Automatic Identity Inference for Smart TVs

Avneesh Saluja and **Frank Mokaya**
Carnegie Mellon University
Moffett Field, CA

Mariano Phielipp
Digital Home Group
Intel Corporation
Chandler, AZ

Branislav Kveton
Technicolor
Palo Alto, CA

Abstract

In 2009, an average American spent 3 hours per day watching TV. Recent advances in TV entertainment technologies, such as on-demand content, browsing the Internet, and 3D displays, have changed the traditional role of the TV and turned it into the center of home entertainment. Most of these technologies are personal and would benefit from seamless identification of who sits in front of the TV. In this work, we propose a practical and highly accurate solution to this problem. This solution uses a camera, which is mounted on a TV, to recognize faces of people in front of the TV. To make the approach practical, we employ online learning on graphs and show that we can learn highly accurate face models in difficult circumstances from as little as one labeled example. To evaluate our solutions, we collected a 10-hour long dataset of 8 people who watch TV. Our precision and recall are in the upper nineties, and show the promise of utilizing our approach in an embedded setting.

1 Introduction

As our interaction with the TV evolves by adopting new technologies, a non-intrusive way of personalizing experiences is still lacking. State-of-the-art solutions to the problem, such as logging into an entertainment profile or creating an avatar, are awkward and people refrain from using them. Ideally, the TV should learn to recognize people seamlessly, based on little or no feedback. Until recently, this was impossible because TVs were not equipped with any sensors. However, this trend has been changing rapidly. For instance, Sony Bravia LX900 has an embedded camera, detects faces, and can adjust its volume based on the distance of people.

In this paper, we put a camera on a TV and use it to recognize people in front of the TV. To make our solution seamless, we mostly rely on unlabeled faces and build a data adjacency graph over these faces to discover the underlying structure of data (Kveton et al. 2010). The adjacency graph is updated in real time using online learning on quantized graphs (Valko et al. 2010). This approach is suitable for our problem because the environment in front of the TV often changes slowly. As a result, it is possible to learn some of these changes. Because the environment is not completely stationary, online learning has a good chance of outperforming offline-trained classifiers.

This paper is not the first attempt to automatically learn the patterns of TV viewers. For instance, both Chang *et al.* (2009) and Phielipp *et al.* (2010) tried to recognize TV viewers based on how they hold the TV remote control and press its buttons. In comparison to these methods, our approach has three main advantages. First, it can learn from as little as a single labeled example. The other two methods require extensive supervised training. Second, our method is non-parametric and can adapt to variable data, such as changes in light or facial expressions. Finally, we utilize a more informative sensor. Thanks to more informative data, both the precision and recall of our solutions reach the upper nineties.

This is the first application of online semi-supervised learning to a real-world problem that is compelling, has a value proposition, and involves both noisy data and imperfect sensors, such as a low-resolution camera on a TV. All previous empirical studies are performed on either heavily-scripted visual data (Grabner, Leistner, and Bischof 2008), synthetic datasets (Goldberg, Li, and Zhu 2008), or both (Valko et al. 2010). Our experiments are performed on realistic data, which were collected using a TV mounted-camera when people watched the TV. Therefore, successful inference on our datasets is a pertinent indicator of successful inference in an actual deployed system.

The paper has the following structure. In Section 2, we relate our paper to the existing work on online semi-supervised learning, face recognition, and identity inference. In Section 3, we introduce the fundamentals of our approach, such as semi-supervised learning on graphs and online manifold tracking. In Section 4, we describe our dataset, experimental setup, and metrics for evaluating our solutions. Results of our experiments are presented in Section 5 and our future research directions are outlined in Section 6.

The following notation is used in the paper. The symbols $\mathbf{x}_i$ and y_i refer to the i-th example (face) and its label, respectively. The examples $\mathbf{x}_i$ are divided into labeled and unlabeled sets, l and u, and labels $y_i \in \{-1, 1\}$ are given for the labeled data only.[1] The total number of training examples is $n = n_l + n_u$, where $n_l = |l|$, $n_u = |u|$, and $n_u >> n_l$.

2 Related Work

In this section, we compare our paper to the existing work on online semi-supervised learning, face recognition, and identity inference in consumer devices.

[1]Our ideas straightforwardly generalize to multi-class classification (Balcan et al. 2005).

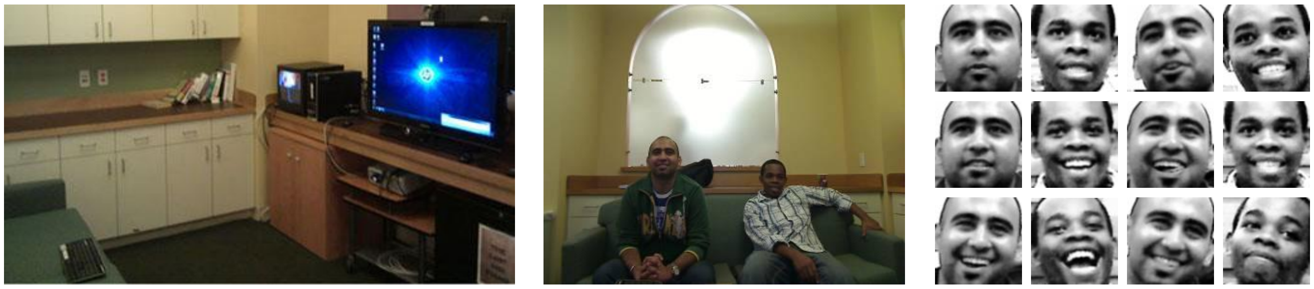
(a) (b) (c)

Figure 1: **a.** Our data collection setup. **b.** An example of an image observed by a TV-mounted camera. **c.** Examples of extracted faces.

2.1 Online Semi-Supervised Learning

Both the computer vision and machine learning communities have recently shown a great interest in online semi-supervised learning. Online manifold regularization of SVMs (Goldberg, Li, and Zhu 2008) and online semi-supervised boosting (Grabner, Leistner, and Bischof 2008) are two examples of recently proposed algorithms. Online manifold regularization of SVMs learns a max-margin classifier, which is regularized by the similarity graph. Online semi-supervised boosting is a form of boosting, where unlabeled data are labeled greedily using the similarity graph and then used in the standard boosting fashion.

The major difference in our work is that we push the limit of online semi-supervised learning in terms of solving a real-world problem. Our problem is compelling, naturally online, and involves both noisy inputs (such as the faces of not necessarily cooperative people) and an imperfect sensor, the TV-mounted camera (Figure 1).

2.2 Face Recognition

Face recognition has been studied in-depth by the computer vision community (Zhao et al. 2003). Most of this research is focused on finding better features. Face recognition in videos is commonly done by combining per-frame predictors with a temporal model of the environment (Lee et al. 2003).

In comparison to the model-based methods, we use neither sophisticated features nor temporal models. Our representation of faces is a similarity graph, which is updated over time by tracking the manifold of data in videos. The advantage of our approach is that it learns from very little human feedback. However, since we do not model the environment, we need to be cautious when extrapolating to new data. We address this issue in Section 3. Finally, we note that our similarity graph (Figure 2) can be constructed using more complex features than those in Section 3.1. Hence, our online solution is essentially complimentary to finding a better set of face recognition features.

In the machine learning community, Balcan *et al.* (2005) applied the harmonic function solution to face recognition. In comparison to our work, their solution was completely offline.

2.3 Identity Inference in Consumer Devices

Another approach to identity inference for TV viewers relies on identifying users based on how they hold a remote control (Chang, Hightower, and Kveton 2009). In contrast to this work, which required extensive training (90% of the data was labeled) and achieved average accuracy levels of 80% in a purely offline setting (and would likely perform worse in an online one), we achieve 95% recall and precision on average (accuracy greater than 90%) across all datasets using just a single labeled example (Section 5). Thus, in addition to achieving better accuracy, our solution minimizes the amount of (explicit) feedback/interactions with the user. As a result, we believe that identity inference for other consumer electronics applications could benefit significantly from our approach.

3 Approach

In this section, we begin by discussing the construction of the similarity graph or matrix for our application. We then review the harmonic function solution, a standard approach to semi-supervised learning on a similarity graph. Next, motivated by the real-time requirements of our application, we look at online learning solutions and present the algorithm that was used to achieve robust real-time identity inference in our experiments.

3.1 Similarity Graph

The *similarity* of faces $\mathbf{x}_i$ and $\mathbf{x}_j$ is computed as:

$$w_{ij} = \exp\left[-d^2(\mathbf{x}_i, \mathbf{x}_j)/(2\sigma^2)\right], \qquad (1)$$

where σ is a tunable heat parameter and $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between the faces in the feature space. The *distance* is defined as $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_{1,\psi}$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are pixel intensities in 96×96 face images, and $\|\cdot\|_{1,\psi}$ is an $\mathcal{L}_1$-norm that assigns higher weights to pixels in the centers of the images. Note that our distance function $d(\cdot, \cdot)$ is fairly simple. In general, this is not an issue because we use manifold tracking to discover additional structure in data.

The heat parameter is set as $\sigma = 0.020$. We empirically select the value of σ so that the faces of different people rarely have a high similarity to each other. For instance, using this setting, the similarity of any two different faces from the SZSL subset of the MPLab GENKI database (MPLab 2009) is at most 10^{-6}. Note that the σ parameter was chosen based on the GENKI database and not the data we obtained (Section 4.1).

The distance function $d(\cdot, \cdot)$ induces a fully connected similarity graph. To prevent arbitrarily small similarities, we turn the similarity graph into an ε-neighborhood graph by deleting edges whenever $w_{ij} < \varepsilon$. In the rest of the paper, we refer to ε as the *minimum edge* and use it to control label propagation on the graph. The higher the value of ε, the lower the number of edges in the graph and the less likely it is that two vertices in the graph can be connected by a path. In Section 5, we study the performance of our solution as a function of ε. Figure 2 shows an example of a similarity graph for $\varepsilon = 10^{-6}$.

At any point in time, the similarity graph is represented by an $n \times n$ matrix W, where n is the number of examples seen thus far. It is helpful to view the similarity matrix as a kernel $k(\mathbf{x}_i, \mathbf{x}_j) = w_{ij}$, as this allows for a fair comparison of semi-supervised learning on graphs to unsupervised and supervised learning.

3.2 Semi-Supervised Learning on Graphs

A standard approach to offline semi-supervised learning is to minimize the quadratic objective function (Zhu, Ghahramani, and Lafferty 2003):

$$\min_{\boldsymbol{\ell} \in \mathbb{R}^n} \boldsymbol{\ell}^\mathsf{T} L \boldsymbol{\ell} \quad \text{s.t. } \ell_i = y_i \text{ for all } i \in l; \tag{2}$$

where $\boldsymbol{\ell}$ is the vector of predictions on both labeled and unlabeled examples, $L = D - W$ is the Laplacian of the similarity graph, which is given by a matrix W of pairwise similarities w_{ij}, and D is a diagonal matrix defined as $d_i = \sum_j w_{ij}$. This problem has a closed-form solution:

$$\boldsymbol{\ell}_u = (L_{uu})^{-1} W_{ul} \boldsymbol{\ell}_l, \tag{3}$$

which is known as the *harmonic function solution (HFS)* because it satisfied the *harmonic property* $\ell_i = \frac{1}{d_i} \sum_{j \sim i} w_{ij} \ell_j$. Due to the random walk interpretation of the solution (Zhu, Ghahramani, and Lafferty 2003), $|\ell_i|$ can be viewed as a *confidence* of assigning the label $\operatorname{sgn}(\ell_i)$ to the vertex i. The confidence $|\ell_i|$ is always between 0 and 1. The closer the value to 0, the more *uncertain* the label $\operatorname{sgn}(\ell_i)$.

The control extrapolation to unlabeled data, we regularize the Laplacian as $L + \gamma_g I$, where γ_g is a non-negative regularization parameter and I denotes the identity matrix. Similarly to the problem (2), the harmonic function solution on $L + \gamma_g I$ can be computed in a closed form:

$$\boldsymbol{\ell}_u = (L_{uu} + \gamma_g I)^{-1} W_{ul} \boldsymbol{\ell}_l. \tag{4}$$

It can also be viewed as a random walk on the graph W with an extra sink. At each step, the probability that the walk ends up in the sink is $\gamma_g / (d_i + \gamma_g)$. Therefore, the confidence $|\ell_i|$ of labeling unlabeled vertices drops with the number of hops from labeled vertices. Moreover, note that $|\ell_i|$ decreases as γ_g increases. Finally, if no labeled example can be reached from the vertex i, note that the confidence $|\ell_i|$ must be 0.

To make the predictor (4) robust to outliers, we refrain from predicting when $|\ell_i|$ drops below a small value, such as 10^{-6}. To simplify the parameterization of the predictor, we link the parameters ε and γ_g as $\gamma_g = 10\varepsilon$. This setting can be viewed as follows. First, the higher the connectivity of the graph, the lower the penalty γ_g for extrapolation. Second, the chance of jumping along the minimum edge in the graph is around 0.1 and rather small. As a result, our predictor does not extrapolate too far along weak edges.

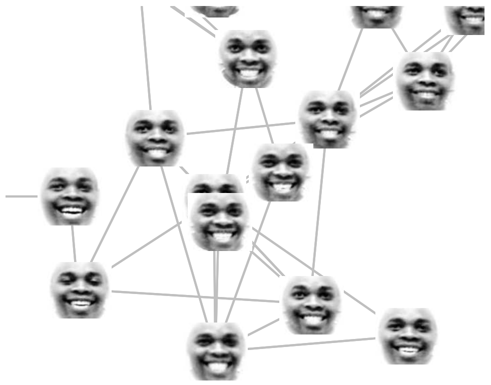
Figure 2: A similarity graph over faces. The minimum edge ε is set to 10^{-6}.

3.3 Online Learning on Graphs

Imagine setting up your new TV. You register a single image of your face, and the TV automatically learns over time to recognize you with high precision and recall. This learning problem can be solved by online semi-supervised learning on a graph. In particular, when you sit in front of the TV, it observes a stream of unlabeled faces and uses their similarity graph to improve its predictor.

How to solve this problem efficiently using offline learning (Section 3.2) is not clear. A trivial way is to update the complete similarity graph at each time step with the new example $\mathbf{x}_t$ and then infer its label. The time complexity of adding $\mathbf{x}_t$ and inferring its labels is $\theta(t)$ and $\Omega(t^2)$, respectively. Therefore, the solution becomes soon impractical and cannot run in real time (Section 5.1).

3.4 Online Learning on Quantized Graphs

One approach to avoiding the dependence on t is to maintain a compact representation of the graph, for instance by approximating it using n_g representative vertices with radius R (Valko et al. 2010). The *doubling algorithm* of Charikar *et al.* (1997) can be used to update the representative vertices incrementally and online. The algorithm works as follows. At each time step, it takes an example $\mathbf{x}_t$, and either merges it with an existing representative vertex or creates a new one. When the number of vertices is n_g+1, the vertices are greedily repartitioned into less than n_g vertices. While doing this, we maintain 2 invariants. First, all vertices are at least R away from each other. Second, any example $\mathbf{x}_t$ is at most $2R$ away from its representative vertex. These invariants are important because they allow for proving bounds on the quality of our approximations. Pseudocode for computing the HFS on quantized graphs is given in Figure 3. Note that the complexity of computing the HFS is $O(n_g^3)$ and hence is independent of t.

The value of n_g is largely problem dependent. In our case, n_g should grow with the number of people that are usually seen by the TV. In the experimental section, we set n_g to 1,000. Based on our experiments, this is sufficient to cover the manifold of faces for two people, which is a common case in our dataset. Finally, note that the computational cost of the HFS is $\Omega(n_g^2)$ and that this is the bottleneck of our algorithm. A practical solution is to replace the exact HFS

```
Inputs:
  an unlabeled example $\mathbf{x}_t$
  a set of representative vertices $C_{t-1}$
  vertex multiplicities $\mathbf{v}_{t-1}$

Algorithm:
  if ($|C_{t-1}| = n_g + 1$)
    $R = 2R$
    greedily repartition $C_{t-1}$ into $C_t$ such that:
      no two vertices in $C_t$ are closer than $R$
      for any $\mathbf{c}_i \in C_{t-1}$ exists $\mathbf{c}_j \in C_t$ such that $d(\mathbf{c}_i, \mathbf{c}_j) < R$
    update $\mathbf{v}_t$ to reflect the new partitioning
  else
    $C_t = C_{t-1}$
    $\mathbf{v}_t = \mathbf{v}_{t-1}$
  find the closest vertex to $\mathbf{x}_t$ in $\mathbf{c}_k \in C_t$
  if ($d(\mathbf{x}_t, \mathbf{c}_k) < 2R$)
    $\mathbf{v}_t(k) = \mathbf{v}_t(k) + 1$
  else
    $\mathbf{v}_t(|C_t| + 1) = 1$
    add $\mathbf{x}_t$ to the position ($|C_t| + 1$) in $C_t$
  build a similarity matrix $\tilde{W}_t$ over the vertices $C_t \cup l$
  build a matrix $V_t$ whose diagonal elements are $\mathbf{v}_t$
  $W_t = V_t \tilde{W}_t V_t$
  compute the Laplacian $L$ of the graph $W_t$
  infer labels on the graph:
    $(\boldsymbol{\ell}_t)_u = (L_{uu} + \gamma_g (V_t)_{uu})^{-1} (W_t)_{ul} (\boldsymbol{\ell}_t)_l$
  find the closest vertex to $\mathbf{x}_t$ in $\mathbf{c}_k \in C_t \cup l$
  make a prediction $\hat{y}_t = \operatorname{sgn}((\boldsymbol{\ell}_t)_k)$

Outputs:
  a prediction $\hat{y}_t$
  a set of representative vertices $C_t$
  vertex multiplicities $\mathbf{v}_t$
```

Figure 3: Harmonic function solution on a quantized graph at time t. The main parameter of the algorithm is the maximum number of representative vertices n_g.

solver with a conjugate gradient optimizer on a sparse representation of W.

4 Experimental Setup

We present the nature of the data and give a brief overview of our data collection methods. We also outline the metrics we used to evaluate the performance of our algorithms.

4.1 Data Collection

We analyze the performance of our algorithm using real-world data. The data was collected by an external camera attached to a TV that captured viewers in a natural state while watching TV content. The camera and TV were linked to a desktop computer which functioned as the storage location for the captured images as well as a processing hub for our identity inference algorithm.

The data was collected in a semi-secluded indoor environment (Figure 1). Video frames are captured at a rate of 3 frames per second (fps) at a 1600×1200 resolution. The fps rate was optimized to maximize the amount of camera data we captured while ensuring that TV content playback was smooth, as the same processor was used for playing content and face data collection.

From each frame, we detect and extract faces using OpenCV, while also maintaining a mapping between the data points (faces) and the frame that they are extracted from. Since multiple people might have been watching television at the time of experiment, multiple faces can also be extracted. The images are turned into grayscale, resized to 96×96 pixels and smoothed using a Gaussian kernel. Finally the image histograms of pixel intensities are also equalized. When labeling faces, we choose a single, symmetric, face-centered image of the subject.

A subset of our data, a part of which is showed in Figure 1c, is chosen as an illustrative dataset D_1. The dataset consists of 1,435 faces of one person and 499 faces of another person in the dataset. One of the 1,435 images is labeled and used to predict the labels for the other 1434. The other 499 images are rightfully predicted as outliers. In short, Dataset D_1 is just 10 minutes long; small when compared to the rest of our data, yet large enough to illustrate our computational trends. We use this dataset to compare our method to the online and offline variants of the HFS.

Dataset D_2 is 10 hours long, was collected over a period of one month and involves all data in which one or more people watched the TV. The dataset consists of 6 TV watching sessions, and we considered each session a subset. Each subset consists of up to two people watching TV, with an occasional outlier, which happens when someone passes by. Only one of the images of each person that watches the TV is labeled. To the end of each subset, we append five minutes of data, consisting of images of people who are not in the dataset. These people are used to test the sensitivity of our predictor to outliers. We perform our experiments separately on each of the subsets and then aggregate all results.

We note that the datasets in this work are very distinct from those used in previous related works. For example, in Valko *et al.* (2010), the experiments were deliberately staged, in front of a computer, and only a few minutes long. In contrast, we conducted completely uncontrolled experiments over a month. There was no limit to the number of users, the duration of a TV-watching session, or in the manner users watched content. All recorded sessions had multiple disruptions (e.g. random people entering/leaving the frames). Lighting conditions differed depending on ambient lighting and time of day. The distance between the viewer and the TV, while typical for TV watching, presented additional challenges for accurate face recognition. Unlike in similar works, there is no difference between the datasets that we evaluated and the real world data the system would see in an embedded setting.

4.2 Evaluation

All compared algorithms are evaluated by their precision and recall. In Dataset D_1, we deal with individual faces, know the true labels, and both the precision and recall can be computed in a straightforward way.

To avoid tedious labeling of all faces in Dataset D_2, we use a trick. Instead of labeling individual faces, we labeled video frames. Since TV viewing sessions are continuous, their true labels can be obtained very easily. The corresponding precision and recall are computed based on our new definitions of true positives, false positives, and false negatives. A true positive occurs when the subject's face is in the frame and we identify it. A false negative occurs when a person's

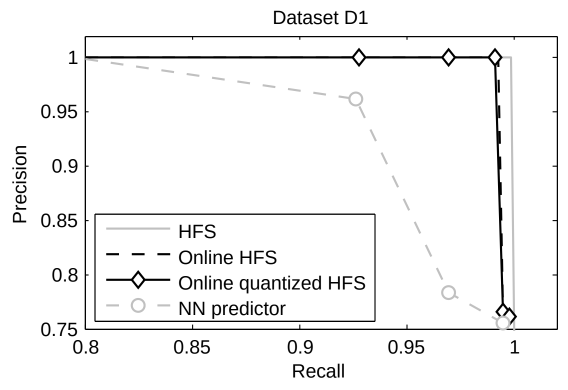

Figure 4: Precision and recall of 4 face recognizers on Dataset D_1. The recognizers are trained by HFS, online HFS, online quantized HFS, and a NN classifier. The points on the curves correspond to various ε. For the harmonic function solutions, ε varies from 10^{-4} to 10^{-12} in multiples of 10^{-2}. For the NN classifier, ε varies from 10^{-4} to 10^{-24} in multiples of 10^{-4}. Different scales are used because the two algorithms have different sensitivities to ε.

face is in the frame and we fail to identify it. A false positive occurs when a viewer's face is not in the frame and we identify it. When a predictor makes multiple identical predictions, we consider it a false positive. This penalizes the predictor for extrapolating too far. Finally, after we compute the per-frame precision and recall for each TV watching session, we aggregate them and report the final numbers.

The precision-recall curves that we generate are obtained by varying the minimum edge. As discussed in Section 3.1, these variations have the effect of loosening or tightening our extrapolation to unlabeled data points. As the minimum edge becomes smaller, recall increases at the expense of decreasing precision.

4.3 Frame Windows

When people sit in front of the TV, they often look away from it for short periods of time, for instance when talking to another person. Therefore, the per-frame evaluation unfairly penalizes face recognizers, such as when no face is extracted from a frame (since the person is looking away from the TV). To mitigate this issue, we look at the precision and recall of face recognizers in 10-second windows (30 frames). The corresponding true positives, false positives, and false negatives are defined as follows.

A false positive occurs if there is at least one false positive in the 10-second window. If this is not the case, a true positive occurs if there is at least one true positive in the window. Finally, if none of the above is true, a false negative occurs if there is at least one false negative in the window. This transformation is only applied to Dataset D_2.

5 Results

First, we compare our online quantized HFS solver to its online and offline counterparts, as well as the nearest-neighbor (NN) classifier, on dataset D_1. We show that the online quantized HFS matches the performance of the online HFS at a fraction of the computational cost and exceeds the performance of the NN classifier. We also compare the cumulative time taken to predict labels on the entire dataset. Next, on the larger dataset D_2, we evaluate precision and recall for the six sub-datasets, take the average of the results to get an aggregated view of the performance, and compare these results to the NN classifier. The online quantized HFS outperforms the supervised classifier at all settings of the minimum edge. Lastly, we compare our results from D_1 with a competition-winning state-of-the-art video face recognition system. We outperform this system, realizing higher precision/recall with significant speed-up.

5.1 Semi-Supervised Learning on Dataset D_1

Figure 4 compares the precision and recall of three harmonic function solutions on Dataset D_1. Based on these results, we can see that the offline HFS has the best precision-recall characteristics (in fact, it has optimal precision and recall) because it has complete knowledge of the entire similarity graph when performing inference. The online quantized HFS maintains very high precision and recall levels and matches the performance of the online HFS. Our solution also outperforms the nearest neighbor classifier at all precision-recall settings, as indicated by the relative positions of the precision-recall curves.

Figure 5 compares the cumulative computation time taken across all three HFS algorithms. The figure shows that for even modest amounts of streaming data (10 minutes worth in this instance), the inference quickly becomes intractable for the online HFS. The only computational "bumps" or "spikes" for the online quantized HFS occur whenever the number of vertices is at or crosses n_g. This is due to the quantization that occurs at these points as described in Section 3.3. Despite these occasional bursts, the online quantized HFS far outpaces the online HFS while yielding similar precision-recall performance as can be seen in Figure 4.

We compared our results on Dataset D1 with a state-of-the-art commercial face recognizer. We achieved 100% precision and recall with our presented approach, however, the commercial recognizer performed poorly on recall and it never exceeded 94%. Additionally, our algorithm is roughly 5 times faster, mostly due to the commercial system's expensive feature extraction and face rotation adjustment steps. Our algorithm's emphasis on real-time inference and improving performance over time shows that identity inference at near-perfect levels can be achieved with just one labeled example, a useful result for deployed and embedded systems.

5.2 Semi-Supervised Learning on Dataset D_2

Figure 6 shows the precision and recall of our solution and the NN baseline on Dataset D_2. Based on these results, our online quantized HFS learner outperforms or is equivalent to a supervised, nearest neighbor learner in all instances. In fact, our solution achieves 100% precision at more than twice the recall of the supervised solution. In addition, with just one labeled data point high precision and high recall are achievable: our algorithm achieves 95% precision at 95% recall.

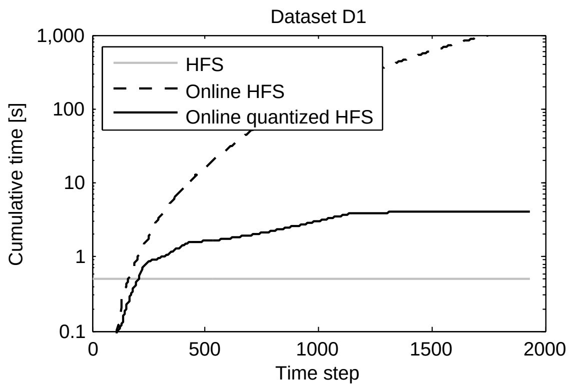

Figure 5: Cumulative computation time of HFS, online HFS, online quantized HFS on Dataset D_1.

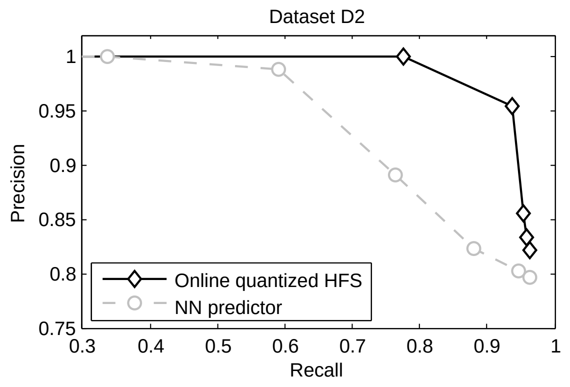

Figure 6: Precision and recall of 2 face recognizers on Dataset D_2. The recognizers are trained by a NN classifier and online quantized HFS. The points on the curves are computed in the same way as in Figure 4.

6 Conclusion

In this paper, we apply a fast algorithm for online learning on graphs to the problem of automatic identity inference of users in front of the TV. We require TV viewers to register only a single image of their face, after which the model is improved over time using unlabeled faces observed by the TV. The approach is evaluated on a 10-hour long TV watching dataset, and both its precision and recall are in the upper nineties. This work is the first instance of using online learning on graphs in the consumer electronics space and it shows a lot of promise.

In the future, we intend to integrate our inference algorithm more closely with the TV functionality so that users can see the benefits of personalization (for example automatically changing the channel to the user's favorite one). We also anticipate running the system for a longer period of time, studying its behavior, and finding practical settings for its parameters, such as the maximum number of representative vertices n_g. Finally, we hope to identify TV viewers based on other features, such as the color of clothes.

7 Acknowledgements

We would like to thank Intel Corporation for funding the equipment to conduct these experiments.

References

Balcan, M.-F.; Blum, A.; Choi, P. P.; Lafferty, J.; Pantano, B.; Rwebangira, M. R.; and Zhu, X. 2005. Person identification in webcam images: An application of semi-supervised learning. In *ICML 2005 Workshop on Learning with Partially Classified Training Data*.

Chang, K.; Hightower, J.; and Kveton, B. 2009. Inferring identity using accelerometers in television remote controls. In *Proceedings of the 7th International Conference on Pervasive Computing*, 151–167.

Charikar, M.; Chekuri, C.; Feder, T.; and Motwani, R. 1997. Incremental clustering and dynamic information retrieval. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, 626–635.

Goldberg, A.; Li, M.; and Zhu, X. 2008. Online manifold regularization: A new learning setting and empirical study. In Daelemans, W.; Goethals, B.; and Morik, K., eds., *Machine Learning and Knowledge Discovery in Databases*, volume 5211 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. 393–407.

Grabner, H.; Leistner, C.; and Bischof, H. 2008. Semi-supervised on-line boosting for robust tracking. In Forsyth, D.; Torr, P.; and Zisserman, A., eds., *Computer Vision ECCV 2008*, volume 5302 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. 234–247.

Kveton, B.; Valko, M.; Philipose, M.; and Huang, L. 2010. Online semi-supervised perception: Real-time learning without explicit feedback. In *Proceedings of the 4th IEEE Online Learning for Computer Vision Workshop*.

Lee, K.-C.; Ho, J.; Yang, M.-H.; and Kriegman, D. 2003. Video-based face recognition using probabilistic appearance manifolds. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 1:313.

MPLab, U. 2009. MPLab GENKI Database. `http://mplab.ucsd.edu`.

Phielipp, M.; Galan, M.; Lee, R.; Kveton, B.; and Hightower, J. 2010. Fast, accurate, and practical identity inference using tv remote controls. In *Innovative Applications of Artificial Intelligence*.

Valko, M.; Kveton, B.; Huang, L.; and Ting, D. 2010. Online semi-supervised learning on quantized graphs. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*.

Zhao, W.; Chellappa, R.; Phillips, P. J.; and Rosenfeld, A. 2003. Face recognition: A literature survey. *ACM Comput. Surv.* 35:399–458.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, 912–919.