Chapter 7

Conclusions

In this concluding chapter we summarize the contributions of this thesis and the possible impact as we see it, and discuss the important directions of future work.

7.1 Summary of thesis contributions

The central problem addressed in this thesis is the problem of modeling a boolean similarity concept, which is conveyed only be means of examples of what constitutes similar and dissimilar pairs under that concept. Before we summarize the specific technical contributions in the remainder of this section, below are the main conclusions we see emerging from our work.

- It is usually beneficial to learn a model for the similarity relevant to the task, be it regression, classification or retrieval. It rarely hurts, and usually improves the performance of the end goal application. Of course, the precise gain of learning similarity for any given application can be assessed by standard validation techniques.

- Such learning can be successfully done directly from examples of similarity judgment specific for the task, with minimal assumptions regarding the properties of the underlying similarity concept. In many cases, for instance when the task involves regression, the learning procedure including labeling similarity examples can be completed fully automatically.

- In some problems, such as pose estimation, example-based methods have been generally overlooked since it is commonly assumed they are computationally infeasible. It does not have to be the case; with suitable embedding technique it may be possible to provide a way of extremely efficient example-based estimation in complex, high-dimensional problems. Our approach, to our knowledge, is the first to combine the power of learning task-specific similarity with the general embedding framework that allows this.

7.1.1 Learning algorithms

The basis of our approach is to construct an embedding

$$H(\mathbf{x}) \;=\; [\alpha_1 h_1(\mathbf{x}), \ldots, \alpha_M h_M(\mathbf{x})],$$

such that low distance between $H(\mathbf{x})$ and $H(\mathbf{y})$ corresponds, with high probability, to positive label assigned by the similarity $\mathcal{S}(\mathbf{x}, \mathbf{y})$. The main advantage of this approach, and what distinguishes it from the alternatives known to us, is that it achieves two important goals:

- It provides us with a set of similarity classifiers on pairs of examples. This set is parametrized by the value of the threshold on distance in the embedding space H.
- It reduces the problem of similarity search to the problem of search for neighbors with respect to the L_1 distance. As a result, we are able to leverage state-of-the-art search algorithms like LSH, that have sublinear running time.

In Chapter 3 we have presented a family of learning algorithms that construct an embedding of the form described above:

Similarity Sensitive Coding (SSC) The algorithm[1] takes pairs labeled by similarity, and produces a binary embedding space H, typically of very high dimension. The embedding is learned by independently collecting thresholded projections of the data. This algorithm improves the performance of example-based methods on some data sets, and has been used however its utility is largely limited to cases when the underlying similarity is close to L_1 distance, with some modifications. This algorithm has been successful in articulated pose estimation domain, as described in Chapters 4 and 5.

Boosted SSC This algorithm[2] addresses the redundancy in SSCby collecting the embedding dimensions greedily, rather than independently. It also introduces weighting on the dimensions of H. We have applied this algorithm to the tasks of pose and orientation estimation for an articulated tracking application, described in Chapter 5.

BoostPro This algorithm is a generalization of Boosted SSCin that the dimensions of the embedding are no longer limited to axis-parallel stumps. We have introduced a continuous approximation for the thresholded projection paradigm in which a gradient ascent optimization becomes possible. This algorithm further improves the performance of example-based methods on standard benchmark data. We also show its performance on articulated pose estimation, in chapter 4. Finally, we have used this algorithm to learn visual similarity of image patches, and have shown significant improvement over standard similarity measures used with two patch descriptors.

[1] Published in [105]; joint work with P. Viola.

[2] Published in [93]; joint work with L. Ren, J. Hodgins, H. Pfister and P. Viola.

Semi-supervised learning For each of these three algorithms we have presented a semi-supervised version which only requires pairs similar under $\mathcal{S}$, in addition to a set of unlabeled individual examples in $\mathcal{X}$.

7.1.2 Example-based pose estimation

In chapters 4 and 5 we have introduced a new approach to pose estimation from single image. Contrary to previously proposed approaches, it does not use a parametric model that is to be fitted to the image. Instead, it uses the learned similarity embedding to search a large database of images with known underlying poses. As a result, the notoriously difficult problem of fitting the articulated pose model is reduced to two much simpler, and much faster, steps: search in a database for (approximately) nearest neighbors, and fitting a local low-order model to the retrieved neighbors. To our knowledge this approach achieves state-of-the-art performance while requiring significantly less time per image than alternative approaches.

7.1.3 Articulated tracking

The main impact of our approach on articulate tracking is in providing a way of automatic initialization of the tracker and, effectively, subsequent re-initialization in every frame. In Chapter 5 we have described two tracking systems that take advantage of this ability. Both systems have been demonstrated to be superior, in terms of combined speed, accuracy and robustness, to state-of-the-art alternatives.

7.1.4 Patch similarity

In Chapter 6 we have described another application of the similarity learning framework: learning visual similarity of natural image patches under rotation and small translation. For two patch descriptors (the sparse overcomplete code coefficients and the very popular SIFT descriptor) we have shown that by learning an embedding of the descriptor with BoostPro and using the distance in the embedding space, we can significantly improve the matching accuracy. The main contributions of this study are:

- This is the first attempt, to our knowledge, to improve the matching accuracy of standard (and widely used) descriptors by learning a similarity model specific to the invariant properties the matching is intended to capture.

- The fact that the learned similarity is measured by the L_1 distance in the embedding space is very significant from the computational point of view, since in a large-scale recognition system we may need to probe databases with millions of patches for similarity to the input set of patches. Our framework allows us to apply algorithms like LSH, and perform this search in sublinear time.

7.2 Direction for future work

Theoretical investigation An open theoretical question that arises from the work presented here pertains to the class of similarity concepts that can be attained by the embedding algorithms presented in Chapter 3. By departing from the framework of LSH to similarity-sensitive framework introduced in Section 2.4.2, we extend the class of similarities from L_1 to a more general family. It would be interesting to characterize the properties of this family, and the connections between the geometry of a similarity concept in $\mathcal{X}^2$ and the extent to which an embedding learned by our algorithms can represent that concept.

Evaluation We believe that a number of interesting additional experiments would be useful to better understand the differences between algorithms and the conditions under which each algorithm is best applicable. Such experiments include an evaluation of boosted SSC on more tasks, in addition to the pose estimation task in Chapter 5, to better understand its capacity and limitations and an investigation into better ways of setting the bound G on the TP-FP gap in SSC. In addition, we are investigating improved strategies of selecting the projection terms (i.e. the dimensions used in a projection) in BoostPro, especially for high-dimensional representation where even approximating the exhaustive search of the space of fixed-size term combinations is impractical.

Another aspect of empirical evaluation that should be improved is in the area of comparing pose estimation algorithms. Although lack of a standard articulated pose benchmark with known ground truth (neither real images nor realistic synthetic ones) makes this difficult, it is important to compare alternative approaches; one approach we are aware of which may provide a suitable alternative has been recently proposed in [2].

Extending the similarity framework In the Introduction we mention definitions of similarity that are more refined than the boolean notion addressed in this thesis. The algorithms presented here are developed to deal with the boolean case, however we believe they can be extended to learning ranking. The main change in the formulation is the transition from the classification of pairs to the classification of triples. Recent work [5] suggests that an embedding can be learned that represents ranking under known distance functions. We believe that it may be possible to extend such an approach to the case when the ground truth ranking is conveyed only by examples, in a spirit similar to our extension of the LSH. One important application of such extension would be in information retrieval, where feedback often is available in the form of ranking rather than just binary labels on the results.

Learning features for visual classification The results presented in Chapter 6 suggest a promising direction of future research in the use of learned similarity. It would be interesting to investigate the effect of embedding the descriptors (and the improved matching accuracy) on classification performance. Below we present an

idea for integration of the similarity learning approach developed in this thesis in a multi-category classification architecture.

Evidence from neuroscience [39] suggests that the majority of cells in the visual pathway may be placed within a computational hierarchy. As the level in the hierarchy increases, which roughly corresponds to retino-cortical direction (away from the retina),

- The invariance increases: features become less sensitive to various transformations.
- Selectivity increases: it takes a more distinctive image elements to activate a feature. Consequently, higher layers should be more overcomplete and sparse.
- Receptive fields become larger.
- Receptive fields become more complex (more non-linear, in particular).

From a computational point of view, the order in the hierarchy corresponds to order of processing: the lowest level corresponds to measures computed directly from the image pixels, and the values in subsequent layers may be computed from the values obtained in the lower levels. However, it is not clear how the flow of sensory information and decisions across the hierarchy is organized in the brain; in particular, there exists a huge number of feedback projections along the visual pathway, the function of which is not fully understood.

It would be interesting to explore a hierarchical representation organized in accordance with the computational principles mentioned above. Finding the learning algorithm for constructing such a hierarchy is the main challenge in designing such an architecture. An interesting approach could be to learn the lower, less selective layers in an unsupervised way, while the higher, more selective layers could be better learned on a per-category basis, perhaps in conjunction with learning object- or part-specific similarity operators, along the lines developed in this thesis.

Figure 7-1 shows a "cartoon" of this approach. An appealing property of it is that lower-level features are necessarily shared between all categories, while higher-level features are more likely to be unique for a given class (although the learning algorithm should probably allow for sharing in later layers as well).

It's important to emphasize the difference between this approach and, say, the standard multi-layer neural network where a designated output layer is the only one affecting the decision. In the proposed hierarchy there is no output layer per se, but rather the entire set of features is considered in similarity calculations. This is achieved by allowing any feature to be used in similarity-reflecting embeddings for the highest (categorical) level.

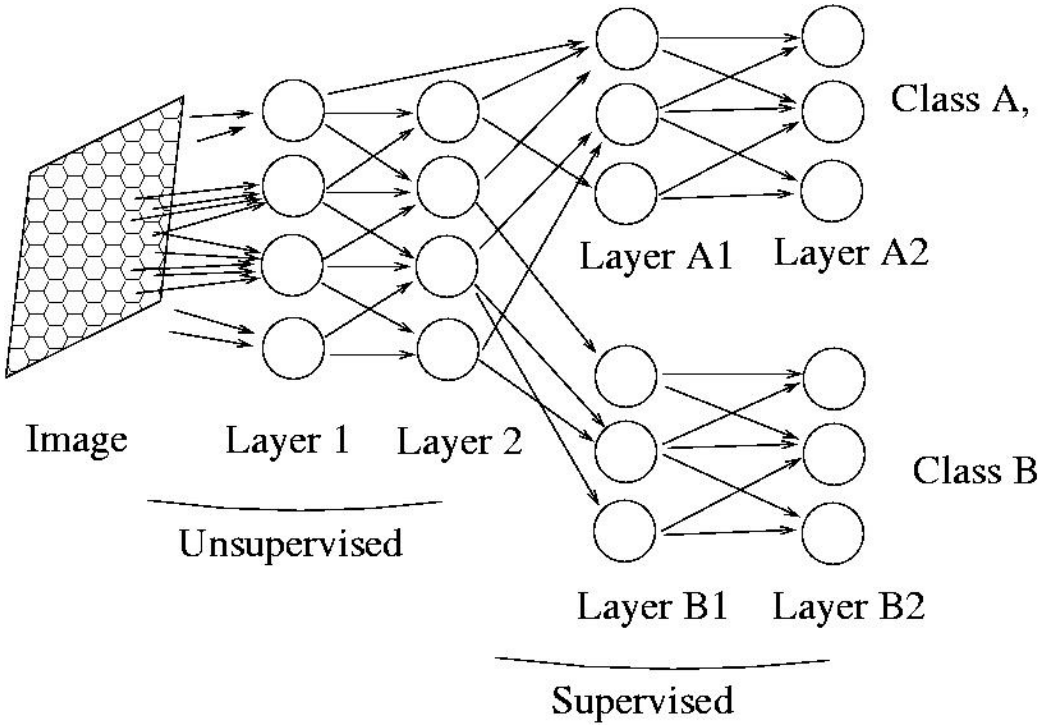

Figure 7-1: A cartoon of the proposed hierarchical representation, showing the sharing of the features and the two-stage learning architecture. A representation for a given image patch may include any of the features from the lower (generic) layers and any of the features from the higher, class-specific layers.