A Decoupled Approach to Exemplar-based Unsupervised Learning

Sebastian Nowozin SEBASTIAN.NOWOZIN@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics, Spemannstrasse 38, 72076 Tübingen, Germany

Gökhan Bakır GHB@GOOGLE.COM
Google GmbH, Brandschenkestrasse 110, 8002 Zurich, Switzerland

Abstract

A recent trend in exemplar based unsupervised learning is to formulate the learning problem as a convex optimization problem. Convexity is achieved by restricting the set of possible prototypes to training exemplars. In particular, this has been done for clustering, vector quantization and mixture model density estimation. In this paper we propose a novel algorithm that is theoretically and practically superior to these convex formulations. This is possible by posing the unsupervised learning problem as a single convex "master problem" with non-convex subproblems. We show that for the above learning tasks the subproblems are extremely well-behaved and can be solved efficiently.

1. Introduction

Methods for unsupervised learning aim at recovering underlying structure from data. In this paper, we are concerned with *exemplar based models* in which this structure is represented by a weighted set of points in input space. Depending on the used model, these points can be interpreted as *clusters*, *codebook vectors* or *mixture components*.

Although the representation is done by a finite point set, the structure being represented – such as a density – is defined on the entire input space by expanding a *smoothing kernel function* around each representing point. In this setting learning simply becomes deciding on the number of points and their weights, as well as their location in input space by means of a suitable *objective*. In EM-learning of mixture models and in k-means clustering one fixes the number of points and adjusts their position by performing descent steps on the objective function starting from a random initialization. This leads to well-behaved but usually non-convex learning problems. Recently, a number of *convex* approaches have been proposed. Our goal in this paper is to improve on these approaches.

In section 2 we review convex formulations for unsupervised learning tasks and discuss two recent methods. We show how convexity is achieved and derive a small experiment whose result suggests a way to improve on the established models. We describe our model in section 3 together with an algorithm and a theoretical justification. The model is validated experimentally in section 4 and we conclude in section 5.

2. Review of convex approaches

We now discuss two convex approaches to unsupervised learning from the literature. We will denote the training set as $X = \{\boldsymbol{x}_i\}_{i=1,\ldots,N}$, with $\boldsymbol{x}_i \in \mathcal{X}$ and usually $\mathcal{X} = \mathbb{R}^d$.

Kernel Vector Quantization (Tipping & Schölkopf, 2001) learns a small set of codebook vectors such that the minimum distance from any training sample to its nearest codebook vector is bounded above by a given *maximum distortion* h. In (Tipping & Schölkopf, 2001), this is done by formulating a linear programming problem, of which the following problem is an equivalent reformulation.[1]

$$\begin{aligned} \max_{\boldsymbol{q},\rho} \quad & \rho \\ \text{sb.t.} \quad & K\boldsymbol{q} \geq \rho\mathbf{1}, \\ & \|\boldsymbol{q}\|_1 = 1, \\ & \boldsymbol{q} \geq \mathbf{0}. \end{aligned} \tag{1}$$

Here K is a (N, N) matrix with $K_{i,j} = I(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq h)$, where $I(\cdot)$ evaluates to one if the predicate is true and to zero otherwise, therefore, $K_{i,j}$ is one if a ball of radius h centered on $\boldsymbol{x}_j$ contains $\boldsymbol{x}_i$. In the solution of (1) the balls selected by $q_j > 0$ form a sparse covering of the training set and the distance of each sample to its closest covering ball is bounded by h.

Convex Clustering (Lashkari & Golland, 2007) was re-

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

[1] Subject to rescaling of $\boldsymbol{q}$.

cently proposed for clustering. In Lashkari and Golland's model, a mixture model is fit to an observed training set, such that a candidate mixture component is centered around each training set exemplar. Using the framework of Bregman clustering (Banerjee et al., 2005), their objective maximizes the log-likelihood subject to the constraint that the resulting model is a proper mixture model. In the optimum solution of the model, a sparse set of exemplars is selected, allowing the interpretation as clusters.

Formally, Lashkari and Golland maximize $\frac{1}{N}\sum_{i=1}^{N}\log\left[\sum_{j=1}^{N} q_j e^{-\beta d_\phi(\boldsymbol{x}_i,\boldsymbol{x}_j)}\right]$ over the mixture parameters $q_j \geq 0,\ j = 1,\ldots,N$ with $\sum_{j=1}^{N} q_j = 1$. The model allows all exponential family distributions with a corresponding Bregman divergence d_ϕ (Banerjee et al., 2005). For the maximization, a multiplicative update is used, which leads to slow convergence once elements of $\boldsymbol{q}$ approach zero. We reformulate the above objective function by introducing a new set of variables γ_i, with $i = 1,\ldots,N$ as follows.

$$\max_{\boldsymbol{q},\boldsymbol{\gamma}} \quad \frac{1}{N}\sum_{i=1}^{N}\log\gamma_i \tag{2}$$

$$\text{sb.t.} \quad K\boldsymbol{q} = \boldsymbol{\gamma}, \tag{3}$$
$$\|q_j\|_1 = 1,$$
$$q_j \geq 0, \qquad j = 1,\ldots,N,$$

where K is a (N,N) matrix and $K_{i,j} = e^{-\beta d_\phi(\boldsymbol{x}_i,\boldsymbol{x}_j)}$. Clearly, problem (2) is equivalent to the previous one because constraints (3) only serve to evaluate the likelihood γ_i for each sample $\boldsymbol{x}_i$.

2.1. Where does Convexity come from?

Models as proposed in (Tipping & Schölkopf, 2001) and (Lashkari & Golland, 2007) achieve convexity by changing the problem parametrization. Instead of learning the coordinates of a fixed number of exemplars $\boldsymbol{z}_j,\ j = 1,\ldots,M$, there is now a larger set of possible candidate exemplars with fixed coordinates. Learning is performed by optimizing over indicator variables, selecting a sparse subset of the candidates.

This reparametrization makes the problem convex but also changes the regularization: whereas usually the number of exemplars M is the main regularization parameter, it is now an implicit guarantee on the quality of the solution. In (Tipping & Schölkopf, 2001) this is the maximum distortion h, whereas in (Lashkari & Golland, 2007) the regularization parameter β controls the smoothness of the density.

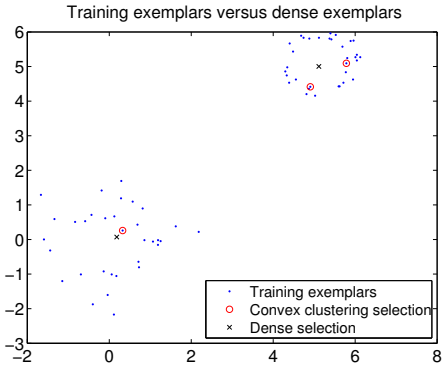

Figure 1. Exemplar selection within the training set versus the finest dense set of 900 exemplars on a regular grid. In this toy example, there are 66 data points.

2.2. Motivating Experiment: More Exemplars

Restricting the set of possible prototype candidates to the training set might result in a suboptimal solution if there is no exemplar close to the true mean of a cluster. If the data is low-dimensional, normal-distributed within each cluster, has low noise and there are enough training examples, this effect is small and can be ignored. But in high dimensions the true mean might be far away from any exemplar.

To demonstrate the effect of restricting the prototype candidate set we perform an experiment. A simple two-dimensional data set is created by sampling from an isotropic Gaussian and a ring of uniform density, forming two well-separated clusters, see Figure 1. We compare convex clustering (Lashkari & Golland, 2007) with a modified model where the objective is changed to $\frac{1}{N}\sum_{i=1}^{N}\log\left[\sum_{j=1}^{M} q_j e^{-\beta d_\phi(\boldsymbol{x}_i,\boldsymbol{z}_j)}\right]$, with N training samples $\boldsymbol{x}_i$ and M cluster center candidates $\boldsymbol{z}_j$. This convex objective still represents the log-likelihood of the training samples under a mixture model. We generate $\boldsymbol{z}_j$ by densely discretizing the $[-2;7]^2$ box on a regular grid. Our hope is that a fine discretization will increase the chance that $\{\boldsymbol{z}_j\}_{j=1,\ldots,M}$ contains exemplars close to the true center of each cluster. For both models we use an isotropic multivariate normal distribution with covariance matrix $\Sigma = \sigma^2 I$, $\sigma = 2.5$.

The clustering result is shown in Figure 1. For the cluster around the origin there is indeed a training set exemplar close to the mean of the generating Gaussian and the difference between the convex clustering and dense selection is small. However, for the ring-like structure, the training set exemplars cannot represent the cluster center adequately. This causes convex clustering to select two exemplars, while in the dense set a single good candidate is selected. A slight perturba-

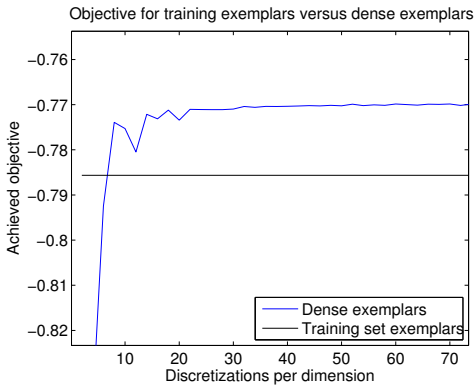

Figure 2. Training set vs. dense set log-likelihood.

tion in the training data would lead to a different selection by the convex clustering method, as all samples bordering to the interior of the ring are roughly equally bad. For this data set, the solution produced by convex clustering is not only qualitatively disappointing but also unstable. The achieved objectives are shown in Figure 2, where the convex clustering objective is drawn as horizontal line and the dense exemplar model forms a curve as the discretization becomes finer and finer. At around eight discretizations per dimension our modified model surpasses the log-likelihood of the convex clustering model. At around 30 discretizations per dimension the log-likelihood levels out and adding more cluster candidates does not improve the solution.

This experiment suggests that a larger set of candidate clusters can lead to higher quality results which are also more robust. While dense discretization is only feasible in case the input space is low-dimensional, ideally we would like to use an infinitely fine discretization and thus use the set of *all* possible input points as candidates. This idea will be the basis for our method.

3. A Decoupled Model

We now introduce our model for unsupervised learning together with an efficient solution algorithm. Essential to the solution is the ability to solve a certain subproblem which we analyze in detail.

3.1. Model

Our model for unsupervised learning generalizes convex clustering (Lashkari & Golland, 2007) and kernel vector quantization (Tipping & Schölkopf, 2001). Let $k_{\boldsymbol{z}}(\cdot)$ be a non-negative smoothing kernel centered at $\boldsymbol{z} \in \mathcal{Z}$, with $\mathcal{Z} \subseteq \mathcal{X}$. Let $\{\boldsymbol{x}_i\}_{i=1,\dots,N}$, $\boldsymbol{x}_i \in \mathcal{X}$ denote the training set. The following semi-infinite convex programming problem learns a convex combination of response functions such that an objective is minimized.

$$\min_{\boldsymbol{q},\gamma,\rho} \quad \Omega(\boldsymbol{\gamma}, \rho) \tag{4}$$

$$\text{sb.t.} \quad \int_{\mathcal{Z}} q_{\boldsymbol{z}} k_{\boldsymbol{z}}(\boldsymbol{x}_i)\, d\boldsymbol{z} = \gamma_i : \alpha_i,\ i = 1, \dots, N \tag{5}$$

$$\rho \le \gamma_i : \omega_i, \quad i = 1, \dots, N, \tag{6}$$

$$q_{\boldsymbol{z}} \ge 0 : \mu_{\boldsymbol{z}}, \quad \forall \boldsymbol{z} \in \mathcal{Z}, \tag{7}$$

$$\int_{\mathcal{Z}} q_{\boldsymbol{z}}\, d\boldsymbol{z} = 1 : \sigma, \tag{8}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\omega}$, $\boldsymbol{\mu}$ and σ are the Lagrange multipliers for the respective constraints. Before discussing the choice of objective function Ω, let us discuss the purpose of the constraints.

- Constraint (5) evaluates a convex combination of responses for each sample. γ_i contains the combined response for sample $\boldsymbol{x}_i$.
- Constraint (6) identifies – if $\nabla_\rho \Omega(\boldsymbol{\gamma}, \rho) < 0$ – the lowest response among all samples. The value of the lowest combined response is ρ.
- Constraints (7) and (8) define the combination simplex of the response functions.

For the special case where $\mathcal{Z}$ is a finite set of points in $\mathcal{X}$, we can replace the integrals and infinite constraints with a finite sum and finite set of constraints, respectively. Constraints (5) can then be compactly written as $K\boldsymbol{q} = \boldsymbol{\gamma}$, where K is a $(N, |\mathcal{Z}|)$ matrix storing the kernel responses. The dual problem of (4) can be derived from the conjugate function $\Omega^*(\boldsymbol{\alpha}, \sigma, \boldsymbol{\omega}, \boldsymbol{\mu})$ and its respective domain (Boyd & Vandenberghe, 2004, result (5.11)). The dual problem is

$$\max_{\boldsymbol{\alpha},\sigma,\boldsymbol{\omega},\boldsymbol{\mu}} \quad -\Omega^*(\boldsymbol{\alpha}, \sigma, \boldsymbol{\omega}, \boldsymbol{\mu}) - \sigma \tag{9}$$

$$\text{sb.t.} \quad (\boldsymbol{\alpha}, \sigma, \boldsymbol{\omega}, \boldsymbol{\mu}) \in \mathbf{dom}(\Omega^*),$$

$$\sum_{i=1}^{N} \alpha_i k_{\boldsymbol{z}}(\boldsymbol{x}_i) \ge \mu_{\boldsymbol{z}} - \sigma, \quad \forall \boldsymbol{z} \in \mathcal{Z}$$

$$\boldsymbol{\omega} \ge \mathbf{0} \tag{10}$$

$$\mu_{\boldsymbol{z}} \ge \mathbf{0}, \quad \forall \boldsymbol{z} \in \mathcal{Z}$$

We propose the following choices of convex objective functions $\Omega(\boldsymbol{\gamma}, \rho)$.

1. $\Omega(\boldsymbol{\gamma}, \rho) = -\rho$

   The objective states that the lowest response among all samples is to be maximized. All other samples have equal or higher responses but are ignored by this objective, hence a single exemplar can have a large influence on the overall objective. The KVQ problem (1) corresponds to this

objective with K chosen as discussed in section 2. The conjugate is $\Omega^*(\boldsymbol{\alpha}, \sigma, \boldsymbol{\omega}, \boldsymbol{\mu}) = 0$ and domain $\mathbf{dom}(\Omega^*) = \{(\boldsymbol{\alpha}, \sigma, \boldsymbol{\omega}, \boldsymbol{\mu}) : \boldsymbol{\omega} + \boldsymbol{\alpha} \leq \mathbf{0}, \boldsymbol{\omega}^\top \mathbf{1} = 1\}$. With (10) we have $\boldsymbol{\alpha} \leq \mathbf{0}$.

2. $\Omega(\boldsymbol{\gamma}, \rho) = -\frac{1}{N} \sum_{i=1}^N \log(\gamma_i)$

   This objective maximizes $\prod_{i=1}^N \gamma_i$. For the special case where the columns of K correspond to evaluations of probability density functions at the training samples this objective maximizes the log-likelihood of the samples under a mixture model, resulting in convex clustering (2). A single exemplar can have a significant effect on the overall objective, but all sample responses are considered, contrasting the previous objective function. The conjugate is $\Omega^*(\boldsymbol{\alpha}, \sigma, \boldsymbol{\omega}, \boldsymbol{\mu}) = -\frac{1}{N} \sum_{i=1}^N \log(-\alpha_i) + \log(N)$ with domain $\mathbf{dom}(\Omega^*) = \{(\boldsymbol{\alpha}, \sigma, \boldsymbol{\omega}, \boldsymbol{\mu}) : \boldsymbol{\alpha} < \mathbf{0}, \boldsymbol{\omega} = \mathbf{0}\}$.

3. $\Omega(\boldsymbol{\gamma}, \rho) = -\rho + \frac{C}{N} \sum_{i=1}^N (\gamma_i - \rho)^2$

   The objective maximizes the margin ρ while penalizing large deviations from the margin, where the penalty strength is determined by $C \geq 0$. The objective may prefer a smaller margin if the corresponding choice of $\boldsymbol{q}$ leads to a more uniform γ_i. This *margin-minus-variance* (MMV) objective was first proposed in (Rückert & Kramer, 2006) for supervised learning.

4. $\Omega(\boldsymbol{\gamma}, \rho) = -\frac{1}{N} \sum_{i=1}^N \gamma_i + \frac{C}{N} \sum_{i=1}^N (\gamma_i - \frac{1}{N} \sum_{i=1}^N \gamma_i)^2$

   The objective maximizes the mean response while penalizing large deviations from it, where the penalty strength is determined by $C \geq 0$. This maximizes the *mean-minus-variance* popular in applications such as portfolio optimization, see for example (Cornuejols & Tütüncü, 2007).

In order to be able to compare our method with established methods from the literature we only use the first two objectives in the experiments.

3.1.1. Relation to existing methods.

Most relevant for our approach is Boosting Density Estimation (Rosset & Segal, 2002). We note the following differences, i) our model includes different objectives, ii) in our solution algorithm, we will use totally-corrective weight updates[2] instead of a simple line-search procedure, and iii) we identify each *weak learner* uniquely with a point in input space. Also related is the hard-margin case of 1-class Boosting (Rätsch et al., 2001). With exemplar-based weak learners it is a special case of our model with the first objective.

[2]Totally-corrective steps update all weights individually in each iteration, leading to faster convergence.

Algorithm 1 Infinite Exemplar Column Generation

$(Z, \boldsymbol{q}) = \textsc{Infex}(X, \epsilon, k, Z_0)$
Input:
  Sample set $X = \{\boldsymbol{x}_i\}_{i=1,\dots,N}$, $\boldsymbol{x}_i \in \mathcal{X}$
  Convergence tolerance $\epsilon > 0$
  Non-negative smoothing kernel $k_{\boldsymbol{z}} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}_+$
  Initial exemplar set $Z_0 = \{\boldsymbol{z}_j\}_{j=1,\dots,|Z_0|}$, $\boldsymbol{z}_j \in \mathcal{Z}$
Output:
  Column exemplar set $Z = \{\boldsymbol{z}_j\}_{j=1,\dots,R}$, $\boldsymbol{z}_j \in \mathcal{Z}$
  Weightings $q_{\boldsymbol{z}_j} \in \mathbb{R}_+$, $j = 1, \dots, R$
Algorithm:
$\boldsymbol{\alpha} \leftarrow -\frac{1}{N}\mathbf{1}$, $Z \leftarrow Z_0$, $R \leftarrow |Z_0| + 1$, $\delta \leftarrow \infty$, $\sigma^* \leftarrow 0$
loop
  $\boldsymbol{z}_R \leftarrow \operatorname{argmax}_{\boldsymbol{z} \in \mathcal{Z}} - \sum_{i=1}^N \alpha_i k_{\boldsymbol{z}}(\boldsymbol{x}_i)$ {(SP)}
  $\delta \leftarrow \sigma^* - \sum_{i=1}^N \alpha_i k_{\boldsymbol{z}_R}(\boldsymbol{x}_i)$ {Compute $\nabla_{\boldsymbol{z}_R}$}
  if $\delta < \epsilon$ **then**
    break {convergence to tolerance}
  end if
  $Z \leftarrow Z \cup \{\boldsymbol{z}_R\}$
  $K \leftarrow \left[k_{\boldsymbol{z}_j}(\boldsymbol{x}_i)\right]_{i=1,\dots,N,\ j=1,\dots,R}$ {response matrix}
  p_R^*, $(\boldsymbol{q}_R^*, \boldsymbol{\gamma}^*, \rho^*)$, $(\boldsymbol{\alpha}^*, \boldsymbol{\omega}^*, \boldsymbol{\mu}_R^*, \sigma^*) \leftarrow$
    objective value, primal- and dual-solution to problem (4) with finite (N, R) matrix K.
  $R \leftarrow R + 1$
end loop

3.2. Algorithm

To solve problem (4), we propose Algorithm 1 (INFEX), a delayed column generation algorithm. The algorithm works with a finite and usually small set of candidate prototypes $\boldsymbol{z}_j$. This set is iteratively enlarged by adding good candidates. Selecting the candidates to add in each iteration becomes a subproblem, which we define now.

Problem 1 (Subproblem (SP)) *Given a set of samples $\boldsymbol{x}_i \in \mathcal{X}$, $i = 1, \dots, N$, a corresponding non-positive sample weighting $\alpha_i \leq 0$, $i = 1, \dots, N$ and a non-negative smoothing kernel $k_{\boldsymbol{z}}(\boldsymbol{x}) : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}_+$, obtain $\boldsymbol{z}^*$ as the solution of*

$$\boldsymbol{z}^* = argmax_{\boldsymbol{z} \in \mathcal{Z}} - \sum_{i=1}^N \alpha_i k_{\boldsymbol{z}}(\boldsymbol{x}_i).$$

The solution to this subproblem provides a candidate $\boldsymbol{z}^*$ that, when added to the set of considered candidates, will reduce the global objective.[3] We will now rigorously derive the subproblem from global optimality conditions of problem (4).

[3]In the optimization literature such columns are referred to as having *negative reduced cost*. The overall decoupled solution approach is closely related to the generalized Benders decomposition (Geoffrion, 1972).

Theorem 1 *Assume that the subproblem (SP) can be solved exactly in each iteration. Then Algorithm 1 solves problem (4) globally to the desired accuracy ϵ.*

Proof. Consider a slightly modified version of problem (4), where a part of the constraints (7) is replaced by equality constraints. We replace (7) by the following constraint set, parametrized by a finite set of points $Z_R = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{|Z_R|}\}$.

$$q_{\boldsymbol{z}} \geq 0 : \mu_{\boldsymbol{z}}, \qquad \forall \boldsymbol{z} \in Z_R, \tag{11}$$
$$q_{\boldsymbol{z}} = v_{\boldsymbol{z}} : \mu_{\boldsymbol{z}}, \qquad \forall \boldsymbol{z} \in \mathcal{Z} \setminus Z_R, \tag{12}$$

where $v_{\boldsymbol{z}} = 0$ is constant for all $\boldsymbol{z} \in \mathcal{Z} \setminus Z_R$. Together, constraints (11) and (12) restrict problem (4) such that only a finite subset of the variables $\boldsymbol{q}$ are used.

For a given finite Z_R, we can obtain an optimal primal $(\boldsymbol{q}^*, \boldsymbol{\gamma}^*, \rho^*)$, and dual $(\boldsymbol{\alpha}^*, \boldsymbol{\omega}^*, \boldsymbol{\mu}^*, \sigma^*)$ solution to the modified problem by solving a finite problem in the restricted set of variables $\{q_{\boldsymbol{z}} : \boldsymbol{z} \in Z_R\}$. Let the optimal function value of this solution be denoted by $p(\boldsymbol{v})$. Because the optimal solution must be feasible, we have $q^*_{\boldsymbol{z}} = v_{\boldsymbol{z}} = 0$ for all $\boldsymbol{z} \in \mathcal{Z} \setminus Z_R$. How would the objective function value $p(\boldsymbol{v})$ change if we force a $q^*_{\boldsymbol{z}}$ to become non-zero? That is, if we increase $v_{\boldsymbol{z}}$ by a very small amount can we improve the solution? The sensitivity theorem (Bertsekas, 1999, Proposition 3.3.3) provides a definite answer, namely we have for all $\boldsymbol{z} \in \mathcal{Z} \setminus Z_R$ the following.

$$\nabla_{v_{\boldsymbol{z}}} p(\boldsymbol{v}) = -\mu^*_{\boldsymbol{z}}.$$

If we have for all $\boldsymbol{z} \in \mathcal{Z} \setminus Z_R$ that $\nabla_{v_{\boldsymbol{z}}} p(\boldsymbol{v}) \geq 0$, then this implies that we can not decrease $p(\boldsymbol{v})$ by making $q_{\boldsymbol{z}} > 0$. Conversely, this observation provides us with a *global optimality condition*: if and only if Z_R contains all relevant (positive $q_{\boldsymbol{z}}$) exemplars, we have $\forall \boldsymbol{z} \in \mathcal{Z} \setminus Z_R : \mu^*_z \leq 0$. Given Z_R and a primal-dual optimal solution we can find an alternative expression for $\mu^*_{\boldsymbol{z}}$. Consider the Lagrangian of the modified problem.

$$\begin{aligned}\mathcal{L}(\boldsymbol{q}, \boldsymbol{\gamma}, \rho, \boldsymbol{\alpha}, \boldsymbol{\omega}, \boldsymbol{\mu}, \sigma) &= \Omega(\boldsymbol{\gamma}, \rho) \\ &+ \sum_{i=1}^{N} \alpha_i \left(\int_{\mathcal{Z}} q_{\boldsymbol{z}} k_{\boldsymbol{z}}(\boldsymbol{x}_i)\, d\boldsymbol{z} - \gamma_i \right) + \boldsymbol{\omega}^\top (\rho \mathbf{1} - \boldsymbol{\gamma}) \\ &- \sum_{\boldsymbol{z} \in Z_R} \mu_{\boldsymbol{z}} q_{\boldsymbol{z}} + \int_{\mathcal{Z} \setminus Z_R} \mu_{\boldsymbol{z}} q_{\boldsymbol{z}}\, d\boldsymbol{z} \\ &+ \sigma\Big(\sum_{\boldsymbol{z} \in Z_R} q_{\boldsymbol{z}} + \int_{\mathcal{Z} \setminus Z_R} q_{\boldsymbol{z}}\, d\boldsymbol{z} - 1 \Big)\end{aligned}$$

Because of optimality of the solution, it must satisfy the Karush-Kuhn-Tucker necessary conditions (Bertsekas, 1999), therefore we must have a zero gradient with respect to the primal variables. Specifically, for all $\boldsymbol{z} \in \mathcal{Z} \setminus Z_R$ we must have $\nabla_{q_{\boldsymbol{z}}} \mathcal{L}(\boldsymbol{q}^*, \boldsymbol{\gamma}^*, \rho^*, \boldsymbol{\alpha}^*, \boldsymbol{\omega}^*, \boldsymbol{\mu}^*, \sigma^*) = \sum_{i=1}^{N} \alpha^*_i k_{\boldsymbol{z}}(\boldsymbol{x}_i) + \mu^*_{\boldsymbol{z}} + \sigma^* = 0$. This allows us to express $\mu^*_{\boldsymbol{z}}$ as

$$\mu^*_{\boldsymbol{z}} = \sigma^* - \sum_{i=1}^{N} \alpha^*_i k_{\boldsymbol{z}}(\boldsymbol{x}_i). \tag{13}$$

Therefore, if for all $\boldsymbol{z} \in \mathcal{Z} \setminus Z_R$ we have dual feasible $\mu^*_z \leq 0$, then the current solution is optimal, despite the restrictions imposed by constraints (12). If we satisfy the optimality condition, then replacing (12) with constraints (11), does not change the solution, which remains optimal in the original problem (4).

What remains to be shown is that Algorithm 1 makes progress in each iteration and thus in the limit will satisfy the optimality condition. Consider the case where the above optimality condition is violated for one or more $\boldsymbol{z} \in \mathcal{Z} \setminus Z_R$. Then, let $\boldsymbol{z}^* = \operatorname{argmax}_{\boldsymbol{z} \in \mathcal{Z} \setminus Z_R} \left(\sigma^* - \sum_{i=1}^{N} \alpha^*_i k_{\boldsymbol{z}}(\boldsymbol{x}_i) \right)$ be the sample corresponding to the most negative partial derivative $\nabla_{v_{\boldsymbol{z}^*}} p(v) < 0$. Because of the sensitivity theorem, adding $\boldsymbol{z}^*$ to Z_R – making $q_{\boldsymbol{z}^*}$ a free variable – and re-solving (4) will reduce the objective value. Therefore, *either* no $\boldsymbol{z}^*$ with $\nabla_{v_{\boldsymbol{z}^*}} p(v) < -\epsilon$ is found and convergence to the tolerance is established, *or* a strict decrease in the objective is obtained. □

Note that in practice, we can add multiple exemplars in each iteration. Suppose during solving the subproblem (SP) we obtain a number of good local maximizers. Then, we can add all these local maximizers in order to obtain a faster convergence. Adding redundant exemplars with $\nabla_{v_{\boldsymbol{z}}} p(v) > 0$ does not have an effect as they will receive a zero weight $q_{\boldsymbol{z}} = 0$.

3.3. On the Nature of the Subproblem

The subproblem (SP) is completely determined by the negative weighting of the training set and the shape of the smoothing kernel function. For further discussion let us define $\eta_i = -\alpha_i$ and rewrite the subproblem as $\operatorname{argmax}_{\boldsymbol{z} \in \mathcal{Z}} \sum_{i=1}^{N} \eta_i k(\boldsymbol{x}_i, \boldsymbol{z})$. From the definition it follows that all η_i are non-negative. Clearly, this problem is non-concave whenever k is non-concave in $\boldsymbol{z}$ which is true for all smoothing functions we consider.

However, for kernel functions of the form $k_{\boldsymbol{z}}(\boldsymbol{x}) = k(\|\boldsymbol{x} - \boldsymbol{z}\|)$, the optima of the subproblem, thus the new candidates, are located at the *modes* of the expansion $\sum_{i=1}^{N} \eta_i k_{\boldsymbol{z}}(\boldsymbol{x}_i)$. It is this fact that can be exploited to efficiently solve the subproblem by standard hill-climbing algorithms. Such algorithms start at a point $\boldsymbol{z}^{(0)}$ in input space and generate iteratively better candidates such that $\sum_{i=1}^{N} \eta_i k_{\boldsymbol{z}^{(t+1)}}(\boldsymbol{x}_i) > \sum_{i=1}^{N} \eta_i k_{\boldsymbol{z}^{(t)}}(\boldsymbol{x}_i)$. In this paper, we use the *weighted*

mean shift procedure which was introduced by (Fukunaga & Hostetler, 1975; Cheng, 1995) and gained popularity due to (Comaniciu & Meer, 2002). Given an initial starting point $\boldsymbol{z}^{(0)}$ the iterates are produced by

$$\boldsymbol{z}^{(t+1)} = \frac{\sum_{i=1}^{N} \alpha_i g\left(\left\|\frac{\boldsymbol{z}^{(t)}-\boldsymbol{x}_i}{h}\right\|^2\right)\boldsymbol{x}_i}{\sum_{i=1}^{N} \alpha_i g\left(\left\|\frac{\boldsymbol{z}^{(t)}-\boldsymbol{x}_i}{h}\right\|^2\right)}, \qquad (14)$$

where $g : \mathbb{R}_+ \to \mathbb{R}_+$ is the negative derivative of the so called *kernel profile*. If for a continuous kernel the function g is convex and non-increasing, then the mean shift procedure is guaranteed to converge to a local maxima (Comaniciu & Meer, 2002). For each of the common continuous smoothing kernels, a unique function g exists and some popular kernels and their profile derivatives are discussed in section 4. For the Gaussian kernel, g is a scaled version of the original kernel profile and thus particularly easy to maximize.[4] Mean shift is popular in computer vision, where specialized procedures have been developed to efficiently find globally good modes, for example the *annealed mean shift* procedure (Shen et al., 2007).

If the smoothing kernel function is a reproducing Hilbert kernel (Schölkopf & Smola, 2002), then problem (SP) is known as the *pre-image problem* (Schölkopf et al., 1999). An important difference which simplifies our subproblem considerably is that all our weights $\boldsymbol{\alpha}$ are of the same sign. In the general pre-image problem the sign is not fixed and procedures such as the one of (Schölkopf et al., 1999) can be unstable and do not have a convergence guarantee.

3.4. Optimality Bound

The proof of global optimality of the solution obtained by Algorithm 1 was based on the assumption that the subproblem (SP) can be solved globally. We now show that even without this assumption, the method can be no worse than methods using a fixed exemplar set.

Theorem 2 *Given $\Omega(\boldsymbol{\gamma}, \rho)$, a set $X = \{\boldsymbol{x}_i\}_{i=1,\dots,N}$, $\boldsymbol{x}_i \in \mathcal{X}$ and a finite set of exemplars $Z_F = \{\boldsymbol{z}_j\}_{j=1,\dots,M}$, the solution obtained by solving problem (4) with $\mathcal{Z} = Z_F$ can not achieve a better objective than the solution obtained by Algorithm 1 with $\mathcal{Z} = \mathcal{X}$, $Z_0 = Z_F$.*

[4]The Gaussian kernel has received special attention in the literature. In (Carreira-Perpiñán, 2000) it was conjectured that the number of modes in a Gaussian mixture is bounded above by the number of components. While this is true in the univariate case, this has been proven wrong in general in (Carreira-Perpiñán & Williams, 2003). See also the counter-example at `http://www.inference.phy.cam.ac.uk/mackay/gaussians/`.

Proof. Let Algorithm 1 be called with $Z_0 = Z_F$. In the first iteration of Algorithm 1, the solved problem is identical to problem (4) with $\mathcal{Z} = Z_F$. Therefore, after the first iteration, the objective of Algorithm 1 is equal to the one obtained by solving problem (4). In all later iterations, the objective can only improve. □

4. Experiments and Results

For the following experiments, we solve the restricted master problem (4) using IpOpt (Wächter & Biegler, 2006), a modern primal-dual interior point solver for non-linear programming available as open-source. For each master problem, we obtain accurate convergence in a few dozen solver iterations. We use tolerances 10^{-10} for the restricted master problem and 10^{-7} for the subproblems for all experiments.[5]

As smoothing kernels we use the unnormalized Gaussian, the unnormalized Epanechnikov, and a simple uniform disc kernel. All are parametrized by a bandwidth parameter h. The following are the kernel functions k and profiles g used in the mean shift procedure.

1. Gaussian, bandwidth h
$$k_{\boldsymbol{z}}(\boldsymbol{x}) = e^{-\frac{1}{2}\left\|\frac{\boldsymbol{x}-\boldsymbol{z}}{h}\right\|^2}, \quad g(y) = \frac{1}{2}e^{-\frac{1}{2}y}$$
2. Epanechnikov, bandwidth h
$$k_{\boldsymbol{z}}(\boldsymbol{x}) = \begin{cases} 1 - \left\|\frac{\boldsymbol{x}-\boldsymbol{z}}{h}\right\|^2 & \left\|\frac{\boldsymbol{x}-\boldsymbol{z}}{h}\right\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$
$$g(y) = \begin{cases} 1 & 0 \le y \le 1 \\ 0 & y > 1 \end{cases}$$
3. Uniform disc, maximum distortion h
$$k_{\boldsymbol{z}}(\boldsymbol{x}) = \begin{cases} 1 & \left\|\frac{\boldsymbol{x}-\boldsymbol{z}}{h}\right\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

The first two kernels are common in non-parametric density estimation, whereas the last one is used by (Tipping & Schölkopf, 2001) for vector quantization. We use the mean shift procedure (14) started from all training samples to solve the subproblem (SP) for the Gaussian and Epanechnikov kernels. We collect the result of each run and add the set of unique local maximizers to the restricted master problem.

However, mean shift cannot be used to solve subproblem (SP) for the non-continuous uniform disc kernel. Instead, when using the uniform disc kernel, we find new codebook candidates by solving the subproblem with the Epanechnikov kernel instead. This is a reasonable approximation as the Epanechnikov kernel response lower bounds the uniform disc kernel response and its maximum lies in the center of the disc.

[5]Our implementation is available at `http://www.kyb.mpg.de/bs/people/nowozin/infex/`.

4.1. Comparison with KVQ

In the first experiment we compare the original Kernel Vector Quantization formulation (1) with all training exemplars as possible prototypes with our Algorithm 1, where the initial set is empty, $Z_R = \emptyset$. We use the first objective $\Omega(\boldsymbol{\gamma}, \rho) = -\rho$ and the uniform disc kernel. As dataset we use a subset of 1100 exemplars from the USPS digit machine learning dataset, with all labels removed and each class sampled equally such that there are 110 exemplars from each class. We evaluate by selecting the maximum allowed distortion h from $\{800, 1000, 1200, 1400, 1600, 1800, 2000\}$, where $\approx$ 2000 is the mean inter-class L_2-distance in the dataset. We compare the achieved margin $\rho^*_{\mathrm{KVQ}}(h)$ with $\rho^*_{\mathrm{INFEX}}(h)$, and the number of codebook vectors $\|\boldsymbol{q}^*_{\mathrm{KVQ}}\|_0$ with $\|\boldsymbol{q}^*_{\mathrm{INFEX}}\|_0$. Figures 3 and 4 show these as the maximum allowed distortion is varied.

The proposed method outperforms KVQ, selecting a smaller number of codebook vectors and achieving a better objective value. Especially for larger allowed distortions, the benefit of selecting an arbitrary point in input space is substantial as due to the high dimensionality of the data set all input samples are relatively far away from each other. Because we use $Z_R = \emptyset$ to initialize our method, the results show that our subproblem approximation using the Epanechnikov kernel is an effective way to find good codebook candidates.

4.2. Comparison with Gaussian Mixture EM

In the second experiment we consider mixture model density estimation and compare our method with Convex Clustering and a homoscedastic Gaussian mixture ($\Sigma = \sigma^2 I$) learned with Expectation Maximization (EM).[6] The log-likelihood objective and the same USPS dataset as before is used. The experimental protocol is as follows. For a range of bandwidths our model and convex clustering are run once per bandwidth. For each run, the number of components of our model is used to fix the number of components in the Gaussian mixture model, which is trained by EM starting 20 times from random initial sample points.

The results are shown in Table 1. Clearly, a single run of our model is consistently the best. The best EM run is always close to our result and Convex Clustering is always the worst. (Lashkari & Golland, 2007) mention that their solution "can be improved in practice with a few extra steps of the EM algorithm". From Table 1, we conclude that the results of convex clustering are qualitatively inferior to plain EM and such *refitting* is actually essential for obtaining good results.

[6] A similar experiment is in (Lashkari & Golland, 2007).

4.3. Subproblem Modes

In the last experiment we show the qualitative behavior of our model with the Epanechnikov kernel with $h = 1500$ and the log-likelihood objective. Because the Epanechnikov kernel has finite support, if we start with $Z_0 = \emptyset$ we could have some samples $\boldsymbol{x}_i$ which have zero response because $k_{\boldsymbol{z}_j}(\boldsymbol{x}_i) = 0$ for all j. Then, the restricted variables $\boldsymbol{q}_j$ are too few and problem (4) would be infeasible. Thus, in order to ensure feasibility of the initial master problems, we use $Z_0 = X$. Some subproblem modes are shown in Figure 5. The modes approximate the "natural" clusters well except for classes such as 3, 8 and 9, which seem to be explained by one joint region with many local modes in it, for example in the first and second row.

5. Discussion and Conclusion

We presented a unifying perspective on existing exemplar based methods that aim at density estimation, clustering and vector quantization. Existing methods were either non-convex or achieved convexity by severe restrictions. In contrast, our approach – although still non-convex as a whole – is provable better than all existing methods. This is achieved by isolating a non-convex but still efficient solvable subproblem. The non-convex subproblem is embedded into a convex master problem steering towards an optimal solution.

One limitation of our model is that one cannot fix $\|\boldsymbol{q}^*\|_0$, the number of components. For problems where guarantees such as maximum distortion or smoothness are more natural constraints, this is not an issue.

There are open questions that result from our work:

1. Does there exists a response function k that is useful for unsupervised learning and at the same time yields a globally solvable subproblem?
2. What is the relation between objective Ω, kernel k and number of components $\|\boldsymbol{q}^*\|_0$?

Table 1. Achieved log-likelihoods. CC is Convex Clustering; for EM the best and mean of 20 runs are shown.

σ	CC	INFEX	EM BEST	EM MEAN
440	−6.3356	**-5.1370**	−5.1442	−5.1485
460	−6.1269	**-4.7424**	−4.7486	−4.7503
480	−5.8705	**-4.3796**	−4.3823	−4.3834
500	−5.5813	**-4.0499**	−4.0507	−4.0520
520	−5.2780	**-3.7499**	−3.7502	−3.7512
540	−4.9779	**-3.4788**	−3.4789	−3.4795

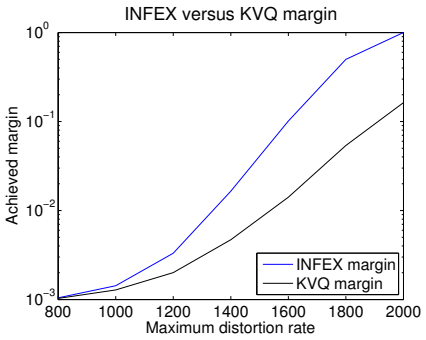

Figure 3. Optimal margin ρ^* as a function of the maximum allowed distortion. Note the log-scale.

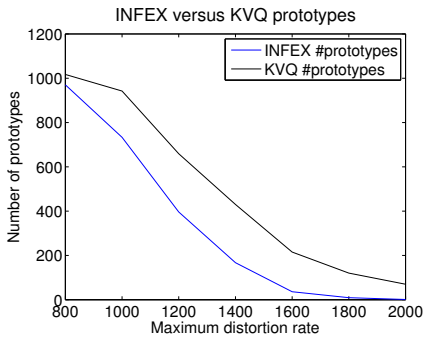

Figure 4. The number of selected prototypes as a function of the maximum allowed distortion.

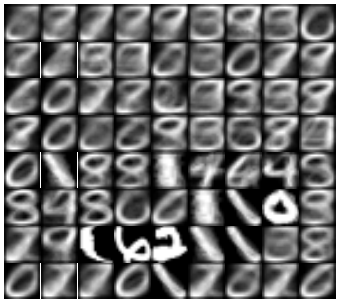

Figure 5. Subproblem modes found in different iterations.

3. Can a decomposition similar to ours yield a training scheme for supervised learning of RBF networks in the line of (Bengio et al., 2005)?

Acknowledgments

This work is funded in part by the EU CLASS project, IST 027978. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

Banerjee, A., Merugu, S., Dhillon, I. S., & Ghosh, J. (2005). Clustering with bregman divergences. *Journal of Machine Learning Research*, *6*, 1705–1749.

Bengio, Y., Roux, N. L., Vincent, P., Delalleau, O., & Marcotte, P. (2005). Convex neural networks. *NIPS*.

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific. 2nd edition.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Carreira-Perpiñán, M. Á. (2000). Mode-finding for mixtures of gaussian distributions. *IEEE Trans. Pattern Anal. Mach. Intell*, *22*, 1318–1323.

Carreira-Perpiñán, M. Á., & Williams, C. K. I. (2003). An isotropic gaussian mixture can have more modes than components.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *17*, 790–799.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell*, *24*, 603–619.

Cornuejols, G., & Tütüncü, R. (2007). *Optimization methods in finance*. Mathematics, Finance and Risk.

Fukunaga, K., & Hostetler, L. D. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory*, *21*, 32–40.

Geoffrion, A. M. (1972). Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, *10*, 237–260.

Lashkari, D., & Golland, P. (2007). Convex clustering with exemplar-based models. *NIPS*.

Rätsch, G., Schölkopf, B., & Mika, S. (2001). SVM and boosting: One class.

Rosset, S., & Segal, E. (2002). Boosting density estimation. *NIPS* (pp. 641–648). MIT Press.

Rückert, U., & Kramer, S. (2006). A statistical approach to rule learning. *ICML* (pp. 785–792).

Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Mueller, K.-R., Raetsch, G., & Smola, A. J. (1999). Input Space versus Feature Space in Kernel-Based Methods. *IEEE-NN*, *10*, 1000.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. MIT Press.

Shen, C., Brooks, M. J., & van den Hengel, A. (2007). Fast global kernel density mode seeking: Applications to localization and tracking. *IEEE Transactions on Image Processing*, *16*, 1457–1469.

Tipping, M., & Schölkopf, B. (2001). A kernel approach for vector quantization with guaranteed distortion bounds. *AISTATS*.

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, *106*, 25–57.