure 2: We show how categories are distributed across the dataset.

3.2 Baseline Predictor

We implement a naive softmax classifier that predicts the ratings category based on five simple features: video age, video length, number of views, number of ratings, and number

of comments. We train using gradient descent on a training set of 1000 randomly sampled videos from the graph, and tune learning rate on a validation set of 30000 randomly sampled videos. The inputs to the classifier are the vectors of concatenated features, where each vector is normalized to have Euclidean norm 1. Because our classifier has so little parameters, there was no evidence of overfitting, as our training and validation accuracies were virtually the same. For view counts, we implement the same type of classifier, except we use video rating as a feature instead of number of views. We used the scikit-learn neural network package to implement the classifier.

We also implement the same type of classifier for both ratings and view prediction, this time using node2vec features as input instead of video metadata features. Our node2vec features have dimension 128. To extract node2vec features, we run the SNAP implementation of node2vec on the directed graph of video relation. If Video A is a related video of video B, that does not necessarily mean video B will be on the related videos for video A. Also, the undirected graph caused the computer to run out of memory.

3.3 Conditional Random Field

Our aim for this project was to leverage the additional supervision of the graph structure to try to perform some more meaningful predictions on the video ratings. In particular, we would like to find a good probabilistic model for the assumption that a video's ratings and view counts are likely to be similar to those of its neighbors. In our analysis, we show that this assumption is valid, although our probabilistic model has more mixed results.

Our probabilistic model is based on conditional random fields from computer vision (Silberman et al., 2012; Nathan Silberman and Fergus, 2012). For video v in our dataset V, let r_v be its predicted rating or view category. Let $f(v)$ be the feature vector of video age, length, etc. We will model the energy function of our CRF as follows:

$$E(R) = \sum_{v \in V} \phi(r_v, f(v)) + \sum_{v \in V} \sum_{v' \in N(v)} \psi(r_v, r_{v'}, f(v), f(v')) \quad (1)$$

$\phi(r_v, f(v))$ will be the negative log probabilities of rating r_v as predicted by our softmax classifier. ψ will be a function imposing a "difference penalty" between related videos; following the computer vision literature (Silberman et al., 2012) we let

$$\psi(r_v, r_{v'}, f(v), f(v')) = \|r_v - r_{v'}\| \alpha \exp\left(\beta \frac{f^*(v) \cdot f^*(v')}{\|f^*(v)\| \|f^*(v')\|}\right) \quad (2)$$

where $\alpha, \beta > 0$ are hyperparameters that we tune. We end up using $\alpha = 0.0003$, $\beta = 3$. To compute $\|r_v - r_{v'}\|$, we compute the number of buckets separating the ratings of v and v'. Finally, f^* is a normalization of f that is necessary because the components of f are on different scales (i.e. the view count will in general be much larger than the number of comments). The choice of $f^*(v)_i$ that worked best on the validation set turned out to be

$$f^*(v)_i = \frac{f(v)_i - \min_{v \in V} f(v)_i}{\max_{v \in V} f(v)_i - \min_{v \in V} f(v)_i}$$

so that $f^*(v)$ has all its components between 0 and 1. We experimented with different choices of ψ and f^*; for example we also tried letting ψ be independent of the features and depend only on the ratings categories, and we also experimented with using unnormalized $f(v)$ in Equation 2. The functions highlighted are the ones that worked best on our validation set.

In Equation 2, we take the cosine similarity between $f(v)$ and $f(v')$; if this is large, then it is more likely that $f(v)$ and $f(v')$ have similar ratings, and we therefore want to penalize dissimilar ratings more.

We wish to find a configuration of ratings which minimizes the energy objective; since this objective is intractable we will use simulated annealing to find a local optimum: we can perform Gibbs-sampling passes through ratings predictions with $R \sim \exp(E(R)/T)$ as the target distribution, where we slowly decrease T according to a schedule that we tune as a hyperparameter. Our initial configuration will be the predicted ratings from applying the classifier alone, without the CRF structure on top of it.

We note that a simple softmax classifier as described in Section 3.2 can be viewed under this CRF framework as the special case when we set the hyperparameter α to 0. In this case, all predictions are made independently from predictions on neighboring nodes.

4 Results and Analysis

We provide numerical results for our experiments in this section. Although our CRF model does not end up improving the accuracy of our predictions, we provide an argument here that models of this type could still be worth exploring in future work.

4.1 Structural Analysis

We analyze the quality of the ratings and views categorizations in the related videos graph to determine how much they "cluster".

The first we metric that we analyze is modularity, which rates the strength of clustering of the network into its different categories (Newman, 2006). The modularity is defined as the fraction of edges that go within a category minus the expected fraction if the edges are distributed at random while preserving vertex degrees. We find that for view count categories, the modularity is 0.0650, and for rating categories, the modularity is 0.0740. The expected modularity if categories were randomly assigned is 0. This means that categories are correlated between neighboring videos, but not to the point of outright clustering.

Category	0	1	2	3
0	0.432	0.238	0.124	0.206
1	0.225	0.370	0.202	0.203
2	0.140	0.229	0.380	0.251
3	0.181	0.210	0.239	0.370

(a) Conditional distribution of ratings categories given majority categories over neighbors.

Category	0	1
0	0.680	0.320
1	0.401	0.599

(b) Conditional distribution of views categories given majority categories over neighbors.

Figure 3: Conditional distribution tables for categories given the majority neighbor category. The majority neighbor category is on the vertical axis, and the node category corresponds to the horizontal axis. For example, from the ratings table, for nodes whose most common neighbor category is 0, 23.8% of these nodes have a category of 1.

Next, we empirically examine how neighboring nodes affect the categorization of a vertex. To do this, we compute the distribution of ratings and views categorizations, conditioned on the class of the majority of the video's neighbors. Figure 3 shows these distributions. Compared to the true distribution over view or ratings categories, these conditional tables clearly show that the categories of the neighbors strongly influence a video's category.

4.2 Experimental Results

We wish to compare our different methods against the baseline of a simple softmax classifier on video metadata. We have not been able to get a classifier based on node2vec features to exhibit reasonable performance, as our training accuracy is roughly around that of random guessing. This is likely because the ratings categorizations do not delineate the

	CRF	Softmax
Test Accuracy	0.460	0.462

(a) Test accuracy on ratings category prediction.

	CRF	Softmax
Test Accuracy	0.767	0.773

(b) Test accuracy on views category prediction.

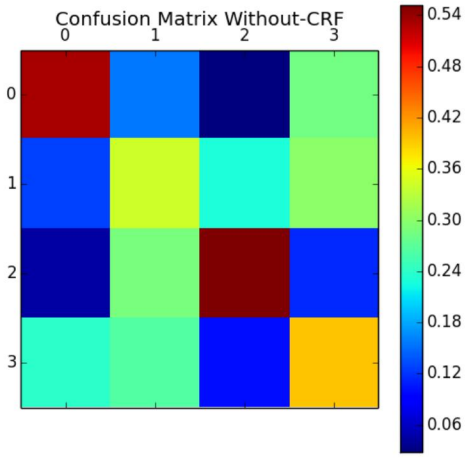
Figure 4: Test accuracies for our prediction tasks.

graph into clear enough clusters, as seen by our modularity scores, which, while positive, are still quite low. Thus, for the remainder of our experiments, we compare just our probabilistic CRF model applied to softmax class probabilities as input with our simple baseline softmax classifier.

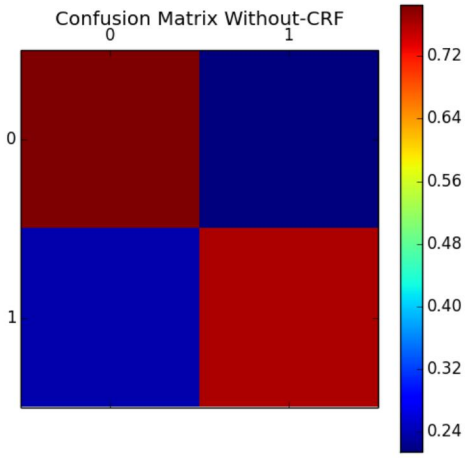
We evaluate performance on the test set, which we define as the entire dataset minus the train and validation sets. Figure 4 shows our accuracy results for the two prediction tasks. As seen from the figure, the conditional random field performs slightly worse for both tasks.

We further analyze our model in order to explain the poorer performance of our conditional random field. In Figure 5, we show confusion matrices for the softmax classifier; the confusion matrices for the classifier with the CRF look virtually identical. One main takeaway from these confusion matrices is that the prediction classes are not balanced: for example, for ratings the classifier tends to predict category 1 much less and for views the classifier also tends to predict category 1 less. We explain why this difference matters in later analysis.

We also require insight into what the conditional random field actually does, since it only changes the prediction accuracy by a few fraction of percentage points. We find that compared to the softmax-only classifier, the CRF



(a) Confusion matrix for ratings category predictions.



(b) Confusion matrix for views category predictions.

Figure 5: Confusion matrices of the softmax classifier for our different prediction tasks. The x axis corresponds to true ratings categories, and the y axis corresponds to predicted categories. We do not show the confusion matrix for the classifier with the CRF on top because there is no visually discernable difference.

on top of softmax classifier actually relabels 2.36% of the videos for view count categorization, and 3.89% of the videos for the ratings categorization. Compared to the amount that the prediction accuracy changes, these values are quite significant, and means that the CRF does make a substantial amount of changes, though most of these changes are wrong.

Figure 6 displays the distribution over class probability predicted by the softmax classifier for videos whose labels are changed by the CRF, and compares that to the class probability distribution over the entire dataset. From the figure, it is apparent that the CRF rarely changes the label of videos that the softmax classifier is already quite certain about. This behavior is what we would expect from the CRF objective and is also desirable because we want the CRF to correct predictions that the softmax classifier is uncertain about.

We can further evaluate whether the CRF is working as desired by comparing the energy objective of the CRF for our different sets of labels, which we display in Figure 8. Recall that the energy objective is modeled as in Equation 1, and the simulated annealing optimization of the CRF aims to minimize the energy objective. We also compute the second sum in Equation 1 involving only the ψ terms (defined in Equation 2) because it does not make sense to apply softmax class probabilites to ground truth or random labels. The random labeling is generated by sampling a label for each video i.i.d. according to the empirical distribution over categories.

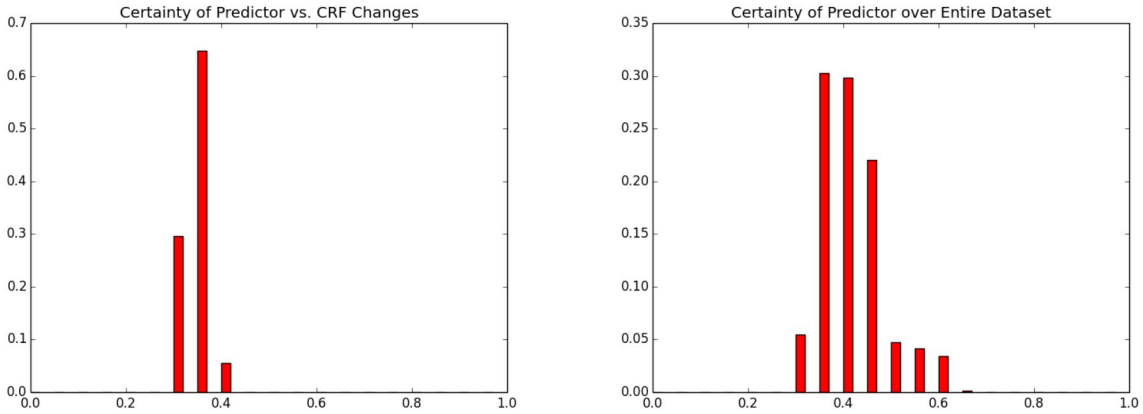
This experiment confirms two things: 1) our optimization strategy is effective in reducing the energy of the CRF, as seen by the fact that the CRF has lower energy values in both categories than all other labelings we look at. This means the CRF model is problematic, not the optimization. 2) The ground truth labels are indeed lower in the CRF objective than randomly generated labels. Intuitively, this could mean that minimizing some objective of this form could be a reasonable heuristic, as long as the objective is right.

We note that the predictions outputted by the softmax classifiers already have lower pairwise energy objectives than the ground truth labels, even without any CRF-specific optimization. One of the reasons here is imbalance in predictions as seen in Figure 5; class imbalance means that there will be more edges going between videos of the same category, thus lowering pairwise terms in the energy objective.

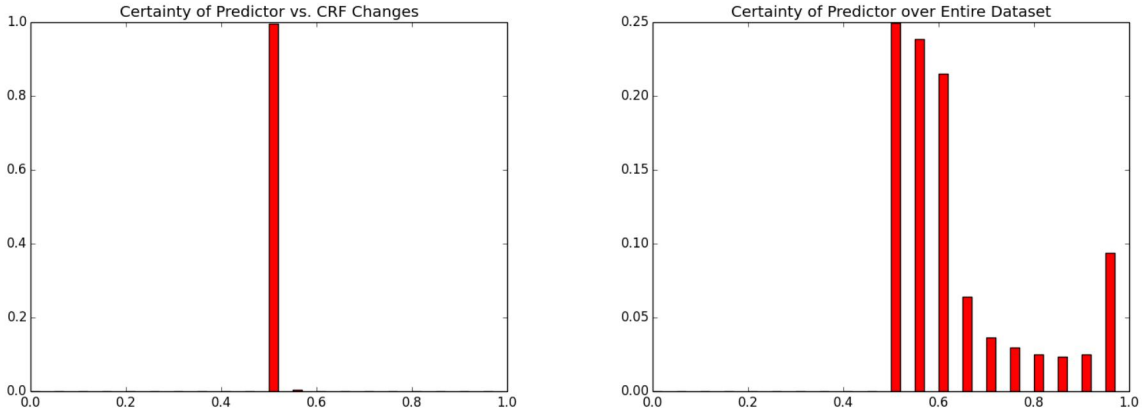
Class imbalance does not entirely explain the lower energy objective of the softmax classifier, however. The modularity objective is more resilient to class imbalance because it pertains to fractions of edges. Figure 7 again confirms the idea that the CRF does increase the amount of clustering in the labels, as we compute the modularity score for CRF-predicted labels as well as labels predicted by only the softmax classifier. However, interestingly the modularity score of the softmax classifier is still closer to the ground truth labels than the modularity score of the CRF classifier. This could be because related videos will also have similar features that are inputted into the softmax classifier, so the softmax classifier will naturally output similar predictions for related videos.

4.3 Main Takeaways

In summary, the main takeaways of our analysis should be as follows: the assumption that related videos will have more similar view counts and ratings is valid, but our current CRF formulation does not leverage this assumption in the right way for prediction. From our analysis of the behavior of the CRF, we know that our algorithm is roughly optimizing the objective as intended, in that the videos it relabels tend to be ones that the softmax classifier is already uncertain about, and the energy objective decreases when the CRF is used. However, because this objective is in-



(a) Distributions over class probability of softmax classifiers for rating categorizations changed by the CRF.



(b) Distributions over class probability of softmax classifiers for view count categorizations changed by the CRF.

Figure 6: We plot the empirical distribution of the class probability for the class predicted by the softmax classifier. We compare the distribution over the entire dataset to the distribution for just the videos where the CRF output does not agree with the softmax prediction. To estimate these distributions, we discretize these class probabilities into 20 evenly spaced buckets and plot the proportion of examples falling into each bucket. The x axis is the class probability predicted by the softmax classifier, and the y axis is the proportion of examples in that bucket.

	CRF	Softmax	G. Truth
Modularity	0.0938	0.0703	0.0740

(a) Modularity of CRF and softmax labels for ratings.

	CRF	Softmax	G. Truth
Modularity	0.0894	0.0638	0.0650

(b) Modularity of CRF and softmax labels for views.

Figure 7: We show the modularity of the labels produced by the CRF and softmax classifiers, respectively.

fluenced heavily by class imbalance, as argued in the previous section, it is questionable whether this objective is the right one. It would be interesting to optimize a modularity-based objective instead (i.e. something akin to maximizing modularity), which will be less influenced by imbalance between classes.

Finally, there is also the question of whether we attempt to cluster ratings and view counts categorizations too much by optimizing a global objective influenced by labels of related videos. This is because similarity between related videos might already be implicitly captured by the fact that related videos have similar feature inputs into the softmax classifier.

5 Conclusion

We explore the task of prediction in the general setting where our labels are related based on social network structure, but not strongly related enough to form clusters. Drawing on computer vision literature, we propose using a conditional random field to model these relations between neighboring videos.

We test our model on a YouTube dataset of related videos, attempting to predict ratings and view counts. While we are unable to report improved performance by our model, from a structural analysis of the YouTube dataset graph we provide evidence that *some* model of this form can work. For example, our model has a lower objective function value when evaluated on the ground truth labels than on randomly generated labels, suggesting that some of the assumptions in our model are accurate. Thus, while the model's performance is not good right now, we believe that it can lead to interesting directions of future work.

For future work, it would be interesting to change our model's objective to be invariant to class imbalances, which is a shortcoming of the current model we used. Another direction of future research could also be to learn a function for pairwise potentials in the CRF directly from the training data, instead of tuning it as a hyperparameter. However, it is unclear whether this approach is computationally feasible.

References

Mingming Chen, Konstantin Kuzmin, and Boleslaw K Szymanski. 2014. Community detection via maximization of modularity and its variants. *IEEE Transactions on Computational Social Systems*, 1(1):46–65.

Xu Cheng, Cameron Dale, and Jiangchuan Liu. 2008. Dataset for "statistics and social network of youtube videos".

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks.

Jianming He and Wesley W Chu. 2010. *A social network-based recommender system (SNRS)*. Springer.

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*.

Mark EJ Newman. 2006. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowl-*

	CRF	Softmax	Ground Truth	Random
Energy Objective	648678	652580	n/a	n/a
Energy Objective, Pairwise Only	30725	35819	50424	69231

(a) Energy objective for different ratings categorizations.

	CRF	Softmax	Ground Truth	Random
Energy Objective	354312	356012	n/a	n/a
Energy Objective, Pairwise Only	24881	27147	28202	35061

(b) Energy objective for different views categorizations.

Figure 8: CRF energy objectives. Since it does not make sense to evaluate the softmax classifier on ground truth labels, we also provide the energy objective due to pairwise potentials only (sum of all the ψ values).

edge discovery and data mining, pages 701–710. ACM.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer.