Semi-supervised Learning with Constraints for Person Identification in Multimedia Data

Martin Bäuml Makarand Tapaswi Rainer Stiefelhagen
Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
{baeuml, tapaswi, rainer.stiefelhagen}@kit.edu

Abstract

We address the problem of person identification in TV series. We propose a unified learning framework for multi-class classification which incorporates labeled and unlabeled data, and constraints between pairs of features in the training. We apply the framework to train multinomial logistic regression classifiers for multi-class face recognition. The method is completely automatic, as the labeled data is obtained by tagging speaking faces using subtitles and fan transcripts of the videos. We demonstrate our approach on six episodes each of two diverse TV series and achieve state-of-the-art performance.

1. Introduction

Automatic identification of characters in TV series and movies is both an important and challenging problem. Person identities are an important source of information in many higher level multimedia analysis tasks, such as semantic indexing and retrieval, interaction analysis and video summarization. Recently, multimedia content providers have started to offer information on cast and characters for TV series and movies during playback[1,2,3], presumably via a combination of face tracking, automatic identification and crowd sourcing.

In this paper, we approach the problem of naming characters in TV series as a transductive learning problem with constraints. Our goal is to automatically identify all characters by training discriminative multi-class classifiers from (i) weakly-supervised track labels, (ii) additional unlabeled data and (iii) automatically generated constraints between tracks. In contrast to other approaches, we integrate all three sources of information (i–iii) into a common learning framework. This allows us to better capture the underlying distribution of the data, resulting in a classifier which subsequently increases recognition performance.

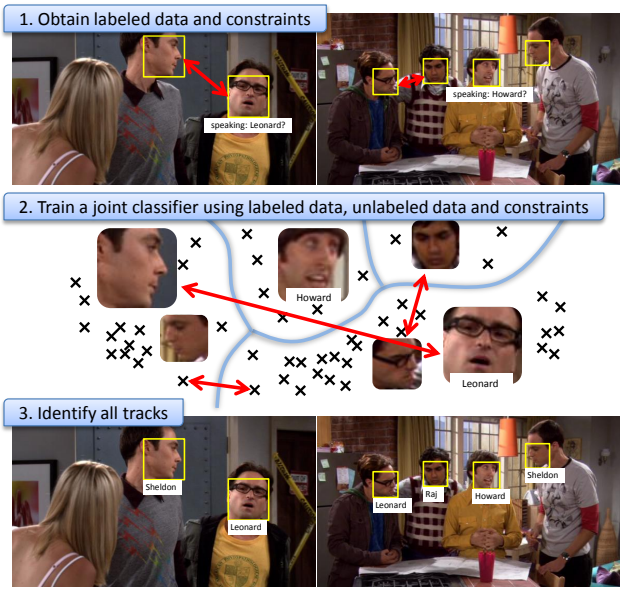


Figure 1: Overview of our approach for character naming.

We assume that we have the entire data available at training time, *i.e.*, we do not need to identify unseen data. For example, the identification can be performed offline beforehand if the goal is to display additional information on characters during the playback of a TV episode.

Our contributions in this work are the following: 1. We propose a multi-class learning framework that takes into account (weakly-)supervised data, unsupervised data and constraints in a joint formulation (Sec. 2). 2. We apply the proposed learning framework to the task of character naming in TV series (Sec. 3) and achieve state-of-the-art results. 3. We provide an extensive data set, consisting of more than 9200 face tracks from a total of 12 episodes over two TV series, together with weakly-supervised labels obtained by matching transcripts and subtitles, to further the research in the field of automatic person identification/character naming and related areas (Sec. 4).

[1] Hulu Face Match: http://www.hulu.com/labs/tagging
[2] Amazon/IMDB X-Ray for movies: http://www.imdb.com/x-ray/
[3] Actor info cards for Google Play Movies & TV

1.1. Related work

Automatic naming of characters in TV series has received increasing attention in the last years. While most work is focused on naming *face* tracks [5, 10, 13, 14], the problem has recently been extended to *person* tracks both to increase coverage and performance [15]. To avoid manual labeling of faces for training person models, Everingham *et al*. [5] proposed an automatic method to weakly label some track identities by detecting speakers, and aligning subtitles and transcripts to obtain identities. This has been adapted and further refined by others [2, 10, 14]. We use a similar method in this work to automatically obtain labels for those tracks which can be detected as speaking. Since speaker detection is a difficult problem by itself, these labels are typically noisy and incomplete (*i.e*., usually only about 20-30% of the tracks can be assigned a name). In order to increase the coverage of the weak labeling, one can treat the names from transcripts as ambiguous labels, *i.e*., assign multiple possible names to a face track when the speaking face cannot be reliably detected (*e.g*., [3, 10]). Different loss functions have been proposed to learn from such ambiguous labels [3, 10]. Köstinger *et al*. [10] further take into account unlabeled data with a cross entropy loss between the expected prior distribution of identities and the model.

Cinbis *et al*. [1] make use of must-link and cannot-link constraints in order to learn a face- and cast-specific metric in order to improve face clustering and identification. However, they rely on supervised labeling of clusters in order to perform the actual identification. In [15], we integrate uniqueness constraints in a second global optimization step. In a different scenario, Yan *et al*. [16] identify persons in a camera network and integrate must-link and cannot-link constraints in an empirical loss in their learning framework.

More generally, many approaches for semi-supervised learning have been proposed (*e.g*., [7]). However, must-link and cannot-link constraints are usually only considered for semi-supervised *clustering* problems, *i.e*., there are no class labels associated with the data, and the clustering is only guided by the constraints (*e.g*., [12]).

In this work, we bring together learning from weakly labeled data, unlabeled data and constraints in a common framework.

2. Semi-supervised learning with constraints

Let $\mathcal{X}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denote training data $\mathbf{x}_i$ with associated labels $y_i \in \mathcal{Y}$. The problem of character naming is inherently a multi-class problem, thus $|\mathcal{Y}| = K$ and, without loss of generality, we assume $\mathcal{Y} = \{1, \dots, K\}$. We further have additional unlabeled data $\mathcal{X}_u = \{\mathbf{x}_i\}_{i=1}^M$ and positive and negative constraints between data points $\mathcal{C} = \{(\mathbf{x}_{i1}, \mathbf{x}_{i2}, c_i)\}_{i=1}^L$, where $c_i \in \{-1, +1\}$, denotes a negative and positive constraint, respectively.

We are interested in learning a classifier, which maps a data point to one of the K classes

$$\mathcal{F}_\theta(\mathbf{x}) : \mathcal{X} \to \mathcal{Y} \quad , \tag{1}$$

where θ denotes the parameter set of the classifier. A common way to learn θ is to define a loss function over the training data, and then obtain the best θ by minimizing the loss:

$$\theta^* = \operatorname*{argmin}_\theta \mathcal{L}(y|\mathcal{X}_l; \theta) \quad . \tag{2}$$

Different choices of $\mathcal{F}$ yield different classifiers, and the definition of $\mathcal{L}$ determines the way in which the parameters of the classifiers θ are learned. In this paper, we propose a combined loss function that takes into account (i) labeled data $\mathcal{X}_l$, (ii) unlabeled data $\mathcal{X}_u$ and (iii) constraints $\mathcal{C}$:

$$\mathcal{L}(\mathcal{X}; \theta) = \mathcal{L}(y_l, y_c; \mathcal{X}_l, \mathcal{X}_u, \mathcal{C}, \theta) \tag{3}$$

$$= \mathcal{L}_l(y_l; \mathcal{X}_l, \theta) + \mathcal{L}_u(\mathcal{X}_u, \theta) + \mathcal{L}_c(y_c; \mathcal{C}, \theta)\,. \tag{4}$$

We will now first introduce our model for $\mathcal{F}$, and then describe the different parts of the loss function in more detail. The influence of different parts of the loss function on a toy example are visualized in Fig. 2.

2.1. Model

Multinomial logistic regression [8] (MLR) belongs to the family of log-linear models and is a classical choice for multi-class classification. One of the advantages of MLR is that it directly models probabilities of a data point belonging to class k with

$$P(y = k|\mathbf{x}; \theta) = \frac{e^{\theta_k^T \mathbf{x}}}{\sum_z e^{\theta_z^T \mathbf{x}}} \tag{5}$$

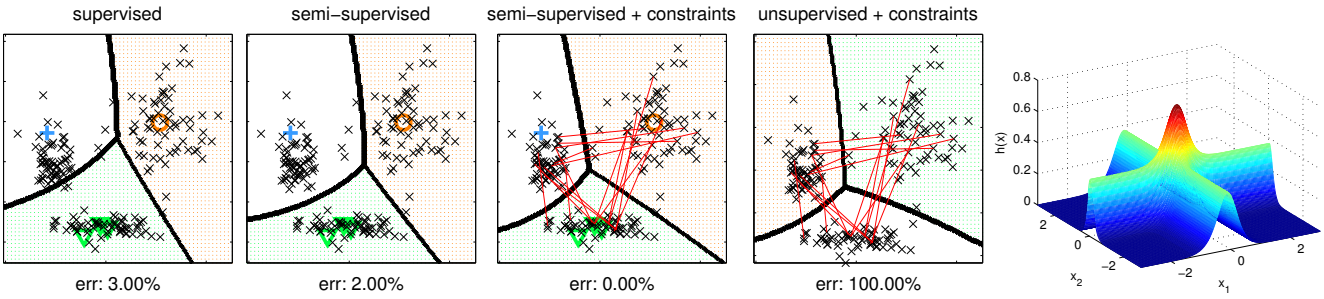
and therefore, $P(y = k|\mathbf{x}; \theta) \in [0, 1]$ and $\sum_k P(y = k|\mathbf{x}; \theta) = 1$. The model is defined by parameter vectors θ_k, one for each class. We denote $\theta = [\theta_1, \cdots, \theta_K]$ for the full parameter set. To classify a sample $\mathbf{x}$ under this model, we compute the most likely class as

$$\mathcal{F}_\theta(\mathbf{x}) = \operatorname*{argmax}_k P(y = k|\mathbf{x}; \theta) \quad . \tag{6}$$

Kernelization Multinomial logistic regression can be extended to non-linear decision boundaries by replacing $\theta_k^T \mathbf{x}$ by a function $f(\mathbf{x})$, which, according to the representer theorem [9], has the form

$$f(\mathbf{x}) = \sum_{i=1}^n \theta_{ki} K(\mathbf{x}, \mathbf{x}_i) \quad , \tag{7}$$

where $K(\cdot, \cdot)$ is a positive definite reproducing kernel.



(a) (b) (c) (d) (e)

Figure 2: Visualization of the effect of the different parts of the loss function on a toy example. The denoted error is the joint error on labeled *and* unlabeled data. (a) Learning from labeled data (colored data points +/○/▽) only. (b) Additionally taking unlabeled data (black ×) into account fits the decision boundaries better to the underlying distribution. (c) With (neg.) constraints the error on the unlabeled data reduces to 0. (d) Even without labels, it is possible to still find meaningful structure in the data using the entropy and constraint loss, however, the assignment to the classes turns out to be wrong. (e) Visualization of the entropy loss.

2.2. Supervised loss

For the sake of notational brevity, let use denote $P_\theta^k(\mathbf{x}) = P(y=k|\mathbf{x};\theta)$ in the following.

In order to learn the parameters θ of $\mathcal{F}$ from labeled training samples $\mathcal{X}_l$, we use the standard negative log-likelihood as loss

$$\mathcal{L}_l(y_l;\mathcal{X}_l,\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbf{1}[y_i{=}k]\ln(P_\theta^k(\mathbf{x}_i)) + \lambda||\theta||^2 \tag{8}$$

and $\mathbf{1}[\cdot]$ the indicator function. In order to prevent overfitting, we add a regularization term $\lambda||\theta||^2$ whose influence is controlled by λ.

For MLR, this loss is convex and can be efficiently minimized with standard gradient descent techniques. The gradient of Eq. 8 with respect to θ is

$$\frac{\partial}{\partial\theta_k}\mathcal{L}_l = 2\lambda\theta - \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\cdot\left(\mathbf{1}[y_i{=}k] - P_\theta^k(\mathbf{x}_i)\right) \quad . \tag{9}$$

2.3. Entropy loss for unlabeled data

While the unlabeled data $\mathcal{X}_u$ does not carry information about its class membership, it can be informative about the distribution of data points in regions without labels. Instead of placing decision boundaries as far as possible between *labeled* samples, we desire that the decision boundaries also respect the distribution of unlabeled data. That is, the class boundaries should preferably lie in low-density regions (see the toy example in Fig. 2 for a visual explanation).

A common way to achieve this is to include an entropy term into the loss function in order to encourage uniformly distributed class membership across the unlabeled data [10, 17]. Instead, we use the entropy function as a penalty on having the decision boundaries close to unlabeled data points (see Fig. 2 (e))

$$h(\mathbf{x}_i) = -\sum_k P_\theta^k(\mathbf{x}_i)\ln(P_\theta^k(\mathbf{x}_i)) \quad . \tag{10}$$

For the loss, we sum over all unlabeled data points

$$\begin{aligned}\mathcal{L}_u(\mathcal{X}_u;\theta) &= \frac{\mu}{M}\sum_{i=1}^{M}h(\mathbf{x}_i) \\ &= -\frac{\mu}{M}\sum_{i=1}^{M}\sum_k P_\theta^k(\mathbf{x}_i)\ln(P_\theta^k(\mathbf{x}_i)) \quad ,\end{aligned} \tag{11}$$

where μ controls the relative influence of the loss. For MLR this leads to the following gradient:

$$\frac{\partial}{\partial\theta_k}\mathcal{L}_u = -\frac{\mu}{M}\sum_{i=1}^{M}\Bigg[\mathbf{x}_i P_\theta^k(\mathbf{x}_i)\cdot \sum_{c=1}^{K}\Big(\mathbf{1}[k=c] - P_\theta^c(\mathbf{x}_i)\Big)\Big(1+\ln(P_\theta^k(\mathbf{x}_i))\Big)\Bigg] \quad . \tag{12}$$

2.4. Constraints

Finally, we include pair-wise constraints between training samples $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$. The constraint $(\mathbf{x}_{i1},\mathbf{x}_{i2},c_i)$ specifies whether $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$ belong to the same class ($c_i = 1$) or not ($c_i = -1$). Such constraints arise for example from temporal relations between face tracks, *i.e.*, two tracks which temporally overlap cannot belong to the same person, and can be automatically generated without manual effort. Note that, in general, we know the class memberships of neither $\mathbf{x}_{i1}$ nor $\mathbf{x}_{i2}$.

Intuitively, for a negative constraint the product of the likelihood of features $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$ belonging to different

classes

$$P(y_{i1} \neq y_{i2}) = \sum_{k=1}^{K} \sum_{\substack{l=1 \\ l \neq k}}^{K} P_\theta^k(\mathbf{x}_{i1}) P_\theta^l(\mathbf{x}_{i2})$$
$$= 1 - \sum_{k=1}^{K} P_\theta^k(\mathbf{x}_{i1}) P_\theta^k(\mathbf{x}_{i2}) \quad (13)$$

should be high. We therefore use the negative log-likelihood of the features belonging to different classes as loss

$$\mathcal{L}_c(c_i; \mathcal{C}, \theta) = -\frac{\gamma}{L} \sum_{i=1}^{L} \ln(P(y_{i1} \neq y_{i2}))$$
$$= -\frac{\gamma}{L} \ln \left(1 - \sum_{k=1}^{K} P_\theta^k(\mathbf{x}_{i1}) P_\theta^k(\mathbf{x}_{i2}) \right) . \quad (14)$$

Again, we need the derivative of the loss for efficient minimization. The derivative with respect to θ_k is

$$\frac{\partial}{\partial \theta_k} \mathcal{L}_c = \frac{\gamma}{L} \sum_{i=1}^{L} \Bigg[\Big(\mathbf{x}_{i1} + \mathbf{x}_{i2} \Big) P_\theta^k(\mathbf{x}_{i1}) P_\theta^k(\mathbf{x}_{i2}) -$$
$$\Big(\mathbf{x}_{i1} P_\theta^k(\mathbf{x}_{i1}) + \mathbf{x}_{i2} P_\theta^k(\mathbf{x}_{i2}) \Big) \frac{P(y_{i1} = y_{i2})}{P(y_{i1} \neq y_{i2})} \Bigg] . \quad (15)$$

3. Automatic character naming

We apply the proposed learning framework to the task of character naming in videos. We consider only *face* tracks for identification similar to [5, 10, 14], in contrast to our previous work [15] which builds on *person* tracks. However, since [15] relies on identities from face recognition as input, we can directly improve those results by providing improved facial identities. We will present some results on this aspect in the evaluation in Sec. 4.2.

3.1. Pre-processing

Face Tracking For tracking faces, we employ a detector-based face tracker based on the Modified Census Transform [6]. Our tracker is able to track faces over a wide range of pose angles (including profile faces and in-plane rotations of up to 45 degrees), which results in a large number of tracks in non-frontal poses.

Speaking-Face Detection Keeping in mind the large amount of multimedia data, we are especially interested in an identification scheme, that does not require manual supervision. Following [5, 10, 14], we align subtitles with transcripts from the web in order to combine the timing component of subtitles with the identities from the transcripts. Using the 9-point facial feature model from [5], we estimate the locations of eyes, nose and mouth in each face track. Based on the estimated mouth position, we determine for each face track whether the person is speaking or not: we follow [5, 14] and compute for each frame the minimum nearest neighbor distance of the (gray scale, histogram equalized) mouth region to the previous frame. By thresholding the distances, we determine whether a person is speaking or not.

Feature Extraction We employ a local-appearance-based method for feature extraction [4]. First, the face is aligned (warped and cropped) to a size of 48×64 pixels. The normalized face is split into 8×8 blocks, and the Discrete Cosine Transform (DCT) is computed over each block. For each block, we ignore the 0th value (average brightness) and retain the next five coefficients, thus obtain a 240 dimensional feature vector for each frame in the track.

3.2. Training

Given the face tracks, speaking information and subtitles associated with names, we obtain three different types of data from the given videos.

Weakly-labeled data When a subtitle (associated with a name from the transcripts) coincides with a "speaking" face track, we label that track with the given identity and take the corresponding features as supervised samples $\mathcal{X}_l = \{(\mathbf{x}_i, y_i)\}$, where y_i corresponds to the identity of the speaker label. Given that both facial feature detection and speaking-face detection are noisy, we do not expect perfectly clean labels from this method. Tbl. 1 shows the precision and recall (in terms of all tracks) of our speaker detection method for all episodes. "#speaking tracks" denotes the number of tracks which were determined as speaking, which is usually less than 30% of the tracks (not all characters speak at the same time). On average, we associate a name to about 22% of the tracks with a precision of 87%, which is similar to the reported performances of [5, 10, 14]. While in [10] the problem of noisy labels is explicitly targeted, the regularization of the parameter vector θ (Eq. 8) penalizes overly complex decision boundaries and prevents overfitting on noisy labels.

Unlabeled data With only 22% of the face tracks labeled by the previous method, we are left with around 78% of the data that has no labels associated with it. We take all features of the unlabeled tracks as $\mathcal{X}_u$.

Constraints We can automatically deduce constraints between data points from face tracks. Negative constraints are formed when two tracks overlap temporally, based on the assumption that the same person cannot appear twice at the same time. This is similar to the uniqueness constraint as used in the model by [15], however, we already employ it at training time. This poses a problem if there actually are two tracks of the same (or very similar looking) person at

	BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6
# characters	6	5	7	8	6	6	12	13	14	15	15	18
# face tracks	622	565	613	581	558	820	764	963	1081	835	786	1084
# unknown tracks	8	2	87	41	82	195	11	126	10	38	94	63
# speaking tracks	206	153	170	163	120	174	178	244	214	227	211	216
spk precision	83.98	91.5	92.35	88.96	90.83	82.76	87.08	85.25	82.24	87.67	89.57	89.35
spk recall	27.81	24.78	25.61	24.96	19.53	17.56	20.29	21.6	16.28	23.83	24.05	17.8

Table 1: Statistics across all videos in the data set showing the number of characters, face tracks and speaker assignment performance.

the same time. Our evaluation data is especially insidious in that sense since there are two *Xanders* in episode 5-03 of *Buffy* (BF-3) (played by identical twins), which actually often appear together in the same shot.

3.3. Training

We first collect training data from all available episodes, and train one joint multi-class classifier from supervised data, unsupervised data and constraints by minimization of the joint loss function (Eq. 4) via L-BFGS [11].

Taking into account *all* available training data from multiple episodes at the same time is unfortunately computationally infeasible, especially for the kernelized version of the multinomial logistic regression. We therefore reduce the data by subsampling, effectively removing features that were temporally nearby and therefore presumably visually similar. For the kernel computation we further randomly select prototypes instead of working with the full kernel matrix. This technically turns the originally transductive learning problem (with all data available at training time) into a semi-supervised learning problem, albeit solely for computational reasons. Scaling the learning so that all available training data can be actually used remains for future research.

3.4. Identification

For determining the identity y_t of a face track t with features $\{\mathbf{x}_i^{(t)}\}_{i=1}^{|t|}$ we apply the learned classifier framewise according to Eq. 6 and compute a class score for the track having identity k as

$$p_t(k) = \frac{1}{|t|}\sum_{i=1}^{|t|} P(y{=}\,k|\mathbf{x}_i^{(t)}) = \frac{1}{|t|}\sum_{i=1}^{|t|} \frac{e^{\theta_k^T \mathbf{x}_i^{(t)}}}{\sum_z e^{\theta_z^T \mathbf{x}_i^{(t)}}} \,. \tag{16}$$

The track is assigned the identity of the most likely class over all frames

$$y_t = \operatorname*{argmax}_k p_t(k) \quad . \tag{17}$$

Although the outputs of the classifier are in the range $[0, 1]$ and could be interpreted as probabilities, we take the sum instead of the product over all frames, which we found to be more robust to outliers in practice.

Assignment to "unknown" Usually some unknowns have small speaking roles, and therefore we can automatically collect some training samples for them. We model unknown characters as one *joint* class in the model, *i.e.*, training data from all unknowns are used as positives for this class. Thus, no special handling for the unknown class is required: a new track is assigned the "unknown" identity, when it is the most likely class according to Eq. 17.

4. Evaluation

4.1. Data set and experimental setup

Our data set[4] consists of 12 full episodes from two TV series. We select episodes 1–6 from season 1 of *The Big Bang Theory (BBT-1 to BBT-6)* (as used in [15]), and episodes 1–6 from season 5 of *Buffy the Vampire Slayer (BF-1 to BF-6)* (as used in [5, 10, 14]). The two series are quite different in their filming style, and therefore also pose different challenges. *The Big Bang Theory* is a sitcom ($\sim$20 minutes per episode) with a main cast of 5-8 people and mostly takes place indoors. It includes many full-view shots which contain multiple people at a time, however the faces are rather small (the average face size is around $75px$). On the other hand, *Buffy* has an average length of $\sim$40 minutes per episode, with a main cast size around 12, while in specific episodes there are up to 18 important characters. Many shots are set outside and at night, resulting in a large range of different lighting conditions. However, it also contains a sizable number of face close-up shots (the average face size is around $116px$).

For an overview on the data set see Table 1. Buffy episodes contain on average less than double the amount of face tracks compared to BBT due to the above mentioned higher number of close-up shots in Buffy. Speaking-face detection and naming performs equally well on both series, with on average around 22% recall (of all face tracks) and around 87% precision.

Table 2 shows the number of face tracks for each identity accumulated over the six episodes of BBT. The precision and recall of the speaking-face naming from subtitles and transcripts reveal that there is a large variation in available training data across the main cast of *Leonard, Sheldon, Penny, Howard* and *Raj*.

[4] Available at http://cvhci.anthropomatik.kit.edu/projects/mma

	#FaceTr	#Speak	Spk-Prec	Spk-Rec
Leonard	1070	281	91.46	24.02
Sheldon	945	323	90.09	30.79
Penny	512	178	87.08	30.27
Howard	299	78	85.90	22.41
Raj	279	43	69.77	10.75
Mary	95	39	100.00	41.05
Leslie	84	9	88.89	9.52
Kurt	32	8	87.50	21.88
Gabelhauser	16	3	100.00	18.75
Doug	8	0	–	–
Summer	4	0	–	–

Table 2: The cast list of BBT, the face tracks across all episodes, and the performance of tagging speaking face tracks automatically.

Guest appearances usually play an important role in the story of an episode, and identifying them correctly is important for applications such as video summarization and multimedia understanding. Thus, we intend to identify all people whose name is mentioned on screen at least once, and label the ground truth accordingly. For example, in BBT there are four minor *named* characters with less than 35 tracks. In BF-3 there is a double of the main character "Xander" (played by his twin brother), and we label him as such "Xander2", since the role he plays and the distinction between the two is important to the story of the episode. All remaining characters (*e.g.*, somebody in the background, extras) are labeled as "unknown".

Performance metric We evaluate our approach in terms of identification accuracy in an *open-set* id context. We require all characters to be identified correctly, even when the automatic speaker assignment does not provide any training data for them. Unknowns should be identified as "unknown". Both assigning a name to an "unknown", and assigning "unknown" to a named character is counted as an error.

4.2. Experiments

We perform a series of experiments in order to compare our approach with other approaches and present results in multiple steps of improvement. Table 3 shows the main recognition results and we will refer back to it in the following.

Baseline results In order to establish a baseline, and also compare with previous approaches, we use the automatically generated weak face labels data to train different supervised-only classifiers.

As the simplest method, we perform Nearest Neighbor (NN) classification comparable to [5]. It achieves an accuracy of 64.2% on BBT and 56.5% on Buffy.

Further, we train Logistic Regression (LR) and Support Vector Machine (SVM) classifiers in a one-vs-all scheme.

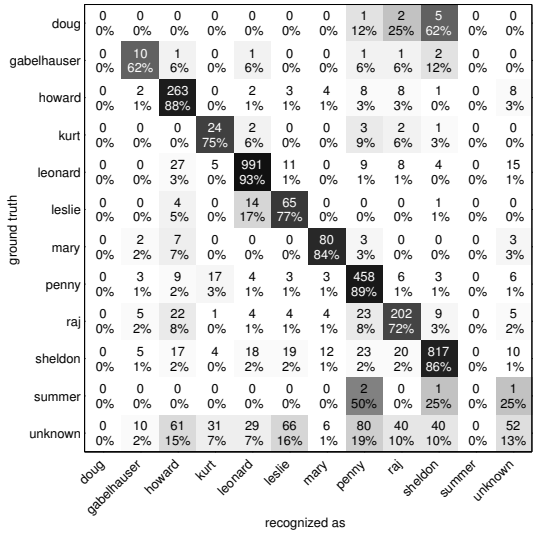


Figure 3: Confusion matrix over all 6 episodes of BBT for MLR + $\mathcal{L}_u$ + $\mathcal{L}_c$. For *Doug* and *Summer*, the automatic labeling did not find any tracks for training (*c.f.* Tbl. 2).

For both, we use a polynomial kernel of degree 2, corresponding to the setting in [15]. LR and SVM perform roughly on par. Note that in [15], where also SVMs are used, face labels were manually supplied, whereas we obtain them automatically from the transcripts. When using our SVM results as input to [15] ("SVM+MRF" in Table 3), we obtain a significant improvement to about 82% accuracy in face recognition. Since we do not have person tracks for Buffy, we perform this evaluation only for BBT.

SS+Constraints MLR We evaluate our method starting with the supervised loss only and then add the other loss terms for incremental improvement. The MLR multi-class classifier already outperforms both LR and SVM for both series (77.4% for BBT and 65.82% for Buffy). By adding additional (unlabeled) data and constraints, we can further increase the identification accuracy. With the full loss term, we reach on average about 79.5% accuracy for BBT (almost 90% on episode 1) and 66.37% on Buffy. In addition, we perform 10 runs on 90% of the data (leave out 1 of 10 folds each) and perform a paired t-test against the baseline (SVM), in which we are able to reject the null-hypothesis of equal means ($p < 0.01$). The big drop in accuracy in BBT-6 can be explained by the large number of unknowns present in that episode (195 tracks, see Table 1), which are harder to identify because there is usually no training data for them. Also, speaking-face precision and recall are significantly lower for BBT-6, which is in parts also caused by unknowns which are incorrectly assumed to be speaking. Figure 3 shows the confusion matrix over all 6 episodes of BBT, which confirms the difficulty in identifying unknowns.

Curiously, while adding constraints helps more for BBT,

	BBT-1	BBT-2	BBT-3	BBT-4	BBT-5	BBT-6	**BBT Avg.**	BF-1	BF-2	BF-3	BF-4	BF-5	BF-6	**BF Avg.**
Max Prior	37.94	33.98	34.42	17.56	24.19	23.66	28.63	29.97	19.31	18.69	25.75	35.24	14.58	23.92
baseline: NN [5]	72.19	71.86	66.88	59.04	59.50	55.98	64.24	60.34	51.92	55.13	58.92	61.96	50.74	56.50
baseline: one-vs-all LR	88.42	84.60	73.57	73.84	70.97	65.73	76.19	69.50	59.29	66.05	65.87	67.56	60.33	64.77
baseline: one-vs-all SVM [15]	87.46	84.96	74.06	74.87	70.25	66.46	**76.34**	69.90	59.71	66.23	66.47	68.07	61.44	**65.30**
baseline: one-vs-all SVM + MRF [15]	94.05	92.21	76.18	79.00	75.63	74.51	**81.93**	–	–	–	–	–	–	–
ours: MLR	88.59	87.61	76.18	74.01	72.76	65.24	77.40	68.85	61.37	65.96	67.19	69.85	61.72	65.82
ours: MLR + $\mathcal{L}_u$	88.59	87.61	76.35	74.01	72.94	65.24	77.46	71.60	60.54	66.42	67.78	70.10	61.44	66.31
ours: MLR + $\mathcal{L}_u$ + $\mathcal{L}_c$	89.23	89.20	78.47	76.59	75.09	68.05	**79.44**	71.99	61.27	66.60	67.07	69.59	61.72	**66.37**
ours: MLR + $\mathcal{L}_u$ + $\mathcal{L}_c$ + MRF [15]	95.18	94.16	77.81	79.35	79.93	75.85	**83.71**	–	–	–	–	–	–	–

Table 3: Evaluation results. The first line shows the accuracy that could be achieved by assigning each track the most often appearing person in the series (*Leonard* for BBT, and *Buffy* for Buffy). In the middle section of the table, we report baseline results of different methods on our data set. The bottom section shows the performance of our approach in multiple steps of improvement. MLR denotes the basic supervised multinomial logistic regression classifier, and $\mathcal{L}_u$ and $\mathcal{L}_c$ denote the additionally incorporated loss terms.

for Buffy adding unlabeled data helps. The importance of constraints in BBT can be explained from the fact that BBT contains many shots with multiple faces, thus allowing constraints such as uniqueness to be useful. On the other hand, Buffy favors close-up face shots, which also results in much fewer and less diverse constraints. The lack of influence of unlabeled data in BBT can be explained by the relatively small cast compared to Buffy, while at the same time having many training samples for each of the main characters.

Finally, if we use the face identification results from our best-performing method as input to the clothing-based MRF model of [15], we can further increase the performance to 83.71% and thus achieve the best results on the BBT data set.

Failure analysis We already identified the naming of unknowns as one of the error sources (see also Fig. 4 (a)). A refusal-to-predict scheme, as used for example in [5, 10], could help to reduce the number of falsely accepted/named unknowns.

Second, our employed DCT features are – despite the pre-processing alignment to a normalized pose – far from pose-invariant. We analyze the identification accuracy depending on the mean pan-angle of the face tracks (see Fig. 4 (b)). The performance drops significantly for greater pan angles to about 50% rank-1 performance for $|pan| > 75$ for BBT. However, at rank 3, we consistently reach around 80% for all pan angles. Pose independent face recognition has been an active research area for many years, and a more robust feature should directly have an impact on our recognition performance.

Curiously, there is a drop in performance for frontal faces. This can be explained by a similar drop in speaker assignment recall for those pose angles (see Fig. 4 (c)), which again can be explained from wide angle shots where usually multiple persons in near-frontal poses are present, but just one is speaking.

Similar to the dependency on the average pan angle, there is a dependency on track length and average face size. We observe that the identification accuracy decreases for shorter tracks or tracks with a small average face size.

Finally, for some minor characters we are unable to find any speaking tracks, *e.g.* for *Doug* and *Summer*, see Tbl. 2. Therefore, we are unable to correctly identify any face track that belongs to these characters (see Fig. 3). However, these only represent a very small portion of the data set (about 0.4%).

5. Conclusion

In this paper, we address the problem of person identification in multimedia data. We propose to use a unified learning framework combining both labeled and unlabeled data, along with their constraints in a principled manner, and apply it to train multinomial logistic regression classifiers. We also set our goal to identify all the people named in the video, thus providing full coverage even for guest appearances. The methods are tested on six episodes each of two TV series – The Big Bang Theory and Buffy the Vampire Slayer – and we obtain state-of-the-art results for person identification.

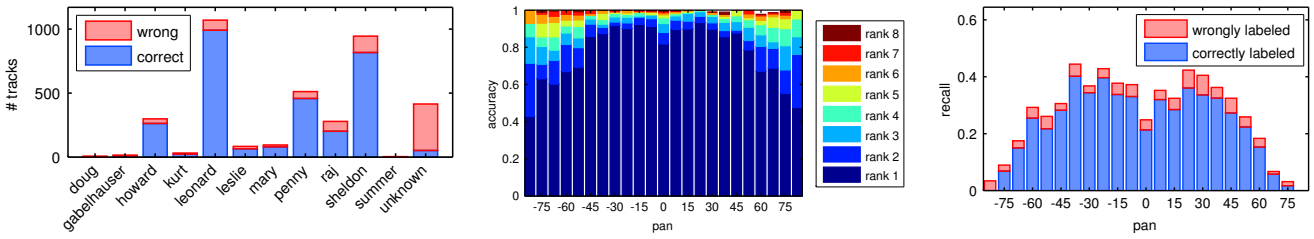
Acknowledgments This work was realized as part of the Quaero Program, funded by OSEO, French State agency for innovation, and was partially funded by the German Federal Ministry of Education and Research (BMBF) under contract no. 01ISO9052E and 13N12063. The views expressed herein are the authors' responsibility and do not necessarily reflect those of OSEO or BMBF.

References

[1] R. Cinbis, J. Verbeek, and C. Schmid. Unsupervised metric learning for face identification in TV video. In *ICCV*, 2011. 2

[2] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *ECCV*, 2008. 2

[3] T. Cour, B. Sapp, and B. Taskar. Learning from Partial Labels. *JMLR*, 12(5):1225–1261, 2011. 2



(a) (b) (c)

Figure 4: Analysis of recognition accuracy over all episodes of BBT (best viewed in color): (a) by identity: "unknowns" are the most often wrongly classified class; (b) by pan angle: tracks with (near-)frontal poses are identified with higher accuracy than tracks in side-views (results are displayed for multiple ranks); (c) speaker assignment recall by pan angle: for frontal poses (around 0°), there is a significantly lower recall of speaker-assigned tracks than for adjacent pan angles.

Figure 5: Sample output results from our system for both TV series (top: BBT, bottom: Buffy). The label "speaking: <name>?" denotes the labeled name from the transcripts, while the Face ID label shows the output of the classifier. One can see, that our system not only correctly identifies unlabeled characters, but also is able to correct wrong speaking labels. We also show an example of a failure case, where the assigned speaker label was correct (*Dawn*) and incorrectly changed to *Harmony*.

[4] H. Ekenel and R. Stiefelhagen. Analysis of Local Appearance-Based Face Recognition: Effects of Feature Selection and Feature Normalization. In *CVPR Biometrics Workshop*, 2006. 4

[5] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – Automatic Naming of Characters in TV Video. In *BMVC*, 2006. 2, 4, 5, 6, 7

[6] B. Fröba and A. Ernst. Face detection with the modified census transform. In *FG*, 2004. 4

[7] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2005. 2

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2009. 2

[9] G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 1971. 2

[10] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof. Learning to Recognize Faces from Videos and Weakly Related Information Cues. In *AVSS*, 2011. 2, 3, 4, 5, 7

[11] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989. 5

[12] Z. Lu and T. Leen. Semi-supervised clustering with pairwise constraints: A discriminative approach. In *International Conference on Artificial Neural Networks*, 2007. 2

[13] D. Ramanan, S. Baker, and S. Kakade. Leveraging archival video for building face datasets. In *ICCV*, 2007. 2

[14] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" – Learning person specific classifiers from video. In *CVPR*, 2009. 2, 4, 5

[15] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. "Knock! Knock! Who is it?" Probabilistic Person Identification in TV-Series. In *CVPR*, 2012. 2, 4, 5, 6, 7

[16] R. Yan, J. Zhang, J. Yang, and A. G. Hauptmann. A discriminative learning framework with pairwise constraints for video object classification. *PAMI*, 28(4):578–93, 2006. 2

[17] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof. Online Semi-supervised Multiple-Instance Boosting. In *CVPR*, 2010. 3