

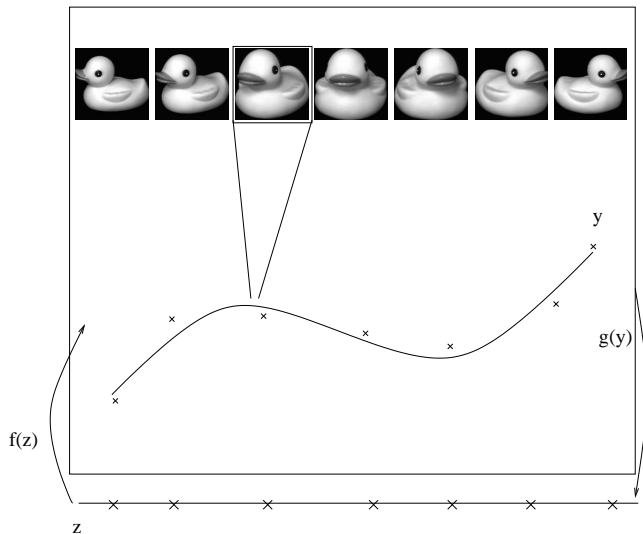
Kernel Information Embeddings

Roland Memisevic

University of Toronto

ICML 2006

Dimensionality reduction



Two issues

- ▶ Learn **mappings** f and g
 - ▶ Sammon mapping: g no, f no
 - ▶ Spectral methods (e.g. LLE, kernel PCA): g yes, f no
 - ▶ GPLVM, Principal curves: g yes, f yes (learning slow)
- ▶ Include **supervision signals**
 - ▶ Factor out things that we know already.
 - ▶ Include knowledge about grouping structure.
 - ▶ Learn the best embedding for a specific task.

Kernel Information Embedding

- ▶ An appealing criterion for embedding: **Mutual information** between y and z :

$$I(Y; Z) = \int p(y, z) \log \frac{p(y, z)}{p(y)p(z)} dy dz$$

- ▶ Intuition: “What, on average, can the low-dimensional code Z tell us about the data Y and vice versa ?”
- ▶ Can write the mutual information in terms of **entropies** $H(\cdot)$:

$$I(Y; Z) = H(Y) + H(Z) - H(Y, Z).$$

Kernel Information Embedding

- ▶ Problem: Difficult to compute in general.
- ▶ Idea: Use **kernel density estimates** to approximate entropies.
- ▶ Using $p(y, z) = \sum_j K(y, y^j)K(z, z^j)$, we can try:

$$\begin{aligned} H(Y) &= - \int p(y) \log p(y) dy \\ &\approx -\frac{1}{n} \sum_i \log p(y^i) \\ &\approx -\frac{1}{n} \sum_i \log \sum_j K(y^i, y^j) \\ &=: \hat{H}(Y) \end{aligned}$$

Kernel Information Embedding

- ▶ We get the MI-estimate:

$$\begin{aligned} & \hat{I}(Y, Z) \\ &= \hat{H}(Y) + \hat{H}(Z) - \hat{H}(Y, Z) \\ &= -\frac{1}{n} \sum_i \log \sum_j K(z^i, z^j) + \frac{1}{n} \sum_i \log \sum_j K(z^i, z^j) K(y^i, y^j) + \Omega \\ &= \frac{1}{n} \sum_i \log \sum_j \frac{K(z^i, z^j)}{\sum_k K(z^i, z^k)} K(y^i, y^j) + \Omega \end{aligned}$$

- ▶ **To train:** Gradient ascent on the z^i !

Complexity control

- ▶ Model can “cheat” by moving all z^i infinitely far apart from one another.
 - ▶ Need to restrict the range of the z^i to a finite interval. Simple way to implement this: Add variance penalty to the objective function. For example:

$$\text{maximize } I(Y; Z) - \lambda \text{Tr}(ZZ^T) \quad (\text{'Power constraint'})$$

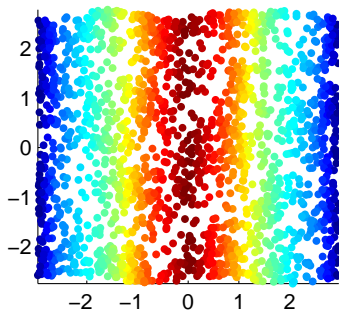
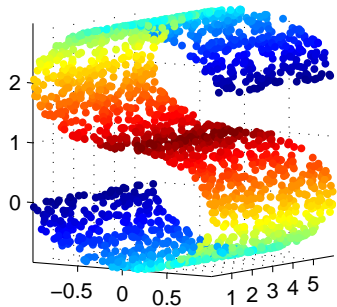
or

$$\text{maximize } I(Y; Z) - \lambda \|Z\|_p$$

or ...

- ▶ Can use deterministic annealing to find good local minima: Gradually decrease λ !

Example



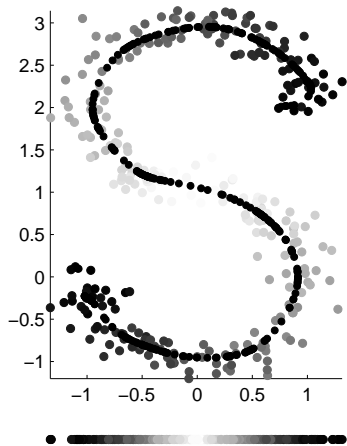
A view from non-parametric regression

$$\frac{1}{n} \sum_i \log \sum_j \frac{K(z^i, z^j)}{\sum_k K(z^i, z^k)} K(y^i, y^j) + \Omega$$

- ▶ Non-parametric regression onto latent variables
[Lawrence, 2004] [Meinicke et al., 2005] [Memisevic, 2003]
- ▶ This motivates the **definitions**:

- ▶ $f(z^{\text{test}}) = \arg \max_y \log \sum_j \frac{K(z^{\text{test}}, z^j)}{\sum_k K(z^{\text{test}}, z^k)} K(y, y^j)$
- ▶ $g(y^{\text{test}}) = \arg \max_z \log \sum_j \frac{K(z, z^j)}{\sum_k K(z, z^k)} K(y^{\text{test}}, y^j)$

De-noising: $f(g(y))$



Conditional embeddings

- ▶ The information theoretic view suggests several ways to extend the method to **semi-supervised** settings.
- ▶ One way to introduce side-information: Replace mutual information by **conditional MI**:

$$I(Y; Z|X) = H(X, Z) - H(X, Y, Z) + \Omega$$

- ▶ Intuition: “What, on average, can Z tell us about Y , given that we already know X ?”

Conditional embeddings

$$\text{maximize: } \hat{I}(Y; Z|X)$$

- ▶ Capture factors Z that are not already represented by X .
- ▶ **Factor out** modes of variability that we already know about.
- ▶ In practice: Reserve some dimensions for known factors and arrange data according to those. Then **learn** the remaining, orthogonal factors.

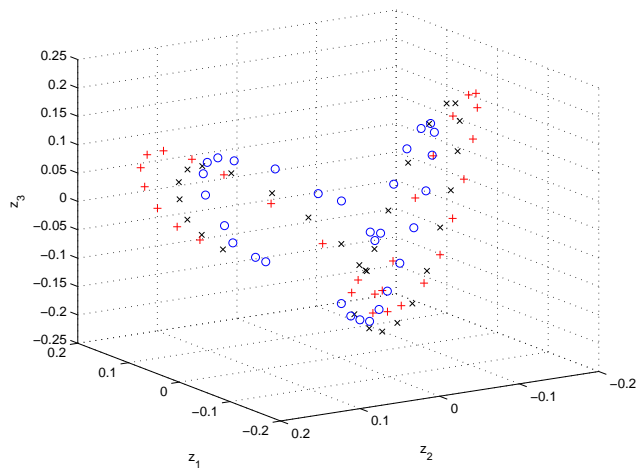
Conditional embeddings

- ▶ Example: Columbia Object Image Library (COIL)-data:



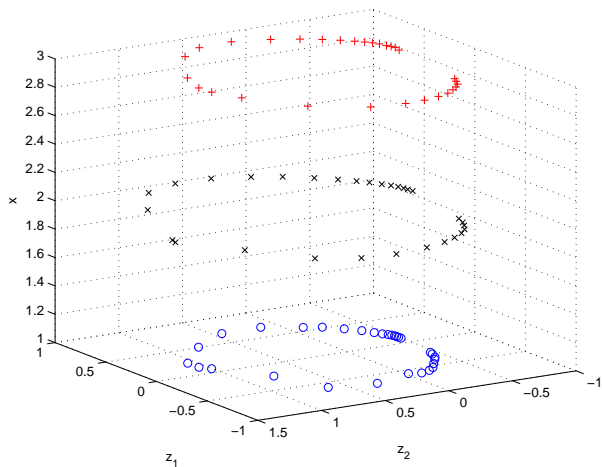
Conditional embeddings

Kernel PCA:



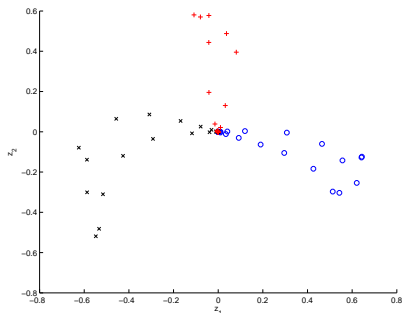
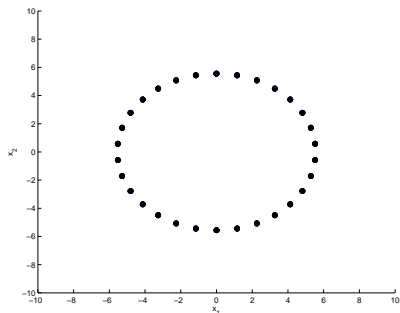
Conditional embeddings

$$\arg \max_Z \hat{I}(Y, Z|X):$$

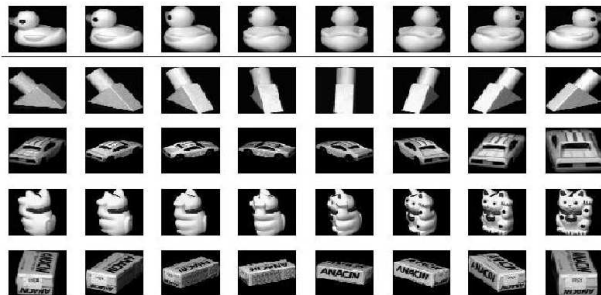


Conditional embeddings

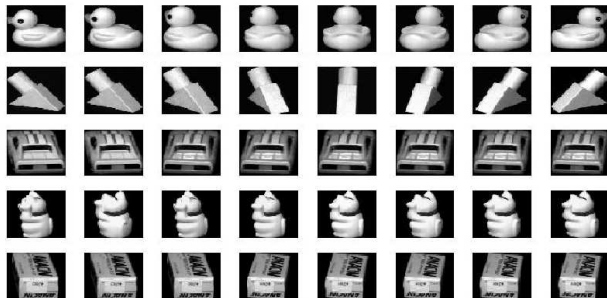
$$\arg \max_Z \hat{I}(Y, Z|X):$$



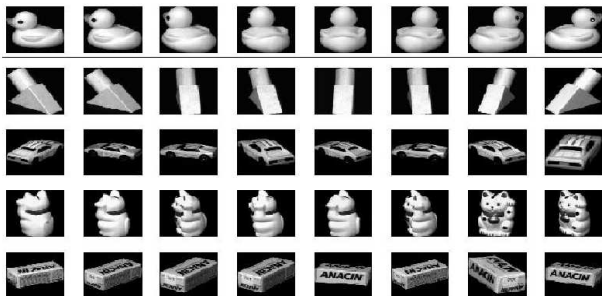
Neighborhood



Neighborhood



Neighborhood



Manifold Alignment

- ▶ Other ways to include side-information:

$$\text{minimize: } \hat{I}(X; Y|Z)$$

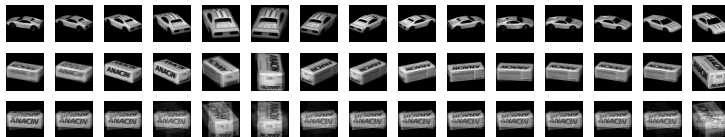
- ▶ Intuition: “Find factors Z , such that given these, X can tell us **as little** as possible about Y and vice versa.”
- ▶ Capture the variability that is **shared** by X and Y : “Let $X - Z - Y$ become a Markov chain.”
- ▶ Feature extraction. Manifold alignment [Ham et al., 2005].

Example: Semisupervised regression

- ▶ Training data:



- ▶ Regression result:



Conclusions

- ▶ Information theoretic embedding.
- ▶ Optimize latent representatives directly, instead of using a parametric model.
- ▶ Forward- and backward mappings available nevertheless.
- ▶ In general: embeddings need to be supplemented by supervision signal(s) to be useful.

References



Ham, J., Lee, D., and Saul, L. (2005).

Semisupervised alignment of manifolds.

In Cowell, R. G. and Ghahramani, Z., editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*.



Lawrence, N. D. (2004).

Gaussian process latent variable models for visualisation of high dimensional data.

In *Adv. in Neural Information Processing Systems 16*.



Meinicke, P., Klanke, S., Memisevic, R., and Ritter, H. (2005).

Principal surfaces from unsupervised kernel regression.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(9):1379–1391.



Memisevic, R. (2003).

Unsupervised kernel regression for nonlinear dimensionality reduction.

Diplomarbeit, Universität Bielefeld.