

High Dimensional Nonlinear Learning using Local Coordinate Coding

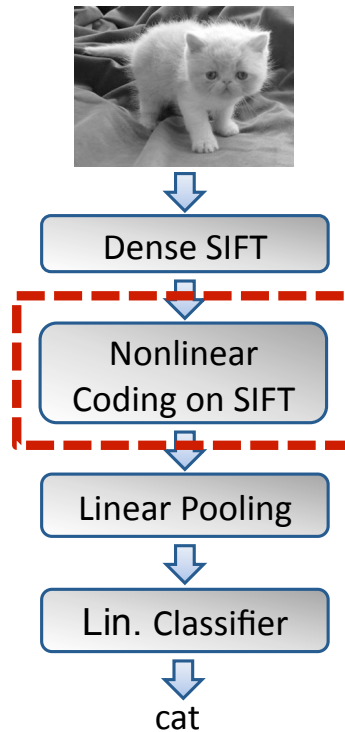
Tong Zhang

Statistics Department
Rutgers University, NJ

Collaborator: Kai Yu (NEC)

General Background

- Basic prediction problem:
 - known input X : a vector in R^d .
 - unknown output Y .
 - classifier $f: Y \approx f(X)$.
- Supervised learning:
 - learn f from training data $(X_i, Y_i)_{i=1, \dots, n}$.
- Example application: hand-written digit recognition
 - input X : image of a digit (from 0 to 9).
 - output Y : the corresponding digit



- X : SIFT feature vector; Y : cat.
- $f(X)$: coding + classifier. A **real-valued function**: how likely is X a cat image.

High Dimensional Data

- Goal: effectively learn a high dimensional nonlinear function.
- Our approach (local coordinate coding):
 - Stage 1: use unlabeled data to learn a nonlinear coding scheme.
 - * encode $X \in R^d$ to **code** $\Phi(X) \in R^L$
 - Stage 2: learn a linear function using the coding from stage 1:

$$f(X) = w^\top \Phi(X).$$

- Non-supervised nonlinear learning + supervised linear learning.

Agenda of the Talk

- Motivations and traditional ideas for nonlinear learning
 - kernel smoothing/k-nn/VQ
 - drawbacks
- Local coordinate coding (LCC)
 - intuition
 - theoretical justification
 - algorithm
- Empirical validation

Traditional Kernel Smoothing

- Training data: $(X_1, Y_1), \dots, (X_n, Y_n)$
- Kernel: $k(x, x')$ (e.g. $k(x, x') = \exp(-\|x - x'\|^2)$)
- Test point X :
- kernel smoothing prediction:

$$f(X) = \frac{\sum_{i=1}^n k(X_i, X) Y_i}{\sum_{i=1}^n k(X_i, X)}.$$

- Intuition: prediction is the weighted average of close-by observations

High Order Kernel Smoothing

- Standard Kernel smoothing is zero-th order:

$$\hat{Y} = \arg \min_y \sum_{i=1}^n k(X_i, X) [y - Y_i]^2.$$

– locally constant regression

- locally first order linear regression:

$$\hat{Y} = \hat{w}_X^T X + \hat{b}_X$$

$$[\hat{w}_X, \hat{b}_X] = \arg \min_{w, b} \sum_{i=1}^n k(X_i, X) [w^T X_i + b - Y_i]^2.$$

Drawbacks of Kernel Smoothing in High Dimension

- Not used in machine learning: why?
- Kernel bandwidth selection is not adaptive
 - solution: k -nearest neighbor (0-th order)
- Overfitting with high order methods
 - too much localization means significant fragmentation of training data in high dimensional local linear learning
 - not robust to contamination of input data noise
- Computationally very expensive
 - especially high order method

Addressing Computation: Basis Learning

- Instead of nearest neighbor, we find a smaller number of anchor points
 - address computational issues
 - anchor points are less noisy than input data
- Learning anchor points (code-book): $C = \{B_j\}$
 - objective: to best represent data (**vector quantization**):

$$\{B_j\}_{j=1}^L = \arg \min_{\{B_j\}} \sum_{i=1}^n \min_j \|B_j - X_i\|_2^2.$$

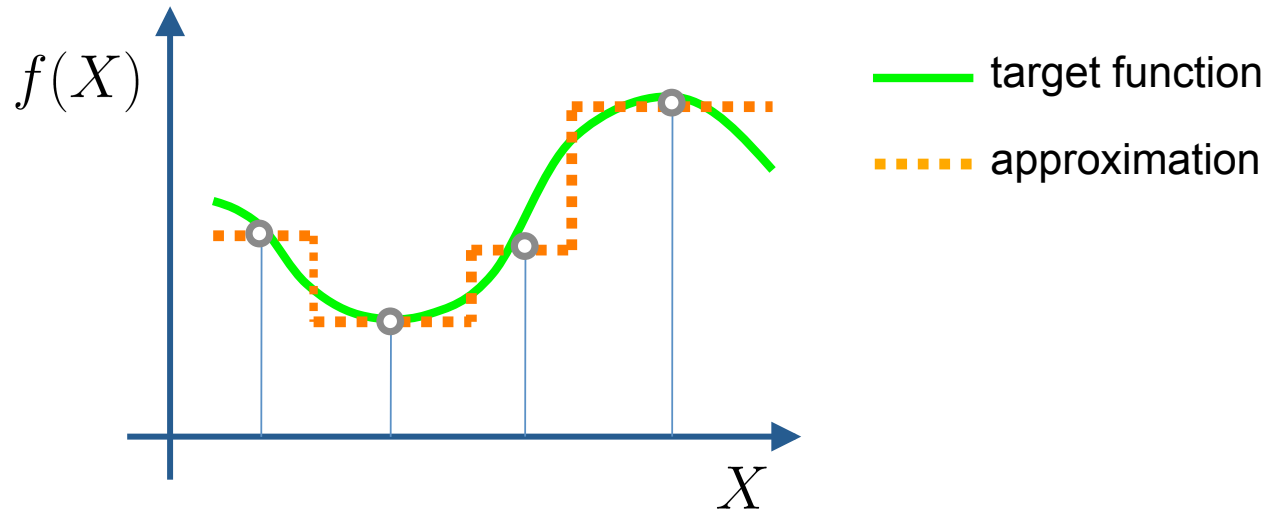
- Using only unlabeled data $\{X_i\}$: semi-supervised Learning

Summary Table

method	dimension	basis learning	approximation power
kernel smoothing	low	no	0th order
local linear regression	low	no	1st order
KNN	high	no	0th order
VQ	high	yes	0th order
proposed (LCC)	high	yes	1st order

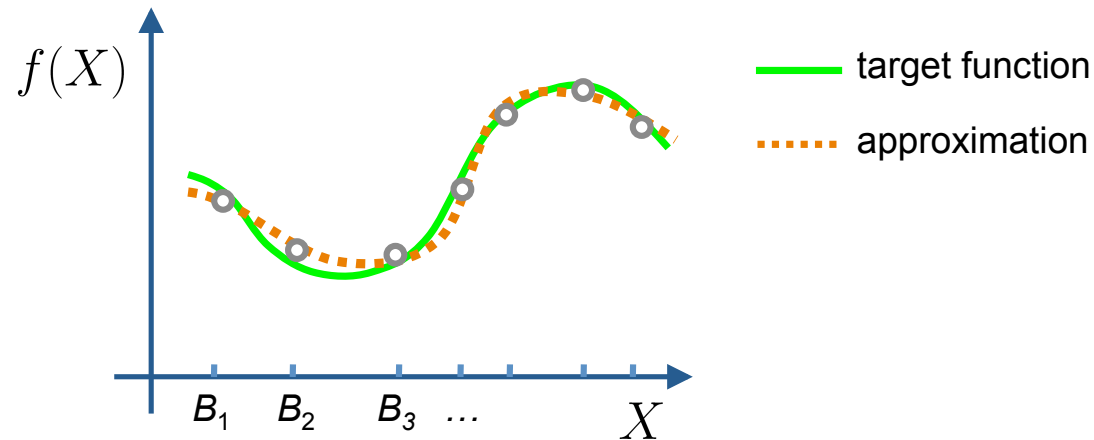
- high dimension:
 - require adaptive bandwidth selection
- basis learning:
 - computationally more efficient; more robust to input data noise
- coding schemes: KNN, VQ, LCC

Vector Quantization (VQ) Coding



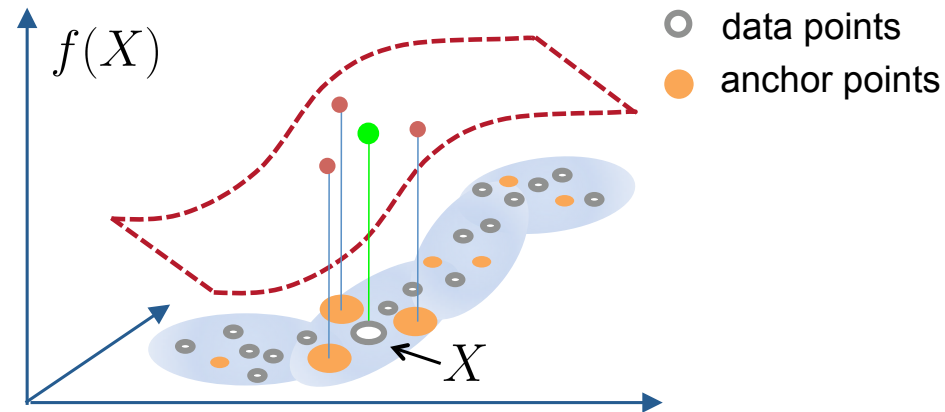
- Approximation function $f(X) = w^T \Phi(X)$
 - piece-wise constant approximation
- $\Phi(X)$: VQ coding of X , map X to the closest anchor point
 - $\Phi(X) = [0, 1, 0, \dots, 0]$ — 2nd anchor point is closest

Local Coordinate Coding (LCC)



- Approximation function $f(X) = w^T \Phi(X)$
 - 1st-order smooth approximation
- $\Phi(X) = [\phi_1, \dots, \phi_L]$ (anchor points B_1, \dots, B_L)
 - good **reconstruction quality**: $X \approx \sum_{j=1}^L \phi_j B_j$
 - good **locality**: $\phi_j \approx 0$ when B_j is far from X

LCC: How it works



- $f(X) = \sum_{j=1}^L w_j \Phi_j(X)$

$$\Phi(X) = \arg \min_{\Phi} \left[\underbrace{\|X - \sum_{j=1}^L \phi_j B_j\|_2^2}_{\text{reconstruction quality}} + \lambda \sum_{j=1}^L \underbrace{|\phi_j| \cdot d(X, B_j)}_{\text{locality}} \right]$$

$d(X, B_j)$: a certain distance from X to B_j

Local Coordinate Coding: formal definition

- Code-book: $C = \{B_j\}_{j=1}^L$: a set of anchor points
- A coordinate coding of X is an approximation of X by linear combination $\Phi(X) \in R^L$ of anchor points in C :

$$X \approx \sum_{j=1}^L \Phi_j(X) B_j.$$

- How to measure its quality:
 - should approximate X well
 - should be localized ($\Phi_j(X) \approx 0$ when B_j is not close to X)

Linearization Lemma

Lemma 1. Let (Φ, C) be an arbitrary coordinate coding on R^d . Let f be a function with smoothness parameters (α, β) . We have for all $X \in R^d$:

$$\underbrace{\left| f(X) - \sum_{j=1}^L w_j \Phi_j(X) \right|}_{\text{linear approximation of nonlinear function}} \leq \alpha \underbrace{\left\| X - \sum_{j=1}^L \Phi_j(X) B_j \right\|}_{\text{approximation quality}} + \beta \sum_{j=1}^L \underbrace{|\Phi_j(X)| \|B_j - X\|^2}_{\text{localization quality}},$$

where $w_j = f(B_j)$.

- nonlinear function can be approximated by a linear function with
 - unknown coefficients $w_j = f(B_j)$.
 - features $\Phi(X)$ from LCC scheme

Remarks of Linearization Lemma

- Importance:
 - change a difficult nonlinear learning problem into a linear learning problem
- Generalization of kernel smoothing to high dimension:
 - 1st order improvement over KNN/VQ
- Quality: depends on approximation and localization properties

Quality of LCC Learning method

Definition 1. [Coding Quality] Given α, β , and coding (Φ, C) , we define

$$Q_{\alpha, \beta}(\Phi, C) = \mathbf{E}_X \left[\alpha \left\| X - \sum_{j=1}^L \Phi_j(X) B_j \right\| + \beta \sum_{j=1}^L |\Phi_j(X)| \left\| X - \Phi_j(X) \right\|^2 \right].$$

Quality of LCC Learning depends on:

- code-book size: L (small linear learning complexity)
- coding quality: $Q_{\alpha, \beta}(\Phi, C)$ (small approximation error)

Intrinsic Dimension

- X lies in R^d , but restricted to a compact smooth manifold M with intrinsic dimension m ($m \ll d$).
 - code-book size: depends on m
 - coding quality: depends on m

Theorem 1. [Manifold Coding Complexity] $\forall \epsilon > 0$, there exist anchor points $C \subset M$ and coordinate coding scheme such that

$$L = |C| \leq \text{const}_1(M) \epsilon^{-m}$$

$$Q_{\alpha,\beta}(\Phi, C) \leq [\alpha \text{const}_2(M) + \beta \text{const}_3(M)] \epsilon^2.$$

Consistency

- Assume data lie in a compact manifold $M \subset R^d$ with intrinsic dimension m .
- When $n \rightarrow \infty$, **LCC can learn any nonlinear function** on M .
- The rate of convergence depends on m , and property of the non-linear function to be learned.

Summary of the LCC idea

- Key idea:
 - Learn a set of anchor points (basis)
 - Approximate each input X as a linear combination of close-by anchor points (local coordinate system)
- Benefits:
 - high dimensional nonlinear function becomes linear with respect to the anchor points
 - 1st order approximation
 - adaptive: complexity depends on intrinsic dimension
- Related to: high order kernel smoothing and basis learning
 - address curse of dimensionality and computation simultaneously

Algorithm Sketch

- Stage 1 (unsupervised): learn LCC coding scheme using unlabeled data by optimizing (Φ, C) :

$$\mathbf{E}_X \left[\left\| X - \sum_{j=1}^L \Phi_j(X) B_j \right\|^2 + \lambda \sum_{j=1}^L |\Phi_j(X)| \left\| X - B_j \right\|^2 \right].$$

using EM-style alternating optimization:

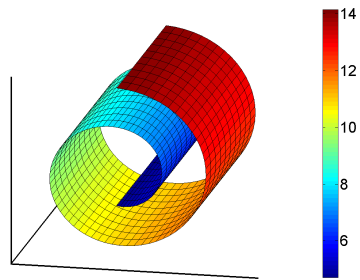
- fix basis $\{B_j\}$, learn coding Φ by solving Lasso
 - fix coding Φ , re-estimate basis by solving a regression problem.
- Stage 2 (supervised): learn linear function $f(X) = w^T \Phi(X)$ using labeled training data.

Relation to Other Methods

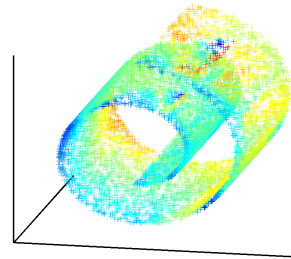
- Related approaches
 - kernel smoothing (low dimension, no basis learning)
 - k-nearest neighbor (high dimension, no basis learning, 0-th order)
 - VQ (high dimension, basis learning, 0-th order)
 - LCC (high dimension, basis learning, 1-th order)
 - sparse coding: a heuristics = our method minus geometry (locality)
- Our method can be regarded as:
 - locality constrained and theoretically correct version of sparse coding
 - high dimensional generalization of 1st order kernel smoothing
 - 1st order extension of VQ.
- Not a traditional statistical method (focus on geometry)
 - no minimax or asymptotic normality types of results

Experiment I: Swiss Roll

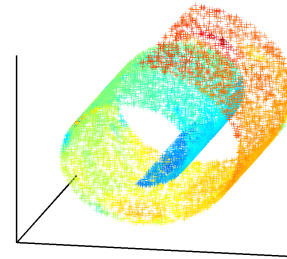
- Simulated data
- Learn three classes on a swiss roll manifold
- To illustrate various aspects of the theory



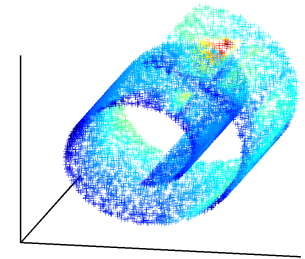
(1) A nonlinear function



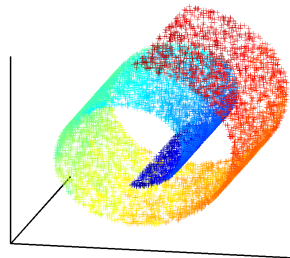
(2) RMSE=4.394



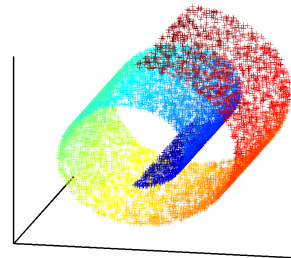
(3) RMSE=0.499



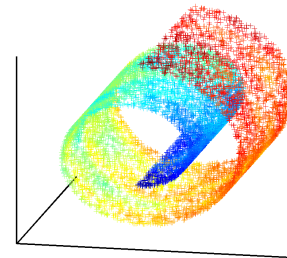
(4) RMSE=4.661



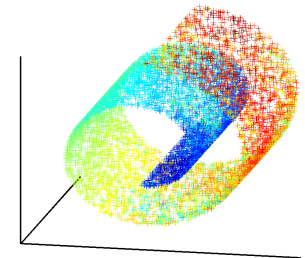
(5) RMSE=0.201



(6) RMSE=0.109

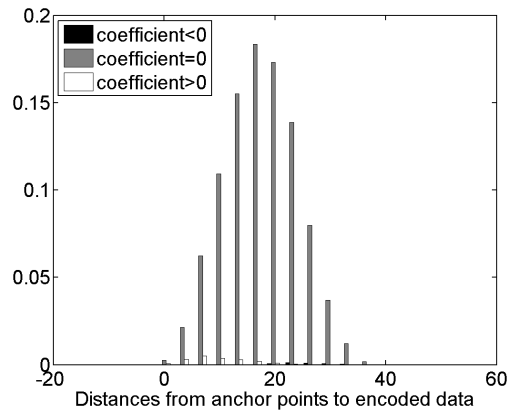


(7) RMSE=0.669

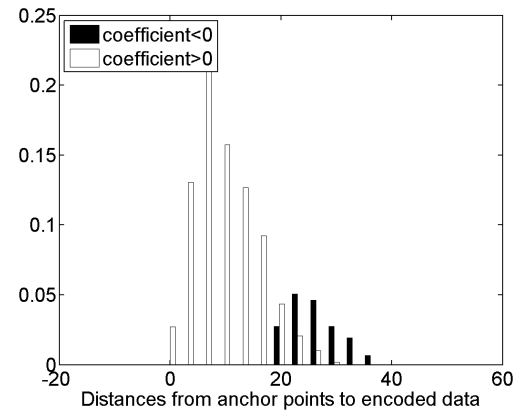


(8) RMSE=1.170

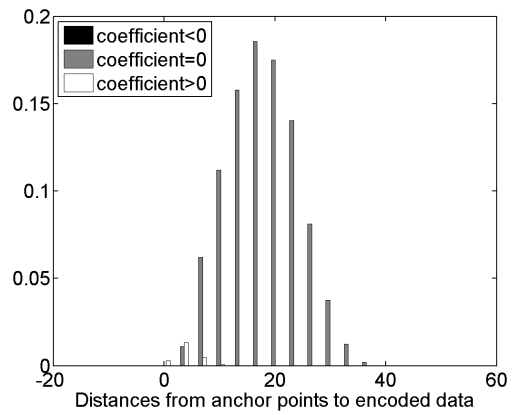
Figure 1: (1) target nonlinear function (color indicates function values); (2) result of sparse coding with fixed random anchor points; (3) result of local coordinate coding with fixed random anchor points; 4) result of sparse coding; (5) result of local coordinate coding; (6) result of local kernel smoothing; (7) result of local coordinate coding on noisy data; (8) result of local kernel smoothing on noisy data.



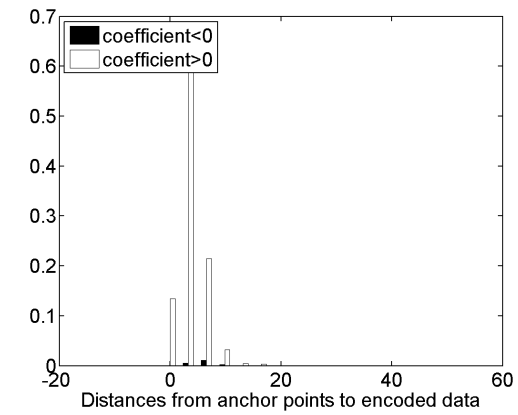
(a-1)



(a-2)



(b-1)



(b-2)

Figure 2: Coding locality on Swiss roll: (a) sparse coding vs. (b) LCC

Experiment II: MNIST

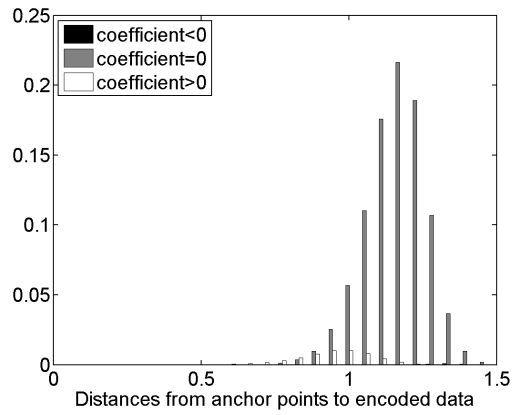
- Hand digit recognition data
- Ten categories: images of 0-9.
- 60K training + 10K testing

Table 1: Error rates (%) of MNIST classification with different methods.

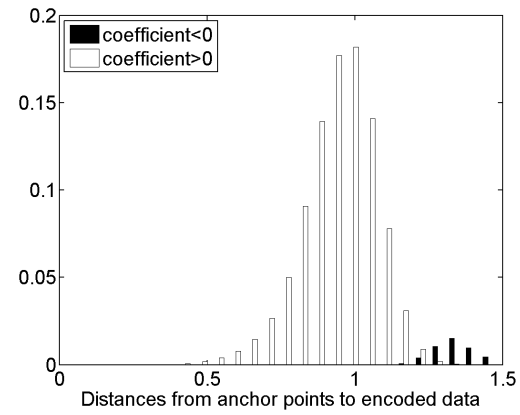
Methods	Error Rate
Linear SVM with raw images	12.0
Local kernel smoothing	3.48
Linear SVM with Laplacian eigenmap	2.73
Linear SVM with VQ coding	3.98
Linear SVM with sparse coding	2.02
Linear SVM with local coordinate coding	1.90
Linear classifier with deep belief network	1.90

Table 2: Error rates (%) of MNIST classification with different $|C|$.

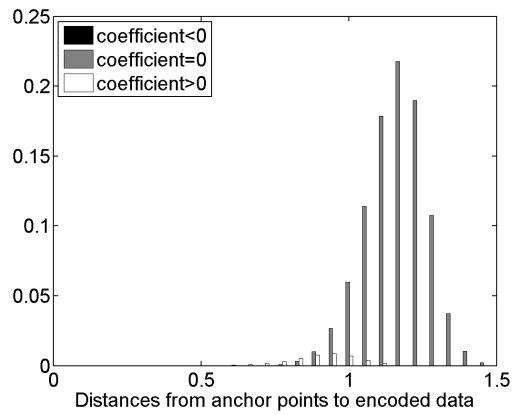
$ C $ (code-book size)	512	1024	2048	4096
Linear SVM with sparse coding	2.96	2.64	2.16	2.02
Linear SVM with local coordinate coding	2.64	2.44	2.08	1.90



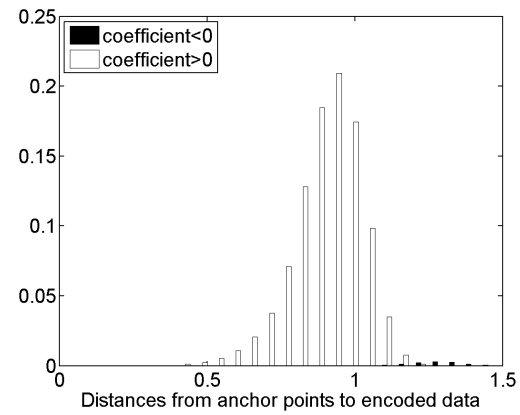
(a-1)



(a-2)



(b-1)



(b-2)

Figure 3: Coding locality on MNIST: (a) sparse coding vs. (b) LCC

2009 Pascal Image Classification Competition

- Classify images into 20 classes
 - Person: person
 - Animal: bird, cat, cow, dog, horse, sheep
 - Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
 - Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor
- Data: \approx 15000 images from flicker.
 - half for training
 - half for blind evaluation
- More than thirty participating teams.
- Winning team: NEC system using LCC.

Example Images



car



monitor/tv



horse



potted plant

System Description

Components	Previous state of the art	NEC System
Feature Extraction	multiple detectors/descriptors	sampling + SIFT (gray)
Feature Aggregation (pooling)	SPM	SPM
Feature Engineering/Coding	VQ	LCC
Classifier	nonlinear	linear

- NEC System
 - minimum feature engineering; new machine learning methods
- Other System
 - complicated feature engineering; tradition machine learning methods

Flow Chart



Dense SIFT



Nonlinear
Coding on SIFT



Linear Pooling



Lin. Classifier



cat

Performance Comparison

class name	LCC single manifold	LCC mixture of multiple manifolds	best performance from other systems
Aeroplane	87.7	88.0	86.6
Bicycle	67.8	68.6	63.9
Bird	68.1	67.9	66.7
Boat	71.1	72.9	67.3
Bottle	39.1	44.2	43.7
Bus	78.5	79.5	74.1
Car	70.6	72.5	64.7
Cat	70.7	70.8	64.2
Chair	57.4	59.5	57.4
Cow	51.7	53.6	46.2
Diningtable	53.3	57.5	54.7
Dog	59.2	59.0	53.5
Horse	71.6	72.6	68.1
Motorbike	70.6	72.3	70.6
Person	84.0	85.3	85.2
Pottedplant	30.9	36.6	39.1
Sheep	51.7	56.9	48.2
Sofa	55.9	57.9	50.0
Train	85.9	85.9	83.4
Tvmonitor	66.7	68.0	68.6
average	64.6	66.5	62.8

Take Home Messages

- Integrate **geometry and learning**
 - novelty: geometrically form local coordinate system
 - other methods?
- Integrate **local and global** learning
 - novelty: nonlinear function is globally linear with respect to LCC
 - traditional: cost function locally weighted using kernels

Summary

- LCC (local coordinate coding):
 - high dimensional nonlinear learning
- Theory:
 - LCC approximates high dimensional nonlinear function by linear function
 - performance depends on intrinsic dimensionality
- Algorithm:
 - unsupervised learning of coding (LCC)
 - supervised learning of linear function