IT & ML: Channels, Quantizers, and Divergences

Yingzhen Li and Antonio Artés-Rodríguez

Department of Engineering

January 16, 2014

Outline

Change this

A Mathematical Theory of Communication

Unsupervised Vector Quantization

Supervised Quantization Design

Connecting the Quantizer and Classifier

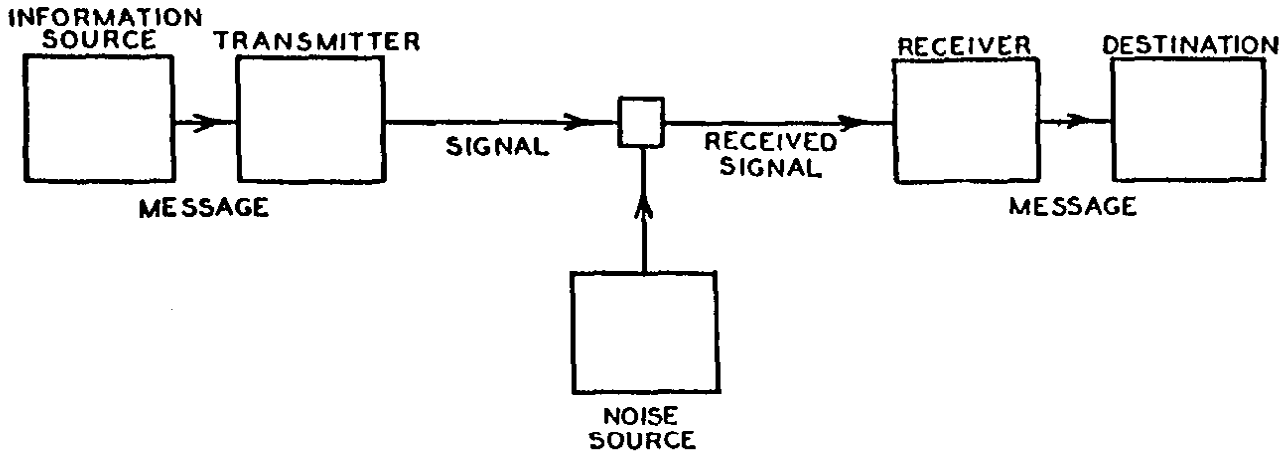
Conclusion

Machine Learning and Information Theory

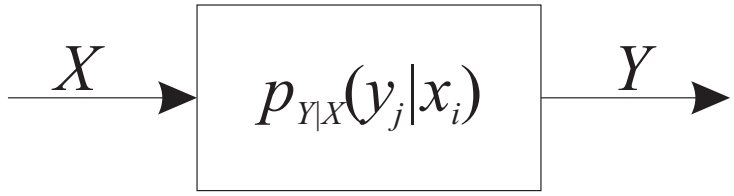
- ML techniques in IT: BP in turbo decoding and LDPC
- IT in ML: Information Bottleneck, feature extraction, ..
- Similar problems: Covariate shift and mismatched decoding, ...

A Mathematical Theory of Communication

A communication system model [Shannon, 1948]

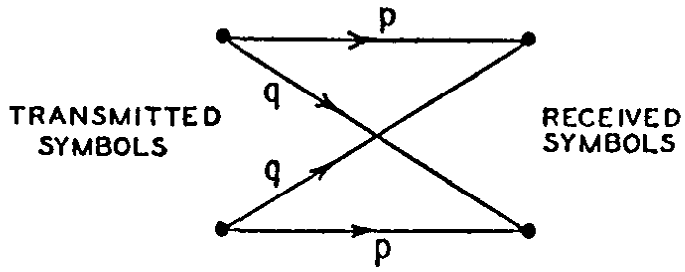


A particular noisy channel model is the DMC (Discrete Memoryless Channel), characterized by $(\mathcal{X}, P(Y|X), \mathcal{Y})$



Reliable communications

Example: transmitting one bit over the BSC Channel



Repetition code and majority voting, $q = 0.15$

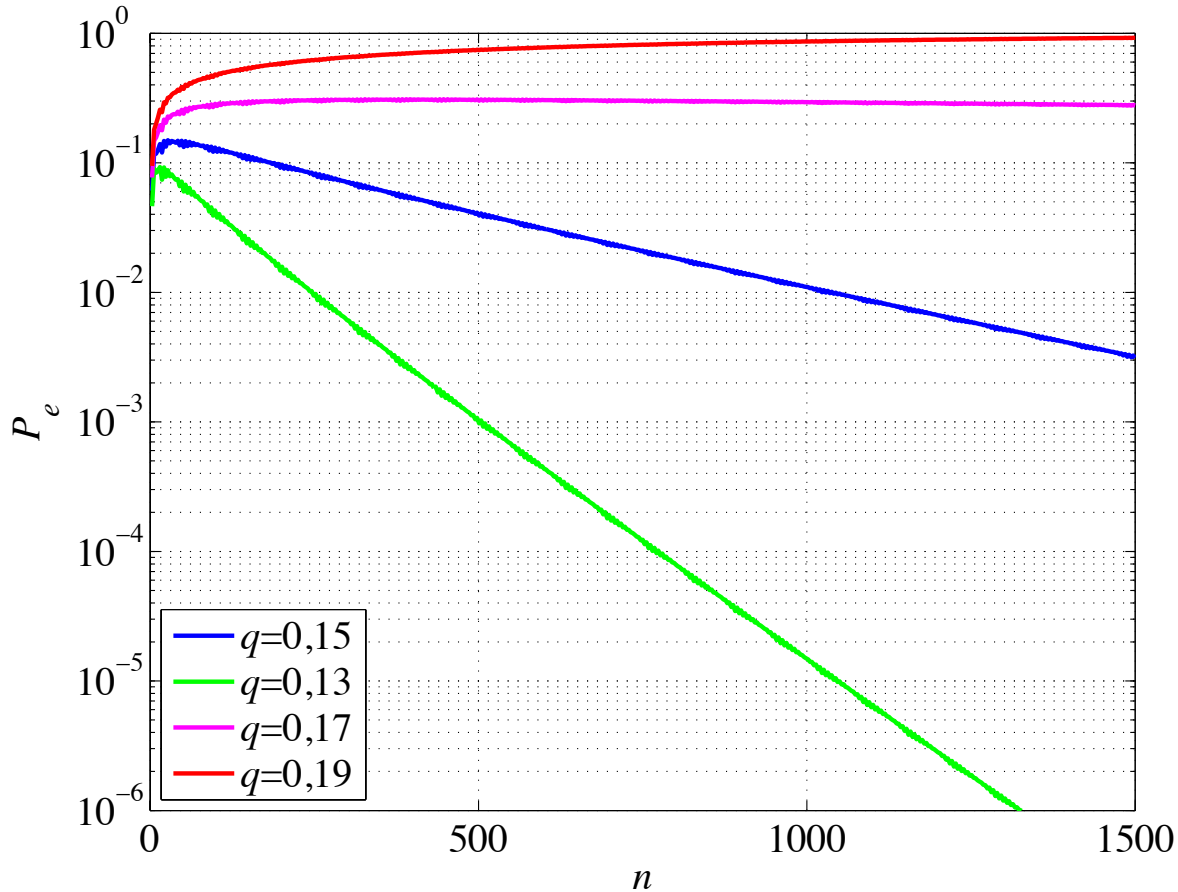
- $n = 1$, $P_e = q = 0.15$
- $n = 3$, $P_e = q^3 + 3q^2p = 0.0607$
- $n = 5$, $P_e = q^5 + 5q^4p + 10q^3p^2 = 0.0266$
- $n \to \infty$, $P_e = \sum_{l=0}^{(n-1)/2} \binom{n}{l} p^l q^{n-l} \to 0$

Limiting the rate

$$R = \frac{k}{n} = \frac{\#\text{ of transmitted symbols}}{\#\text{ of channel uses}}$$

It is possible to lower P_e fixing R and increasing $n = Rk$?

Example: Best Pe with $R = \frac{1}{3} = \frac{2}{6} = \frac{3}{9} = \cdots$



Achievable rate and channel capacity

- **Code**: a (k, n) code for the channel $(\mathcal{X}, P(Y|X), \mathcal{Y})$ is composed by:
  1. An index set $\{1, \ldots, 2^k\}$
  2. An encoding function $f_n : \{1, \ldots, 2^k\} \to \mathcal{X}^n$
  3. A decoding function $g_n : \mathcal{Y}^n \to \{1, \ldots, 2^k\}$
- $P_e(n, \max) = \max_{i \in \{1,\ldots,2^k\}} Pr(g_n(Y^n) = i | X^n = f_n(i))$
- **Achievable rate**: a rate R is achievable if, $\forall \epsilon > 0$ there exists a sequence of $(\lceil nR \rceil, n)$ codes and an n_0 such that $P_e(n, \max) < \epsilon$ when $n > n_0$
- **Channel capacity**: the capacity of a channel, C is the supremum of all achievable rates

Random coding bound

- f_n picked randomly: $f_j(i)$ iid from $p(z)$
- g_n: ML (MAP) decoder
- P_e averaged over all the possible encoders can be bounded [Gallager, 1965]

$$P_e \leq e^{-n(-\rho R + E_0(\rho, p(z)))} \leq e^{-nE(R, p(z))} \leq e^{-nE(R)}$$

$$E(R, p(z)) = \max_{\rho} \left[-\rho R + E_0(\rho, p(z))\right]$$

$$E(R) = \max_{p(z)} E(R, p(z))$$

- If $R < I(X;Y)|_{p(x)=p(z)}$, then $E(R, p(z)) > 0$. If we define $C_I = \max_{p(x)} I(X;Y)$, then

$$C = C_I$$

Mutual Information, KL Divergence and Entropy

- **Relative entropy** or **Kullback-Leibler divergence**

$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = E_p \left\{ \log \frac{p(x)}{q(x)} \right\}$$

- **Mutual information**:

$$I(X;Y) = D(p(x,y)||p(x)p(y)) = E_{X,Y} \left\{ \log \frac{p(x,y)}{p(x)p(y)} \right\}$$

Convex function of $p(y|x)$ if $p(x)$ is fixed.
Concave function of $p(x)$ if $p(y|x)$ is fixed.

- **Autoinformation** or **entropy** (diferential entropy):

$$H(X) = I(X;X) = E_X \left\{ \log \frac{1}{p(x)} \right\} \qquad (h(X) = I(X;X))$$

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

The IT game

1. Performance bounds
   - The goal is to obtain lower or upper bound on the performance on hard (sometimes NP-hard) problems
   - Bounds based on statistical measures (D, I, H, ...)
   - Some of the bounds are asymptotic
   - Generally, it does not say anyting about the solution to the problem
2. Optimization of statistical measures
   - Bounded or unbounded optimization of the statitical measure to achieve the bound
   - Generally, it provides some intuition about the original problem

Non-asymptotic bounds

- Non-asymptotic bounds like

  $$R^*(n,\epsilon) = \max\{R : \exists\, (\lceil nR \rceil, n) \text{ such that } P_e(n, \max) < \epsilon\}$$

  also rely on C_I [Polyanskiy, 2010] $\Longrightarrow$ **Optimization of statistical measures also makes sense in the non-asymtotic regime**
- For example, in the BSC

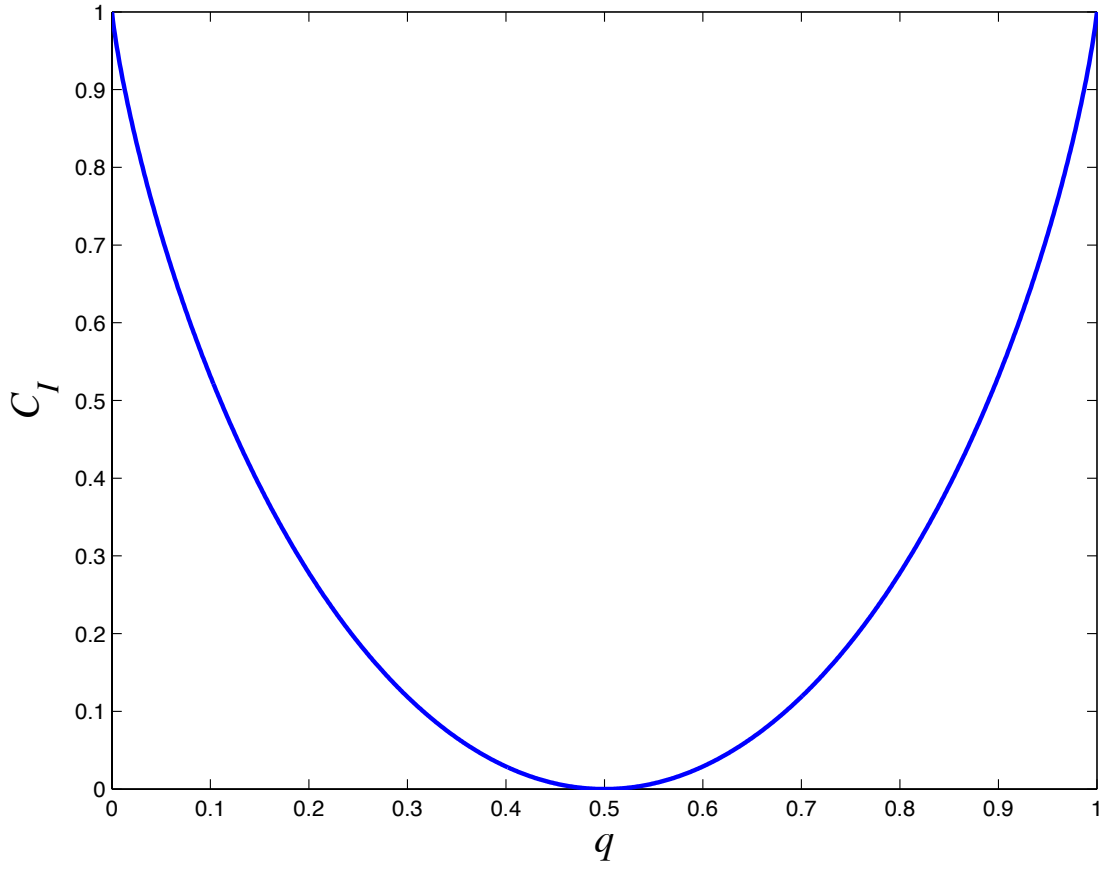
$$R^*(n,\epsilon) = nC_I - \sqrt{nV}\,Q^{-1}(\epsilon) + \frac{1}{2}\log n + O(1)$$

$$V = q(1-q)\log^2\frac{1-q}{q} \quad \text{(Channel dispersion)}$$

Bounds and optimization

MI maximization: KL minimization

Convexity



Channels in ML

Intrepretations of $P(Y|X)$

- Prior $P(X)$ and posterior $P(Y|X)P(X)$
- Latent information (M) and observations (X)
- Active learning (parallel channels)
- ...

Similar problems in IT

- Rate distortion theory (dual of noisy channel coding)
- Gambling and betting [Kelly, 1956]
- Portfolio management
- Compressed sensing [Donoho, 2006]

Rate Distortion Theory

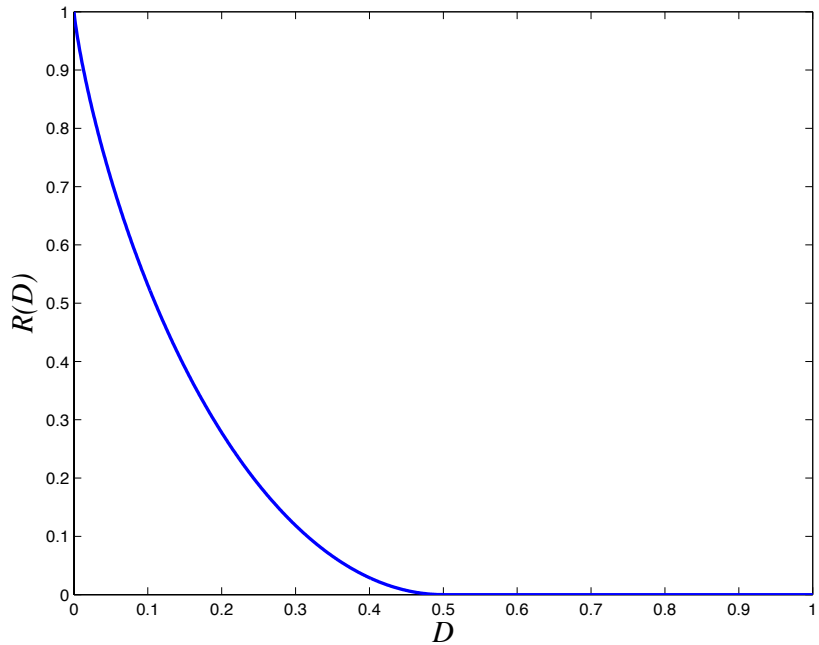
- **Rate-distortion code**: a $(2^{nR}, n)$ code for source X is composed by:
  1. An encoding function $f_n : \mathcal{X}^n \to \{1, \ldots, 2^{nR}\}$
  2. A decoding function $g_n : \{1, \ldots, 2^{nR}\} \to \hat{\mathcal{X}}^n$
- **Distortion**: $D_n = E_X \{d(X^n, g_n(f_n(X^n)))\}$
- **Distortion measure**: $d : \mathcal{X} \times \hat{\mathcal{X}} \to \mathcal{R}^+$
- **Achievable rate distortion**: a pair (R, D) is achievable if, $\forall \epsilon > 0$ there exists a sequence of $(2^{nR}, n)$ codes and an n_0 such that $D_n - D < \epsilon$ when $n > n_0$
- **Rate distortion region**: the closure of the set of all achievables (R, D)
- **Rate distortion curve** $R(D)$: the infimum of R such that (R,D) is in the rate distortion region for a given D

Rate Distortion function

$$R(D) = R_I(D) = \min_{p(\hat{x}|x):E_{X,\hat{X}}\{d(x,\hat{x})\}\leq D} I(X;\hat{X})$$

Example: binary source, *Bernouilli*(p), and Hamming distance

$$R_I(D) = \begin{cases} H_b(p) - H_b(D) & \text{if } 0 \leq D \leq \min(p, 1-p) \\ 0 & \text{if } D > \min(p, 1-p) \end{cases}$$



Distortion and divergences

- All the distances can be used directly as a distortion measures: euclidean, Mahalanobis, ...
- Divergences between x and $\hat{x} = g(f(x))$ can also be used as distortion measures: Bregman divergences

$$d_\phi(x, \hat{x}) = \phi(x) - \phi(\hat{x}) - \langle x - \hat{x}, \nabla \rangle$$

  - $\phi(x) = ||x||^2$: euclidean distance

Quantizer design

- It is easy to obtain g if f is fixed
- It is easy to obtain f if g is fixed
- Lloyd algorithm make iterative estimation of f and g

Quantization

- As channel coding
  - Transmitter encodes the (continuous) signal to (discrete) codes
  - Receiver decodes the and recovers that signal
- As feature extraction
  - Data pre-processing: to extract features for learning
  - To represent the density of data

Vector Quantization[1]

[1] See [Villmann and Haase, 2011] for details.

Unsupervised Vector Quantization

- Data points $\boldsymbol{v} \in V \subseteq \mathbb{R}^n$
  prototypes $W = \{\boldsymbol{w}_k\}, k \in Z, \boldsymbol{w}_k \in \mathbb{R}^n$
- Representing data points by clustering, via the distance ξ

$$\boldsymbol{v} \mapsto s(\boldsymbol{v}) := \arg\min_{k \in Z} \xi(\boldsymbol{v}, \boldsymbol{w}_k) \tag{1}$$

- The quantization error (loss function)

$$E_{VQ} = \frac{1}{2} \int P(\boldsymbol{v}) \xi(\boldsymbol{v}, \boldsymbol{w}_{s(\boldsymbol{v})}) d\boldsymbol{v} \tag{2}$$

- Online learning update

$$\Delta \boldsymbol{w}_{s(\boldsymbol{v})} = -\epsilon \cdot \frac{\partial \xi(\boldsymbol{v}, \boldsymbol{w}_{s(\boldsymbol{v})})}{\partial \boldsymbol{w}_{s(\boldsymbol{v})}} \tag{3}$$

Self-organizing Map (SOM)

- Introducing topological structure A in the cardinality Z
- The neighbourhood function

$$h_{\sigma}^{SOM}(\boldsymbol{r}, \boldsymbol{r}') = \exp\left(\frac{-||\boldsymbol{r} - \boldsymbol{r}'||_A}{2\sigma^2}\right) \tag{4}$$

- The prototype mapping

$$\boldsymbol{v} \mapsto s(\boldsymbol{v}) := \arg\min_{\boldsymbol{r} \in A} \sum_{\boldsymbol{r}' \in A} h_{\sigma}^{SOM}(\boldsymbol{r}, \boldsymbol{r}') \xi(\boldsymbol{v}, \boldsymbol{w}_{\boldsymbol{r}'}) \tag{5}$$

Self-organizing Map (SOM)

- The loss function

$$E_{SOM} = \frac{1}{K(\sigma)} \int P(\boldsymbol{v}) \sum_{\boldsymbol{r} \in A} \delta_{\boldsymbol{r}}^{s(\boldsymbol{v})} \sum_{\boldsymbol{r}' \in A} h_{\sigma}^{SOM}(\boldsymbol{r}, \boldsymbol{r}') \xi(\boldsymbol{v}, \boldsymbol{w}_{\boldsymbol{r}'}) d\boldsymbol{v} \tag{6}$$

- Online learning update

$$\Delta \boldsymbol{w}_{\boldsymbol{r}} = -\epsilon h_{\sigma}^{SOM}(s(\boldsymbol{v}), \boldsymbol{r}) \frac{\partial \xi(\boldsymbol{v}, \boldsymbol{w}_{\boldsymbol{r}})}{\partial \boldsymbol{w}_{\boldsymbol{r}}} \tag{7}$$

Exploration Machine (XOM)

- A 'reverse SOM' with an embedding space $\boldsymbol{w}_k \in S$:

$$E_{XOM} = \frac{1}{K(\sigma)} \int_S P_S(\boldsymbol{s}) \sum_{k=1}^{N} \delta_k^{k^*(\boldsymbol{s})} \sum_{j=1}^{N} h_\sigma^{XOM}(\boldsymbol{v}_k, \boldsymbol{v}_j) \xi_S(\boldsymbol{s}, \boldsymbol{w}_j) d\boldsymbol{s} \tag{8}$$

- $k^*(\boldsymbol{s}) := \arg\min_{k=1,\ldots,N} \sum_{j=1}^{N} h_\sigma^{XOM}(\boldsymbol{v}_k, \boldsymbol{v}_j) \xi_S(\boldsymbol{s}, \boldsymbol{w}_j)$
- $\boldsymbol{w}_i = -\epsilon h_\sigma^{XOM}(\boldsymbol{v}_i, \boldsymbol{v}_{k^*(\boldsymbol{s})}) \frac{\partial \xi_S(\boldsymbol{s}, \boldsymbol{w}_i)}{\partial \boldsymbol{w}_i}$

SOM Simulations

Experiment:

- A chain lattice with 100 units $\boldsymbol{r}$, and $\boldsymbol{w_r} \in \mathbb{R}^2$
- 10^7 data points $\boldsymbol{v} \in [0,1]^2$, $\boldsymbol{v}_1 + \boldsymbol{v}_2 = 1$
- Prior $P(\boldsymbol{v_1}) = 2\boldsymbol{v}_1$
- Decreasing learning rate & neighbourhood range: $\epsilon_{final} = 10^{-6}$, $\sigma_{final} = 1$

SOM Simulations

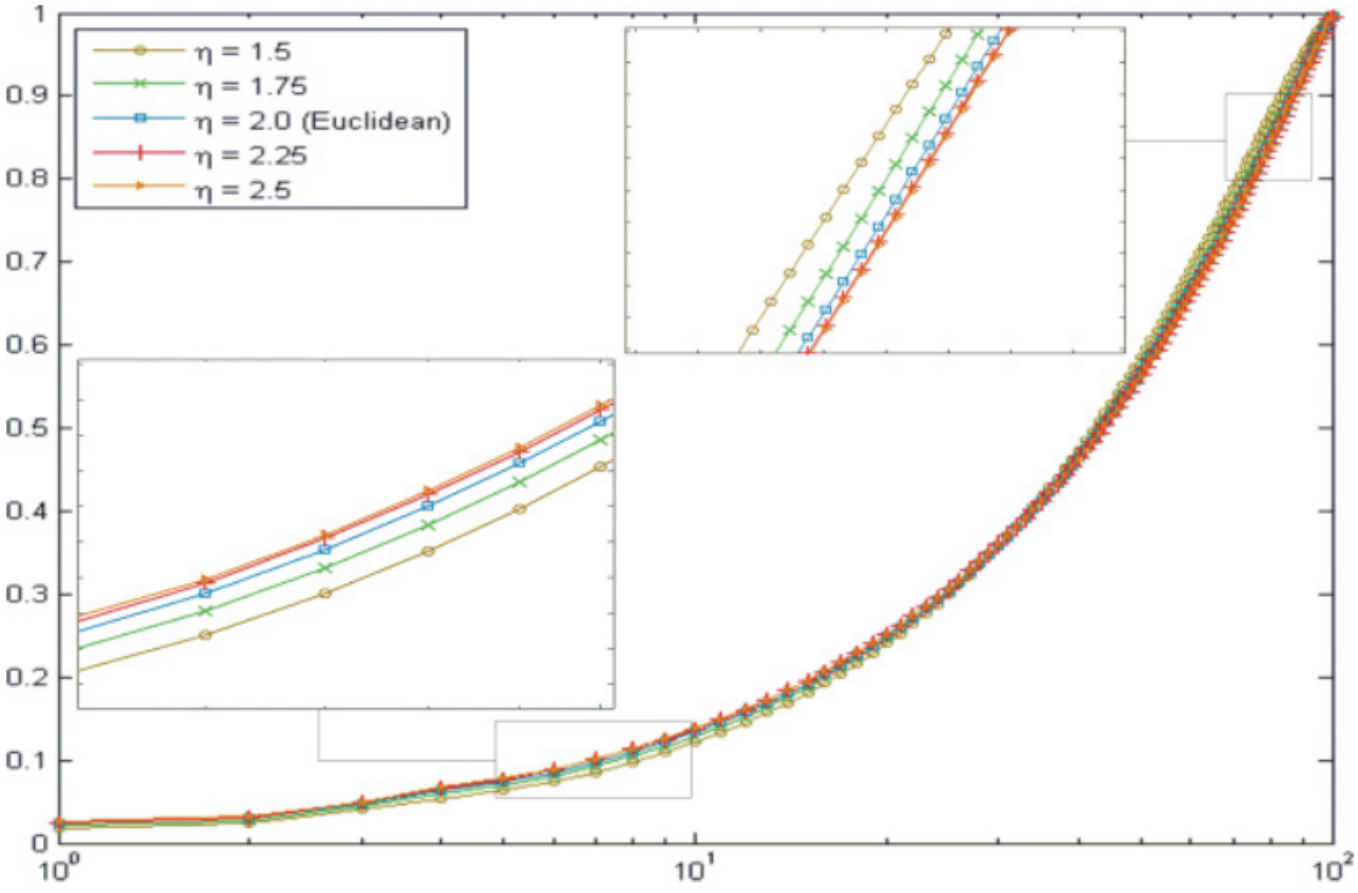


Figure: Prototype distribution for η-divergence-based SOM. (prototype index v.s. w_1 of the prototypes $w = (w_1, w_2)$)

SOM Simulations

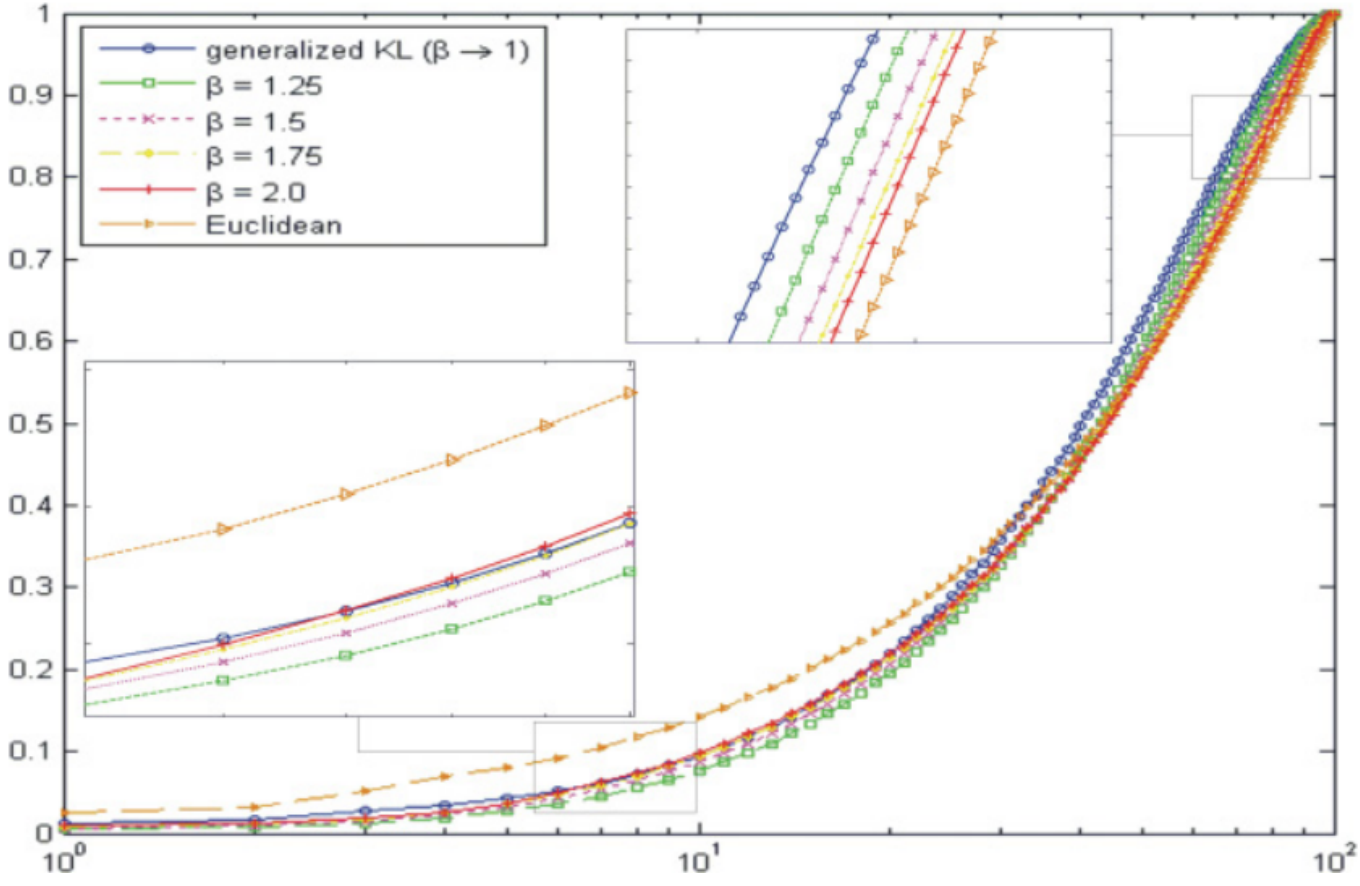


Figure: Prototype distribution for β-divergence-based SOM. (prototype index v.s. w_1 of the prototypes $w = (w_1, w_2)$)

SOM Simulations

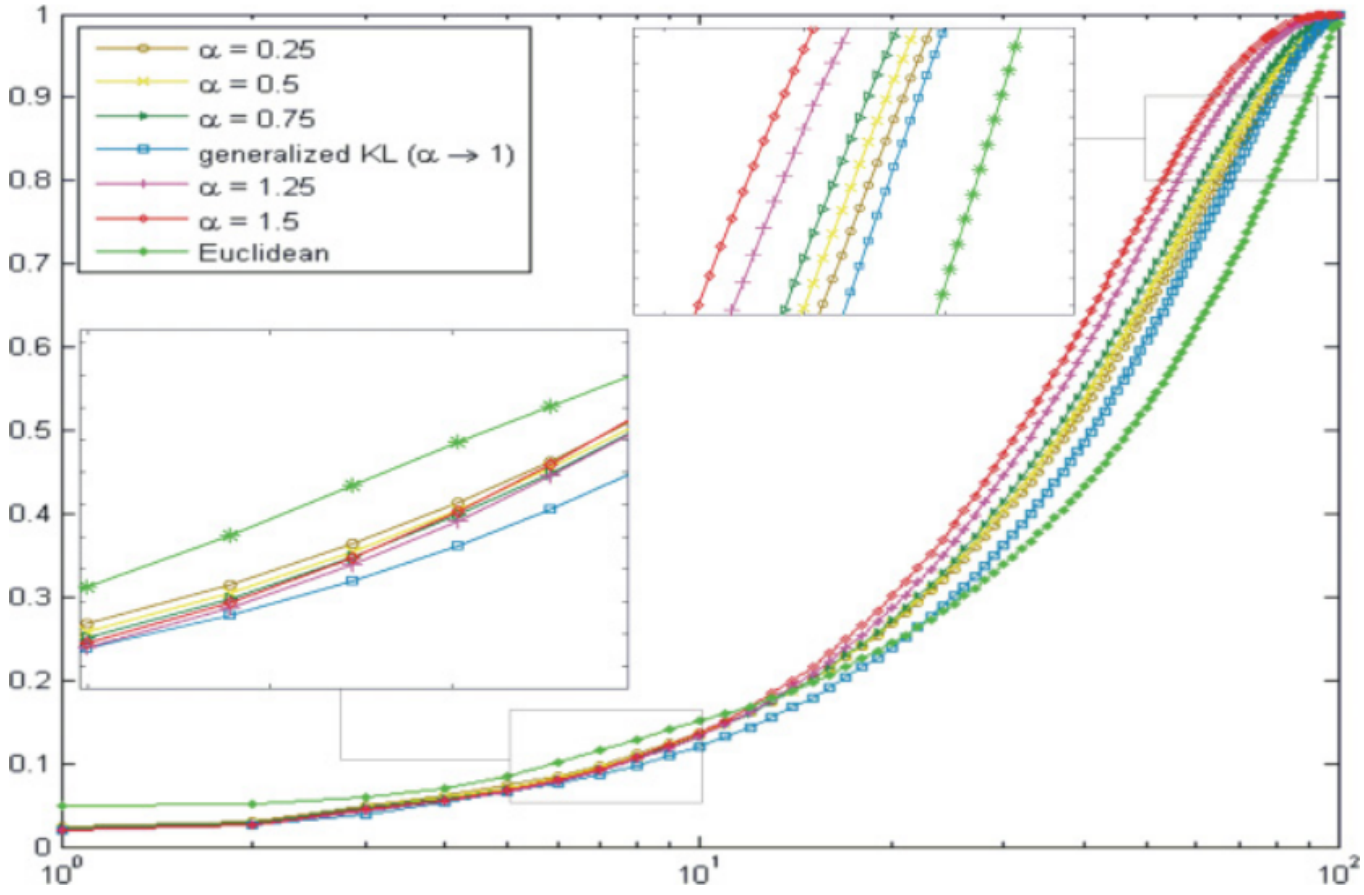


Figure: Prototype distribution for Tsallis divergence-based SOM. (prototype index v.s. w_1 of the prototypes $w = (w_1, w_2)$)

SOM Simulations

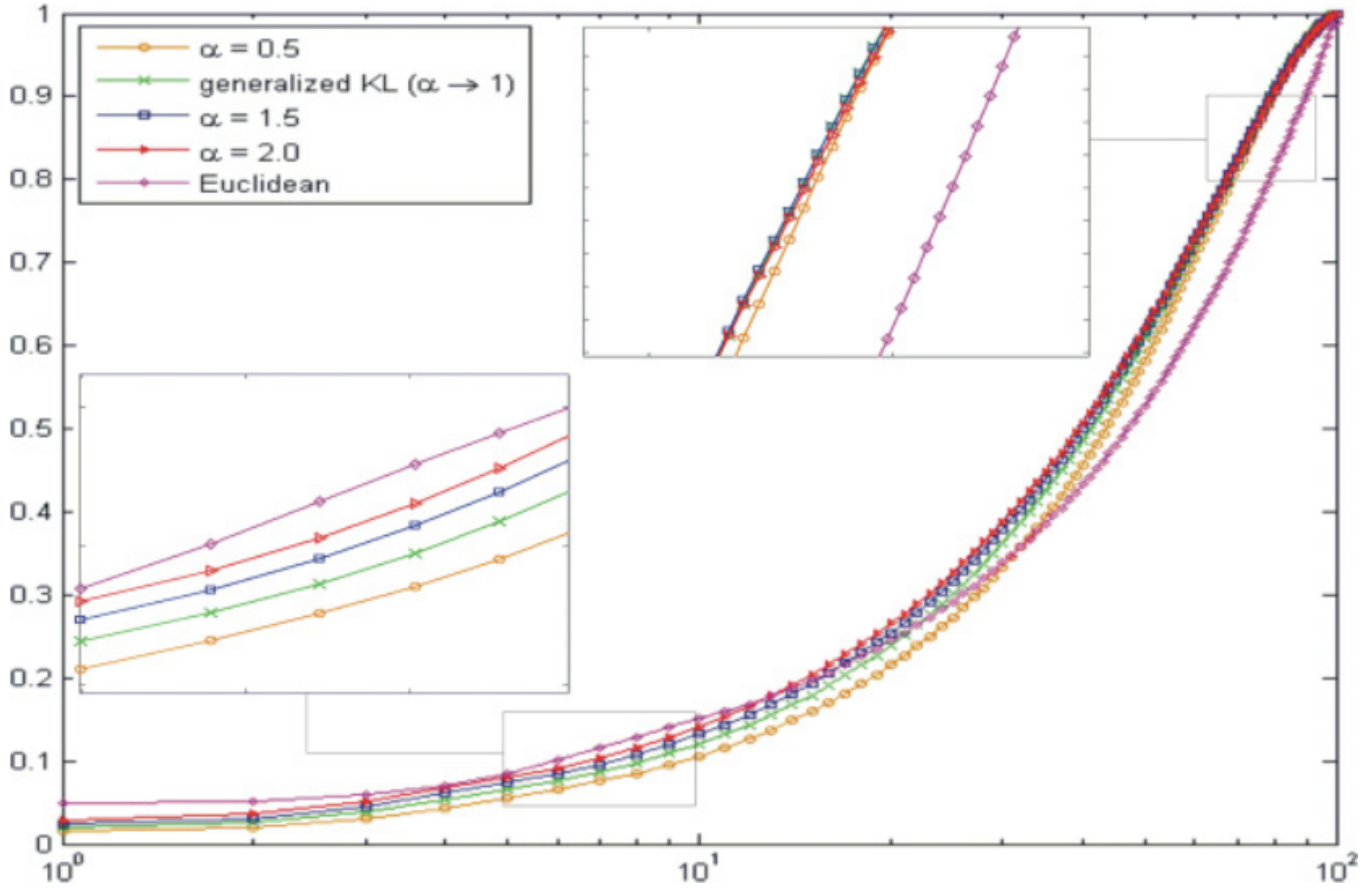


Figure: Prototype distribution for Renyi divergence-based SOM. (prototype index v.s. w_1 of the prototypes $w = (w_1, w_2)$)

SOM Simulations

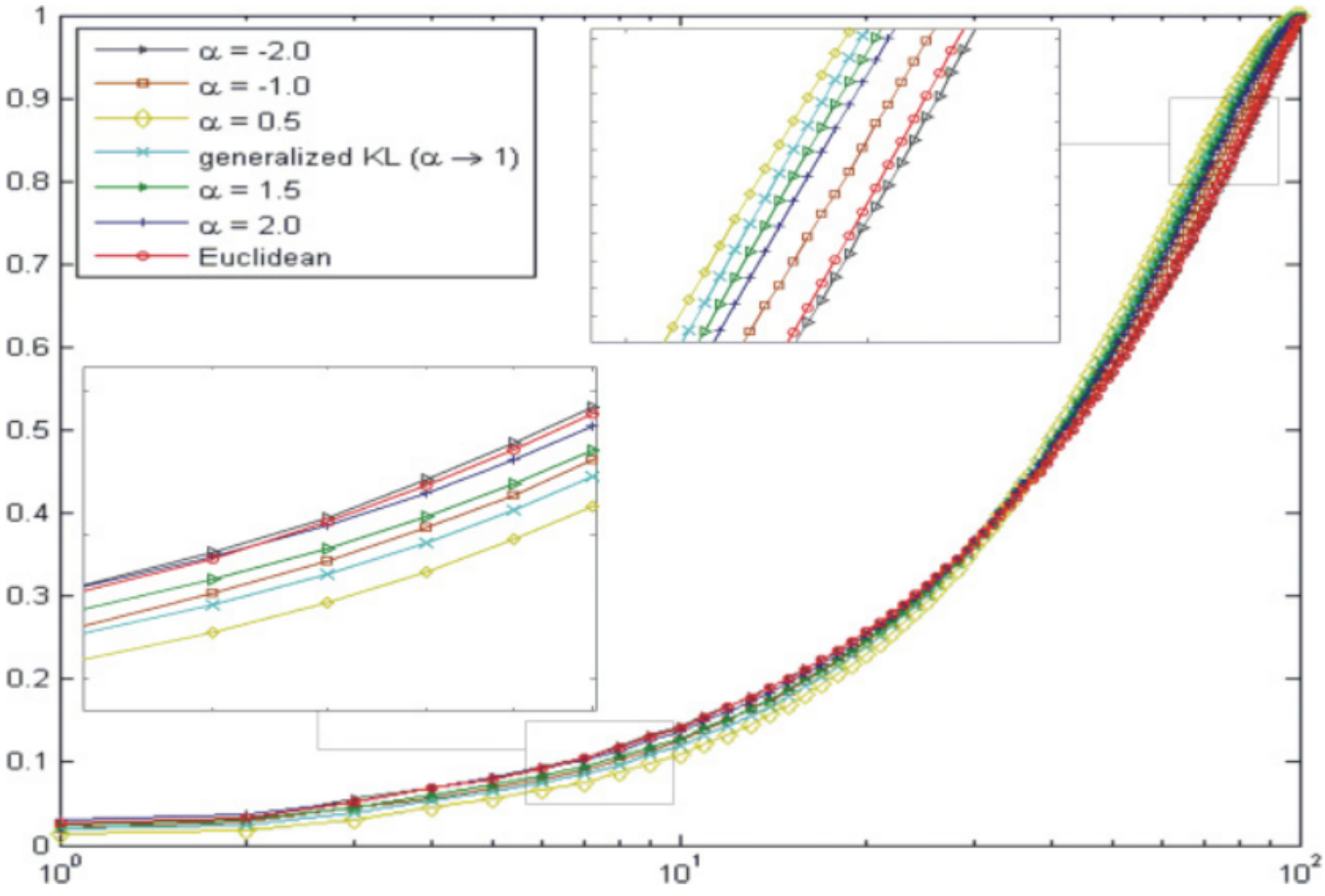


Figure: Prototype distribution for α-divergence-based SOM. (prototype index v.s. w_1 of the prototypes $w = (w_1, w_2)$)

SOM Simulations

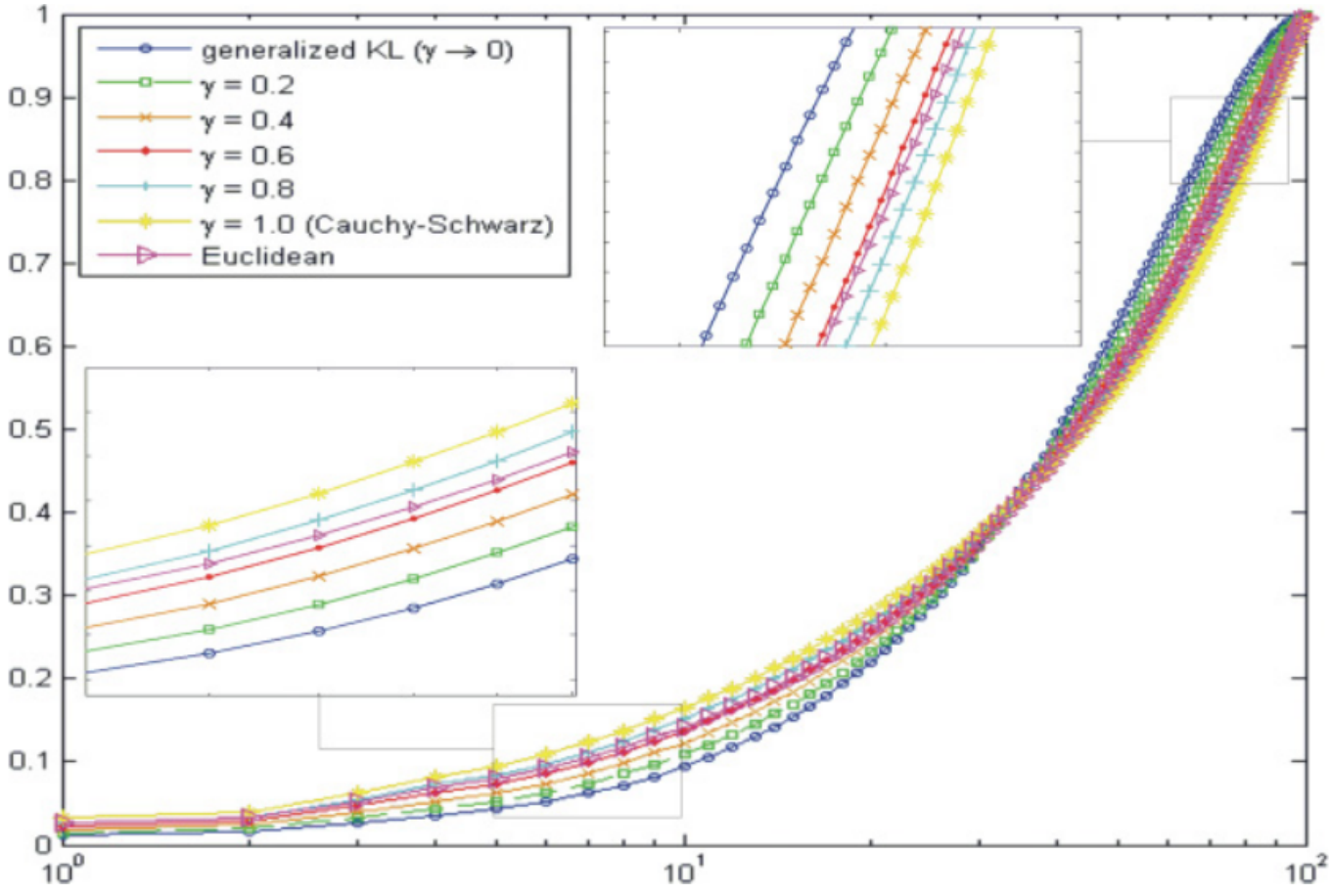


Figure: Prototype distribution for γ-divergence-based SOM. (prototype index v.s. w_1 of the prototypes $w = (w_1, w_2)$)

SOM Simulations

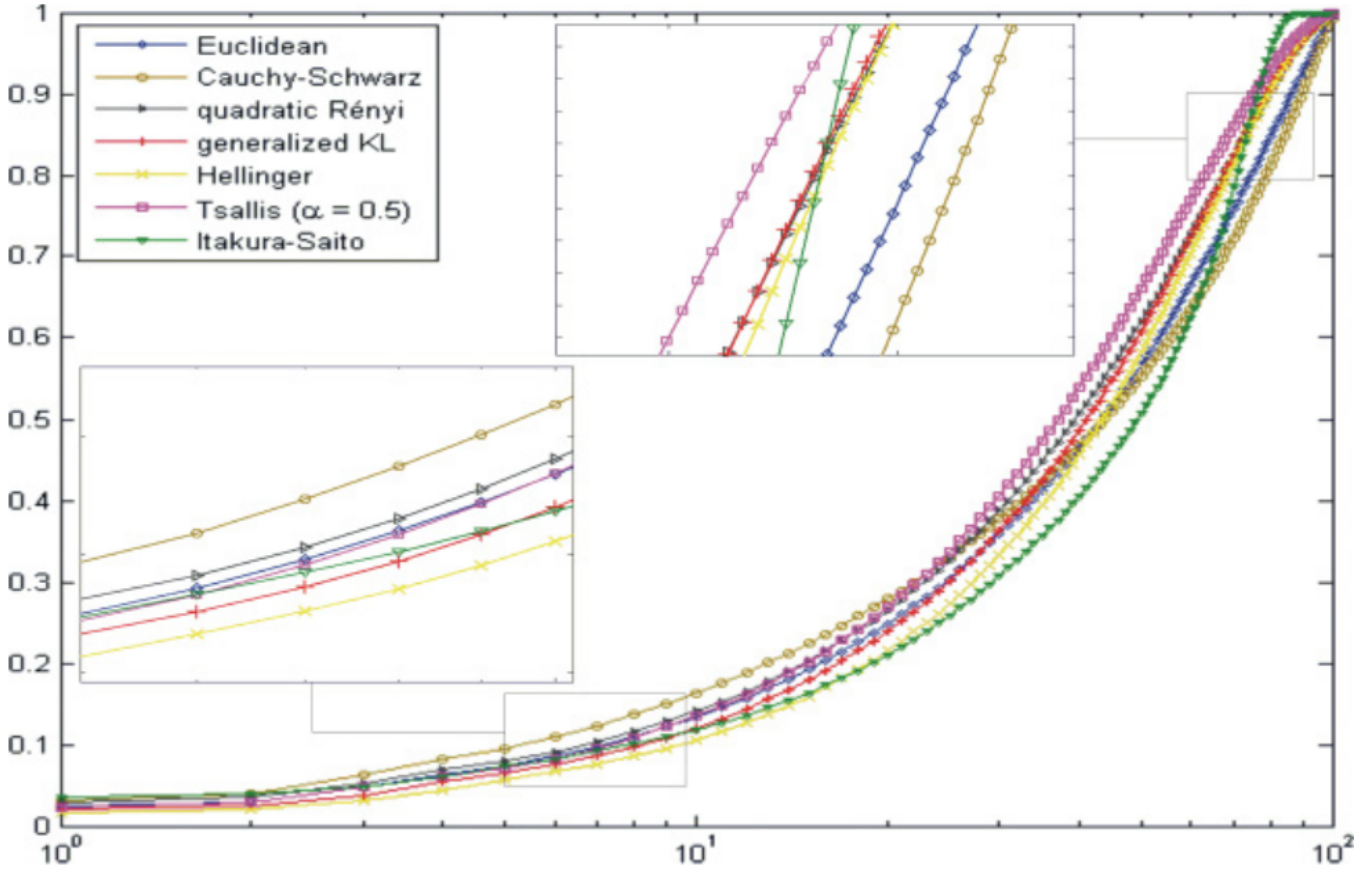


Figure: Prototype distribution for divergence-based SOM. (prototype index v.s. w_1 of the prototypes $w = (w_1, w_2)$)

Learning Vector Quantization (LVQ)

- Suppose we have K classes, define the label of $\boldsymbol{v}$

$$\boldsymbol{c_v} \in \{0,1\}^K, \sum_j \boldsymbol{c_{vj}} = 1$$

- The prototypes $\boldsymbol{w}_j$ are also labelled (represented by $\boldsymbol{y}_j$)
- Online learning update

$$\Delta \boldsymbol{w}_{s(\boldsymbol{v})} = -\alpha \cdot \epsilon \cdot \frac{\partial \xi(\boldsymbol{v}, \boldsymbol{w}_{s(\boldsymbol{v})})}{\partial \boldsymbol{w}_{s(\boldsymbol{v})}} \tag{9}$$

- $\alpha = 1$ iff. $\boldsymbol{c_v} = \boldsymbol{y}_{s(\boldsymbol{v})}$, otherwise $\alpha = -1$

Generative Learning Vector Quantization (GLVQ)

- Closest correct prototype

$$\boldsymbol{w}_{s^+(\boldsymbol{v})} = \arg\min_{k \in Z} \xi(\boldsymbol{v}, \boldsymbol{w}_k) \; s.t. \; \boldsymbol{y}_k = \boldsymbol{c}_{\boldsymbol{v}}$$

- Closest incorrect prototype

$$\boldsymbol{w}_{s^-(\boldsymbol{v})} = \arg\min_{k \in Z} \xi(\boldsymbol{v}, \boldsymbol{w}_k) \; s.t. \; \boldsymbol{y}_k \neq \boldsymbol{c}_{\boldsymbol{v}}$$

- The loss function

$$E_{GLVQ} = \sum_{\boldsymbol{v}} \frac{\xi(\boldsymbol{v}, \boldsymbol{w}_{s^+(\boldsymbol{v})}) - \xi(\boldsymbol{v}, \boldsymbol{w}_{s^-(\boldsymbol{v})})}{\xi(\boldsymbol{v}, \boldsymbol{w}_{s^+(\boldsymbol{v})}) + \xi(\boldsymbol{v}, \boldsymbol{w}_{s^-(\boldsymbol{v})})} \tag{10}$$

Generative Learning Vector Quantization (GLVQ)

- Online learning update by differentiating E_{GLVQ}

$$\Delta \boldsymbol{w}_{s^+(\boldsymbol{v})} = -\epsilon^+ \cdot \theta^+ \cdot \frac{\partial \xi(\boldsymbol{v}, \boldsymbol{w}_{s^+(\boldsymbol{v})})}{\partial \boldsymbol{w}_{s^+(\boldsymbol{v})}}$$
$$\Delta \boldsymbol{w}_{s^-(\boldsymbol{v})} = -\epsilon^- \cdot \theta^- \cdot \frac{\partial \xi(\boldsymbol{v}, \boldsymbol{w}_{s^-(\boldsymbol{v})})}{\partial \boldsymbol{w}_{s^-(\boldsymbol{v})}} \tag{11}$$

with scaling factors

$$\theta^+ = \frac{2\xi(\boldsymbol{v}, \boldsymbol{w}_{s^-(\boldsymbol{v})})}{(\xi(\boldsymbol{v}, \boldsymbol{w}_{s^+(\boldsymbol{v})}) + \xi(\boldsymbol{v}, \boldsymbol{w}_{s^-(\boldsymbol{v})})^2}$$

$$\theta^- = -\frac{2\xi(\boldsymbol{v}, \boldsymbol{w}_{s^+(\boldsymbol{v})})}{(\xi(\boldsymbol{v}, \boldsymbol{w}_{s^+(\boldsymbol{v})}) + \xi(\boldsymbol{v}, \boldsymbol{w}_{s^-(\boldsymbol{v})})^2}$$

GLVQ Simulations [Mwebaze et al., 2010]

WBC	train	test	AUC (train)	AUC (test)
Euclidean	85.00 (0.040)	84.46 (0.041)	0.924	0.918
CS	86.35 (0.003)	85.33 (0.007)	0.923	0.916
Renyi	84.44 (0.059)	84.17 (0.059)	0.916	0.910

Table: Test on the Wisconsin Breast Cancer (WBC) data set.

LC	train	test	AUC (train)	AUC (test)
Euclidean	77.99 (0.006)	75.70 (0.004)	0.809	0.787
CS	74.06 (0.005)	69.70 (0.009)	0.825	0.796

Table: Test on the lung cancer (LC) data set.

Hyperparameter Learning in GLVQ

- Denote θ as the parameter of the divergence ξ
- Updating the parameter θ

$$\begin{aligned}\Delta\theta &= -\epsilon \cdot \frac{\partial E_{GLVQ}}{\partial \xi} \cdot \frac{\partial \xi}{\partial \theta} \\ &= -\epsilon \left(\theta^{+} \cdot \frac{\partial \xi(\boldsymbol{v}, \boldsymbol{w}_{s^{+}(\boldsymbol{v})})}{\partial \theta} + \theta^{-} \cdot \frac{\partial \xi(\boldsymbol{v}, \boldsymbol{w}_{s^{-}(\boldsymbol{v})})}{\partial \theta} \right)\end{aligned} \tag{12}$$

Hyperparameter Learning in GLVQ

- Recall the γ-divergence

$$D_\gamma(p||q) = \frac{1}{\gamma+1}\log F_1 - \frac{1}{\gamma}\log F_2 \tag{13}$$

where $F_1 = (\int p^{\gamma+1}dx)^{\frac{1}{\gamma}}(\int q^{\gamma+1}dx)$, $F_2 = \int p\cdot q^\gamma dx$

- $\gamma \to 0$: Kullback-Leibler divergence
- $\gamma = 1$: Cauchy-Schwarz divergence

$$D_{CS}(p||q) = \frac{1}{2}\log\frac{(\int p^2dx)(\int q^2dx)}{(\int p\cdot qdx)^2} \tag{14}$$

Hyperparameter Learning in GLVQ

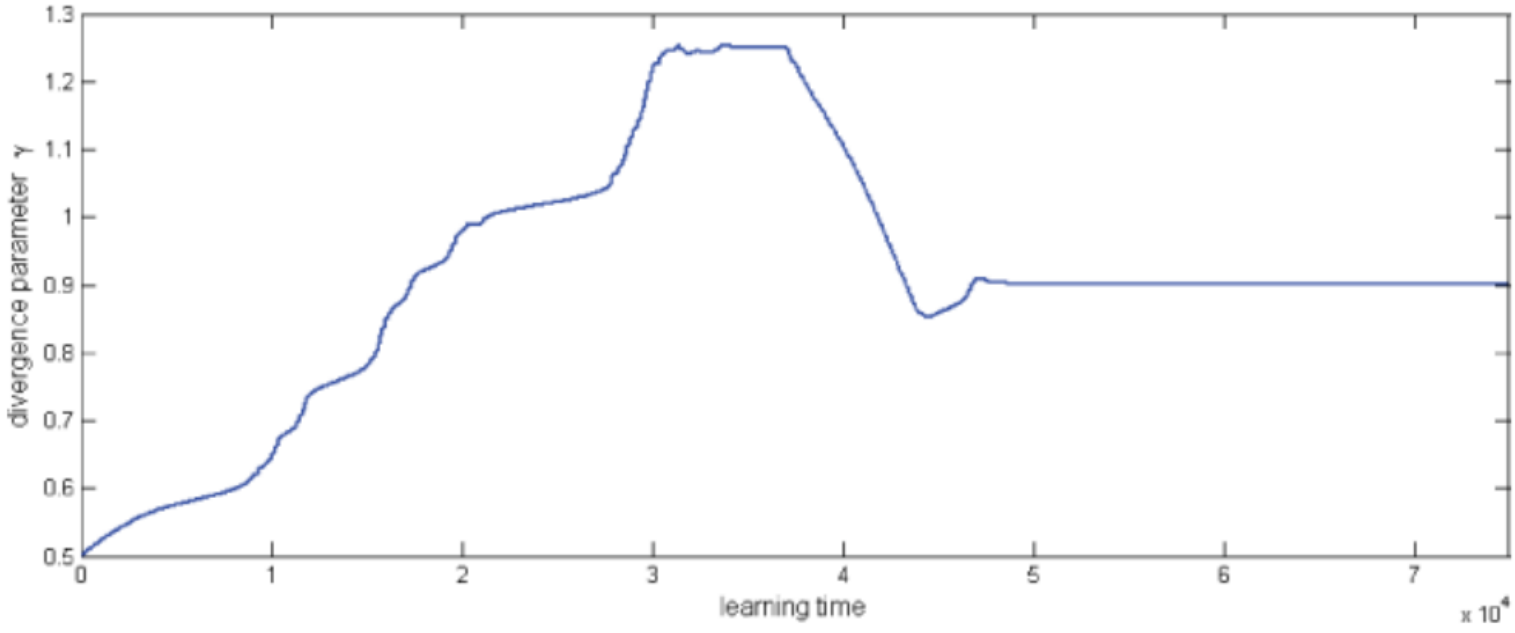


Figure: γ control on the IRIS dataset

- Average classification accuracy: 78.34% (KL) v.s. 95.16% (CS)
- Best value $\gamma_{final} = 0.9016$, yielding average accuracy 95.89%

Connecting the Quantizer and Classifier[2]

[2] See [Nguyen et al., 2009] for details.

Bayes Risk

- Given the data points $\boldsymbol{v} \in V$ with labels $y_{\boldsymbol{v}} \in \{-1, 1\}$:
  - the quantizer $\boldsymbol{w} = Q(\boldsymbol{v})$, $Q \in \mathcal{Q}$
  - the classifier $y = \gamma(\boldsymbol{w})$, $\gamma \in \Gamma$
- ϕ-risk (ϕ is a margin-based convex loss function)

$$R_\phi(\gamma, Q) = \mathbb{E}\phi(y_{\boldsymbol{v}}\gamma(\boldsymbol{w})) \tag{15}$$

- empirical ϕ-risk:

$$\hat{R}_\phi(\gamma, Q) = \frac{1}{|V|}\sum_{\boldsymbol{v}}\sum_{\boldsymbol{w}}\phi(y_{\boldsymbol{v}}\gamma(\boldsymbol{w}))Q(\boldsymbol{w}|\boldsymbol{v}) \tag{16}$$

- optimal ϕ-risk: $R_\phi(Q) = \inf_{\gamma\in\Gamma} R_\phi(\gamma, Q)$
- Bayes Risk (0-1 loss): $R_{Bayes}(\gamma, Q) = \mathbb{E}\left[\mathbb{I}(y_{\boldsymbol{v}}\gamma(\boldsymbol{w}) < 0)\right]$

Connecting the ϕ-risk and f-divergence

- Define measures with prior $p = P(y = 1)$ and $q = P(y = -1)$

$$\mu(\boldsymbol{w}) := P(y = 1, \boldsymbol{w}) = p \int_{\boldsymbol{v}} Q(\boldsymbol{w}|\boldsymbol{v}) dP(\boldsymbol{v}|y = 1)$$

$$\pi(\boldsymbol{w}) := P(y = -1, \boldsymbol{w}) = q \int_{\boldsymbol{v}} Q(\boldsymbol{w}|\boldsymbol{v}) dP(\boldsymbol{v}|y = -1)$$

- Rewrite the f-divergence as

$$I_f(\mu, \pi) := \sum_{\boldsymbol{w}} \pi(\boldsymbol{w}) f\left(\frac{\mu(\boldsymbol{w})}{\pi(\boldsymbol{w})}\right) \tag{17}$$

- $R_\phi(Q) = -I_f(\mu, \pi)$ with $f(u) := -\inf_\alpha(\phi(-\alpha) + \phi(\alpha)u)$

Approximation & Estimation Error

- Restricting the searching space

$$\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq ... \subseteq \mathcal{C}_n \subseteq \Gamma, \mathcal{D}_1 \subseteq \mathcal{D}_2 \subseteq ... \subseteq \mathcal{D}_n \subseteq \mathcal{Q}$$

- empirical solution

$$(\gamma_n^*, Q_n^*) := \underset{(\gamma, Q) \in (\mathcal{C}_n, \mathcal{D}_n)}{\arg\min} \hat{R}_\phi(\gamma, Q) \tag{18}$$

- Minimum Bayes risk:

$$R_{Bayes}^* := \inf_{(\gamma, Q) \in (\Gamma, \mathcal{Q})} R_{Bayes}(\gamma, Q) \tag{19}$$

- Excess Bayes risk:

$$R_{Bayes}(\gamma_n^*, Q_n^*) - R_{Bayes}^* \tag{20}$$

Approximation & Estimation Error

- Approximation error

$$\mathcal{E}_0(\mathcal{C}_n, \mathcal{D}_n) := \inf_{(\gamma, Q) \in (\mathcal{C}_n, \mathcal{D}_n)} \{R_\phi(\gamma, Q)\} - R_\phi^* \tag{21}$$

with $R_\phi^* := \inf_{(\gamma, Q) \in (\Gamma, \mathcal{Q})} R_\phi(\gamma, Q)$

- Estimation error

$$\mathcal{E}_1(\mathcal{C}_n, \mathcal{D}_n) := \mathbb{E} \sup_{(\gamma, Q) \in (\mathcal{C}_n, \mathcal{D}_n)} |R_\phi(\gamma, Q) - \hat{R}_\phi(\gamma, Q)| \tag{22}$$

approximation condition : $\lim_{n \to \infty} \mathcal{E}_0(\mathcal{C}_n, \mathcal{D}_n) = 0$

estimation condition : $\lim_{n \to \infty} \mathcal{E}_1(\mathcal{C}_n, \mathcal{D}_n) = 0$ *in probability*

Bayes Consistency

- B1: ϕ is continuous, convex, and classification-calibrated
- B2: $M_n := \max_{y\in\{-1,1\}} \sup_{(\gamma,Q)\in(\mathcal{C}_n,\mathcal{D}_n)} \sup_{\boldsymbol{w}\in W} |\phi(y\gamma(\boldsymbol{w}))| < \infty$ for every n

Theorem (Bayes Consistency)

Let ϕ a loss function satisfying B1 and B2 and inducing the f-divergence of the form $f(u) = -c\min(u, 1) + au + b$ with some $c > 0$ and $a, b \in \mathbb{R}$ s.t. $(a - b)(p - q) \geq 0$. If $\{\mathcal{C}_n\}$ and $\{\mathcal{D}_n\}$ satisfy the approximation & estimation conditions, then the empirical estimation procedure is universally consistent:

$$\lim_{n\to\infty} R_{Bayes}(\gamma_n^*, Q_n^*) = R_{Bayes}^* \text{ in probability} \tag{23}$$

Bayes Consistency (Proof)

Lemma

Let ϕ satisfies B1 and B2 and induces the f-divergence of the form $f(u) = -c\min(u, 1) + au + b$ for some $c > 0$ and $a, b \in \mathbb{R}$ s.t. $(a-b)(p-q) \geq 0$. Then for any $(\gamma, Q) \in (\Gamma, \mathcal{Q})$

$$\frac{c}{2}[R_{Bayes}(\gamma, Q) - R^*_{Bayes}] \leq R_\phi(\gamma, Q) - R^*_\phi \tag{24}$$

- From B2 (we sample n data points to estimate the empirical risk), $\sup_{(\gamma,Q)\in(\mathcal{C}_n,\mathcal{D}_n)} |\hat{R}_\phi(\gamma, Q) - R_\phi(\gamma, Q)|$ varies by at most $2M_n/n$ if we change one training example $(\boldsymbol{v}_i, \boldsymbol{y}_v)$ to $(\boldsymbol{v}_i^{'}, \boldsymbol{y}_v^{'})$

Bayes Consistency (Proof)

- Using McDiamid's Inequality (with probability at least $1-\delta$):

$$\left| \sup_{(\gamma,Q)\in(\mathcal{C}_n,\mathcal{D}_n)} |\hat{R}_\phi(\gamma,Q) - R_\phi(\gamma,Q)| - \mathcal{E}_1(\mathcal{C}_n,\mathcal{D}_n) \right| \leq M_n \sqrt{\frac{2\log(1/\delta}{n}} \tag{25}$$

- Suppose $(\gamma_n^\dagger, Q_n^\dagger) := \arg\inf_{(\gamma,Q)\in(\mathcal{C}_n,\mathcal{D}_n)} R_\phi(\gamma,Q)$, define the error

$$err(\gamma,Q) := |R_\phi(\gamma,Q) - \hat{R}_\phi(\gamma,Q)|$$

Bayes Consistency (Proof)

- Apply the lemma:

$$
\begin{aligned}
&\frac{c}{2}[R_{Bayes}(\gamma_n^*, Q_n^*) - R_{Bayes}^*] \leq R_\phi(\gamma, Q) - R_\phi^* \\
\leq &err(\gamma_n^*, Q_n^*) + err(\gamma_n^\dagger, Q_n^\dagger) + |R_\phi(\gamma_n^\dagger, Q_n^\dagger) - R_\phi^*| \\
&+ \hat{R}_\phi(\gamma_n^*, Q_n^*) - \hat{R}_\phi(\gamma_n^\dagger, Q_n^\dagger) \\
\leq &2\mathcal{E}_1(\mathcal{C}_n, \mathcal{D}_n) + 2M_n\sqrt{\frac{2\log(2/\delta)}{n}} + \mathcal{E}_0(\mathcal{C}_n, \mathcal{D}_n)
\end{aligned}
\tag{26}
$$

- The theorem is proved by $n \to \infty$.

Conclusion

- Some IT stuff is still valid in ML
- Quantizer also influences performance of learning tasks
-


Donoho, D. L. (2006).
Compressed sensing.
IEEE Transactions on Information Theory, 52(4):1289–1306.


Gallager, R. G. (1965).
A simple derivation of the coding theorem and some applications.
Institute of Electrical and Electronics Engineers. Transactions on Information Theory, IT-11(1):3–18.


Kelly, J. L. (1956).
A new interpretation of information rate.
Information Theory, IRE Transactions on, 2(3):185–189.


Mwebaze, E., Schneider, P., Schleif, F.-M., Haase, S., Villmann, T., and Biehl, M. (2010).
Divergence based learning vector quantization.
In *ESANN*.


Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2009).
On surrogate loss functions and f-divergences.
The Annuals of Statistics, 37(2):876 – 904.


Polyanskiy, Y. (2010).
Channel coding: Non-asymptotic fundamental limits.
PhD thesis, Princeton University.


Shannon, C. E. (1948).
A mathematical theory of communication.
The Bell System Technical Journal, 27:379–423, 623–656.


Villmann, T. and Haase, S. (2011).
Divergence-based vector quantization.
Neural Computation, 23(5):1343–1392.