Learning from examples in Neural Gas and Vector Quantization

Michael Biehl, Anarta Ghosh, and Aree Witoelaar

Institute for Mathematics and Computing Science
University of Groningen, PO Box 800, 9700 AV Groningen, The Netherlands
m.biehl@rug.nl, re/pre-prints available at [12]

DY 46.9

Abstract

The dynamics of training a neural gas for vector quantization in high dimensions is studied by means of methods from statistical physics. Prototype vectors for the representation of the data are updated either off-line from a given, fixed set of example data, or on-line from a sequence of single data. Here we concentrate on the latter case. For the on-line learning scenario, a description of the learning dynamics in terms of ordinary differential equations for a set of order parameters becomes exact in the limit of infinite-dimensional data. We explain the method, present first results and discuss possible extensions.

1. Introduction

- **Vector Quantization (VQ)**
  unsupervised identification of prototype vectors
  for the **representation** of data (e.g. clusters) by means of **distance-based competitive learning**, see, e.g. [1]
- **Neural Gas (NG)**
  representation of data by (many) prototypes
  determined by **rank based competitive learning** [2]
- **Self-Organizing Maps** (SOM)
  representation of data by a grid of prototypes
  competitive learning + **pre-defined topology** [3]

Aims:
– understand **typical properties** of VQ and NG
in terms of **model situations** with high-dim. data
– dynamics of on-line learning
– typical equilibrium properties of off-line training
– evaluate the **performance of training schemes**
– develop novel and **efficient algorithms**
– extensions to Self-Organizing Maps and similar systems

2. The Model

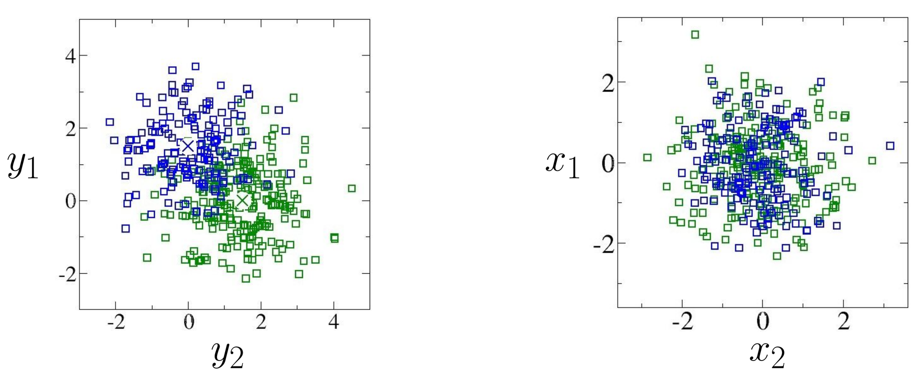
2.1 The example data

Independent random inputs $\vec{\xi}^\mu \in \mathbf{R}^N$, $\mu = 1, 2, \ldots P$,
drawn from a **mixture density** (index μ omitted)

$$P(\vec{\xi}) = \sum_{m=1}^{M} p_m P(\vec{\xi} \mid m) \text{ with } P(\vec{\xi} \mid m) = \frac{1}{(2\pi v_m)^{N/2}} \exp\left[-\frac{1}{2v_m}\left(\vec{\xi} - \ell\vec{B}_m\right)^2\right].$$

Mixture of Gaussians
spherical clusters with variance v_m, centered at $\ell\vec{B}_m$,
prior weights $\sum_m p_m = 1$

orthonormal vectors $\vec{B}_m \in \mathbf{R}^N$ with $\vec{B}_m \cdot \vec{B}_n = \delta_{m,n}$.
Note: $(\vec{B}_m)^2 = 1$, but $\left\langle (\vec{\xi})^2 \right\rangle_m = v_m N + \ell^2 \sim N$



Example: Monte Carlo data ($N = 200$) with $M = 2$, $v_1 = v_2 = 1$
left: *weak separation apparent in $y_m = \vec{B}_m \cdot \vec{\xi}$ centers marked by "×"*
right: *projections $x_k = \vec{w}_k \cdot \vec{\xi}$ on independent random $\vec{w}_k$ ($\vec{w}_k^2 = 1$)*

Notation: conditional averages $\langle \cdots \rangle_m$ over $P(\vec{\xi} \mid m)$

averages $\langle \cdots \rangle = \sum_{m=1}^{M} p_m \langle \cdots \rangle_m$ over the full $P(\vec{\xi})$

2.2 Training

Rank based training of prototypes

- initialize a **set of prototypes** $\left\{\vec{w}_k \in \mathbb{R}^N\right\}_{k=1}^{K}$
  e.g. random $\vec{w}_k(0) \approx 0$

1) present a **single example vector** $\vec{\xi}^\mu$

2) evaluate the (squared) **distances** $d(k,\mu) = \left(\vec{\xi}^\mu - \vec{w}_k\right)^2$
sort: $d(j(1),\mu) \geq d(j(2),\mu) \geq \ldots \geq d(j(K),\mu)$
here: $r(j)$ is the **rank** of prototype j w.r.t. $\vec{\xi}^\mu$
$j(r)$ is the index of the prototype with rank r

3) update all prototypes
$$\vec{w}_k(t+1) = \vec{w}_k(t) + \frac{\eta}{N} f(r(k)) \left(\vec{\xi}^\mu - \vec{w}_k(t)\right)$$

– **learning rate** η controls the step size
– update is **towards the data**
– **rank function** $f(r)$ defines the magnitude:

- $f(r) = e^{-r/\sigma}$ rank based Neural Gas as in [2]
- $f(r) = \begin{cases} 1 & \text{if } r = 1 \\ 0 & \text{else} \end{cases}$ (Winner-Takes-All)
- $f(r) = K - r$ linear rank function $(K - 1 \geq f(r) \geq 0)$
  considered in the following

associated cost function quantization error

$$H(\{w_k\}) = \sum_\mu \sum_k f(r(k)) \left(\vec{\xi}^\mu - \vec{w}_k\right)^2$$

3. The analysis

training from a sequence of **independent examples** ($t \equiv \mu$),
consider **macroscopic overlaps as order parameters**

$$R_{km}(\mu) = \vec{w}_k(\mu) \cdot \vec{B}_m \quad \text{and} \quad Q_{jk}(\mu) = \vec{w}_j(\mu) \cdot \vec{w}_k(\mu)$$
$$\text{for} \quad k = 1, 2, \ldots, K, \quad \text{and} \quad m = 1, 2, \ldots, M.$$

training algorithm $\rightarrow KM + K(K+1)$ recursion relations

$$\frac{R_{km}(\mu) - R_{km}(\mu-1)}{1/N} = \eta f(r(k)) \left(y_m^\mu - R_{km}(\mu-1)\right)$$
$$\frac{Q_{kl}(\mu) - Q_{kl}(\mu-1)}{1/N} = \eta f(r(k)) \left(x_k^\mu - Q_{kl}(\mu-1)\right) + \eta f(r(l)) \left(x_l^\mu - Q_{kl}(\mu-1)\right) + \eta^2 f(r(k)) f(r(l)) + \mathcal{O}\left(\frac{1}{N}\right)$$

with $x_k^\mu = \vec{w}_k(\mu-1) \cdot \vec{\xi}^\mu$ and $y_m^\mu = \vec{B}_m \cdot \vec{\xi}^\mu$

Central Limit Theorem $\rightarrow$ **Gaussian** $P(\{x_k^\mu\}, \{y_m^\mu\})$ given by

$$\langle x_k^\mu \rangle_m = \ell R_{km}(\mu-1), \quad \langle y_m^\mu \rangle_n = \ell \delta_{mn}, \quad \langle x_k^\mu x_l^\mu \rangle_m - \langle x_k^\mu \rangle_m \langle x_l^\mu \rangle_m = v_m Q_{kl}(\mu-1)$$
$$\langle x_k^\mu y_n^\mu \rangle_m - \langle x_k^\mu \rangle_m \langle y_n^\mu \rangle_m = R_{kn}^{\mu-1}, \quad \langle y_m^\mu y_n^\mu \rangle_q - \left\langle y_m^\mu \right\rangle_q \left\langle y_n^\mu \right\rangle_q = v_q \delta_{mn}.$$

thermodynamic limit $N \rightarrow \infty$:

- **averages over latest example** $\rightarrow$ r.h.s. as functions of overlaps
- $\rightarrow$ **coupled ODEs** in *continuous time* $\alpha = \mu/N$.
- $\{R_{km}, Q_{kl}\}$ **self–averaging** w.r.t. random examples, e.g. [6]
  fluctuations vanish in the limit $N \rightarrow \infty$

Analytical evaluation possible with the following simplifications

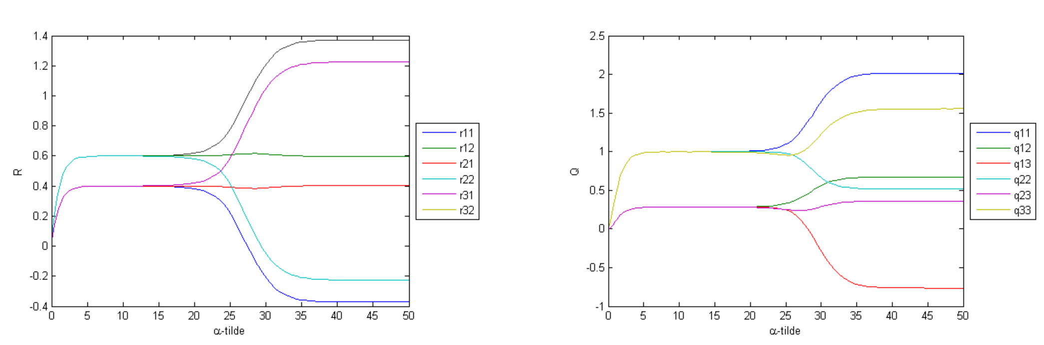
- **limit of small learning rates** $\rightarrow$ neglect term $\propto \eta^2$
  $\eta \rightarrow 0$, $\alpha \rightarrow \infty$, such that $\tilde{\alpha} = \eta\alpha = \mathcal{O}(1)$
- **linear rank function**
  note: rank can be obtained from pair-wise comparison
  $r(k) = \sum_{j \neq k} \Theta\left[d(k,\mu) - d(j,\mu)\right]$ with $\Theta(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{else} \end{cases}$

numerical integration of ODE, from initial $R_{km}(0), Q_{kl}(0)$
$\rightarrow$ **learning curves** $R_{km}(\tilde{\alpha}), Q_{kl}(\tilde{\alpha})$
$\rightarrow$ **typical dynamics of learning** in high dimensions
fixed point analysis,...

4. First results

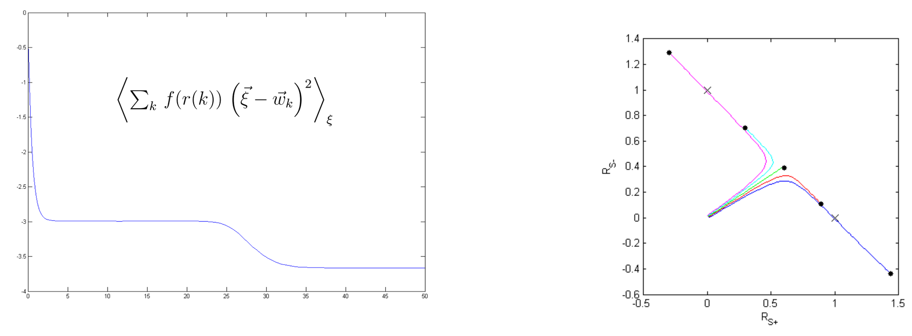
4.1 Learning curves

Example situation
data from two clusters, $p_1 = 0.6, p_2 = 0.4$, variances $v_1 = v_2 = 1$,
separation $\ell = 1$, 3 prototype vectors
evolution of R_{jm}, Q_{ij}:



intial phase: **unspecialized units** in the space $\perp \{B_j\}_{m=1}^{M}$.
asymptotic configuration: lowest quantization error

Left: evolution of the average (weighted) **quantization error**

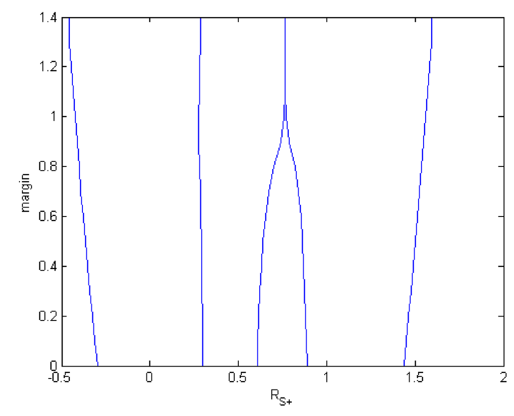


Right: trajectory of **5 prototypes**, projected into the $B_{1,2}$-plane

4.2 Modified rank evaluation

NG rank function $f(r) \propto e^{-r/\sigma}$:
system approaches *Winner-Takes-All* behavior for $\sigma \rightarrow 0$

Modified rank function:
similar effect by introduction of a *margin* c
$f(\hat{r}(j)) = \sum_{j \neq k} \Theta\left[d_j - d_k - c\right]$ (count only *clear wins*)



margin vs. $\alpha \rightarrow \infty$ asymptotic configuration

- few prototypes and/or small margin:
  all prototype along connection of clusters
- many prototypes and/or large margin:
  projections of several $\vec{w}_i$ merge
  actual position differs in the space $\perp$ to $\left\{\vec{B}_m\right\}$

5. Outlook

- detailed fixed point analysis
  (repulsive/attractive configurations, plateaus in the learning curves)
- investigation of the *magnification factor* of NG
  (*density* of prototypes vs. density of the data)
- optimized training schedules: learning rate, margins
- analysis of *batch Neural Gas* algorithms

References

[1] J.A. Hertz, A. Krogh, R.G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley (1991)

[2] T. Martinetz, K. Schulten, *A Neural Gas Network learns topologies*, in: T. Kohonen et al. (eds), Artificial Neural Networks, Elsevier (1991)

[3] T. Kohonen, *Self-Organizing Maps*, Springer (1997)

[4] D. Saad (ed.), *On–line Learning in Neural Networks* , Cambridge University Press (1998)

[5] M.Biehl, A.Freking, G.Reents, Europhys.Lett. **38** (1997) 73

[6] G. Reents and R. Urbanczik, Phys. Rev. Lett. **80** (1998) 5445

[7] M. Biehl, A. Ghosh, and B. Hammer, *The dynamics of learning vector quantization*, In: Michel Verleysen (ed.), Proc. ESANN 2005, d-side (2005),

[8] M. Biehl, A. Ghosh, and B. Hammer, Neurocomputing 69 (7-9), 660 (2006),

[9] A. Ghosh, M. Biehl, and B. Hammer, *Performance analysis of LVQ algorithms: a statistical physics approach*, Neural Networks, in press (2006)

[10] A. Ghosh, M. Biehl, and B. Hammer, Dynamical Analysis of LVQ type learning rules, in: Proc. of WSOM'05, M. Cottrell (ed.), Univ. Paris (I) 2005

[11] A. Ghosh, M. Biehl, A. Freking, and G. Reents, Technical Report 2004-9-02, Math. and Comp. Science, Univ. Groningen, see [12]

[12] **visit** http://www.cs.rug.nl/~biehl **for further information, re- and preprints.**