DARPA PPAML Challenge Problem #6: Image Labeling

Version 14, 7 July 2015

Summary

Multimedia data such as video and pictures are being produced and shared at an unprecedented and accelerating pace in recent years. For example, on YouTube, video data is currently being uploaded at the rate of approximately 30 million hours a year. This drives a strong need to develop automated tools to help users understand, organize, and retrieve images and videos from very large collections.

The goal of the proposed challenge problem is to assign labels (e.g. "tree", "people", or "baby") to images based on relationships found in a rich social multimedia database containing visual features and metadata such as user information (e.g. username, location, network of contacts), comments, user image gallery, uploader defined groups, and links between shared content. When all this information is used collectively in a suitable fashion, it may be able to advance the state-of-the-art in image labeling.

Tags: [empty]
Labels: plant_life, sky, structures, tree

Tags: Paris, Ile-de-France, France, Act Up-Paris
Labels: female, people, plant_life, sky, structures, tree

Tags: Dashboard, steering, wheel, Knox Cruise Night
Labels: car, night, structures, transport

Tags: Tuscon, flower, Om
Labels: people, plant_life

Sample images, first four associated user-supplied tags, and ground-truth labels, from [Huiskes2008].

Problem Specification

Overview

The problem of multimedia retrieval is to develop the scientific methodology to understand and discover images/videos with particular content from a complex, large, and growing collection of multimedia. Real-world multimedia, especially as shared on the Internet, can be challenging to retrieve using only visual information, due to complex content, partial occlusion, and diverse styles and quality. The most common solution to this problem is to annotate media with keywords that describe the content and then perform keyword search against these annotations. The problem of annotating images consists of inferring content labels, L, conditioned on an image, I, and other related metadata information, M, e.g. $P(L|I,M)$.

In Challenge Problem 6 (CP6), we will solve the task of automatic image annotation or labeling by exploiting the metadata, M, in addition to the visual information, I. Some types of metadata (i.e. EXIF tags) are generated by the camera when the image is taken; others (i.e. user-provided tags, comments from viewers) are generated after the image is uploaded to an image-sharing service such as Flickr. We will use a subset of the MIRFLICKR [Huiskes2008] dataset to supply the ground-truth image labels, image features, and related metadata. The MIRFLICKR data is available under Creative Commons licenses. The subset is the one used by [McAuley2012], which we will henceforth refer to as the MIR14k dataset.

Within the PPAML taxonomy of challenge problems, this CP is related to the Intelligence Analysis domain; the data structures are a hybrid of discrete (categorical) and continuous (features and feature distances) presented in both relational and vector forms. The basic parametric probabilistic model is an undirected graphical model over a fixed model structure with latent variables. Queries are formulated as marginal *maximum a posteriori* MAP for individual images, or joint MAP for the entire graph. The query timing is one-shot with slow tempo and stationary parameters.

Problem Statement

Some definitions:

- A label is a string representing the "ground truth" of concepts present in an image. MIR14k provides 24 labels, such as "river", "dog", and "baby". For the purposes of CP6, the set of available labels is fixed and independent of any particular set of images (although we assume it is applicable to whatever set of images we are considering.)
- An image I is a matrix of pixel values. The raw images will be made available; however, instead of requiring the teams to work with raw images, we will summarize the images in terms of various image features such as histograms, texture measures, bag-of-word descriptors, and other standard descriptors, as well as more specialized features such the output of detectors tuned to specific real-world objects such as cars or people.
- Each image is associated with metadata M, which may be divided into two types: **intrinsic**, based on the EXIF data collected from the camera, and **relational**, describing the context of the image on Flickr. The following list is provided as guidance; precise file

formats may be found in the "Solution I/O Specification" section. The following fields will be provided when present on the original image:

- Date and time the picture was taken
- Whether or not the flash fired
- Geo-location (uncommon)

Relational metadata includes:

- The Flickr user ID of the photo's owner
- Photo title (a string)
- The free-form text tags associated with the image, and who provided the tag. These text tags are unrelated to the labels in the label set L.
- The Flickr **groups** to which the photo belongs. A group may contain photos belonging both to the group's owner and other Flickr users. In the context of this dataset, groups provide information via the title of the group. For example, a photo of a palm tree may be in two groups, one titled "Vacation photos" and another titled "Hawaii". Other photos in the group will not generally be available.
- The Flickr **galleries** to which the photo belongs. A gallery may only contain photos belonging to other Flickr users.. Galleries provide both a title and a description, as ancillary information such as the number of photos in the gallery, the number of times it has been viewed, etc.

Not every photo will have every item of metadata.

Given these definitions, the problem can be stated as follows:

Given:

- A database of images
- Metadata for those images
- Labels for a subset of those images (the training set)

Find:

- The labels of all other images in the database

The following sections describe the baseline model, data package and formats, and evaluation methodology.

Probability Model

The probability model for CP6 is based on a conditional random field (CRF). This captures both unary dependencies between image labels, $L = \{l_1, l_2, \ldots, l_N\}$, and the input features (e.g. image features and metadata), as well as the pairwise dependencies between pairs of labels and the input features to produce the conditional probability $P(L|I, M) = P(L|\boldsymbol{x}, f, M)$, where $\boldsymbol{x}$ and f are features derived from the image set, I. The observed input features can include data from three sources:

- raw image features
- outputs from specific object detectors or classifiers

- metadata

The raw image features, $\boldsymbol{x}$, include low-level descriptors such as histogram of oriented gradients and color histograms. The classifier outputs, $f_c(\boldsymbol{x}); c = \{1,2,\dots,C\}$, are posterior probabilities or scores that represent how well the data matches a set of class models, which have been previously trained using image features. These classifiers characterize classes such as scene categories (building, grass, road), object categories (person, bicycle, vehicle), and image type (birthday party, nature, dancing). The metadata, *M*, is a vector of entries in a lookup table that indicates the occurrence of *words* (derived from titles, descriptions, and comments), *groups,* and *tags* for a single image.

The metadata is also used to define the cliques in the CRF prior to the parameter learning process. The cliques represent collections of labels that are dependent on each other based on having common properties (i.e. assigned to the same gallery or group).

The image labels, $L = \{l_1, \dots, l_n, \dots l_N\}$, for each of the *N* images are binary values indicating if the image has this class label, $l_n = 1$, or not, $l_n = 0$. The labels are treated as binary hidden nodes in the CRF and the image features, $\boldsymbol{x}$, classifier outputs, f, and metadata, M, are used in the observation nodes. The conditional probability of the CRF is:

$$P(L|I,M) = p(L|\boldsymbol{x},f,M) = \cdots$$

$$\frac{1}{Z}\exp\left(\sum_{n=1}^{N} A_n(l_n,\boldsymbol{x},f,M) + \sum_{n=1}^{N}\sum_{m\in\mathcal{N}(n)} B_{n,m}(l_n,l_m,\boldsymbol{x},f,M)\right), \quad (1)$$

where Z is the normalization constant that depends on $\boldsymbol{x}$, f, and M, while A and B are the unary and pairwise potential functions, and $\mathcal{N}(n)$ is the clique neighborhood. The unary potentials are single image potentials, while the pairwise potentials are between pairs of images. For simplicity, a separate binary CRF model can be learned for each label (e.g., airplane).

The implementation of the CRF can follow the approach used in [Kumar2006, Domke13], where a fixed feature function is used to calculate the unary potential, *A*:

$$A_n(l_n,\boldsymbol{x},f,M) = l_n w^T h_n(\boldsymbol{x}), \quad (2)$$

where

$$h_n(\boldsymbol{x}) = [1, \mathrm{f}_1(\boldsymbol{x}), \dots, \mathrm{f}_C(\boldsymbol{x})], \quad (3)$$

and $\mathrm{f}_c(\boldsymbol{x})$ is the classifier output feature vector, but can include the image features, and metadata, while *w* is a vector of learned weights. The feature vector can be normalized for faster convergence and possibly more accurate results.

The pairwise potential, B, from equation (1) can be modeled using a discriminative model similar to equation (2):

$$B_{n,m}(l_n, l_m, \boldsymbol{x}) = l_n l_m v^T \mu_{n,m}(x), \quad (4)$$

where v is the parameter to be learned for the pairwise potential.. The pairwise potential for image *n* and *m* is denoted as $\mu_{n,m}(x)$ for simplicity, but refers to the features that co-exist for the two images, i.e. $[h_n(\boldsymbol{x})\ h_m(\boldsymbol{x})]$, which can be concatenated image features and/or classifier outputs from the two images. The relational metadata can also be added to the pairwise potential by calculating common properties between the two images. These include:

- Number of common tags
- Flag indicating if both images were taken by the same user
- Temporal separation between two images

Kitware will provide the input features and metatdata that will be used as observations in the CRF as well as correct labels for each image in the training set as part of a data package.

Data Source

CP6 will use a subset of the MIRFLICKR dataset [Huiskes2008] as defined by McAuley [McAuley2012], which contains 14,460 images uploaded to Flickr and licensed under a Creative Commons license. Various logistical constraints for CP6 reduce this to 12,690 images. See "Solution I/O specification" for file format details.

Each image has:

1. One or more label annotations for each image drawn from a vocabulary of 24 concepts. See Figure 1 for label frequency and per-image co-occurrence data.
2. The photo's title, description, and location (any or all may be empty)
3. The owner's user ID
4. EXIF metadata, providing the time the photo was taken, whether the flash was used, etc. when these were present on the original image.
5. Text tags associated with the photo (supplied by the owner or other Flickr users)
6. Information on the Flickr group and galleries the photo belongs to.
7. The comments associated with the photo

(Note that not every image will have all of 5, 6, or 7.)

Additionally, Kitware will provide a suite of features for all the images. The precise set of features is still being developed, but it will include standard features such as bag-of-words, color and edge histograms, and/or wavelet textures. Regardless of the precise feature set, all of these features will be presented as a vector of integers or floats together with a distance metric such as inner product or chi-squared distance. We will also provide more specialized classifier outputs as described above for specific object detections or scene classifications. These will be provided as a per-image vector of likelihoods, one for each classifier type.

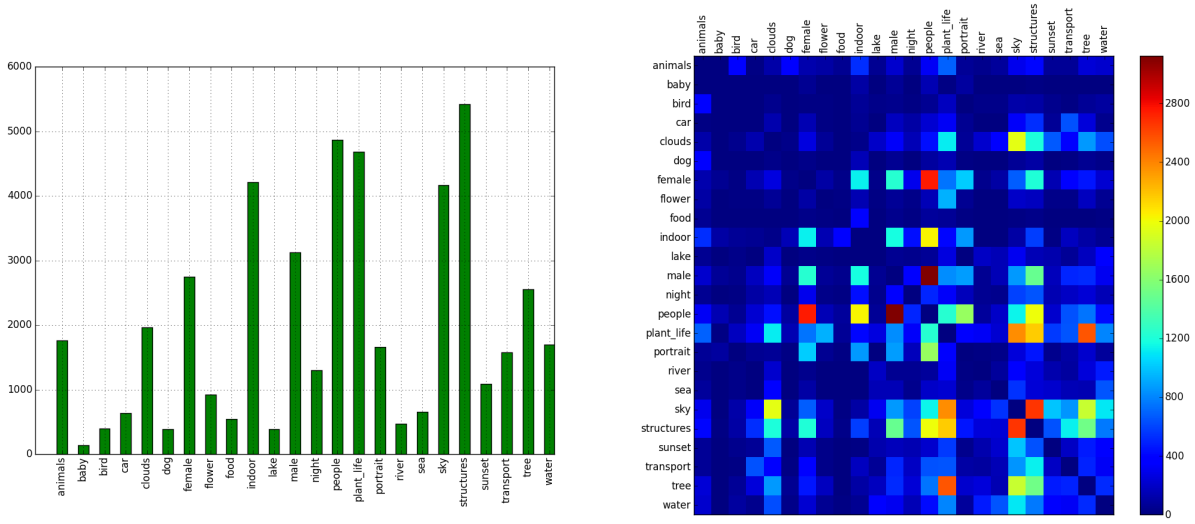

Figure 1; Left, label frequency histogram; right: per-image label co-occurrence heatmap.

The dataset will scale in size from Round 1 to Round 2. The intent is that each round will define training (available) and testing (sequestered) subsets of the MIR14k data, scheduled as follows:

- For both rounds, data will be partitioned into test and train sets based on EXIF timestamps: data taken before December 2007 will be training, data after December 2007 will be test. Data with no EXIF timestamp (approximately 2000 images) will be used for training. Figure 2 shows the distribution of Round 1 and Round 2 data versus timestamp.
- Round 1 (introduced July 2015, evaluated January 2016): This data drop will focus on images which do **not** include the label "structures", which results in a set of 7468 images (3345 train, 4123 test)
- Round 2 (introduced January 2016, evaluated July 2016): This data drop will add in the 5421 images with the label "structures", for a total of 12889 images (5619 train, 7270 test.)

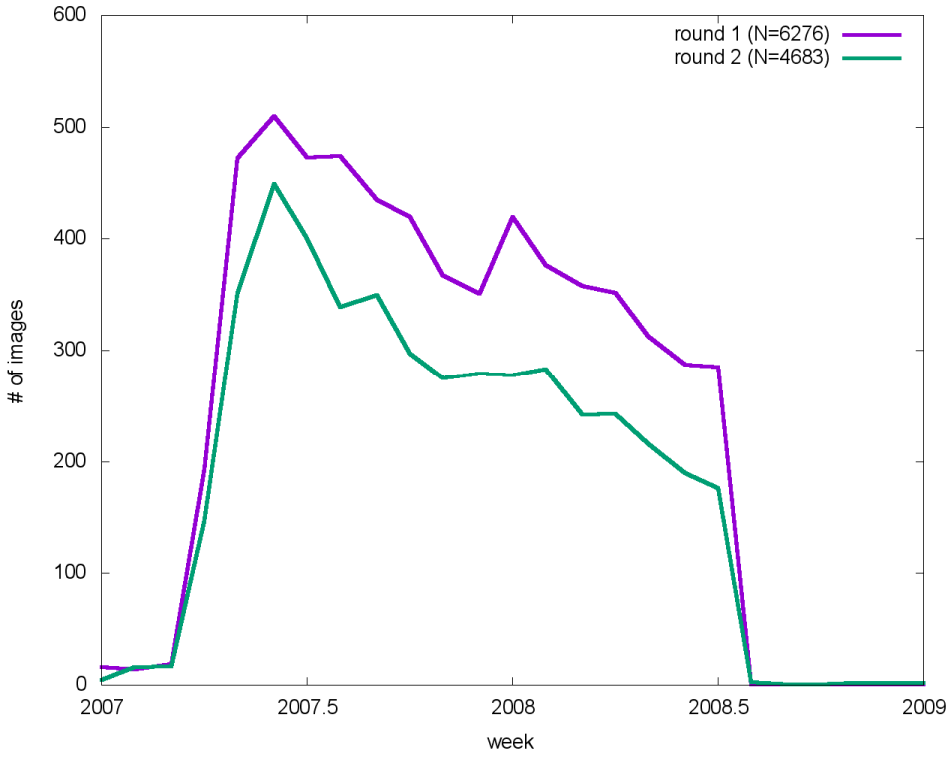

Figure 2; EXIF-timestamp distribution of round 1 and round 2 data

Evaluation

The performance of the image label classifiers will be determined using the test data, which represents about half of the overall dataset. The performance will be measured using both Mean Average Precision (mAP) and Balanced Error Rate [McAuley2012].

The mAP is a single value metric that summarizes the quality of a ranked list of classified images based on their associated classifier probability/score. More precisely, the average precision (AP) is the average of the precision values that are calculated at all true positives in a descending ranked list. The AP is calculated for each class (i.e. label) and then averaged over all label experiments to obtain the mAP.

The Balanced Error Rate is designed to assign equal importance to false positives and negatives, which McAuley et al. believe more accurately represents the performance when simultaneously making binary label predictions for the entire dataset. The balanced error rate is calculated as follows:

$$\Delta(Y, Y_c) = \frac{1}{2}\left[\frac{\left|Y^{Pos} \backslash Y_c^{Pos}\right|}{\left|Y_c^{Neg}\right|} + \frac{\left|Y^{Neg} \backslash Y_c^{Neg}\right|}{\left|Y_c^{Pos}\right|}\right] \quad (5)$$

where Y^{Pos} is the set of images with positive predicted labels, Y^{Neg} is the set of images with negative predicted labels, and Y_c^{Pos} and Y_c^{Neg} are the sets of correct positive and negative images, respectively.

A reference implementation of the CRF model will be provided, as well as McAuley's alternative approach using max-margin optimization [McAuley 2012].

Solution I/O Specification

This section first describes the text files which contain the training and test data, then how input and output files are formatted.

Dataset Overview

The dataset will be provided as a set of "flat" text files. Lines are space-separated sequences of values. Strings are UTF-8 encoded, enclosed in double-quotes. Any double-quotes, carriage returns, or linefeeds in the string are replaced by spaces. Data fields of type 'char' are single characters (without double-quotes.)

Vectors of numbers (either integers or doubles) are sequences of numbers separated by commas, without spaces, e.g. '1,0,-1,1,-2'. If the vector is empty, the (unquoted) string 'none' is used.

Label Table

This file contains 24 lines, one line per label; the fields are:

`label_id label_string`

label_id	unsigned	The ordinal of this label (range 0-23).
label_string	string	The text of the label.

The labels are the 24 identifiers associated to the MIRFLICKR image set by the MIRFLICKR team.

Image Table

This file contains one line per image; the fields are:

`index image_id owner_id title description label_vector`

index	unsigned	The ordinal of this image in the dataset.
image_id	unsigned	The Flickr ID of the image.
owner_id	string	The Flickr ID of the owner.
title	string	The title of the image, or "none"

description	string	The description of the image, or "none"
exif_date	date string	The EXIF date ("YYYY:MM:DD"), or "none" if not available.
exif_time	time string	The EXIF time ("hh:mm:ss"), or "none" if not available.
exif_flash	char	"Y" if the flash fired, "N" if not, "U" if not known.
flickr_locality	string	Locality string returned by flickr API, or "none" if not available.
label_vector	vector of double	A 24-element vector of doubles; the i'th element indicates the likelihood that label i applies to the image. See below for special values.

It is intended that the image_id is unique and stable across test and train datasets for both rounds.

Valid entries for the label_vector are shown below:

[0..1]	testing / training	Values from [0..1] indicate the likelihood that the label applies to the image.
-1	testing / training	Indicates that the label does not apply (for example, the label is not being used in this round.) In training, indicates that no examples of this label will be found in the entire dataset. In testing, indicates that this label should not be estimated.
-2	testing only	Indicates that this label likelihood is to be estimated as part of testing.

A label_vector containing only -1 or -2 values is said to be "empty".

Image Feature Table

Each image feature type (edge density histogram, color histogram, etc.) will generate a table of fixed-length vector of floats for each image. The length of the vector will vary with each feature, but all image feature tables will have one line per image:

```
image_id N feature_vector
```

image_id	unsigned	The ID of the image to which this feature vector refers.
N	unsigned	The dimensionality of this feature vector. (Fixed across each file.)
feature_vector	N x double	An N-long vector of doubles representing the image feature.

Image Detector Table

In addition to image features, a bank of specialized object detectors (person, vehicle, etc.) will be run against each image. Each detector returns a likelihood that the image contains the object. The image detector file format has a header line (highlighted in blue) followed by one output line per image:

```
N-detectors detector-label-1 detector-label-2 ... detector-label-N
image_id detector-output-1 detector-output-1 ... detector-output-N
```

N-detectors	unsigned	(header line) Number of detector outputs per line.
detector-label-*i*	string	(header line) The label of the target of the i'th detector.
image_id	unsigned	The ID of the image to which this feature vector refers.
detector-output-*i*	double	The likelikehood output of the i'th detector on this image

Image Indicator Lookup Table (LUT)

This defines the "dictionary", used by McAuley's baseline solution, which identifies "the 1000 most popular words, groups, and tags across the entire dataset, as well as any words, groups, and tags that occur at least twice as frequently in positively labeled images compared to the overall rate." [McAuley, section 5]. He further states "As word features we use text from the image's title, description, and its comment thread, after eliminating stopwords." McAuley does not use gallery information, probably because shared galleries are much less common than shared groups. The file format has a header line (highlighted in blue) followed by one entry per line.

```
N-groups N-words
entry_id group-or-word entry-string
```

N-groups	unsigned	(header line) Number of groups
N-words	unsigned	(header line) Number of words
entry_id	unsigned	The ordinal of the entry. Groups and Words are independently numbered (i.e. there is both a group 0 and a word 0.)
group-or-word	char	The entry is a "G" (group) or "W" (word)
entry-string	string	Either a Flickr Group ID or a word from the word sources as described above.

Image Indicator Table

This table lists how the groups, tags, and text associated with each image are represented in the Image Indicator Lookup Table. There is one line per image of the format:

```
image_id group_indicator word_indicator
```

image_id	unsigned	ID of the image.
group_indicator	vector of unsigned / 'none'	Vector of LUT group entry_ids; each element is the entry_id of a group associated with this image. If there are no groups, the (unquoted) string 'none' is present.
word_indicator	vector of unsigned / 'none'	Vector of LUT word entry_ids; each element is the entry_id of a word associated with this image. If there are no words, the (unquoted) string 'none' is present.

Edge Table

This table indicates the shared properties between any two images *A* and *B*. When two images share no properties, the edge is omitted. Edges are non-directional and the assignment of images to *A* or *B* is arbitrary. There is one line per edge with the following format (broken across multiple lines for legibility):

```
image_A_id image_B_id
  N_shared_groups N_shared_words
  shared_group_id_vector shared_word_id_vector
  same_user_flag same_location_flag shared_contact_flag
```

image_A_id	unsigned	ID of image A.
image_B_id	unsigned	ID of image B.
N_shared_groups	unsigned	Number of shared groups
N_shared_words	unsigned	Number of shared words
shared_group_id_vector	vector of unsigned / string	Vector of LUT group entry_ids; each of the N_shared_groups elements is a group shared between images A and B. If N_shared_groups is 0, the unquoted string 'none' is present.
shared_word_id_vector	vector of unsigned / 'none'	Vector of LUT word entry_ids; each of the N_shared_words elements is a word shared between images A and B. If N_shared_words is 0, the unquoted string 'none' is present.
shared_word_type_vector	vector of unsigned / 'none'	See below; if N_shared_words is 0, the unquoted string 'none' is present.
same_user_flag	char	'1' if the same user took images A and B, '0' if different users took the images, or '.' if unknown
same_location_flag	char	'1' if images A and B share a "locality" (as defined by Flickr), '0' if different localities are specified, or '.' if one or both images has no locality
shared_contact_flag	char	'1' if images A and B are taken by users who share a contact, '0' if it is determined that they share no contacts, or '.' if no determination can be made

The elements of shared_word_type_vector are bit flags, each describing the source text fields of the word shared between images A and B;

Image A title	0x01
Image A description	0x02
Image A tags	0x04
Image A comments	0x08
Image B title	0x10
Image B description	0x20
Image B tags	0x40
Image B comments	0x80

For example, suppose the following:

- The value of shared_word_id_vector[3] is 119.
- According to the Image Indicator Lookup Table, word 119 is "dog"
- The value of shared_word_type_vector[3] is 106 (hexadecimal 6A, binary 0110 1010)

This means that "dog" appears in Image A's description and comments, and in Image B's description and tags.

Training vs. Testing data

For each round, the training data will be supplied as a set of tables described above. Testing data will be the same tables augmented with the test images in the Image Table (with empty label_vectors); the other tables will be modified as appropriate to indicate connections to the training data.

Test Input format

The test input will be an instance of the data tables, set as above for testing; test images may be detected by their empty label_vectors in the Image Table (those containing only -1 or -2 values).

Test Output format

The output shall be a copy of the Image Table with the test image's label_vectors filled in (i.e. all values of -2 have been replaced with label likelihoods.)

References

- [McAuley2012] Julian McAuley and Jure Leskovec, Image Labeling on a Network: Using Social-Network Metadata for Image Classification, in ECCV 2012
- [Chua2009] T.S. Chua, J. Ang, R. Hong, H. Li, Z. Luo, Y.T. Zheng, “NUS-WIDE: A real-world web image database from the National University of Singapore,” CIVR, 2009
- [Huiskes2008] M. Huiskes, M. Lew, “The MIR Flickr retrieval evaluation,” CIVR, 2008
- [Nowak2010] S. Nowak, M. Huiskes, “New strategies for image annotation: Overview of the photo annotation task at ImageCLEF 2010,” CLEF, 2010
- [Denoyer2010] L. Denoyer, P. Gallinari. “A ranking based model for automatic image annotation in a social network,” ICWSM, 2010
- [Lindstaedt2008] S. Lindstaedt, V. Pammer, R. Morzinger, R. Kern, H. Mulner, C. Wagner, “Recommending tags for pictures based on text, visual content and user context,” Internet and Web Applications and Services, 2008
- [Sigurbj ornsson2008] B. Sigurbjornsson, R.V. Zwol, “Flickr tag recommendation based on collective knowledge,” WWW, 2008.
- [Sawant2010] N. Sawant, R. Datta, J. Li, J. Wang, “Quest for relevant tags using local interaction networks and visual content,” MIR, 2010

- [Stone2008] Z. Stone, T. Zickler, T. Darrell, “Autotagging Facebook: Social network context improves photo annotation,” CVPR Workshop on Internet Vision, 2008
- [Kumar2006] S. Kumar and M. Hebet, “Discriminative Random fields,” IJCV, 2006
- [Domke13], Justin Domke, “Learning Graphical Model Parameters with Approximate Marginal Inference,” TPAMI, vol. 35, no. 10, pp. 2454-2467, 2013