TRAINING IMAGE CLASSIFIERS WITH SIMILARITY METRICS, LINEAR PROGRAMMING, AND MINIMAL SUPERVISION

Karl Ni[†], *Ethan Phelps*[†]. *Katherine L. Bouman*[‡], *and Nadya Bliss*[†]

MIT Lincoln Laboratory[†], Massachusetts Institute of Technology[‡]

E-mails: {karl.ni, ethan.phelps, nt}@ll.mit.edu, klbouman@mit.edu

ABSTRACT

Image classification is a classical computer vision problem with applications to semantic image annotation, querying, and indexing. Recent and effective generative techniques assume Gaussianity, rely on distance metrics, and estimate distributions, but are unfortunately not convex nor keep computational architecture in mind. We propose image content classification through convex linear programming using similarity metrics rather than commonly-used Mahalanobis distances. The algorithm is solved through a hybrid iterative method that takes advantage of optimization space properties. Our optimization problem uses dot products in the feature space exclusively, and therefore can be extended to non-linear kernel functions in the transductive setting.

1. INTRODUCTION

Image classifiers and content recognition is an age-old computer vision problem, the most prominent applications being labeling and retrieving images semantically. The literature has consistently employed learning algorithms involving parameter estimation built from training sets. Training and classification methods almost universally rely on two components: feature extraction and matching.

Both feature extraction and matching require low noise levels in the training data, and therefore, significant manual involvement in either labeling or segmentation. Additionally, extensive cross-validation procedures must drive down false alarms. Finally, there may be multiple instances of a single concept that are not addressed. To ensure relevant and accurate features at such massive scales, training data fidelity and segmentation truth is often manually performed with crowd-sourcing tools like Antonio Torralba's LabelMe [1],the now-retired Google labels, and most face/object detection/recognition training sets [2, 3]. While effective, the gain in accuracy has not yet offset the needed throughput.

This has inspired a more recent push towards multi-instance, unsupervised learning [4, 5, 6], in which the proposed algorithm is grouped. The paradigm reflects the notion that with enough quantity, where current data rates and accessibility are outpacing processing capabilities, training quality can be improved naturally via large numbers and through noise integration.

Popular multi-instance learning techniques approach classifier construction generatively by modeling the conditional distributions of various semantic classes [7, 8, 9]. The most mature parameter estimation for distribution parameters were effected with multi-modal Gaussian mixtures (GMM's). Unfortunately, without correct choices in the number of clusters, assumptions on noise behavior, and good initialization, maximum likelihood parameter estimates through expectation maximization (EM, a.k.a, iterative annealing, [10]) will produce irrecoverable and incorrect feature prototypes. Furthermore, GMMs have small sample bias and are often instable with respect to parameterization. Subsequently, iteratively determined optimal values are sensitive to initialization. Online or incremental clustering is also limited through EM and may require respecification of variables. The problem is augmented by the number of parameters to be updated, which significantly impacts the objective function. Finally, convergence speed depends on dimensionality as GMMs and similar techniques traditionally (and logically) utilize difference metrics, often the Mahalanobis distance.

Instead of modeling the representation generatively, we propose to determine prototype features for comparing images *directly* by finding a small subset through sparsity constraints in a linear programming (LP) problem. Replacing distance metrics and using only dot products, nonlinearity may be introduced with kernel matrices representing a positive definite kernel space. The resultant system classifier relies on normalized cross-correlation (similarity) between features derived from a query image and those from a trained subset of prototypes. The implementation as matched filter bank will fit many system architectures.

This paper will discuss these issues and the resultant classifying prototypes along with the many practical aspects that allow minimal supervision. Sec. 2 describes the convex problem required to train images. Sec. 3 discusses associated theoretical results and problems, and Sec. 4 promotes more practical procedures.

This work is sponsored by the Department of the Air Force under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

2. TRAINING IMAGE CLASSIFIERS WITH SIMILARITY METRICS

The empirical determination of optimal filters in a training set is based on solutions that find the best prototype or feature set that is least redundant. The optimal features are used as templates, that can eventually serve as matched filters during runtime. This section details the convex optimization problem that can be used to determine both linear and nonlinear filters for discrimination.

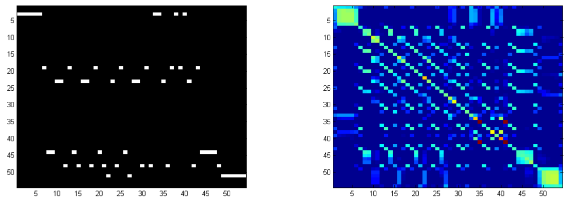
2.1. Linear Classification in Euclidean Space

Let a d dimensional feature be denoted by $\mathbf{x}$, and X be the collection of N features, and $\mathbf{x}_i$ is organized as the i^{th} column of X. Then an LP problem that determines the vectors for a data set that are the most representative and least redundant can be specified as follows.

$$\begin{aligned} \underset{\boldsymbol{\beta},\mathbf{t}}{\arg\min} \quad & -tr\left(X^T X \boldsymbol{\beta}\right) + \lambda \cdot \sum_i t_i \\ \text{such that} \quad & 0 \leq \beta_{ij} \leq t_i \leq 1 \\ \text{and} \quad & \boldsymbol{\beta}^T \mathbf{1} = \mathbf{1} \end{aligned} \tag{1}$$

In (1), the selector matrix to determine which features to use as prototypes is embedded in $\boldsymbol{\beta}$; the tuning parameter λ determines the extent to how much we'd like to reduce redundancy by inducing sparsity; and the training set for a single class is written in matrix form, $X \in \mathbb{R}^{d \times N}$. Intuitively, the solution matrix $\boldsymbol{\beta}$ will indicate the smallest set of features in X that best represent it by indicating them with nonzero values. Each column vector $\beta_i \in \boldsymbol{\beta}$ selects the candidate prototypes for every $\mathbf{x}_i$. As the full paper will discuss, the natural tendency of the elements of $\boldsymbol{\beta}$ will tend toward 1 or zero, but occasionally it can take on a value v in between. In such cases, a single "best" vector is chosen through maximum likelihood. Regardless of the values in β_i, the final matrix $\boldsymbol{\beta}$ will have a rank equal to the number of classes as its optimum value.

Correctly framed sparsity solutions not only induce efficiency in computation and class depiction, but can reduce noise and improve error rates. Optimization in (1) is reminiscent of research on sparse *feature representation* (i.e., dictionary learning techniques)[1]. While classifying images is often formulated with the construction of learned features, recent surveys on such work has proven such methodology unnecessary and inefficient. Nevertheless, similar techniques to enforce sparse *class structure* [2], have appeared in convex grouping problems [11], though are less efficient and intuitive as seen in Fig. 2.1.



(a) min $\sum \ell_\infty$ rows (b) min $\sum \ell_2$ columns

Fig. 1. Fig. 1(a) is the proposed algorithm while Fig. 1(b) is the ℓ_2 Group Lasso penalty

2.2. Nonlinear Classification via Kernel Matrix

Note that (1) consists solely of dot products with respect to vectors in X, which we can use to improve and extend the proposed problem in (1). Similar to SVM's, nonlinearity may be introduced in the form of kernel optimization with the reproducing kernel Hilbert space (RKHS) in the general form of (2), where K is a positive definite kernel function (or convex grouping of kernel functions) in the RKHS. The logical extension to (1) is the straightforward assignment of $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, where we can bypass the calculation and knowledge of the high-dimensional mapping of $\phi : \mathbb{R}^d \to \mathbb{R}^t$ in the transductive setting. When $t \gg d$, (2) is exceedingly useful.

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\arg\min} \quad & \sum_{ij} \beta_{ij} K(\mathbf{x}_i, \mathbf{x}_j) + \lambda \cdot \sum_i t_i \\ \text{such that} \quad & 0 \leq \beta_{ij} \leq t_i \leq 1 \\ \text{and} \quad & \boldsymbol{\beta}^T \mathbf{1} = \mathbf{1} \end{aligned} \tag{2}$$

3. CLASSIFIER ANALYSIS

For most learning frameworks, an instance $\mathbf{x}$ is classified by comparing to prototypes or probabilistic models to determine the likeliest solution based on a distribution in some feature space. That is, the feature vector $\mathbf{x}$ belongs to class i of C classes if it is closest to the prototypes in the set $\{\mathbf{y}_j\}_i$ characterizing the i^{th} class. Take a simplistic view of classification in GMMs, where each mixture component relates to a single class:

$$\underset{i \in \{1,\ldots,C\}}{\arg\max} \quad K \exp\left[(\mathbf{x} - \mathbf{y}_i)^T \Sigma^{-1} (\mathbf{x} - \mathbf{y}_i)\right] \tag{3}$$

$$= \underset{i \in \{1,\ldots,C\}}{\arg\min} \quad -2\mathbf{x}^T \Sigma^{-1} \mathbf{y} + \|\mathbf{y}_i\|^2_{\Sigma^{-1}}. \tag{4}$$

Solutions to (4) are the same as (3); that is, the classification of an input $\mathbf{x}$ relies on the Mahalanobis distance to all class prototypes. It is not uncommon to normalize $\mathbf{y}_i^T \Sigma^{-1} \mathbf{y}_i$ to a scalar value (say unity) for every class, though we constrain feature vectors to the unit ellipse (or ball, depending on Σ).

[1] It is important to note that the proposed algorithm does *not* solve this problem

[2] As opposed to feature representation, our problem addresses this problem instead.

One will find normalized class representations in many applications in biological datasets, image processing applications, detection-theory, etc., where a signal processing paradigm places significant emphasis on the relationship *between* feature dimensions rather than the actual values themselves. For example, pre-processing in images for computer vision-based applications often involves DC subtraction and division by pixel variance.

Under such an assumption, (4) can be written as the dot product of $\mathbf{x}$ and $\mathbf{y}_i$:

$$\underset{i\in\{1,\dots,C\}}{\arg\max} \quad < \mathbf{x}, \mathbf{y}_i > \tag{5}$$

This is an important result because the classifier is broken down to a simple cross-correlation between $\mathbf{x}$ and $\mathbf{y}_i$, where $\mathbf{y}_i \in \{\mathbf{y}_1, \mathbf{y}_2 \cdots \mathbf{y}_C\}$, each vector a known prototype of a given class. The process of matching $\mathbf{x}$ with a bank of filters is frequently called categorization by matched filters, where the Cover and Hart inequality holds, $R^* \leq R \leq R^* \left(2 - \frac{C}{C-1} R^*\right)$, where R^* is the Bayes error rate.

3.1. Asymptotic Consistency

The infinity norm regularization in the proposed optimization relies on naturally clustered events, where $\mathbf{x}_i$ is not unique within X, suggesting inconsistent (and initialization-dependent) $\boldsymbol{\beta}$ estimators. For example, take $X(\xi) = Y + \xi$, where ξ is additive noise. If Y contains several instances of the same vector, then $\hat{\boldsymbol{\beta}}$ can represent $X(\xi)$ with any $\mathbf{y}_i$, where $\hat{\boldsymbol{\beta}}$ is the estimated solution. Or, it can represent *all* of them in the unlikely event that $X(\xi)^T Y = \mathbf{1}\mathbf{1}^T$ with λ improperly chosen. This scenario is rare for sufficiently large λ since the ℓ_1-norm of ℓ_∞-norms tends toward a single selector value as opposed to the 2-norm, seen in Fig. 2.1.

However, there are sufficient conditions for asymptotic consistency, which may not necessarily satisfy $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2 = o_P(1)$, but may guarantee properties about the *grouping* of features for a given λ and the total number of clusters $C(\lambda)$. Under our penalization in (1), $\boldsymbol{\beta}$ promotes a unique and consistent grouping, namely that $rank(\boldsymbol{\beta}_g) = 1$, with $\boldsymbol{\beta}_g$ being a submatrix of $\boldsymbol{\beta}$ for group g. Therefore, the number of clusters $C(\lambda)$ equals $rank(\boldsymbol{\beta})$.

4. APPLICATION CONSIDERATIONS

In order to apply the algorithm to discriminate between classes (in either one versus all scenario) and at scale, we can apply simple yet effective common methodologies. Previously, Sec. 2 proposes a solution to create within-class filters. This section discusses best filters to use *between* the classes as well as how to train filters hierarchically.

4.1. Between Class Filter Optimization

Clustering for each class will naturally yield similar recurrent filters among classes that, while representative of a portion of a single class, are not discriminative between them. For example, one will often find that a large portion of most images contain the sky. This is true whether or not one wishes to differentiate between images of, say, mountains or buildings, two completely unrelated concepts that happen to share a similar feature in the images. Analogously, the discriminating power in "sky features", which the within feature optimization will invariably produce, will be low because $P(\text{mountain}|\text{sky})$ and $P(\text{buildings}|\text{sky})$ values are small.

Deletion of similar filters is then a logical step, and the choices of which filters to remove are simply those with high correlation occurring across a pair of classes. We can define a threshold t_{keep} for features that we wish to keep. Let $X^{(r)} \subseteq X$ and $Y^{(r)} \subseteq Y$ be the collection of *within-class* representative features for classes c_1 and c_2, respectively, where $\mathbf{x}_i \in X^{(r)}, \mathbf{y}_j \in Y^{(r)}$. The final set of pair-wise *between-class* filters discriminating c_1 and c_2 is:

$$\{(f_{c_1}, f_{c_2})\} = \left\{ (\mathbf{x}_i, \mathbf{y}_i) : \ \left(\max_{\mathbf{y}_j} \mathbf{x}_i^T \mathbf{y}_j \right) < t_{keep} \ \right\} \tag{6}$$

4.2. Hierarchical Filters

As discussed in [8], hierarchical methods are especially useful for groupings that may appear different in different situations. *Filter hierarchies* address scenarios where groupings reflect some semantic organization. In [12], image patch-based clustering of objects taken at several angles, times of days, etc., may appear different for each instance. Furthermore, mixture hierarchies are useful for complexity reasons because we have relied on the covariance matrix, where memory can grow according to MN. Since the proposed algorithm aims to remove redundancy, we prune especially large data sets (M and N on the order of millions) to a few relevant features to take advantage of central limit behavior, a property enabling [8] to automatically segment images without explicitly specifying boundaries.

Hierarchical training operates over several data subsets (e.g., images), effectively partitioning the class data. We optimize over each subset, and then between each subset. According to [9], irrelevant features (noise) will occur infrequently while class features will arise; normalization will asymptotically integrate noise to zero in distribution. The procedure is, then, to first find $\boldsymbol{\beta}$ in data subsets and between data subset. After this optimization, the rows of $\boldsymbol{\beta}$ corresponding to the highest frequency features relate to class structure.

5. RESULTS

Of the large set of features to choose from (e.g. SIFT [13], SURF, Cosine Transforms, GIST [14]), our classification leverages uniformly extracted, multi-channel (RGB/YBR) DCT feature vectors, much like Carneiro et al. [4]. Though it is an isometric transform (with DCT/pixel ℓ_2-distance equal), we take advantage of DCT's energy compaction property with the first 45 dimensions while weighting color components higher to improve illumination-invariance. Classification accuracy for individual image patches are shown in Table 1. A separate application in Table 2, the localization of images, stresses the multiple instance learning potential of the proposed algorithm by classifying images into particular locations. The latent features (which we have conceptually labeled) underscore another capability that by training for semantic concepts, image segmentation is gained for free. The segmentation and labeling of a location is visually shown in Fig. 5. This is further evidenced by the automatic extraction of faces in Fig. 5.

Table 1. Classification accuracy for synthetic and corel image datasets [5]. Below are the performances under various initialization conditions. The metrics are probability values of **Correct Detection, Correct Rejection, False Alarm, and Misses**.

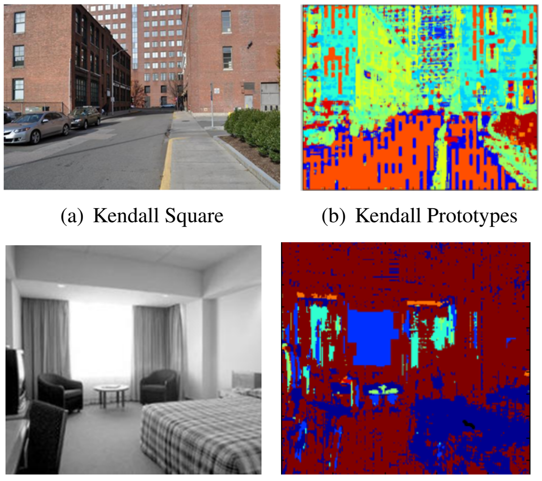
		H_{C1} vs H_{C2} Performance (%)			
	Methodology	P_{det}	P_{rej}	P_{fa}	P_{miss}
Synth	GMM X-Val'd Init	97.24	92.51	7.49	2.76
	GMM Under-Init	87.84	17.95	82.05	12.16
	GMM Over-Init	85.24	86.25	13.75	14.76
	Best k-means	89.84	90.49	9.51	10.16
	Group Lasso	93.17	90.05	9.95	6.83
	LP Estimate	94.84	92.04	7.96	5.16
Corel	Best GMM (26 inits)	87.24	76.76	23.24	12.76
	GMM Under-Init	70.36	64.09	35.81	29.64
	GMM Over-Init	75.57	65.92	34.18	24.43
	LP Estimate	87.32	74.51	25.49	12.68

Table 2. An example application of semantic or concept image classification is the geo-location problem, where a collection of images at various locations are gathered for training. The performance is based on how well a classifier places the images at the correct locations in the training set. Below is the **Multi-class Cross-validation Confusion Matrix.**

	Training Set Locations			
Test Set	MIT Kendall	Lubbock, TX	Dubrovnik	Vienna
MIT Kendall	0.930	0.062	0.028	0.083
Lubbock, TX	0.019	0.902	0.039	0.041
Dubrovnik	0.014	0.024	0.879	0.038
Vienna	0.036	0.013	0.057	0.838

6. CONCLUSIONS

We have proposed a sparse data representation procedure that can determine prototypes quickly and efficiently. This representation can be used for clustering, classification, and feature selection with the advantages of fast matched filtering. The algorithm has several contributions which are enumerated as follows.


(a) Kendall Square (b) Kendall Prototypes

(c) Indoor Room (d) Room Prototypes

Fig. 2. In the classification of scenes, different prototypes typically dominate in identifying different portions of a scene. The segmentation seen in the scene is a natural result of correlation and relevancy. The top scene is classified as the Kendall Square area of Cambridge, MA, 1,237 low-resolution images trained from Table 2. The bottom scene is an example derived from training data in the Corel data set [5].

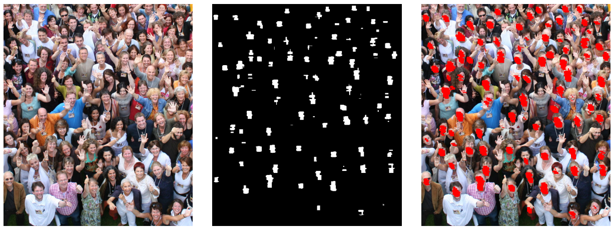

(a) Single image (b) Class/Cluster #14 (c) Overlaid Class

Fig. 3. Completely unsupervised clustering trained on a *single image* in Fig. 3(a) of a crowd producing several selected features: including one of faces Fig. 3(b) and Fig. 3(c)

- An approximation to the LP relaxation solves an optimization problem to obtain representative features.
- Class prototypes based on their covariance matrix are sparse and can be tuned with a λ parameter.
- Filter hierarchies can be built and similar filters between classes should be removed.
- Results generalize well to several data sets.

Further research is still warranted in understanding and characterizing our solution, most notably selection of λ and consistency modeling. Rigor and statistical justification will also be the subject of extended papers in the future. Finally, as evident in recent talks by Google and Torralba et. al, classification performance is directly associated with the context in which is applied, where assessing global properties of the feature space could boost performance.

7. ACKNOWLEDGEMENTS

We would like to thank Andrew Bolstad at MIT Lincoln Laboratory for all the help, advice, and good ideas he has given us in regard to convex optimization for sparse regularization techniques.

8. REFERENCES

[1] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “Labelme: a database and web-based tool for image annotation,” *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157–173, May 2008.

[2] Paul Viola and Michael Jones, “Rapid object detection using a boosted cascade of simple features,” 2001, pp. 511–518.

[3] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, “Discriminatively trained deformable part models, release 4,” http://people.cs.uchicago.edu/ pff/latent-release4/.

[4] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos, “Supervised learning of semantic classes for image annotation and retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 394–410, March 2007.

[5] Pinar Duygulu, Kobus Barnard, Nando de Freitas, and David Forsyth, “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary,” *Seventh European Conference on Computer Vision*, vol. IV, pp. 97–112, 2002.

[6] K. Barnard and D. Forsyth, “Learning the semantics of words and pictures,” *Proceedings of the International Conference on Computer Vision*, vol. 2, pp. 408–415, 2001.

[7] D. Blei and M. Jordan, “Modeling annotated data,” *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003.

[8] Nuno Vasconcelos and Andrew Lippman, “Learning mixture hierarchies,” in *Neural Information Processing Systems*, Denver, Colorado, 1998, vol. 11.

[9] N. Rasiwasia and N. Vasconcelos, “Holistic context modeling using semantic co-occurrences,” *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[11] Han Liu and Jian Zhang, “Estimation consistency of the group lasso and its applications,” *Journal of Machine Learning Research - Proceedings Track*, vol. 5, pp. 376–383, 2009.

[12] Nuno Vasconcelos, “Image indexing with mixture hierarchies,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 2001.

[13] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[14] Aude Oliva and Antonio Torralba, “Modeling the shape of the scene: a holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.