Supervised Learning of Quantizer Codebooks by Information Loss Minimization

Svetlana Lazebnik, Maxim Raginsky

IEEE

Presented by Kunhao Li, Yuhong Li

December 6, 2019

Overview

1. Motivation
   - X: Continuous random variable, feature vector
   - K: Discrete random variable, codebook
   - Y: Discrete variable, label of X
   - Goal: Find an ideal quantizer to compute a sufficient statistic K of X for Y
   - $X \to K \to Y$
2. Approach
   - Empirical Approach
   - Constrain Encoder

Motivation

Since we are trying to find a quantizer that computes a sufficient stat of X for Y. The K we found, ideally, should contain as much information as X does. In other words, this quantizer minimizes the information loss at the quantization step, meaning:

$$\text{minimize } I(X;Y) - I(K;Y)$$

Approach

- Now our goal has become an optimization problem, the objective function is $I(X;Y) - I(K;Y)$
- Assume X, Y are jointly distributed, but the underlying distribution is unknown.
- The training samples are i.i.d drawn from the joint distribution of X and Y
- The suggested approach stated in the paper needs some background knowledge with empirical information loss minimization. So, we start from there.

Empirical Approach

Denote $P_x = P(y|X = x)$, $\mu_x = P(x)$.
Denote the marginal of Y: $P = \int_X P_x \mu_x$
Then the mutual information: $I(X;Y) = \int_X D(P_x||P)$
Suppose the set of X is partitioned into C disjoint sets, $R_1 ... R_C$
Then $I(K;Y) = \sum_{k=1}^{C} p_k(K) D(P_k||P)$

Empirical Approach

Since we don't know the underlying distribution, use the empirical version of the distributions on last slide. Use $\hat{P}$ to denote:

$$\hat{\mu}_{x_i} = T(x_i|N)/N \tag{1}$$

$$\hat{P}_{x_i} = T(y_i|T_k)/k \tag{2}$$

$$\hat{P} = \frac{1}{N}\sum_{i=1}^{N}\hat{P}_{x_i} \tag{3}$$

where T is the counting function, T_k is the set of k nearest neighbors of x_i in Knn rule.
(2) is obtained by counting the labels of k nearest neighbors around x_i

Empirical Approach

We get the empirical mutual information between X and Y

$$\hat{I}(X;Y) = \frac{1}{N}\sum_{i=1}^{N} D(\hat{P}_{x_i}||\hat{P}) \tag{4}$$

Empirical Approach

Suppose the set of X is partitioned into C disjoint sets, $R_1 \ldots R_C$
And define $K(x_i) = k$ if $x_i \in R_k$
Let $N_k = |R_k|$

$$\hat{P}_k = \pi_k = \frac{1}{N_k} \sum_{x_i \in R_k} \hat{P}_{x_i} \quad (5)$$

$$\hat{I}(K;Y) = \sum_{i=1}^{C} \frac{N_k}{N} D(\hat{P}_k || \hat{P}) \quad (6)$$

Empirical Approach

Rewrite (4) as

$$\hat{I}(X;Y) = \sum_{k=1}^{C} \frac{N_k}{N} \sum_{x_i \in R_k} \frac{1}{N_k} D(\hat{P}_{x_i} || \hat{P}) \tag{7}$$

Then

$$\begin{aligned}
&\sum_{x_i \in R_k} \frac{1}{N_k} D(\hat{P}_{x_i} || \hat{P}) - D(\hat{P}_k || \hat{P}) && (8) \\
&= \sum_{x_i \in R_k} \frac{1}{N_k} D(\hat{P}_{x_i} || \hat{P}) - \sum_{x_i \in R_k} \sum_{y \in Y} \frac{1}{N_k} \hat{P}_{x_i} log(\frac{\pi_k}{\hat{P}}) && (9) \\
&= \frac{1}{N_k} \sum_{x_i \in R_k} D(\hat{P}_{x_i} || \pi_k) && (10)
\end{aligned}$$

Empirical Approach

The objective function becomes

$$\hat{I}(X;Y) - \hat{I}(K;Y) = \frac{1}{N}\sum_{k=1}^{C}\sum_{x_i \in R_k} D(\hat{P}_{x_i} || \pi_k) \tag{11}$$

Empirical Approach

Clearly,

$$\pi_k = argmin_{\pi} \sum_{x_i \in R_k} D(\hat{P}_{x_i} || \pi_k) \tag{12}$$

And for fixed distribution of π, q_1 .. q_C, the best partition is

$$R_k \triangleq \{x_i : D(\hat{P}_{x_i} || q_k) \leq D(\hat{P}_{x_i} || q_j), j \neq k\}, k = 1..C \tag{13}$$

In other words, the right code has smallest divergence.
The optimization can be found by descent algorithm with some initialized π_k

Drawbacks

- Does not take advantage of the continuous structure of the feature space
- Encoding depends on the conditional distribution of training set P_x.

Constrain the Encoder

- Assume the data X comes from a compact subset X of Euclidean space R^d
- Encoding does not depend on the distribution of a given point.

Constrain the Encoder

The optimization problem can be rephrased as follows:

Seek a codebook $M = \{m_1, ...m_C\}$, and a set of associated posterior distribution $\prod = \pi_1, ..., \pi_C$ that jointly minimize:

$$\sum_{k=1}^{C} \sum_{x_i \in R(m_k)} D(P_{x_i} || \pi_k) \tag{14}$$

And the encoding rule becomes:

$$R(m_k) \triangleq \{x \in X : ||x - m_k|| \leq ||x - m_j||, \forall j \neq k\} \tag{15}$$

Constrain the Encoder

Note the new encoding rule does not involve label of x, and is thus suitable to encode unlabelled data, using MAP criterion:

$$\hat{Y} = argmax_{y \in Y} \pi_k(y) \tag{16}$$

Weakness

Although the objective function defined by (14) is a big improvement over (11), since it gives us a simple encoding rule that extends to unlabeled data, it is still unsatisfactory for computational reasons:

Optimize M for a given $\prod$ is a difficult combinatorial problem

A Suggested Method

Introduce a differentiable relaxation to the objective function, so the partition of sample set can be "soft".

Let $w_k(x)$ denote the "weight" of assignment of a point $x \in X$ to $R(m_k)$ with

$$\sum_{k=1}^{C} w_k(x) = 1 \tag{17}$$

As suggested by Rao, a natural choice of these weights is the Gibbs distribution:

$$w_k(x) = \frac{e^{-\beta||x-m_k||^2/2}}{\sum_j e^{-\beta||x-m_j||^2/2}} \tag{18}$$

β corresponds to the fuzziness of assignments. Smaller β corresponds to soft clustering, and infinite β yields hard clustering.

A Suggested Method

Now we get a suboptimal version of the objective function:

$$E(M, \prod) = \sum_{k=1}^{C} \sum_{x_i \in R(m_k)} w_k(x) D(P_{x_i} || \pi_k) \tag{19}$$

A Suggested Method

Find the local minimum for E using alternating minimization. First hold $\prod$ and update M using gradient descent:

$$m_k^{(t+1)} = m_k^{(t)} - \alpha \sum_{i=1}^{N} \sum_{j=1}^{C} D(P_{X_i} || \pi_j^{(t)}) \frac{\partial w_j^{(t)}(X_i)}{\partial m_k^{(t)}} \tag{20}$$

Where $\alpha > 0$ is the learning rate found using line search, and

$$\frac{\partial w_j^{(t)}(X_i)}{\partial m_k^{(t)}} = \beta[\delta_{jk} w_k(x) - w_k(x) w_j(x)](x - m_k) \tag{21}$$

Where δ_{jk} is 1 if $j = k$, and 0 otherwise

A Suggested Method

Then hold M and update $\prod$ using Lagrange multiplier:

$$J(M, \prod, \Lambda) = E(M, \prod) + \sum_k \lambda_k \sum_y \pi_k(y) \tag{22}$$

Set the partial derivative of J with respect to π_k to zero and solve λ_k. The resulting update of π_k is:

$$\pi_k^{(t+1)}(y) = \frac{\sum_{i=1}^N w_k^{(t+1)}(X_i) P_{X_i}(y)}{\sum_{y'} \sum_{i=1}^N w_k^{(t+1)}(X_i) P_{X_i}(y')}, \forall y \in Y \tag{23}$$

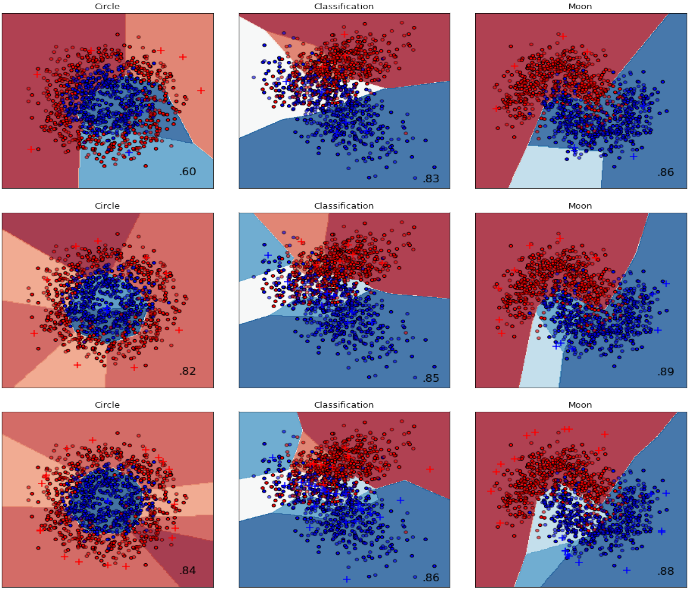

Figure: The results.

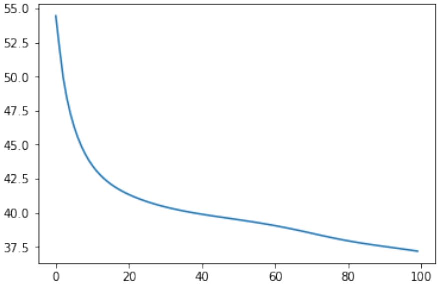

Figure: The Loss in first 100 epochs.