Local Distance Functions: Some Thoughts and Experiments

Deva Ramanan
UC Irvine

Basis of much of this talk

Local Distance Functions: A Taxonomy, New Algorithms, and an Evaluation

Deva Ramanan and Simon Baker

***Abstract*—**
We present a taxonomy for local distance functions where most existing algorithms can be regarded as approximations of the geodesic distance defined by a metric tensor. We categorize existing algorithms by how, where, and when they estimate the metric tensor. We also extend the taxonomy along each axis. How: We introduce hybrid algorithms that use a combination of techniques to ameliorate over-fitting. Where: We present an exact polynomial time algorithm to integrate the metric tensor along the lines between the test and training points under the assumption that the metric tensor is piecewise constant. When: We propose an interpolation algorithm where the metric tensor is sampled at a number of references points during the offline phase. The reference points are then interpolated during the online classification phase. We also present a comprehensive evaluation on tasks in face recognition, object recognition, and digit recognition.

***Index Terms*—Nearest Neighbor Classification, Metric Learning, Metric Tensor, Local Distance Functions, Taxonomy, Database, Evaluation**

I. Introduction

The K-nearest neighbor (K-NN) algorithm is a simple but effective tool for classification. It is well suited for multi-class problems with large amounts of training data, which are relatively common in computer vision. Despite its simplicity, K-NN and its variants are competitive with the state-of-the-art on various vision benchmarks [6], [23], [43].

A key component in K-NN is the choice of the distance function or metric. The distance function captures the type of invariances used for measuring similarity between pairs of examples. The simplest approach is to use a Euclidean distance or a Mahalanobis distance [14]. Recently, a number of approaches have tried to learn a distance function from training data [5], [15], [23], [24], [38]. Another approach is to define a distance function analytically based on high level reasoning about invariances in the data. An example is the tangent distance [37].

The optimal distance function for 1-NN is the probability that the pair of examples belong to different classes [28]. The resulting function can be quite complex, and generally can be expected to vary across the space of examples. For those motivated by psychophysical studies, there is also evidence that humans define categorical boundaries in terms of local relationships between exemplars [34]. The last few years have seen an increased interest in such local distance functions for nearest neighbor classification [12], [13], [30], [43], though similar ideas were explored earlier in the context of local discriminant analysis [10], [19] and locally weighted learning [3], [35]. Recent approaches have also leveraged local metrics for Gaussian process regression [39] and multiple kernel learning [16].

Our first contribution is to present a taxonomy for local distance functions. In particular, we show how most existing algorithms can be regarded as approximations of the geodesic distance defined by a metric tensor. We categorize existing algorithms in terms of **how**, **where**, and **when** they estimate the metric tensor. See Table I for a summary. In terms of **how**, most existing algorithms obtain a metric either by dimensionality reduction such as principle components analysis (PCA) or linear discriminant analysis (LDA) [14], or by explicit metric-learning [5], [15], [24]. In terms of **where**, existing algorithms sample the metric tensor: (a) for the whole space ("global"), (b) for each class ("per-class") [24], (c) at the training points ("per-exemplar") [12], [13], or (d) at the test point [3], [19]. In terms of **when**, existing algorithms either estimate the metric tensor: (a) offline during training or (b) online during classification.

Our second contribution is to extend the taxonomy along each dimension. In terms of **how**, we introduce hybrid algorithms that use a combination of dimensionality reduction and metric learning to ameliorate over-fitting. In terms of **where**, we consider algorithms to integrate the metric tensor along the line between the test point and the training point. We present an exact polynomial time algorithm to compute the integral under the assumption that the metric tensor is piecewise constant. In terms of **when**, we consider a combined online-offline algorithm. In the offline training phase, a representation of the metric tensor is estimated by sampling it at a number of reference points. In the online phase, the metric tensor is estimated by interpolating the samples at the reference points.

Our third contribution is to present a comprehensive evaluation of the algorithms in our framework, both prior and new. We show results on a diverse set of problems, including face recognition using MultiPIE [18], object recognition using Caltech 101 [11], and digit recognition using MNIST [27]. To spur further progress in this area and allow other researchers to compare their results with ours, we will make the raw feature vectors, class membership data, and training-test partitions of the data available on a website with the goal of defining a standard benchmark[1].

II. Taxonomy and Algorithms

We now present our framework for local distance functions. We begin in Section II-A by describing the scenario and class of functions that we consider. We introduce the metric tensor and explain how it defines the geodesic distance. In Section II-B we describe how the core distance functions can be learnt using either dimensionality reduction or metric learning and extend the framework to include hybrid algorithms. In Section II-C we

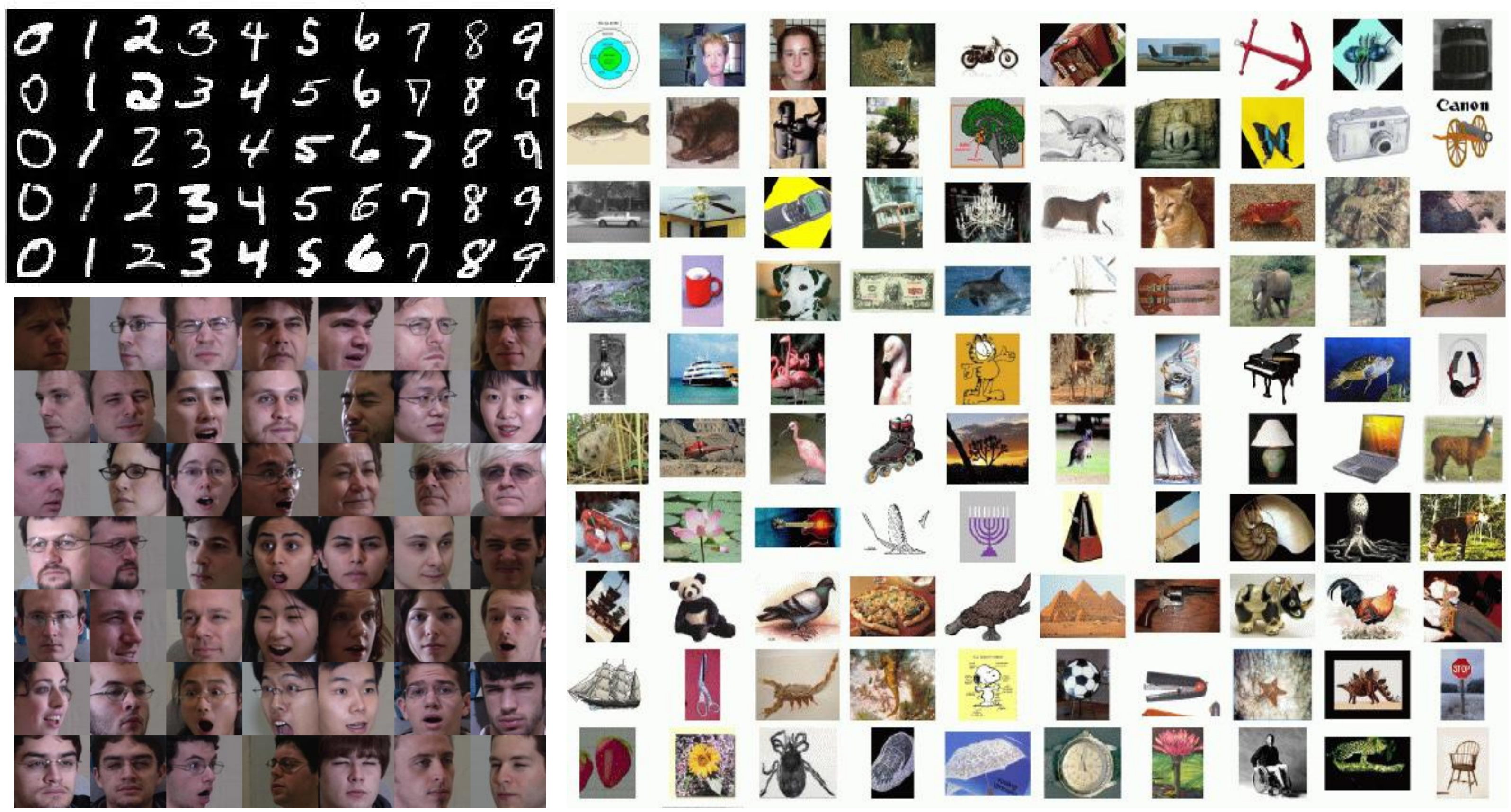
Deva Ramanan is with UC Irvine.
Simon Baker is with Microsoft Research.
A preliminary version of this paper appeared in the IEEE International Conference on Computer Vision [32].

[1]Available at http://www.ics.uci.edu/~dramanan/localdist/

Joint with Simon Baker, PAMI 2011

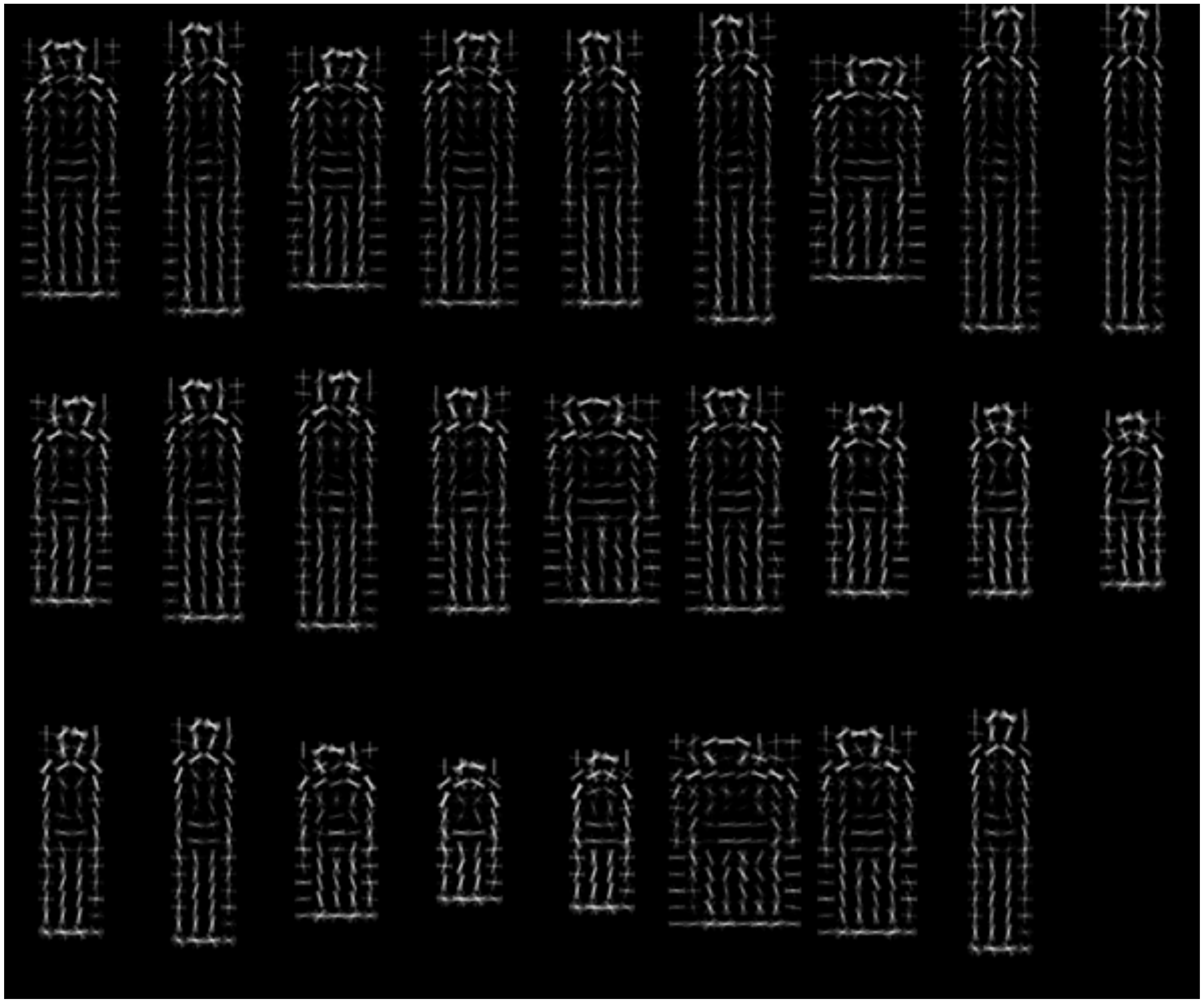
Nearest neighbor is a workhorse of vision

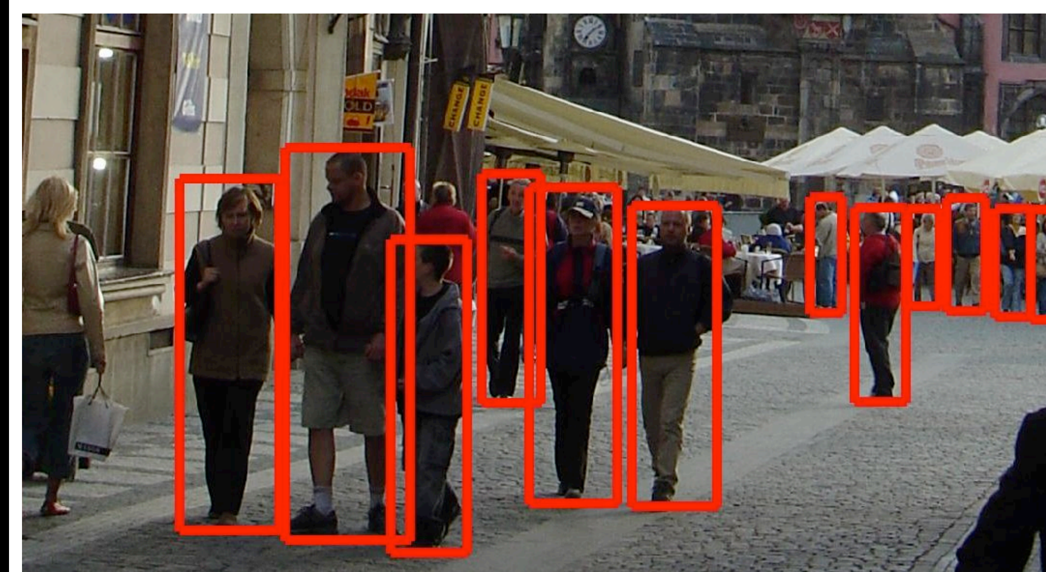
Works well for multiclass problems with lots of data



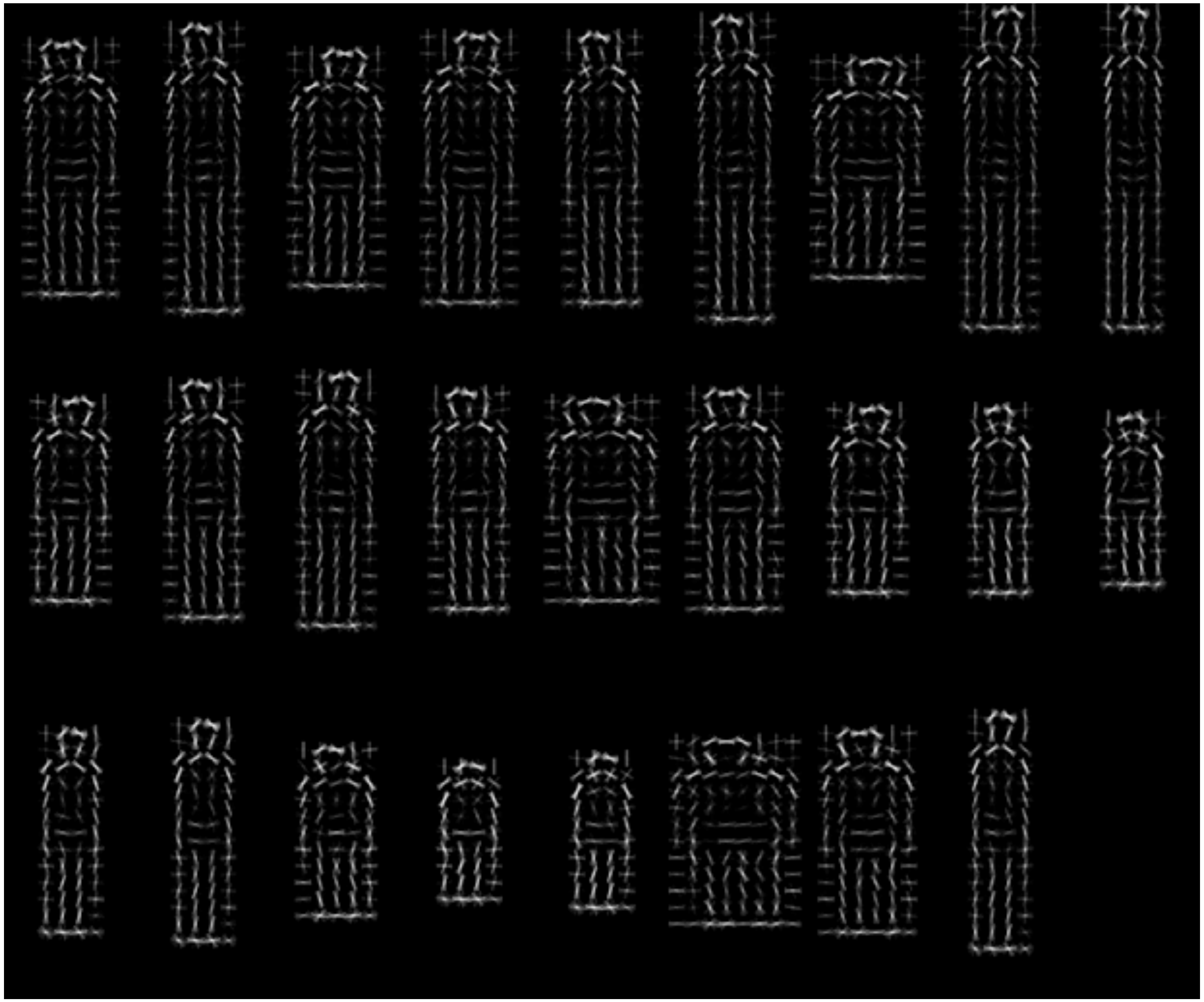
Competitive with state-of-the-art for:
Digit classification
Face recognition
Image classification (Caltech)

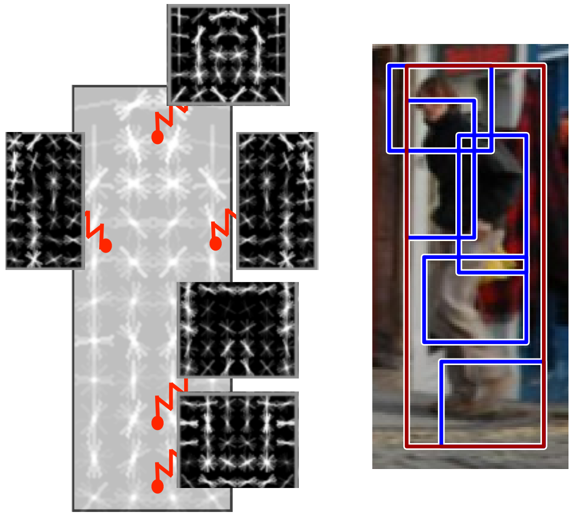
Indexing a large set of examples





Indexing a large set of examples

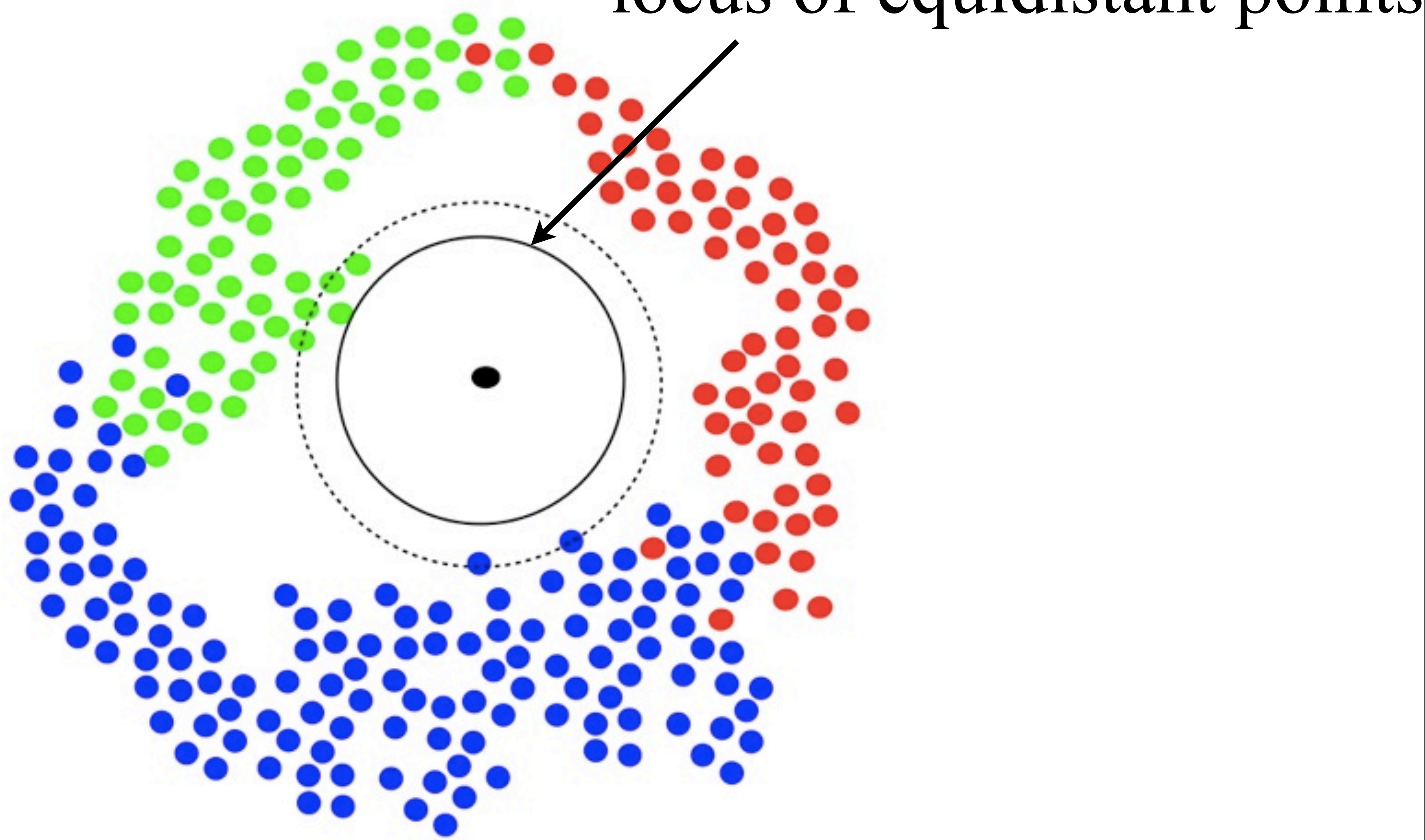




Parts allow us to index a (exponentially) large set of deformed templates

Basic NN

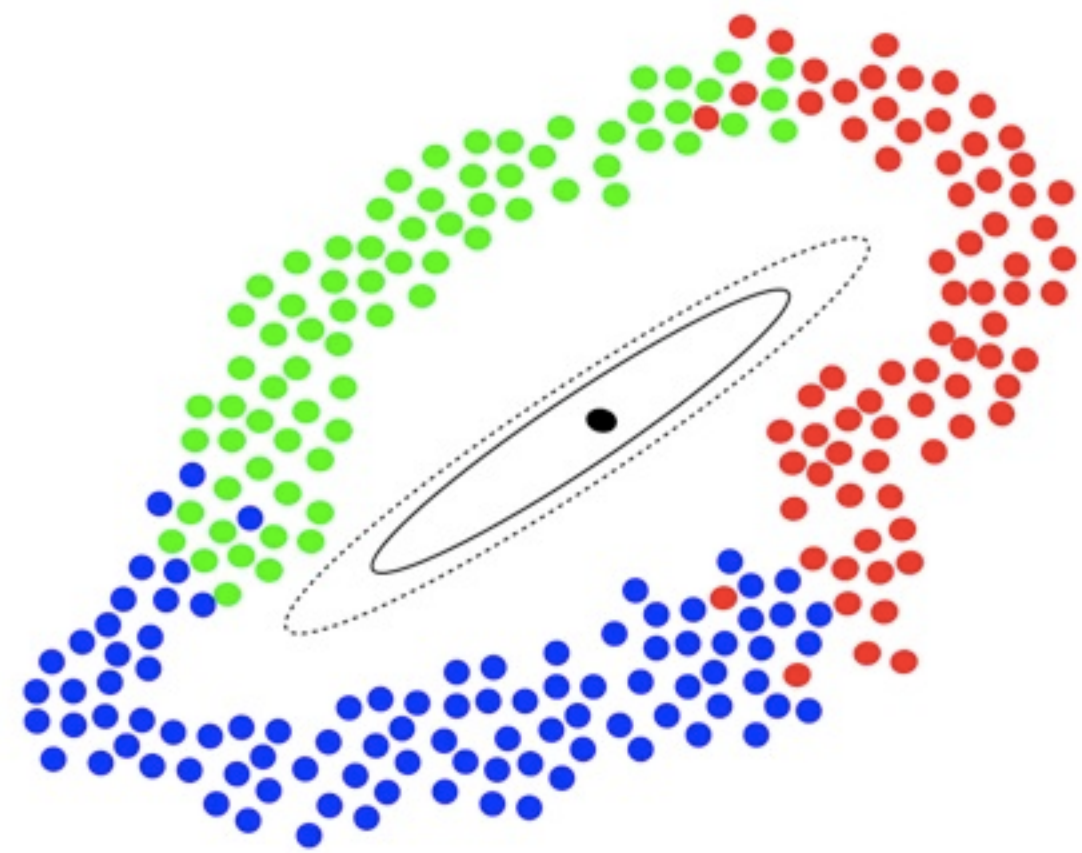
- Test point
- Class 1
- Class 2
- Class 3



Metric learning

Warp space to emphasize particular directions

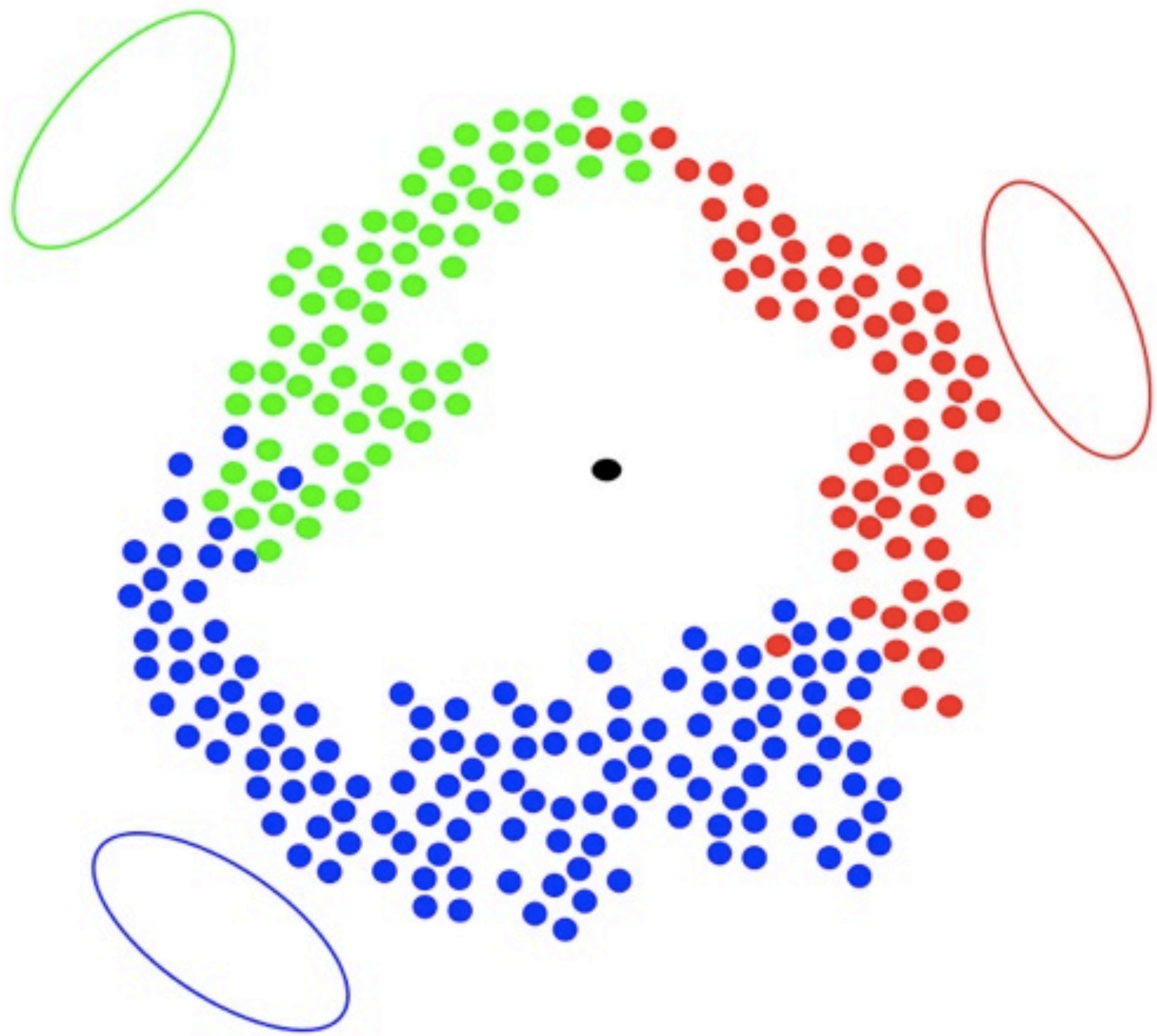
Golberger et al. NIPS 05
Torresani & Lee NIPS 07
Kumar et al. ICCV 07
Bar-Hillel et al. JMLR 05
Weinberger et al. JMLR 09



Past work:

Neighborhood Component Analysis (NCA)
Relevant Component Analysis (RCA)
Large Margin Nearest Neighbors (LMNN)

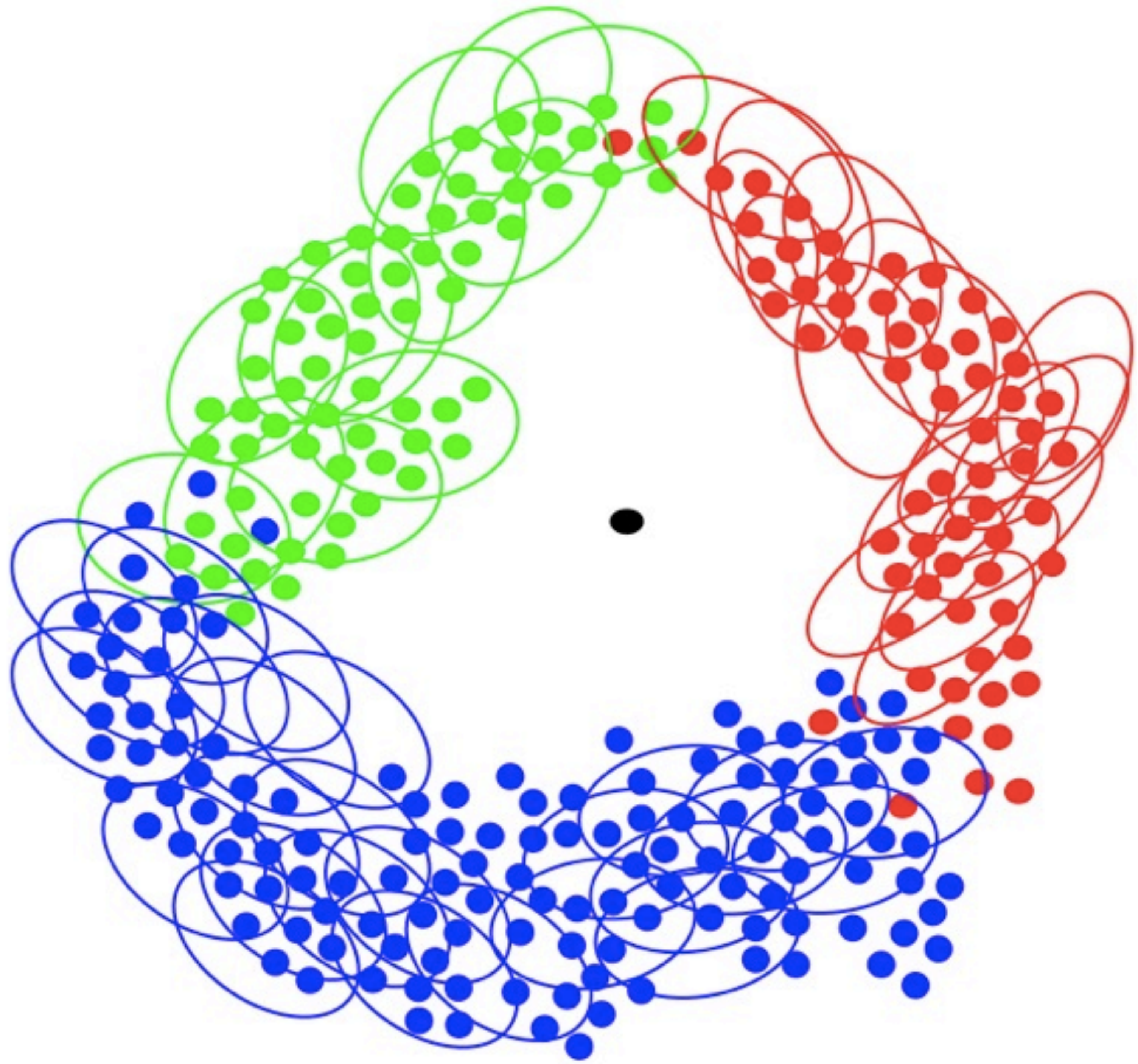
Local distance functions



Use a separate metric for each class

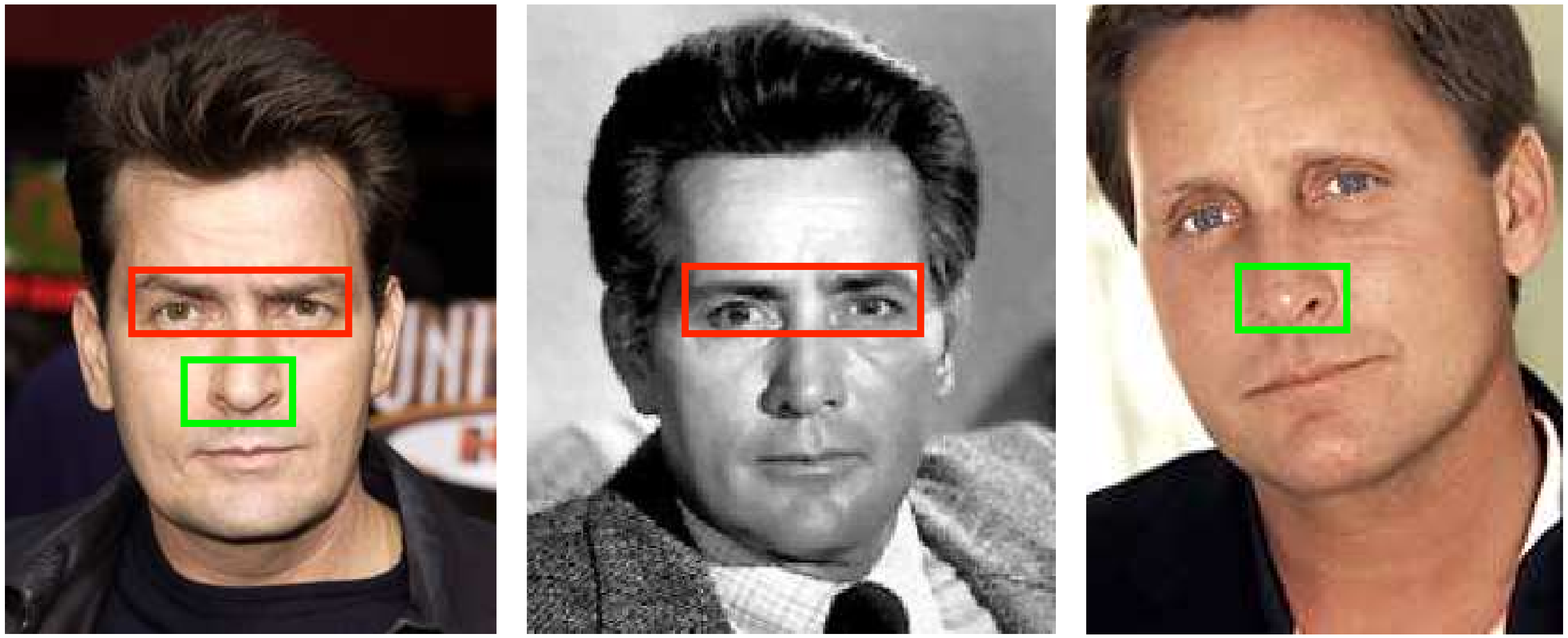
“Per-class” metrics

Local distance functions



Use a separate metric for each training example
"Per-exemplar" metrics

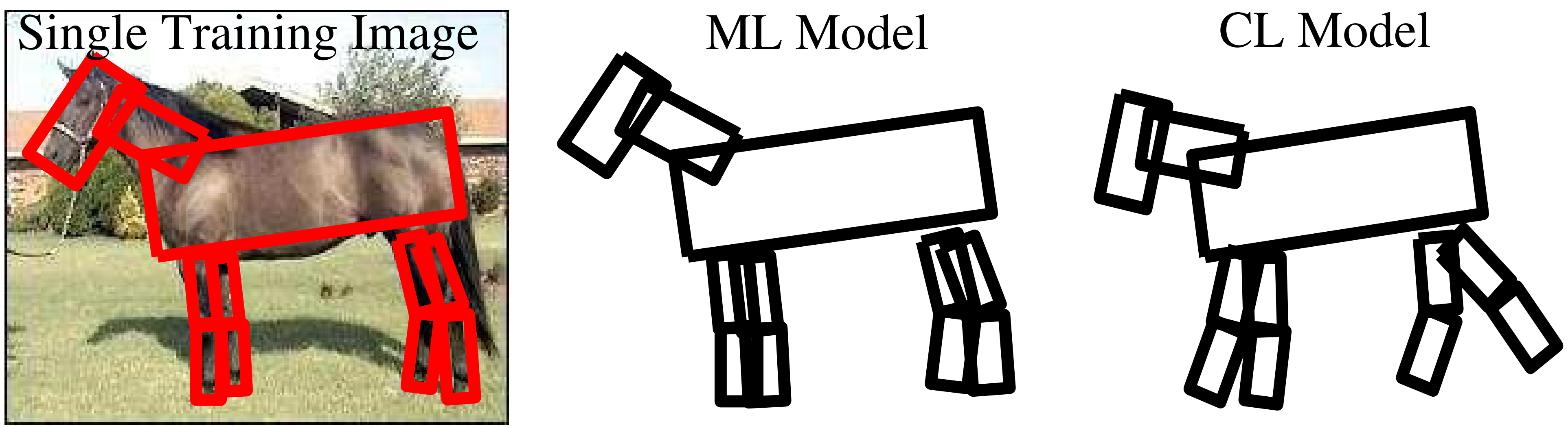
Family resemblences



Andrea Frome's Thesis, 07
"Learning Local Distance Functions for Exemplar Based Object Recognition"

"Categorization of natural objects"
Mervis and Bosch
Annual Review of Psychology 1981

“Caricature”-exemplars



Spatially-weight and geometrically warp each exemplar such that it is easier to recognize

Can be formulated as discriminative learning of a deformable part model given single training image

Ramanan & Sminchisescu, CVPR 06

Growing literature on local distance functions

(c.f. previous talk)

Inspiration for our work:

Hastie and Tibshirani PAMI 96
Atkeson, Moore, & Schaal AI 97
Domeniconi, Peng, & Gunopolus PAMI 02
Zhang, Berg, Maire, & Malik CVPR 06
Frome, Singer, & Malik NIPS 07
Frome, Singer, Sha, & Malik ICCV 07
Davis et al ICML 07
Gonen & Alpaydin ICML 08
Malisiewicz & Efros CVPR 08
Urtasun & Darrell CVPR 08
Babenko, Branson, & Belongie ICCV 09
Malisiewicz & Efros ICCV11

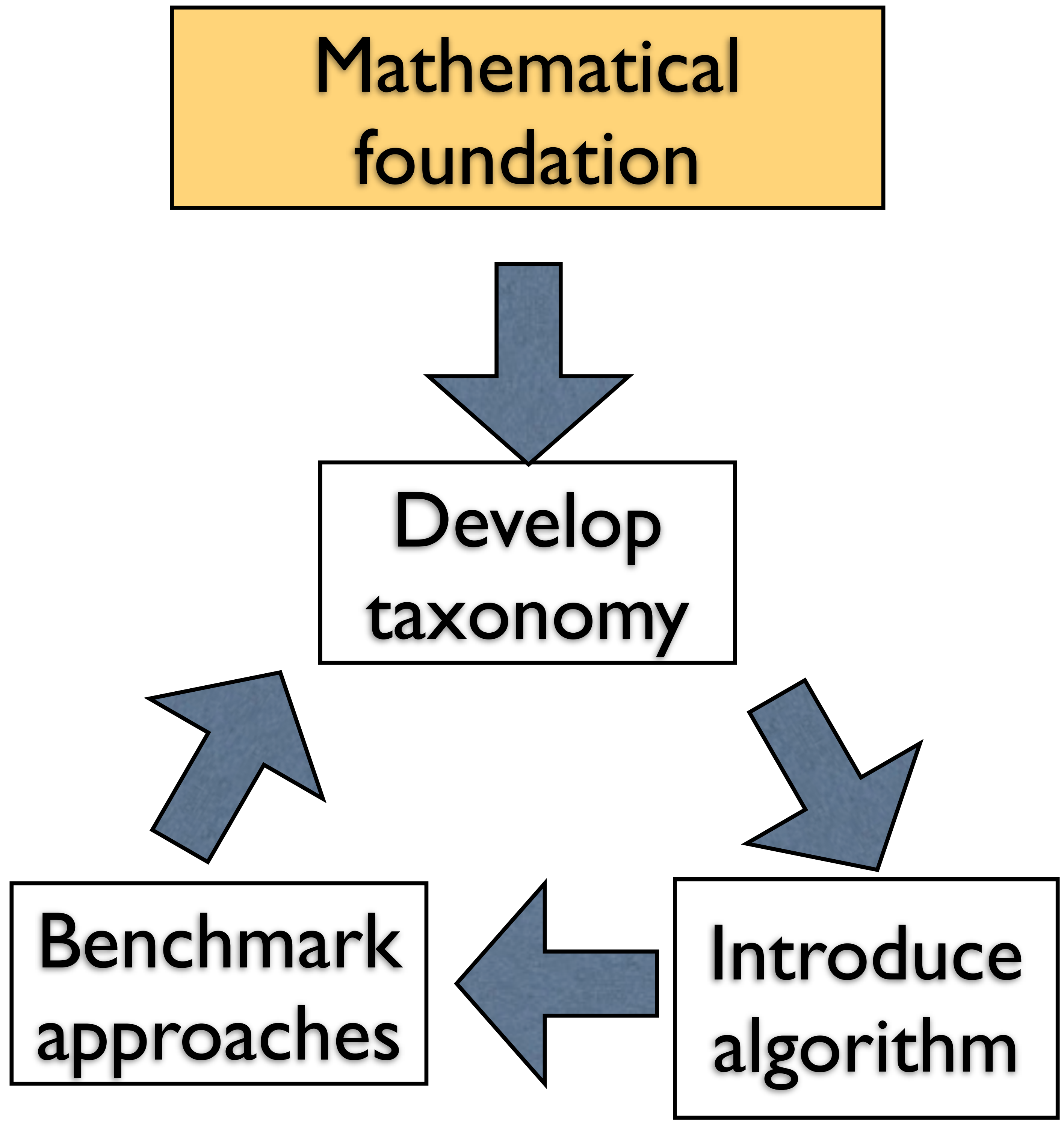
Goals for this talk

1. How do different approaches relate to one another?

2. How do they perform on benchmark vision datasets?

Along the way, we'll develop some new algorithms and test them as well

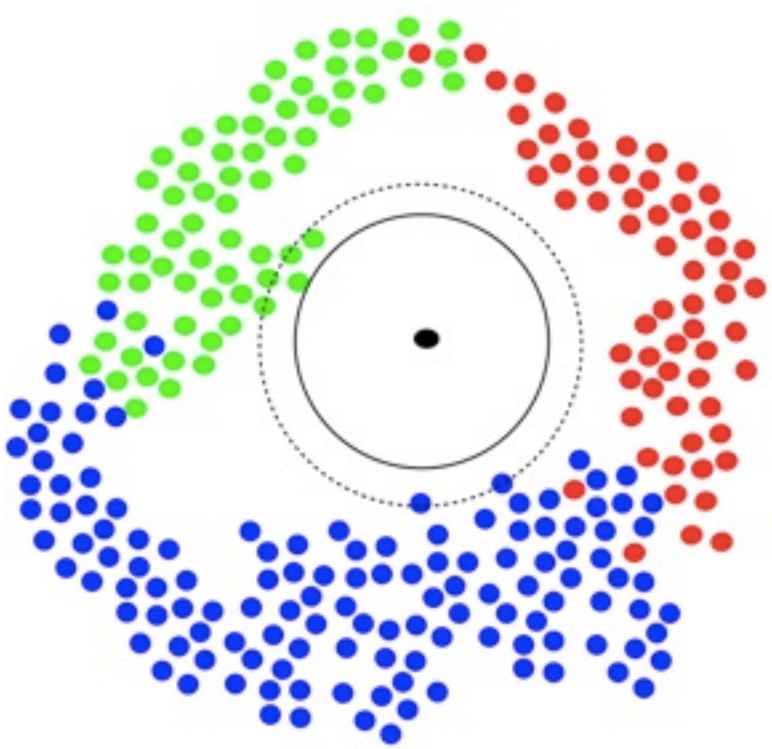
Roadmap



Mathematical foundation
Develop taxonomy
Introduce algorithm
Benchmark approaches

Caveats

We assume data is embedded in finite dimensional vector space



Examples:

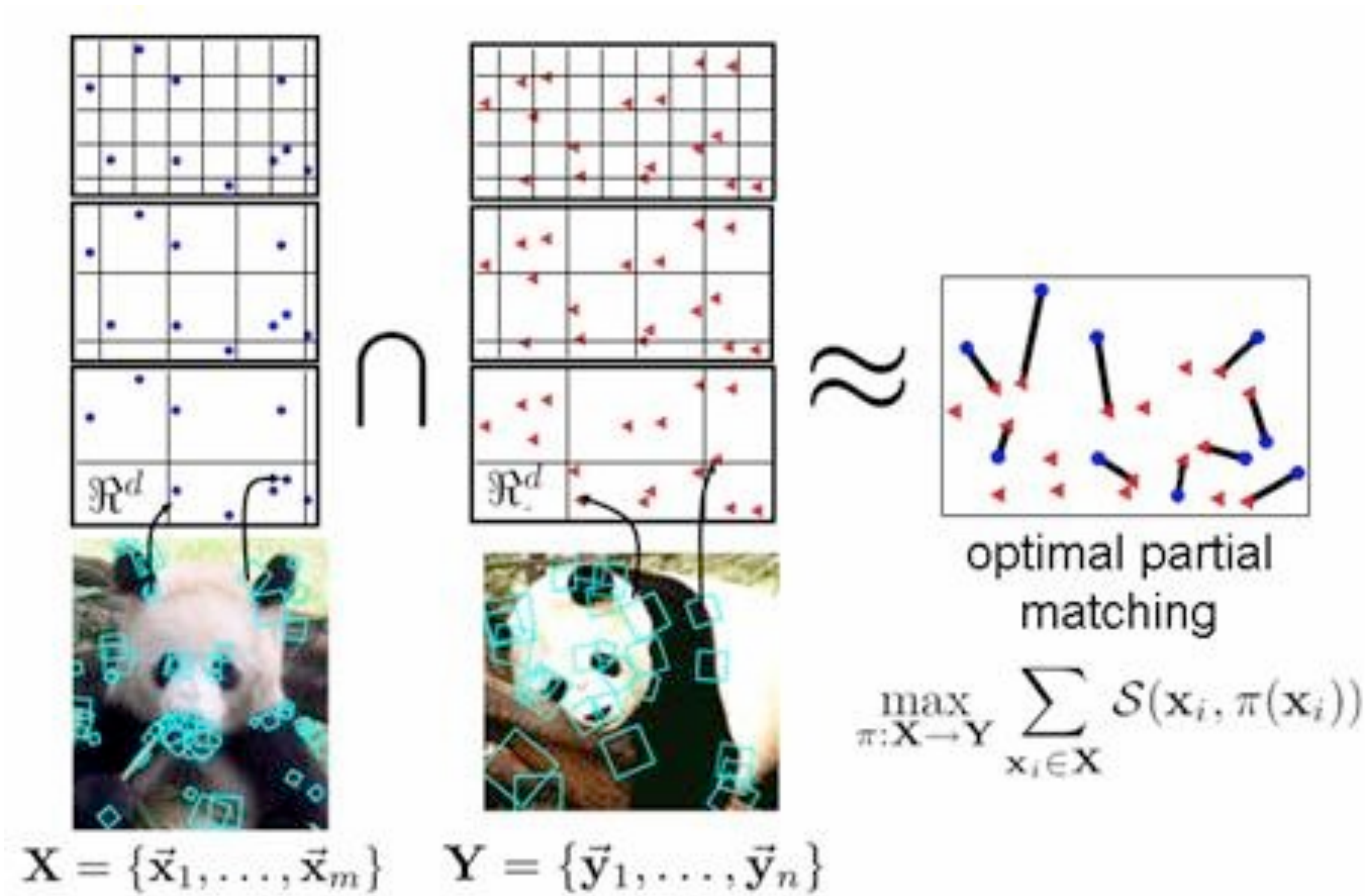
pyramid match kernels (Grauman & Darrell JLMR 07)

intersection kernels (Maji & Berg ICCV 09)

Exceptions:

Gaussian kernels

Approximate correspondence



pyramid match kernels (Grauman & Darrell JLMR 07)

Class-based distance functions

Naive-Bayes Nearest Neighbors (NBNN)

(Boiman et al CVPR 08)
(Tuytelaars et al ICCV11)

1) For each feature in image, find closest matching feature from database of class 'i'
2) x[i] = average distance of features from image to class 'i'
3) Train linear 1-vs-all SVM

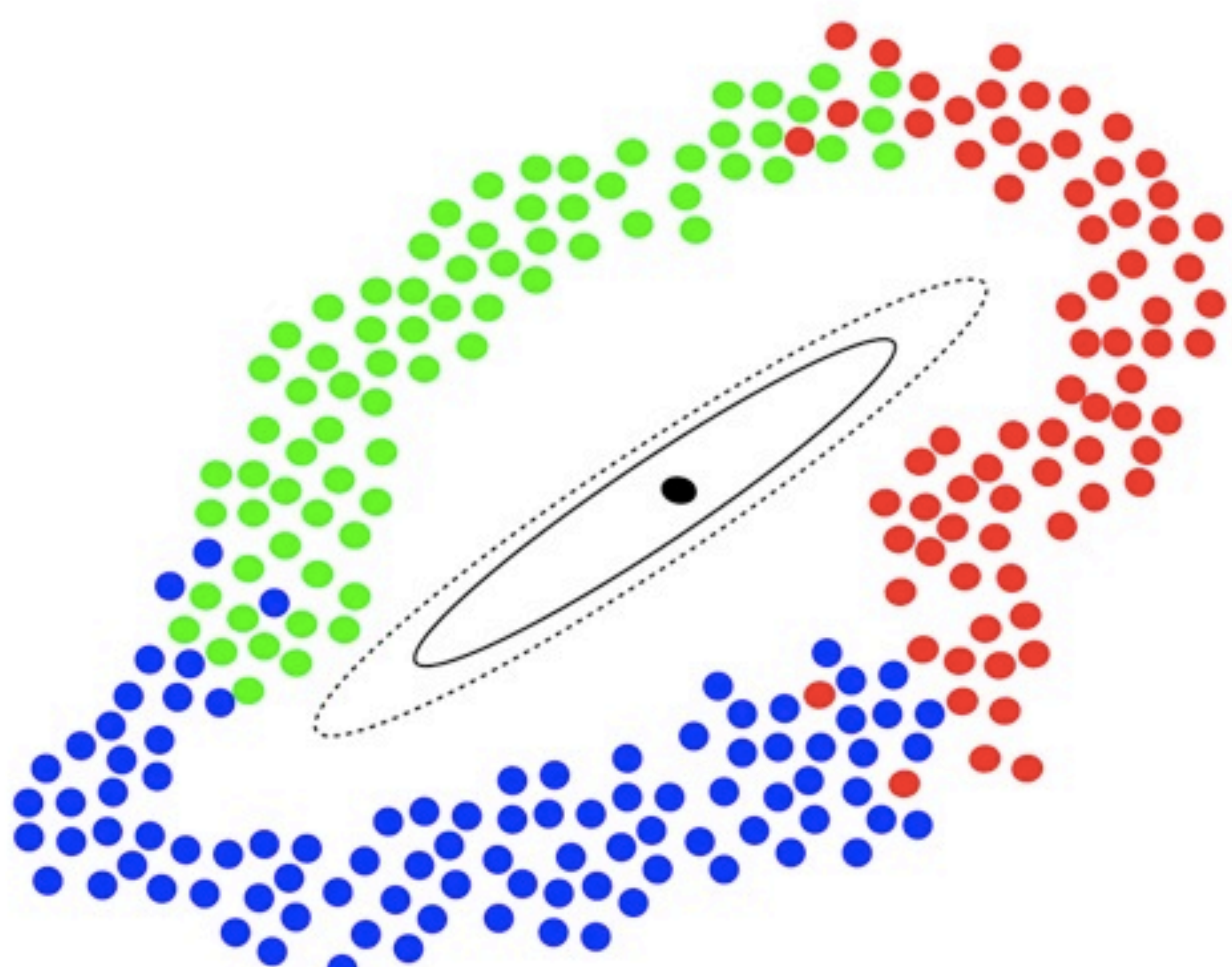
Classeme features (Torresani et al ECCV 2010)

1) Train 1 vs all classifier for base class i
2) x[i] = score of classifier for base class i
3) Train linear (intersection kernel) 1 vs all SVM

Metric representation

$$\mathrm{Dist}(\mathbf{x},\mathbf{y}) = (\mathbf{y}-\mathbf{x})^T M (\mathbf{y}-\mathbf{x})$$

$$\mathbf{x},\mathbf{y} \in R^N \qquad M \in R^{N\times N}$$



Learning a Metric: Large Margin Nearest Neighbors (LMNN)

Weinberger & Saul JMLR 09

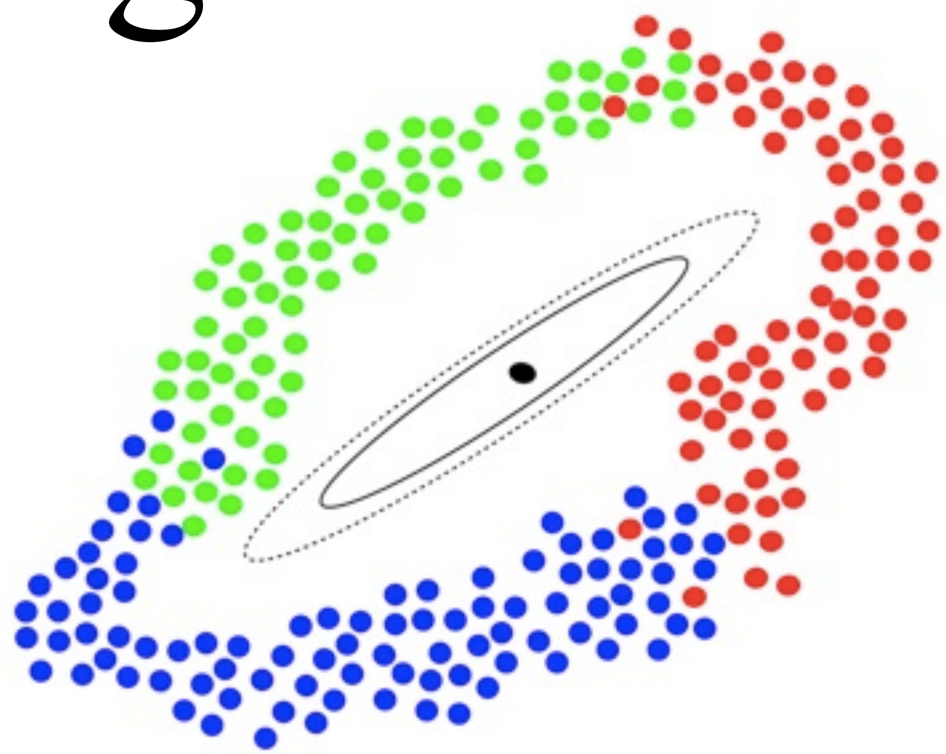
State-of-the-art metric learning algorithm

Given training data $\{x_i, y_i\}$, find M that minimizes a "NN-loss" on that data

1) Discriminative
2) Requires solving a semi-definite program (SDP)
3) Tends to require lots of training data to avoid over-fitting

Code available online

Learning a metric: LDA



Let V be the k-dimensional basis
returned by (regularized) LDA

$$V \in R^{N \times k}$$

$$M = VV^T$$

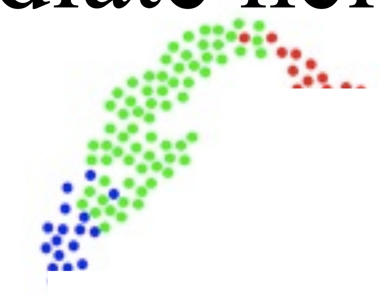
1) Generative
2) Requires solving an eigenvalue problem (relatively fast)
3) Straightforward to regularize (add diagonal to within-class covariance matrix)

How do we learn local metrics?

Use a subset of the training data!

e.g.

1) Look at all points from a particular class and their immediate neighbors

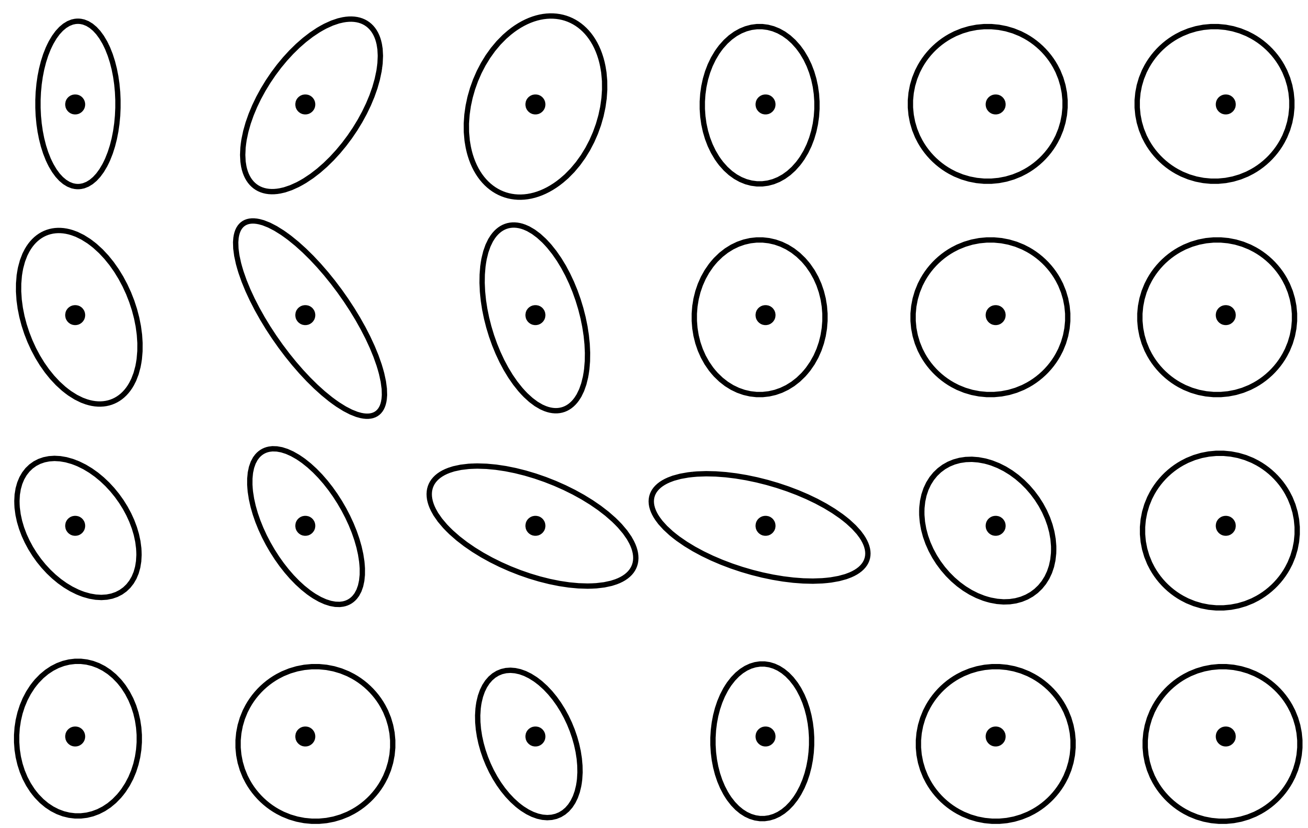


2) Look at immediate neighbors of a point

Mathematical framework I: metric tensor

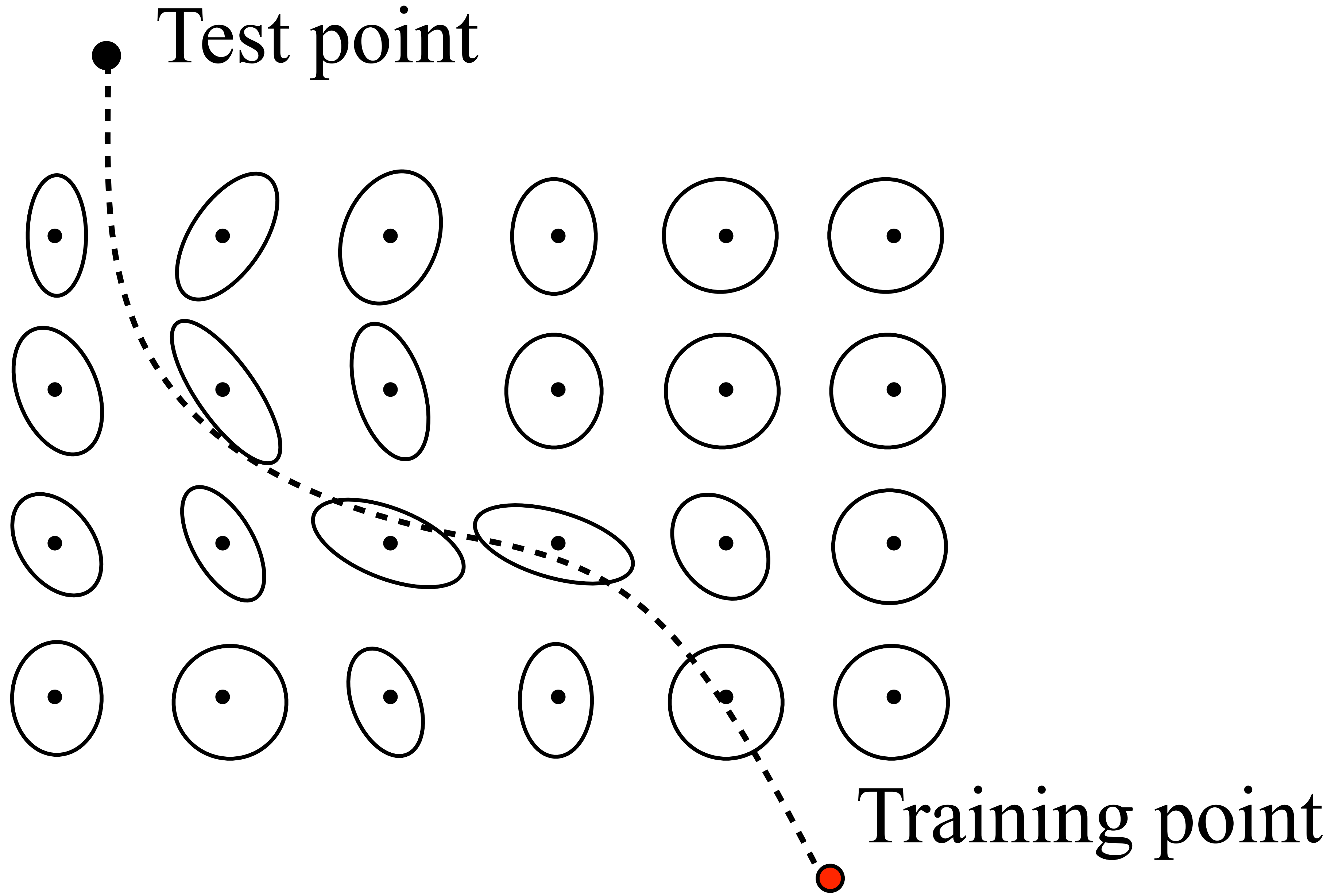
$$MT(\mathbf{x}) \in R^{N \times N}$$

$$\mathbf{x} \in R^N$$



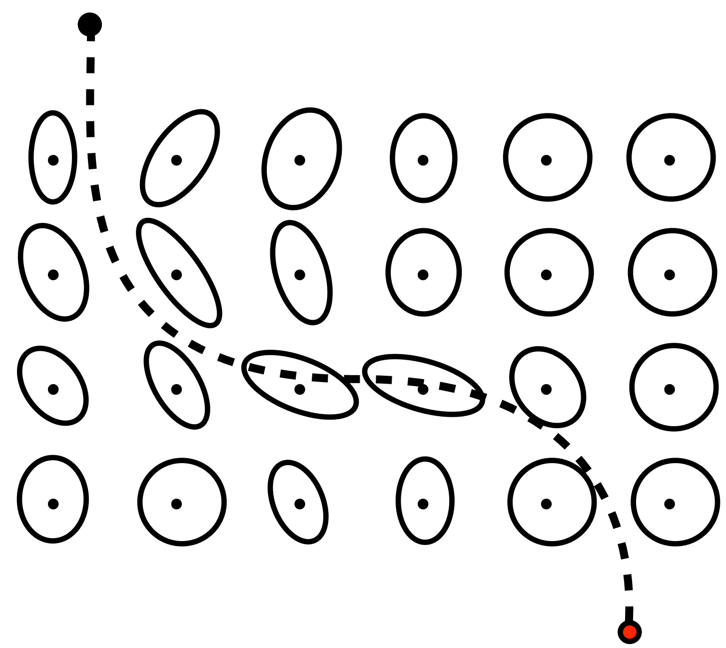
Metric varies over R^N

thematical framework II: geodesic distances



1. The length of curve **c** = integral of MT(**x**) along **c**
2. Dist(**x**,**y**) = minimum length curve from **x** to **y**

Approximations

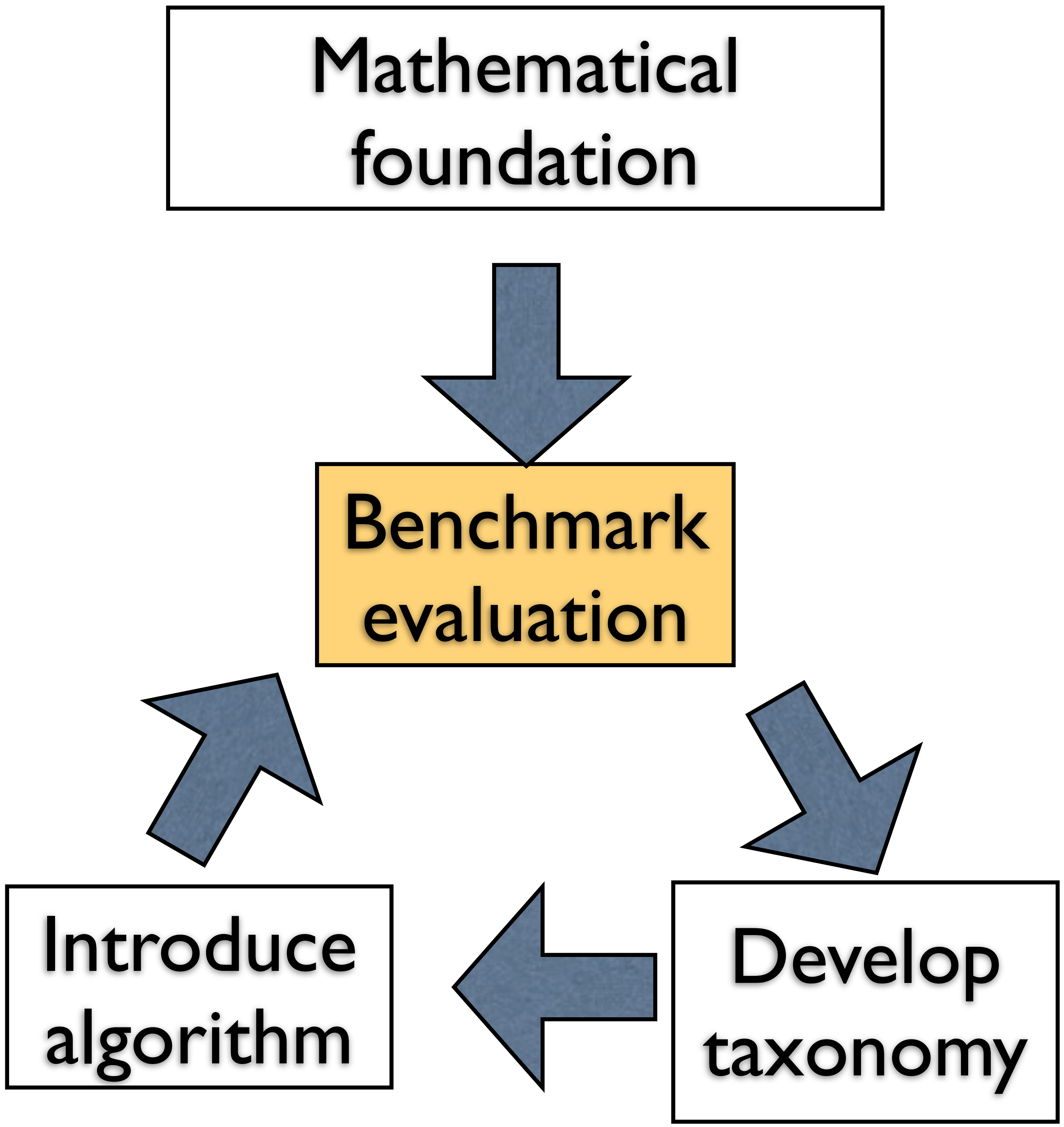


1. Geodesic distances cannot be efficiently computed for large dim.
   Use distance transform for small dimensions:

Toivanen 96
Yatsiz et al. 06
Criminisi et al. ECCV 08

2. We will present a taxonomy of local distance functions in terms of when, where, and how they approximate MT(**x**)

Roadmap


Mathematical foundation
Benchmark evaluation
Introduce algorithm
Develop taxonomy

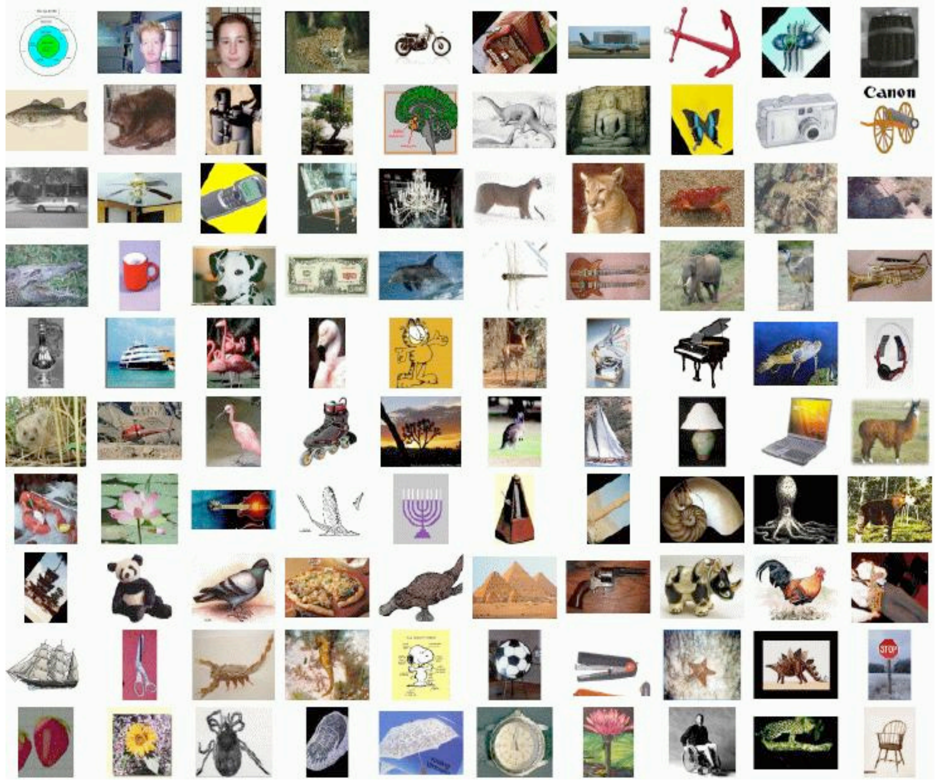
Benchmark data

Multi-PIE (700K images, 300 classes)
eigenface features

Gross et al. FG08

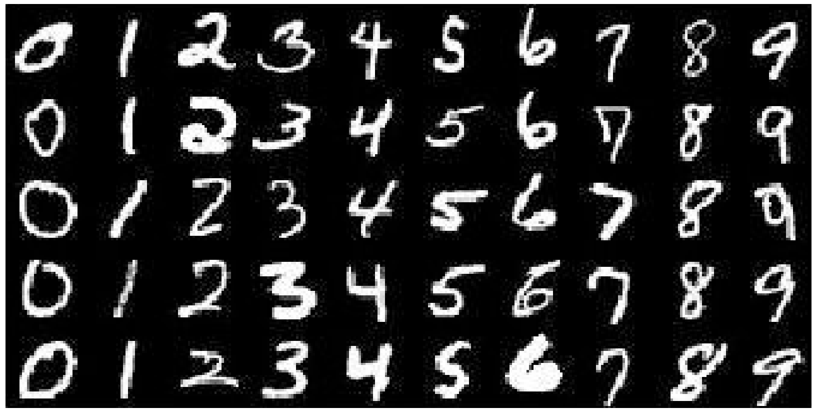
Caltech 101 (9K images, 101 classes)
spatial pyramid features

Fei Fei et al. GMBV04,
Lazebnik et al. CVPR06



MNIST (70K images, 10 classes)
normalized pixel features

Lecun & Cortes



Reporting results

We’ll give a taste of the results here

http://www.ics.uci.edu/~dramanan/localdist/index.html

(includes computed features for download)

"Local Distance Functions: A Taxonomy, New Aglorithms, and an Evaluation"

We present a taxonomy for local distance functions where most existing algorithms can be regarded as approximations of the geodesic distance defined by a metric tensor. We categorize existing algorithms by **how**, **where**, and **when** they estimate the metric tensor. We also extend the taxonomy along each axis. **How**: We introduce hybrid algorithms that use a combination of techniques to ameliorate over-fitting. **Where**: We present an exact polynomial time algorithm to integrate the metric tensor along the lines between the test and training points under the assumption that the metric tensor is piecewise constant. **When**: We propose an interpolation algorithm where the metric tensor is sampled at a number of references points during the offline phase. The reference points are then interpolated during the online classification phase. We also present a comprehensive evaluation on tasks in face recognition, object recognition, and digit recognition.

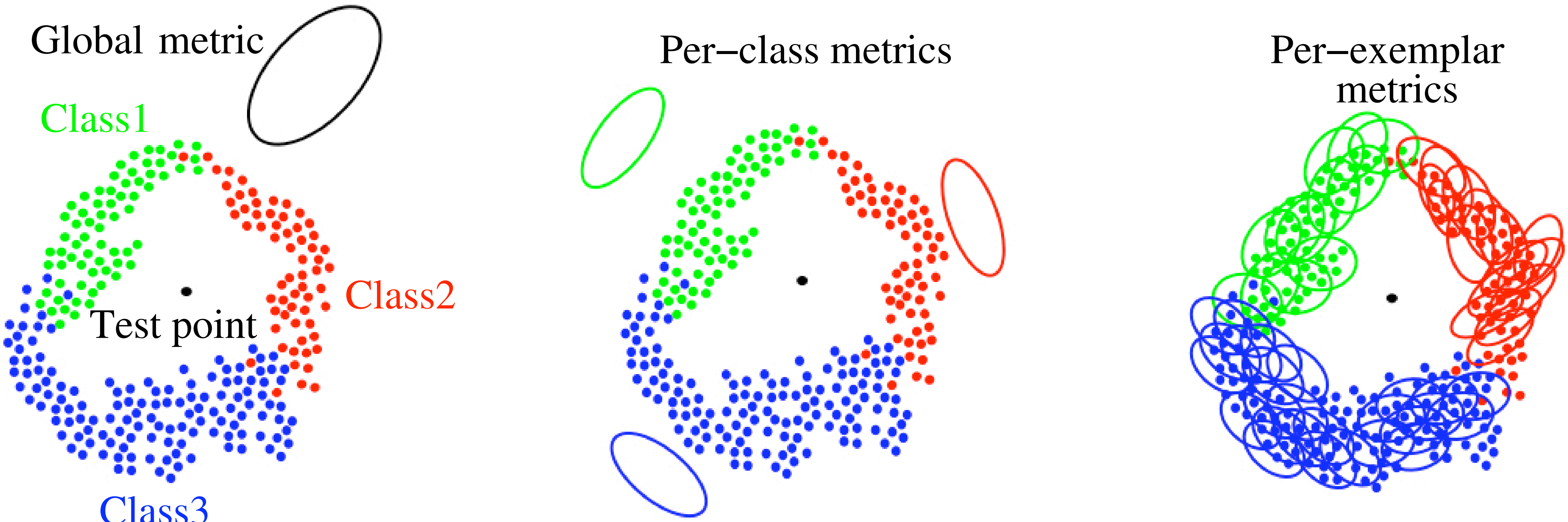
D. Ramanan, S. Baker. **"Local Distance Functions: A Taxonomy, New Algorithms, and an Evaluation"** *International Conference on Computer Vision* (ICCV) Kyoto, Japan, Sept. 2009.

PDF Supplementary Results

References for datasets and feature vector construction

Where

$$\mathrm{Dist}_M(\mathbf{x},\mathbf{y}) = (\mathbf{y}-\mathbf{x})^T M(\mathbf{y}-\mathbf{x})$$



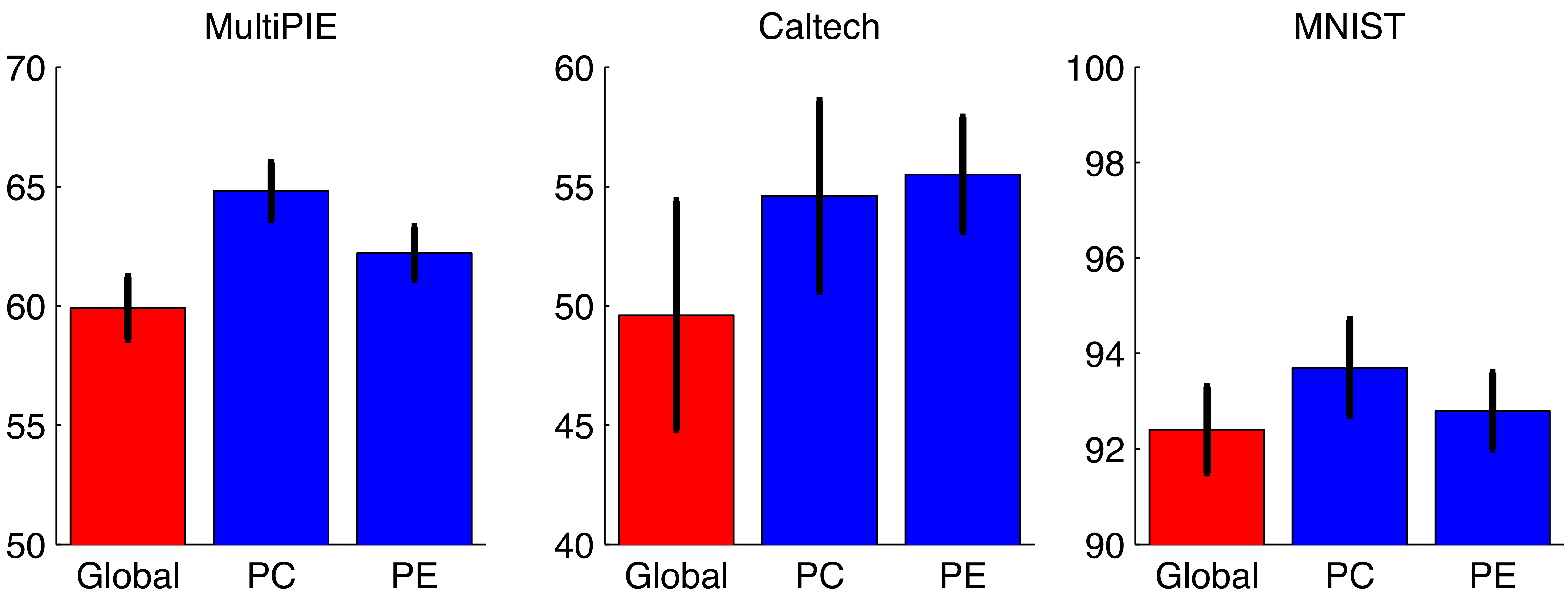
Weinberger & Saul. JMLR 09

Frome, Singer, & Malik NIPS 07
Frome et al. ICCV 07

Per-class (PC): Use class-specific metric when computing distance to training example

Per-exemplar (PE): Use example-specific metric when computing distance to training example

Where

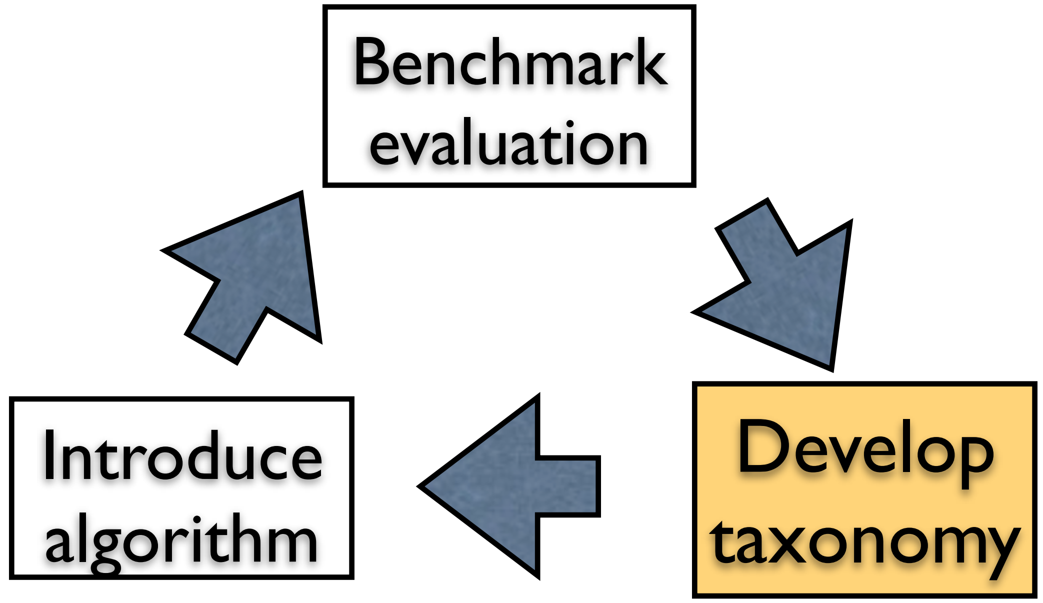


Conclusion: Per Exemplar (PE) does not always perform better then Per Class (PC)

Hypothesis: Multiple metrics can over-fit and require normalization

Joint training of metrics is hard: Frome et al ICCV 07, Weinberger & Saul JMLR 09

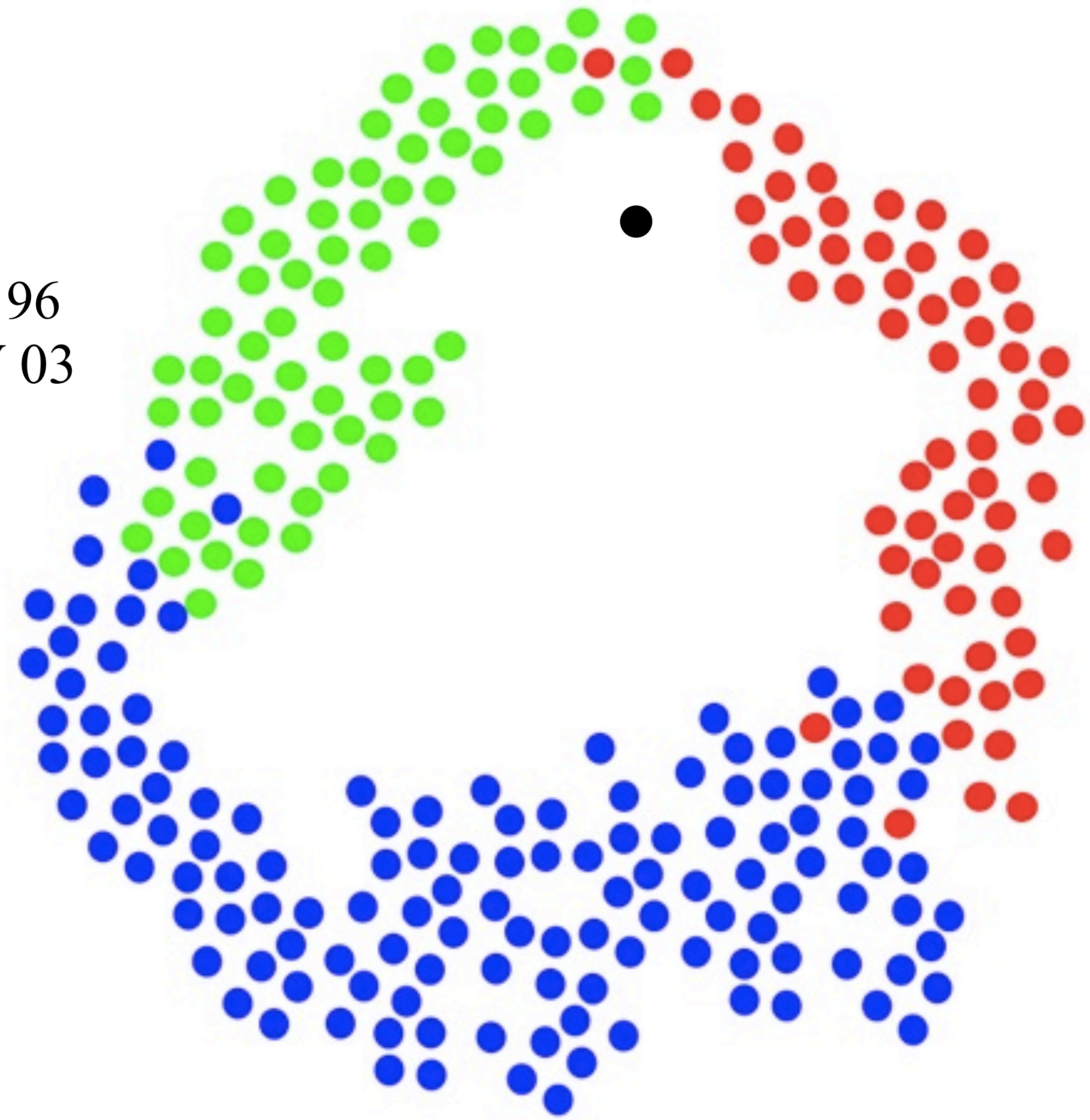
Taxonomy

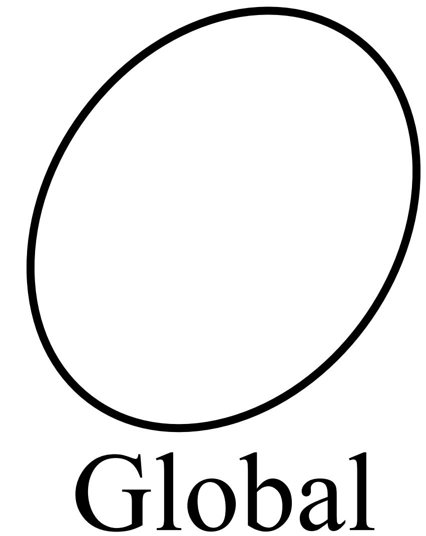


Where	How	When
Global	LDA	Offline
Per-Class	LMNN	Online
Per-Exemplar	Hybrid	Interp
Lazy		
Line		

Lazy learning: Estimate metric at test point

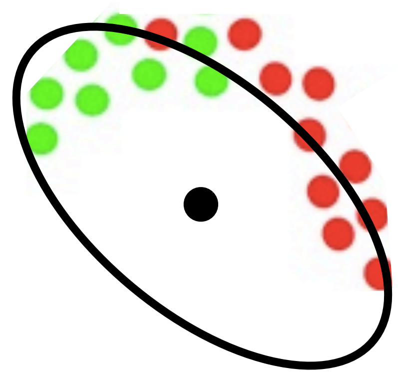
Atkeson et al. 97
Hastie & Tibshirani PAMI 96
Shaknarovich et al. ICCV 03
Zhang et al. CVPR06





Build initial short list of neighbors using global metric

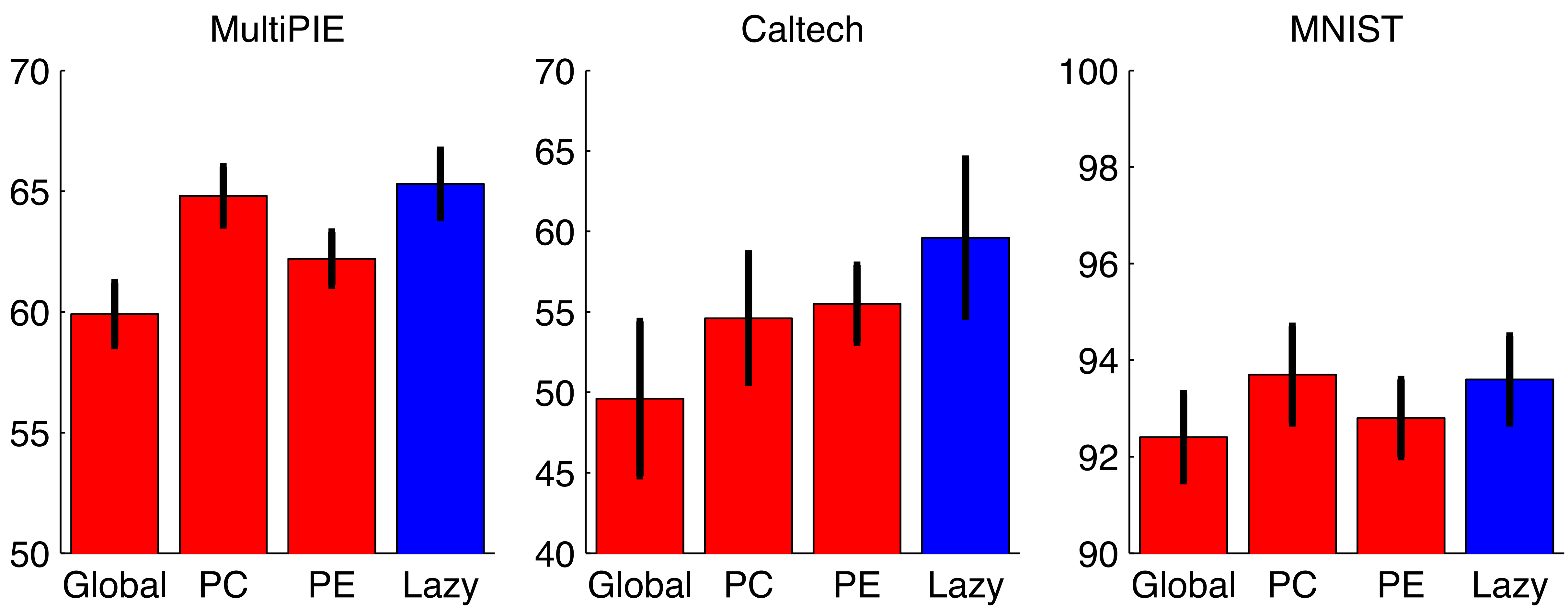
Lazy learning: Estimate metric at test point



Atkeson et al. 97
Hastie & Tibshirani PAMI 96
Shaknarovich et al. ICCV 03
Zhang et al. CVPR06

1. Build initial short list of neighbors using global metric
2. Learn local metric from neighbors
3. Re-rank neighbors using local metric

Lazy learning

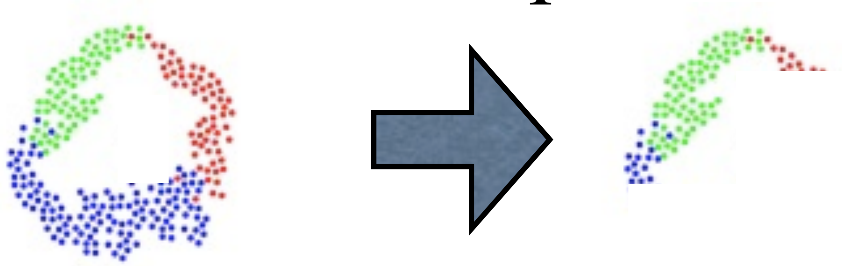


Conclusion: Lazy-learning consistently performs well

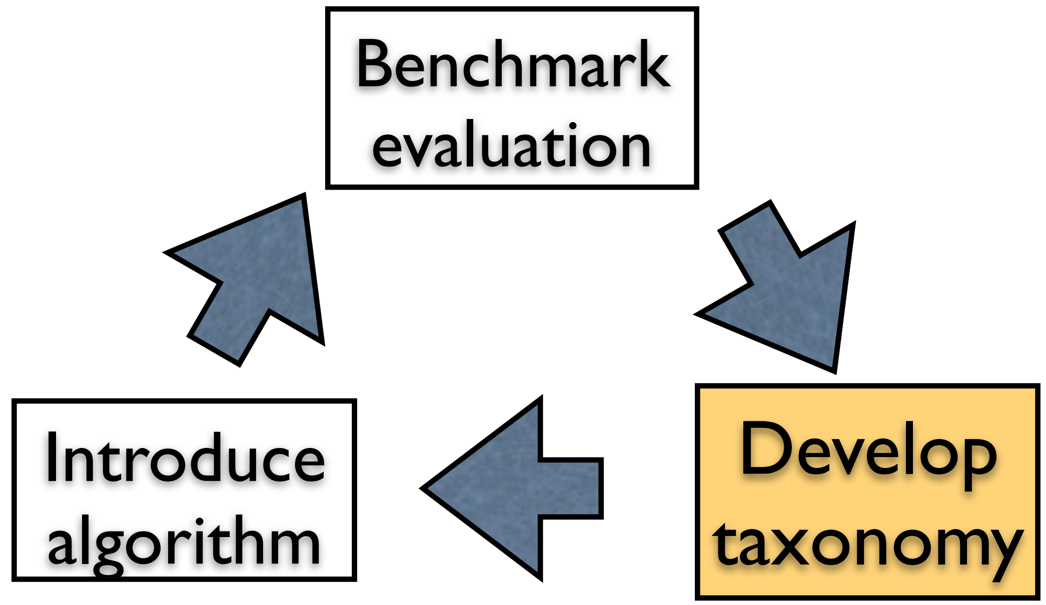
Hypothesis: We avoid over-fitting + normalization issues by estimating metric at test point

Why does 'Lazy' work well?

Local metric learned from 50-NN
tailored to specific expression and pose

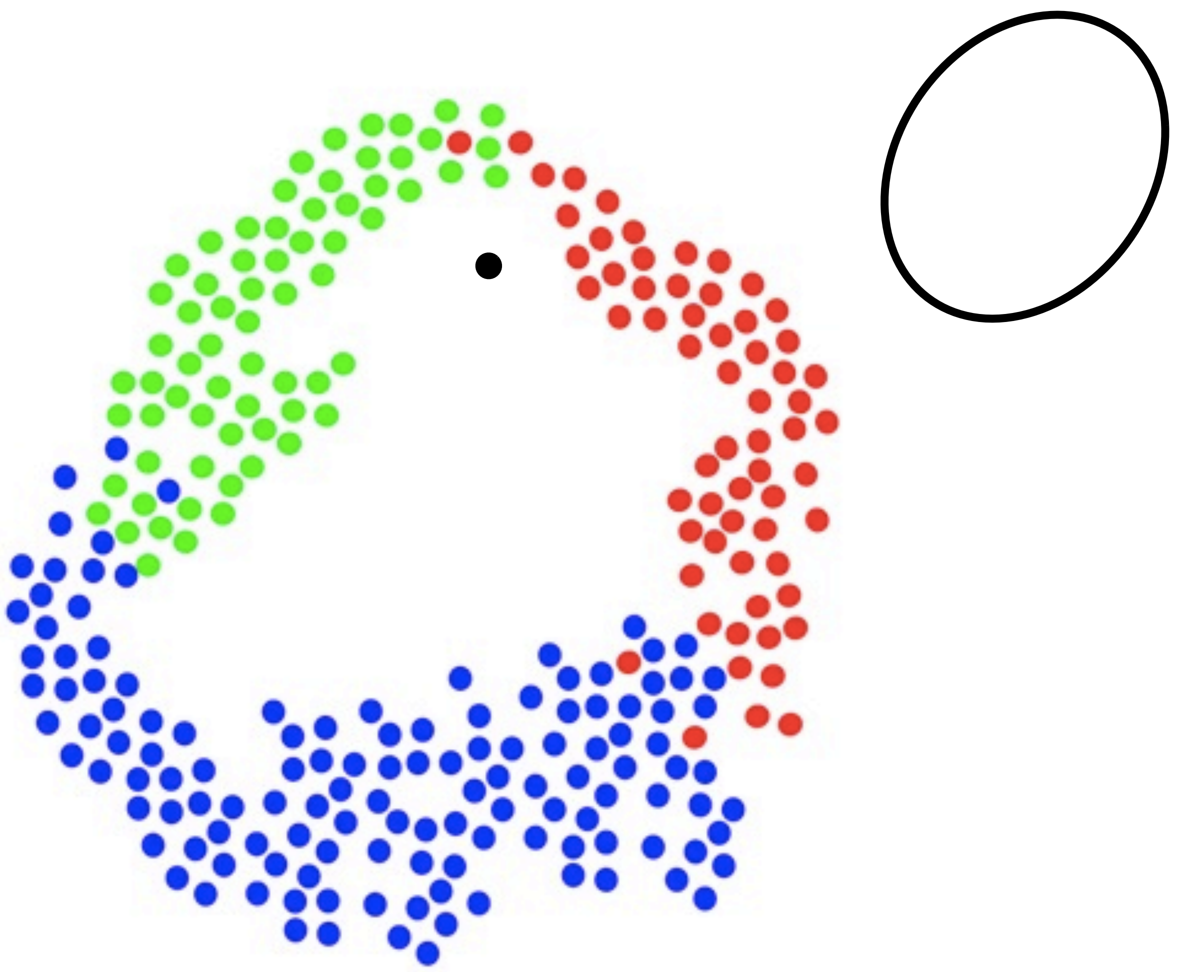


Taxonomy

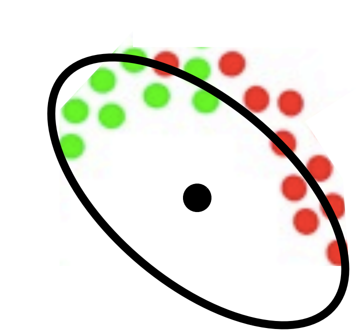


Where	**How**	**When**
Global	LDA	Offline
Per-Class	LMNN	Online
Per-Exemplar	Hybrid	Interp
Lazy		
Line		

How



Global LMNN

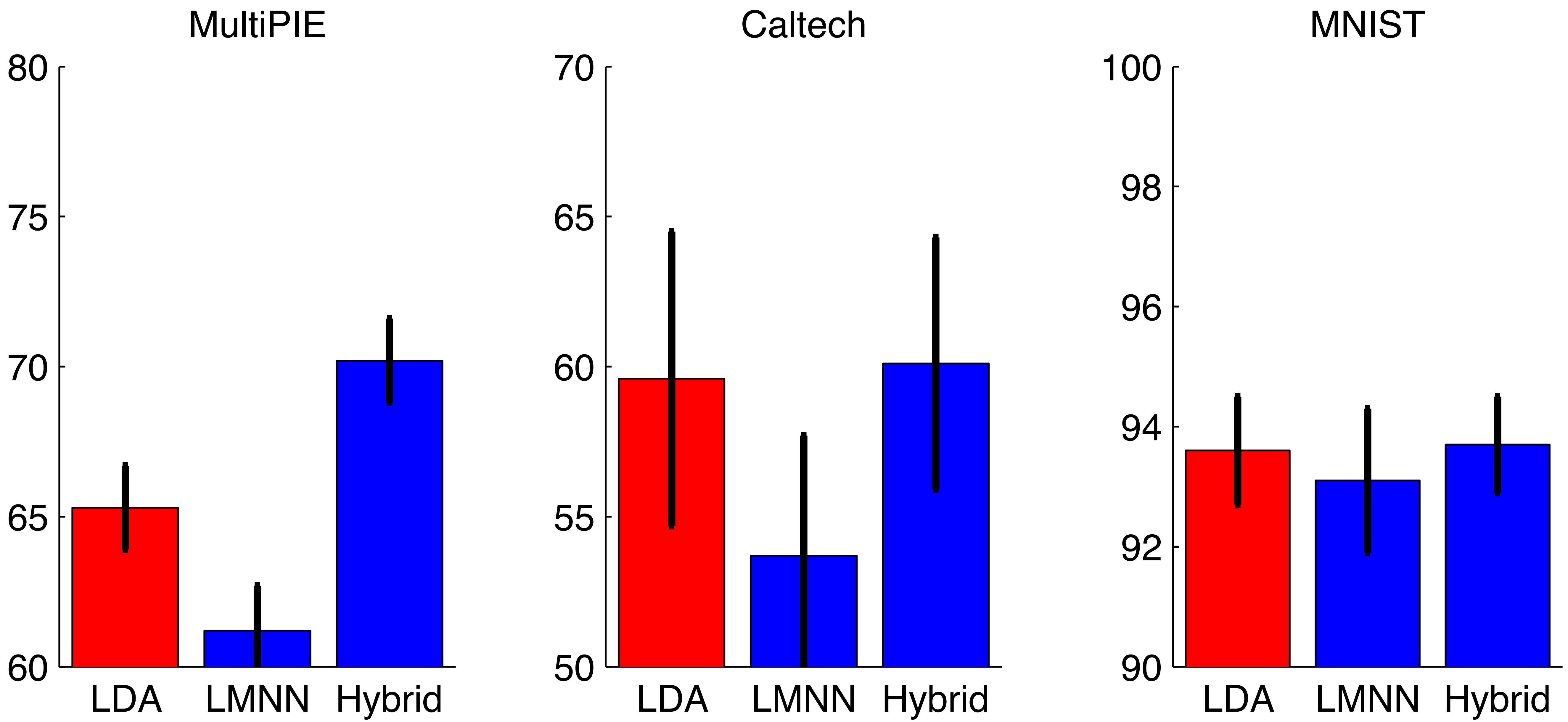


Local LDA

Hybrid learning of metric

1) Learn Global metric with LMNN (more training data)
2) Learn Local metric with LDA (easier to regularize)

How

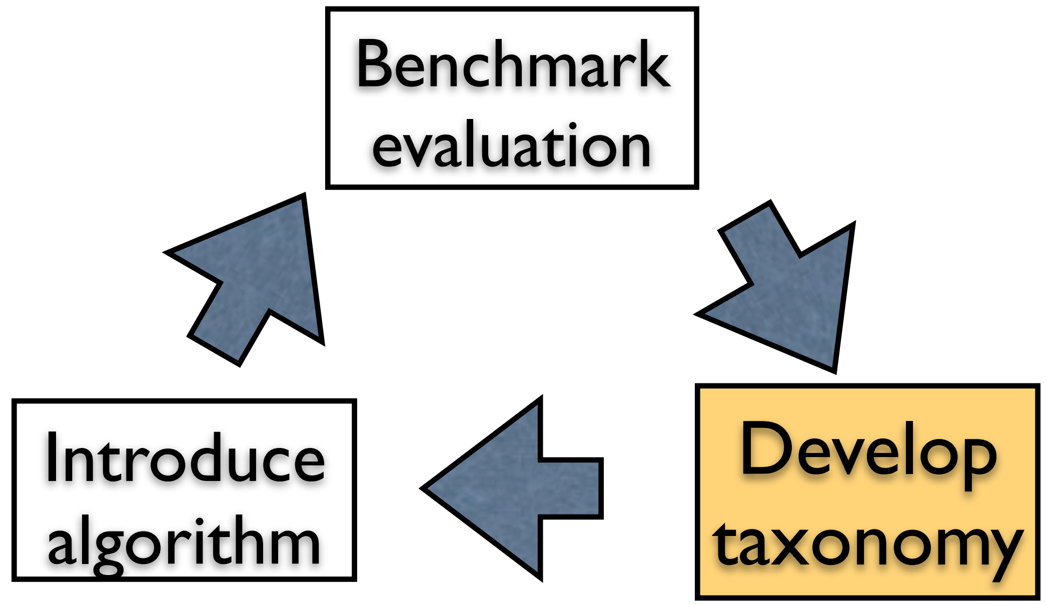


Conclusion: Global LMNN + local LDA exploits strengths of both

LMNN works well when trained on lots of data

LDA easier to regularize

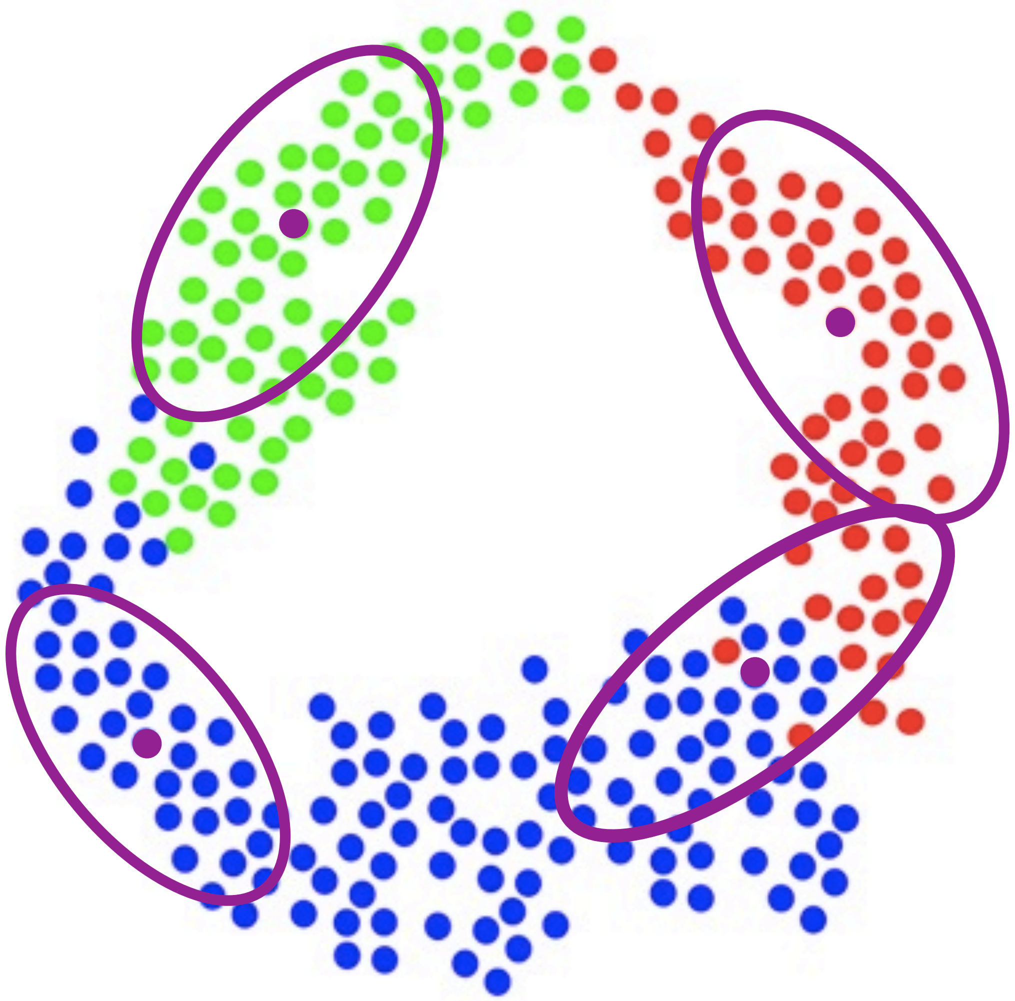
Taxonomy



Where	**How**	**When**
Global	LDA	Offline
Per-Class	LMNN	Online
Per-Exemplar	Hybrid	Interp
Lazy		
Line		

(novel algorithms in red)

When

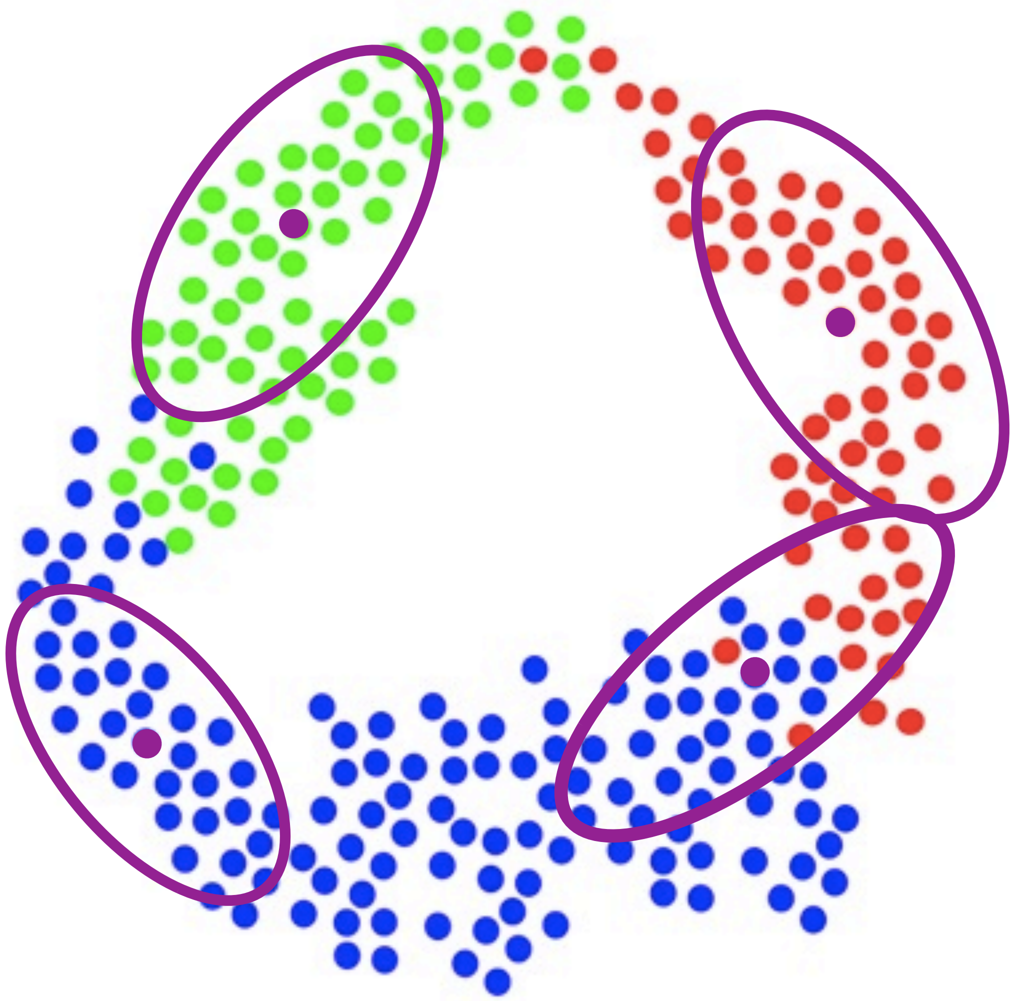


Offline
Learn metrics at reference points

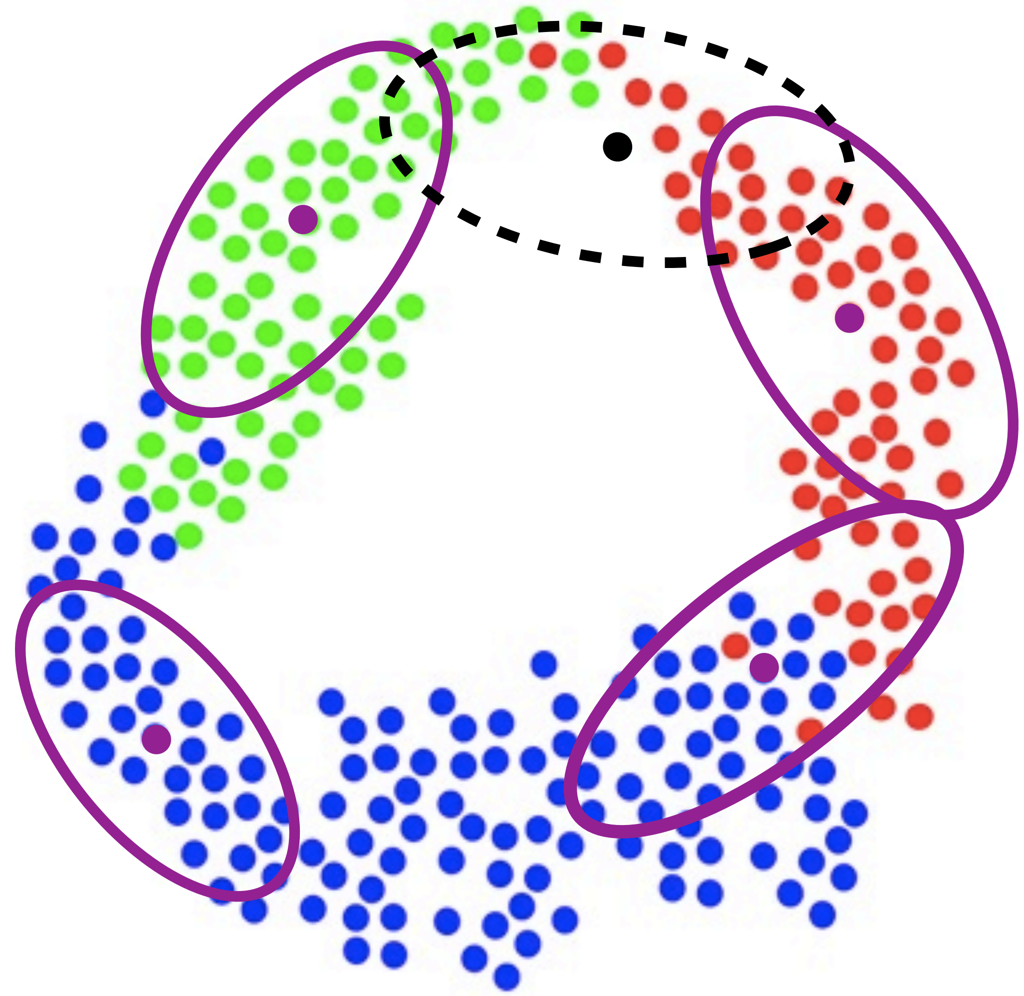
Interpolated metrics

Shift computational load of lazy algorithm off-line

When



Offline
Learn metrics at reference points

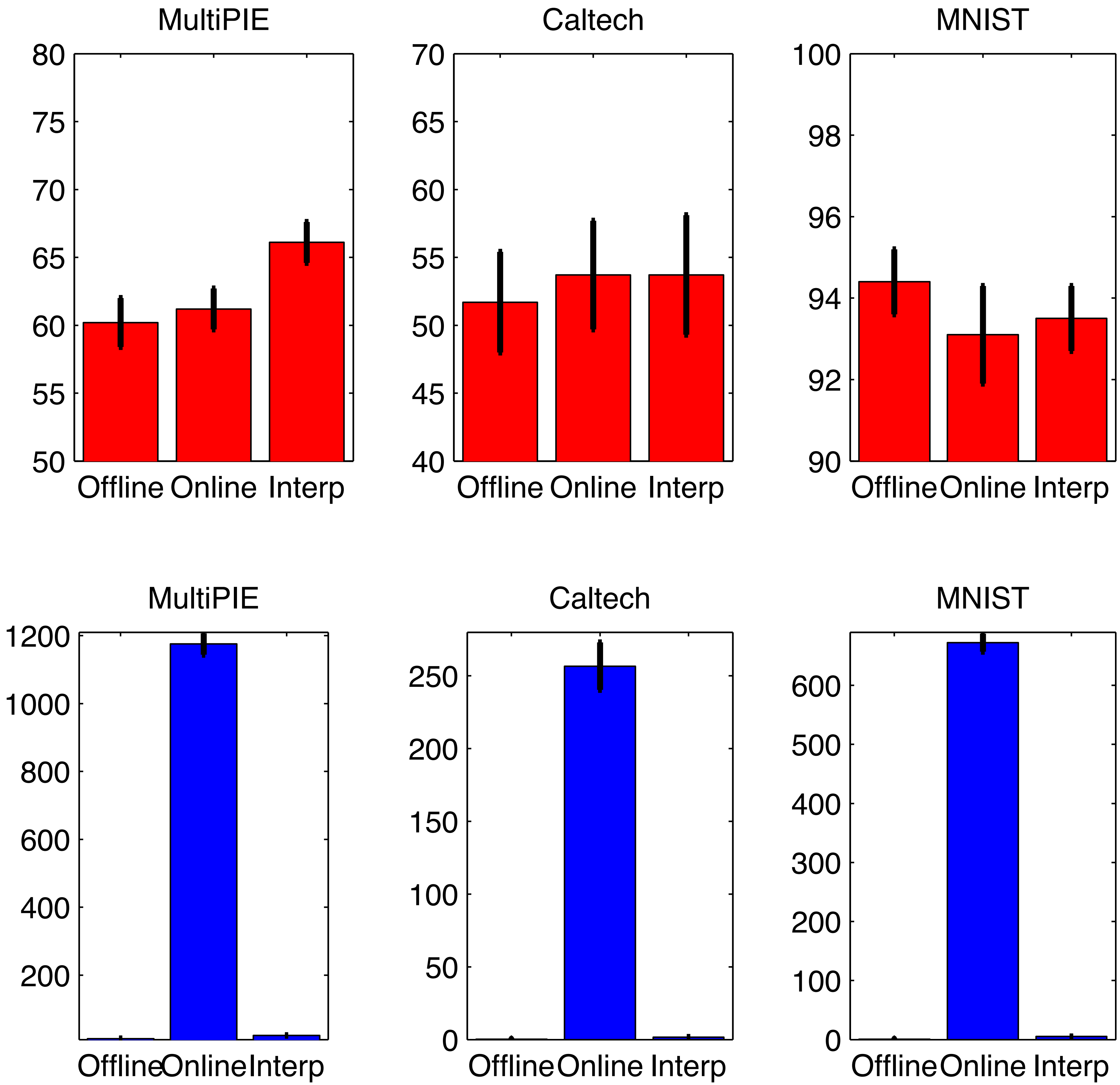


Online
Interpolate metrics at test point

Interpolated metrics

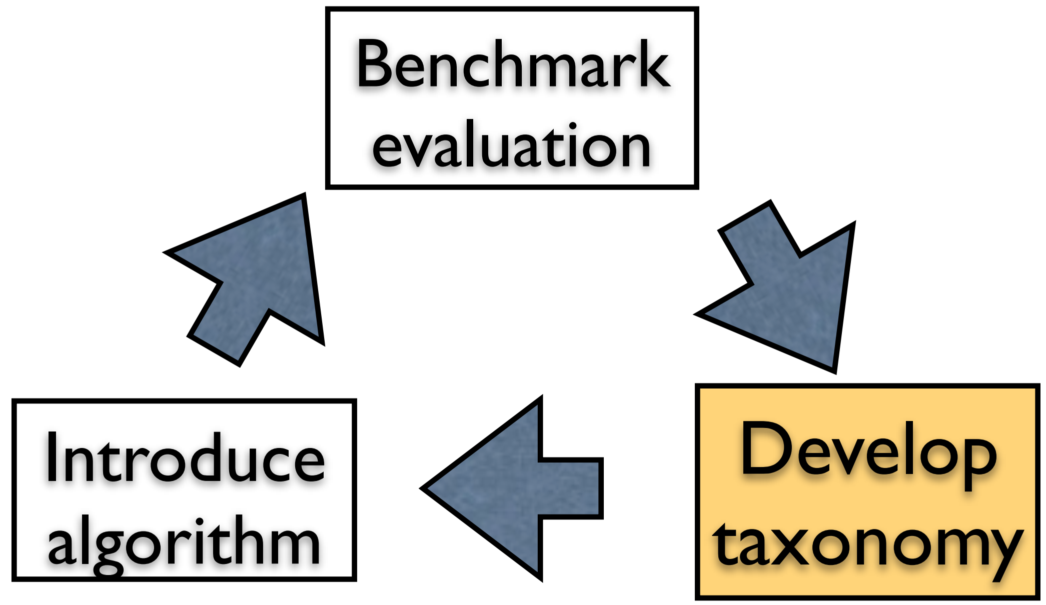
Shift computational load of lazy algorithm off-line

Metric Learning Recognition Rate



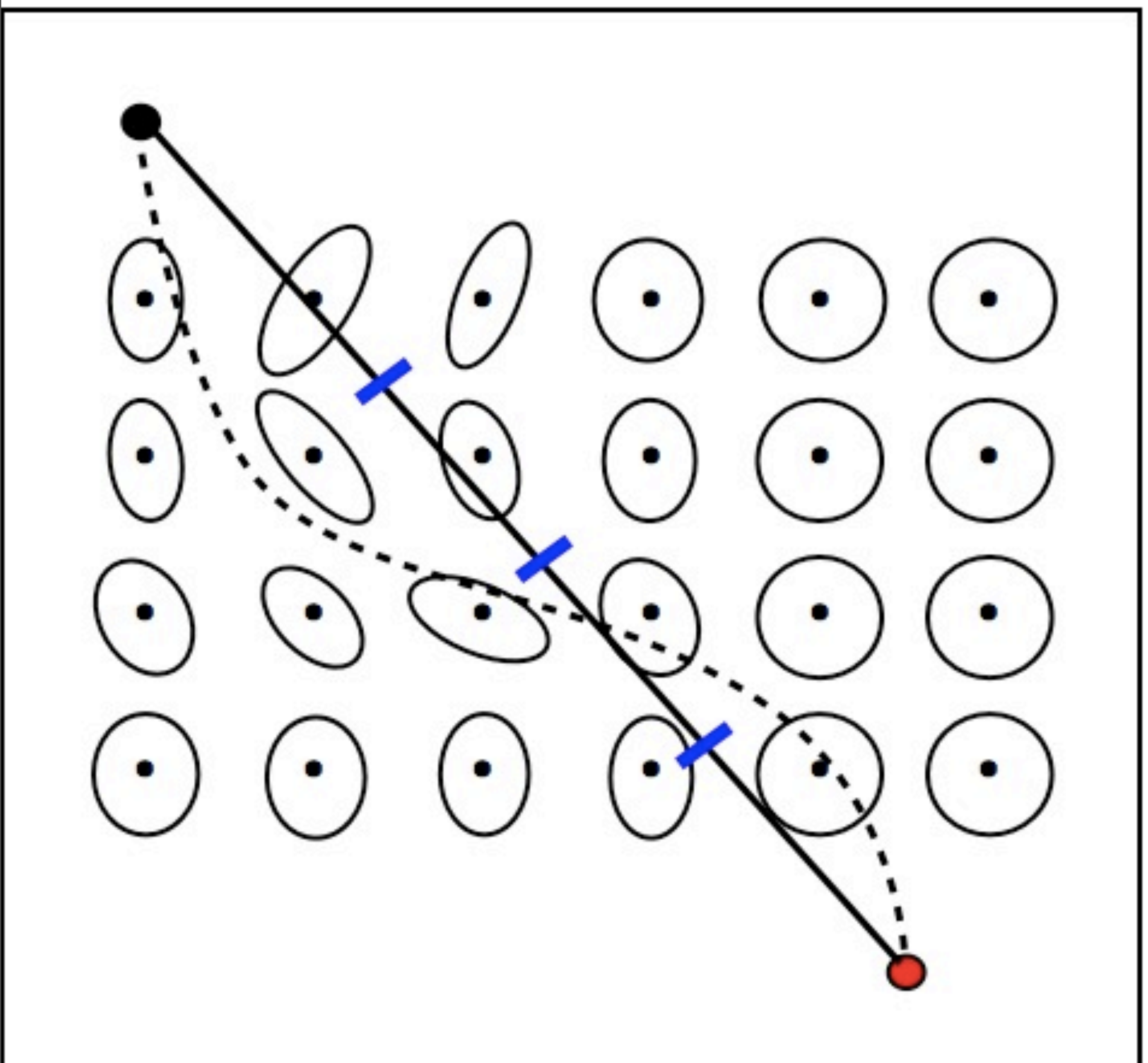
Conclusion: Interpolated metrics often perform similarly to online lazy-approaches, but are much faster.

Taxonomy

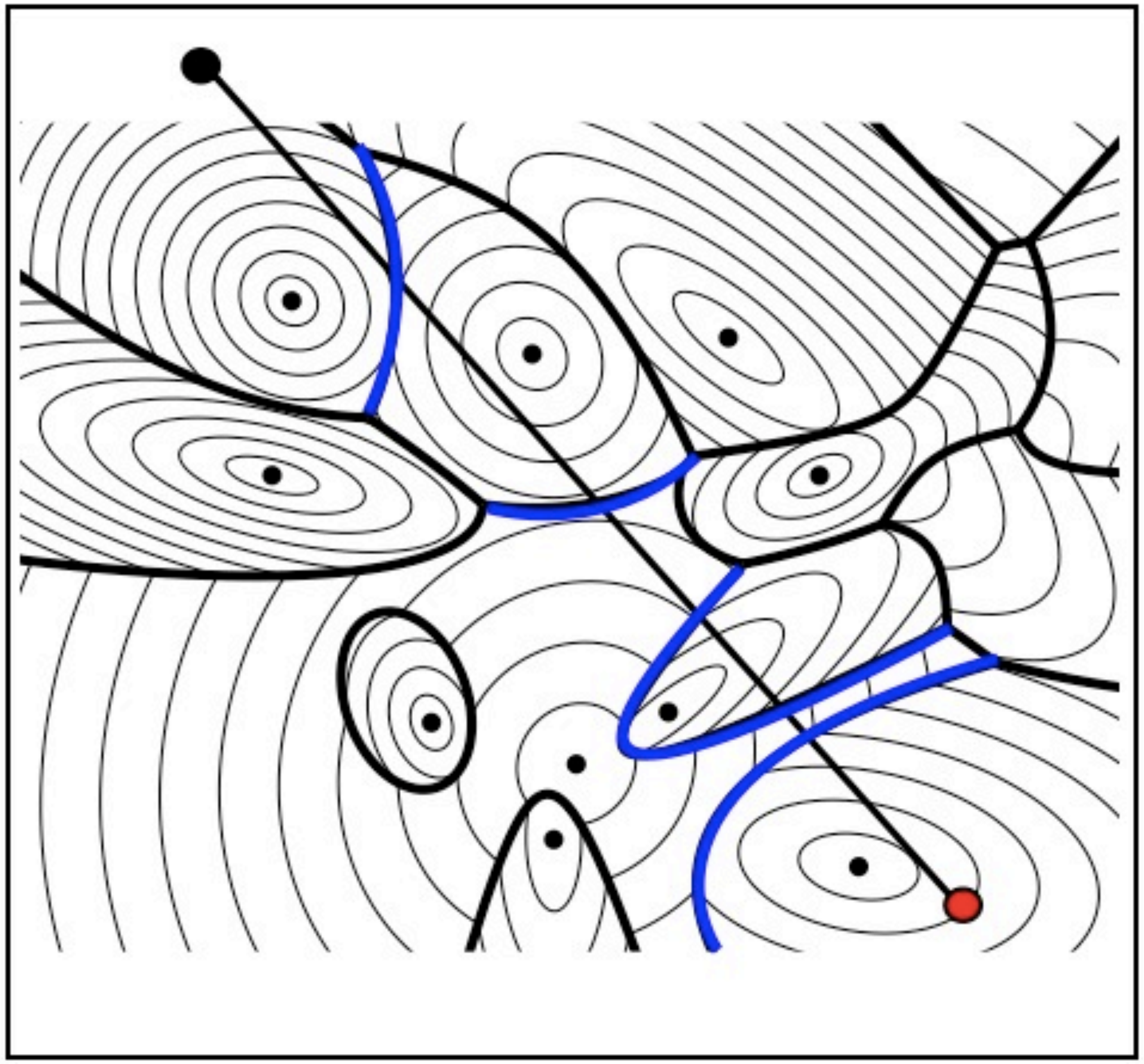


Where	**How**	**When**
Global	LDA	Offline
Per-Class	LMNN	Online
Per-Exemplar	Hybrid	Interp
Lazy		
Line		

Where: Line integral distance



Straight line approx. of geodesic distance

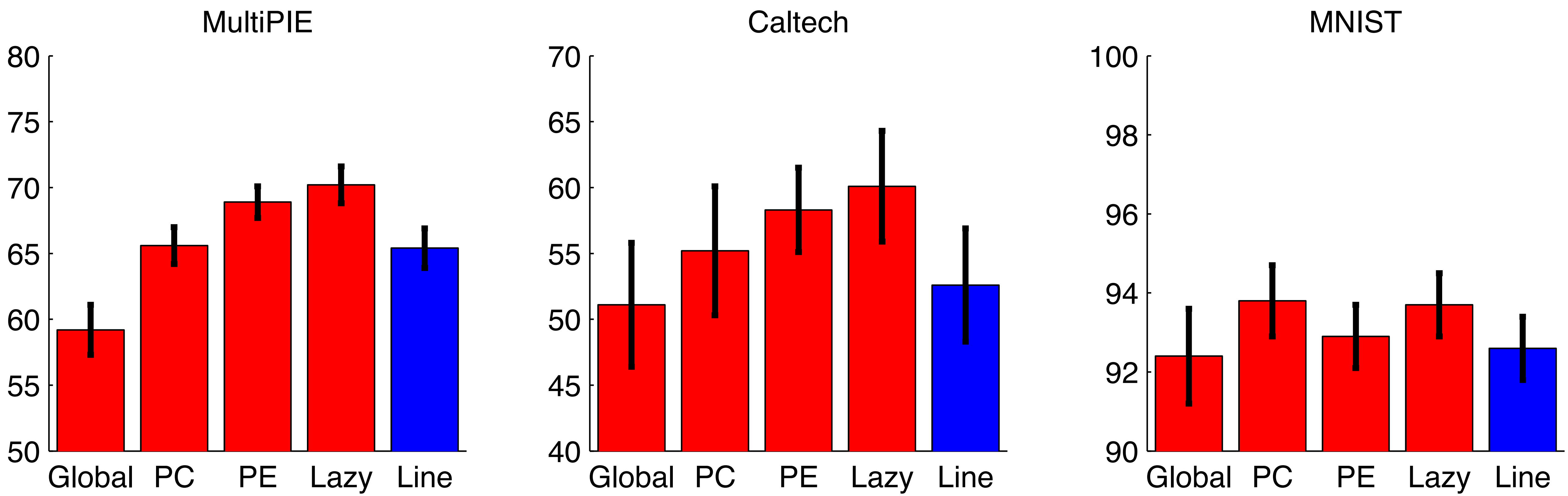


Anisotropic Voronoi diagram
Labelle & Shewchuk SOCG 03

Metric tensor is piecewise constant along line

One can compute transition points in polynomial time

Line integral with hybrid metrics



Line integral better than Global but not as good as Lazy or Per-Class

Review of taxonomy

Where	**How**	**When**
Global	LDA	Offline
Per-Class	LMNN	Online
Per-Exemplar	Hybrid	Interp
Lazy		
Line		

Taxonomy of local distance functions based on how, where, and when they estimate metric tensor

Evaluation on 3 large-scale diverse datasets

Conclusions

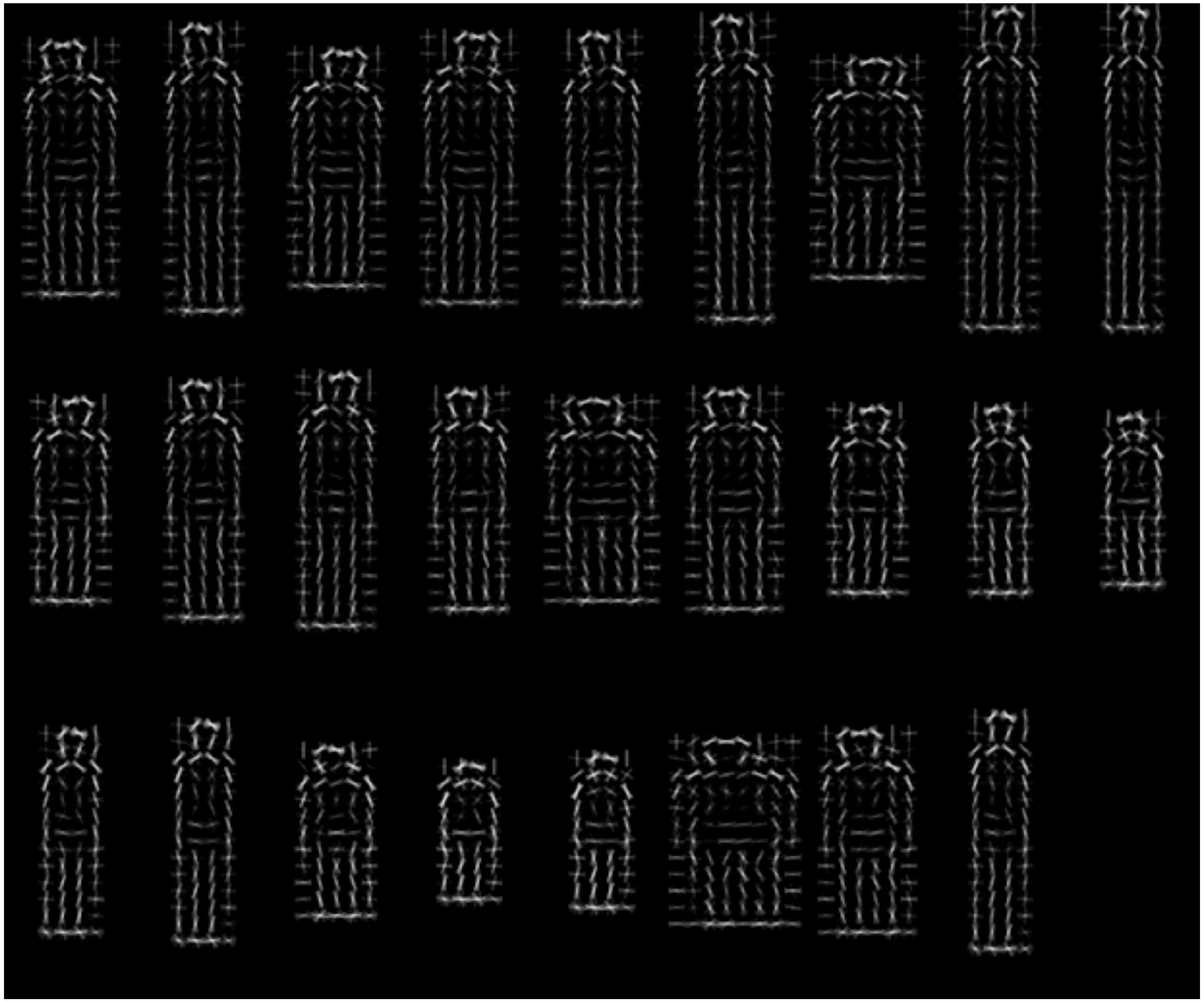
Overall, Lazy algorithms performed the best, but high run-time cost

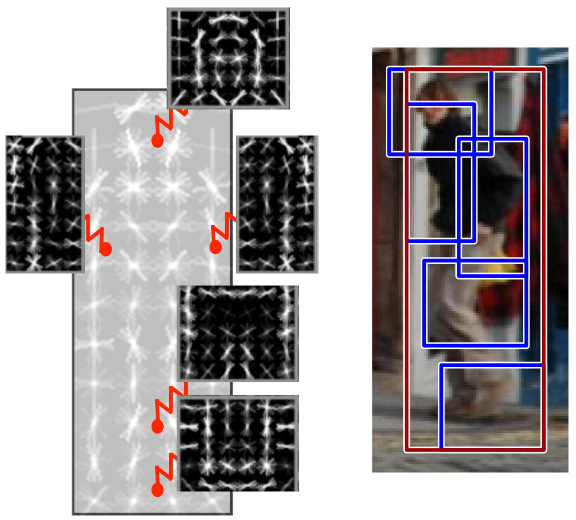
Interpolation and Per-class also perform well with low run-time cost

Regularization in metric learning is open issue, making hybrid algorithms based on combined generative/discriminative approaches a good alternative

Take-home message:
Local distance functions are useful!

Can we make use of shared representations?

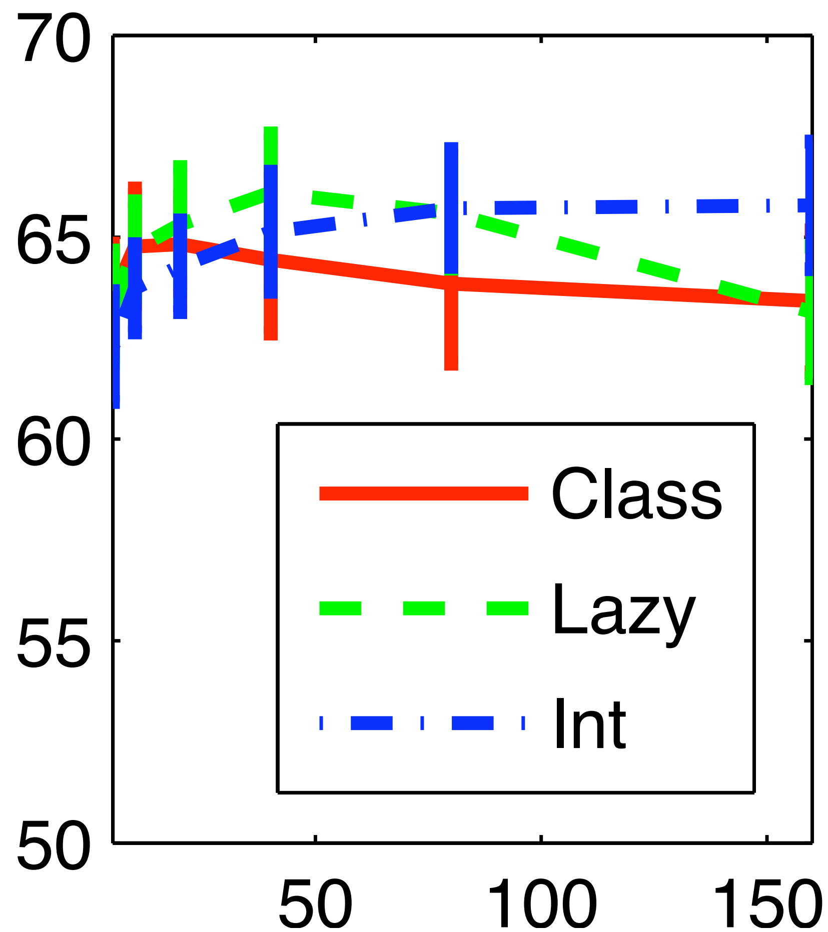




Should allow us to generate new “exemplars” never seen in training

Thanks!

Varying the shortlist parameter



Efficient indexing