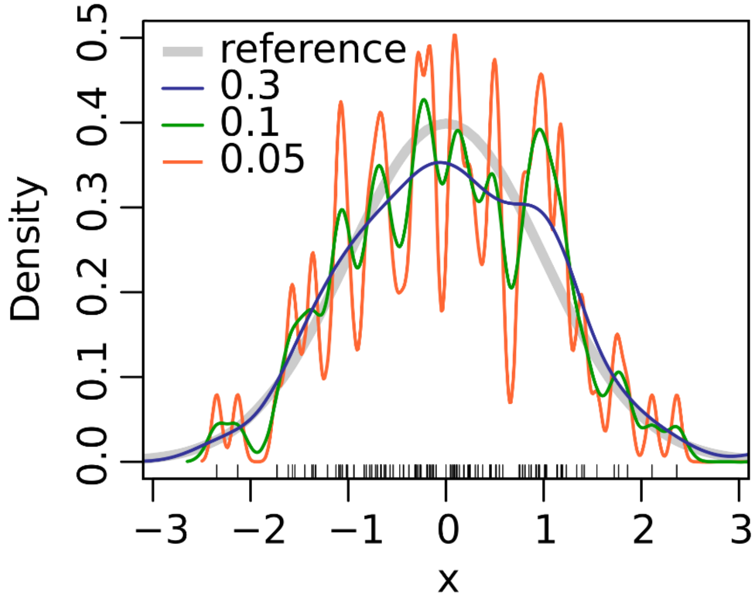
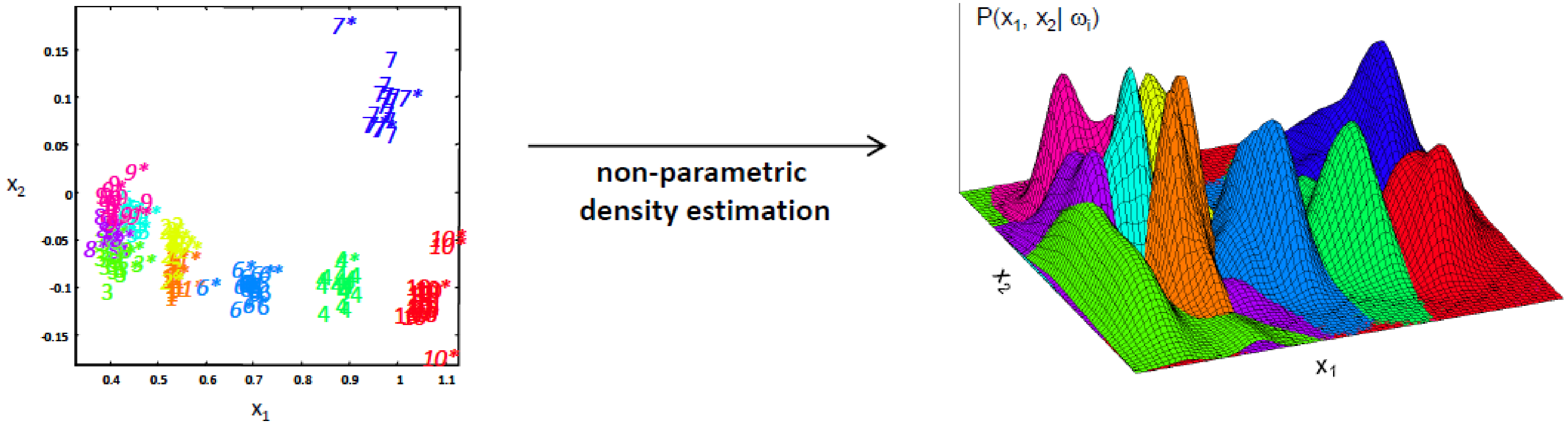


Anomaly Detection:
Parzen Window Density Estimation

Pilsung Kang
School of Industrial Management Engineering
Korea University

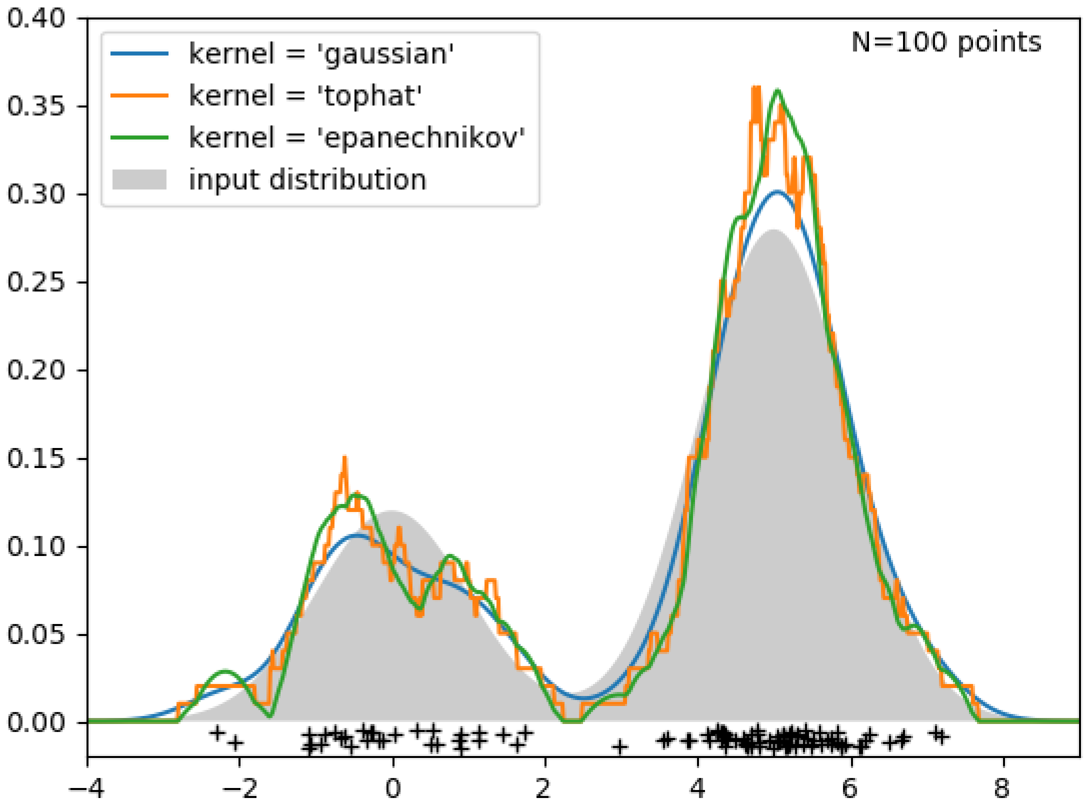
Kernel-density Estimation

- Kernel-density Estimation
  - ✓ Attempts to estimate the density directly from the data without assuming a particular form for the underlying distribution



Kernel-density Estimation

- Kernel-density Estimation: 1-D example



Kernel-density Estimation

- Kernel-density Estimation
  - ✓ The probability that a vector x, drawn from a distribution p(x), will fall in a given region R of the sample space

$$P = \int_R p(x')dx'$$

  - ✓ Suppose that N vectors {$\mathbf{x}^1$, $\mathbf{x}^2$, …, $\mathbf{x}^n$} are drawn from the distribution; the probability that k of these N vectors fall in R is given by

$$P(k) = \binom{N}{k} P^k (1-P)^{N-k}$$

  - ✓ It can be shown that (from the binomial distribution) the mean and variance of the ratio k/N are

$$E\left[\frac{k}{N}\right] = P, \qquad Var\left[\frac{k}{N}\right] = \frac{P(1-P)}{N}$$

Kernel-density Estimation

- Kernel-density Estimation
  - ✓ As N → ∞, the distribution becomes sharper (the variance gets smaller), so we can expect that a good estimate of the probability P can be obtained from the mean fraction of the points that fall within R

$$P \cong \frac{k}{N}$$

  - ✓ If we assume that R is so small that p(x) does not vary appreciably within it, then

$$P = \int_R p(x')dx' \cong p(x)V$$

  - ✓ where V is the volume enclosed by region R
  - ✓ Merging the two previous results

$$P = \int_R p(x')dx' \cong p(x)V = \frac{k}{N}, \qquad p(x) = \frac{k}{NV}$$

Kernel-density Estimation

- Kernel-density Estimation

$$p(x) = \frac{k}{NV}, \quad where \begin{Bmatrix} V: \text{volume surrounding } x \\ N: \text{the total number of examples} \\ k: \text{the number of examples inside } V \end{Bmatrix}$$

✓ Estimation becomes more accurate as we increase the number of sample points N and shrink the volume V

✓ In practice, the total number of examples is fixed so that we have to find a compromise for V

- Large enough to include enough examples within R
- Small enough to support the assumption that p(x) is constant within R

✓ Fix V and determine k from the data: Kernel-density estimation

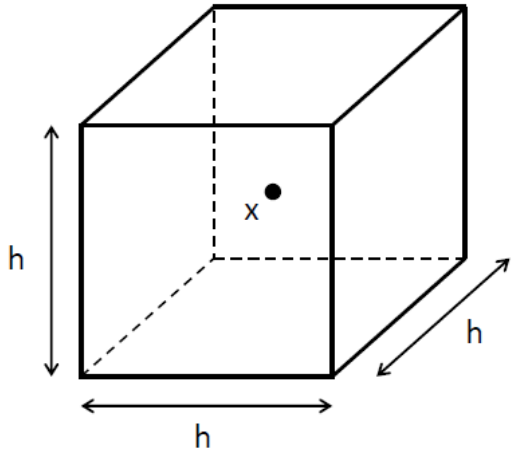
✓ Fix k and determine V from the data: k-nearest neighbor density estimation

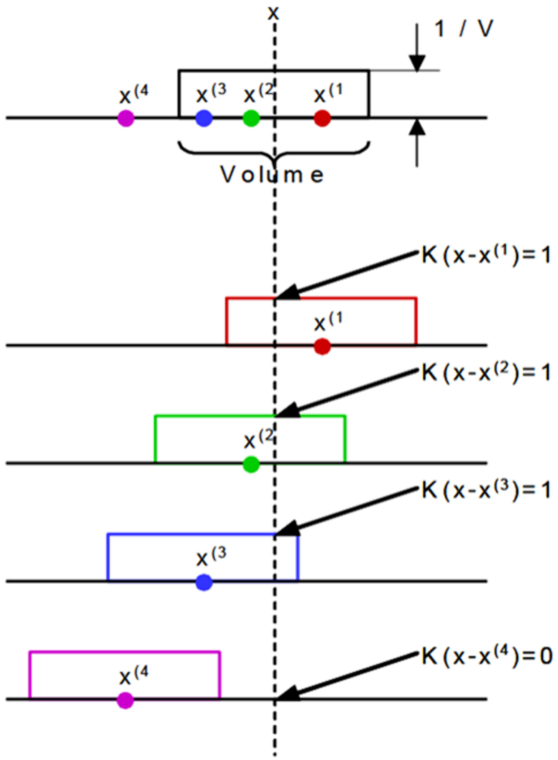
Parzen Window Density Estimation

- Parzen Window Density Estimation
  - ✓ Assume that the region R that encloses the k examples is a hypercube with sides of length h centered at x
    - Its volume is given by $V = h^d$, d: N. dimensions
  - ✓ Define a kernel function K(u)

$$K(u) = \begin{cases} 1 & |u_j| < \dfrac{1}{2} \ \forall j = 1, \dots d \\ 0 & otherwise \end{cases}$$

$$k = \sum_{i=1}^{N} K\left(\frac{\mathbf{x}^i - \mathbf{x}}{h}\right) \qquad p(x) = \frac{1}{Nh^d} \sum_{i=1}^{N} K\left(\frac{\mathbf{x}^i - \mathbf{x}}{h}\right)$$



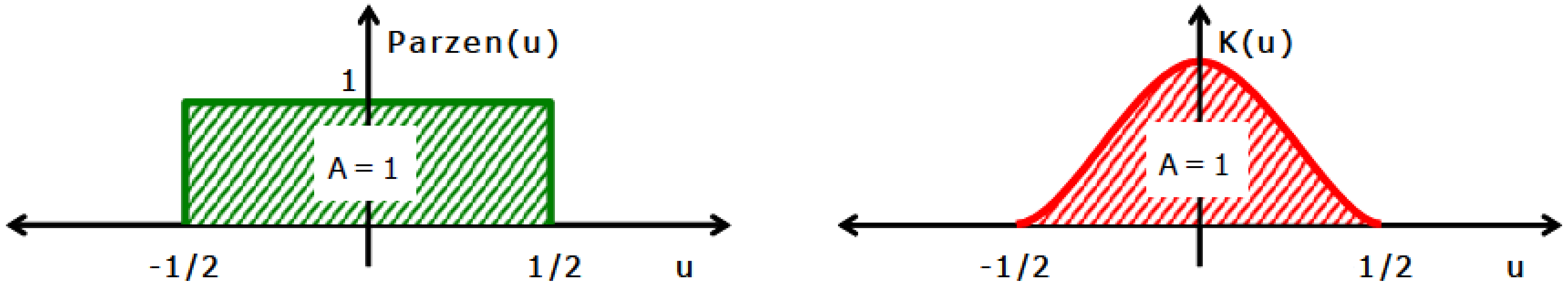


Parzen Window Density Estimation

- Drawbacks of K(u)
  - ✓ Yields density estimate that have discontinuities
  - ✓ Weights equally all points $\mathbf{x}^i$, regardless of their distance to the estimation point $\mathbf{x}$
- Smooth kernel function

$$P = \int_R K(x)\, dx = 1$$

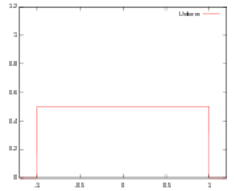
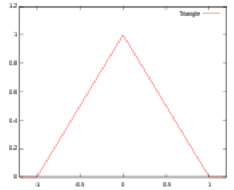
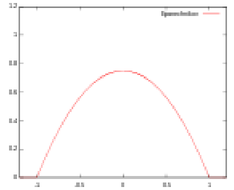
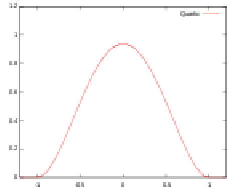
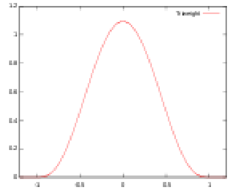
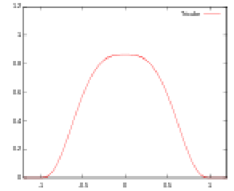
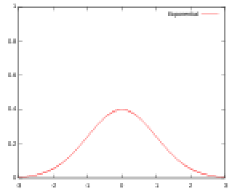
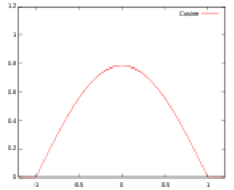
  - ✓ Commonly use a radially symmetric and unimodal pdf, such as Gaussian



$$p(x) = \frac{1}{N} \sum_{i=1}^{N} K\left(\frac{\mathbf{x}^i - \mathbf{x}}{h}\right)$$

Parzen Window Density Estimation

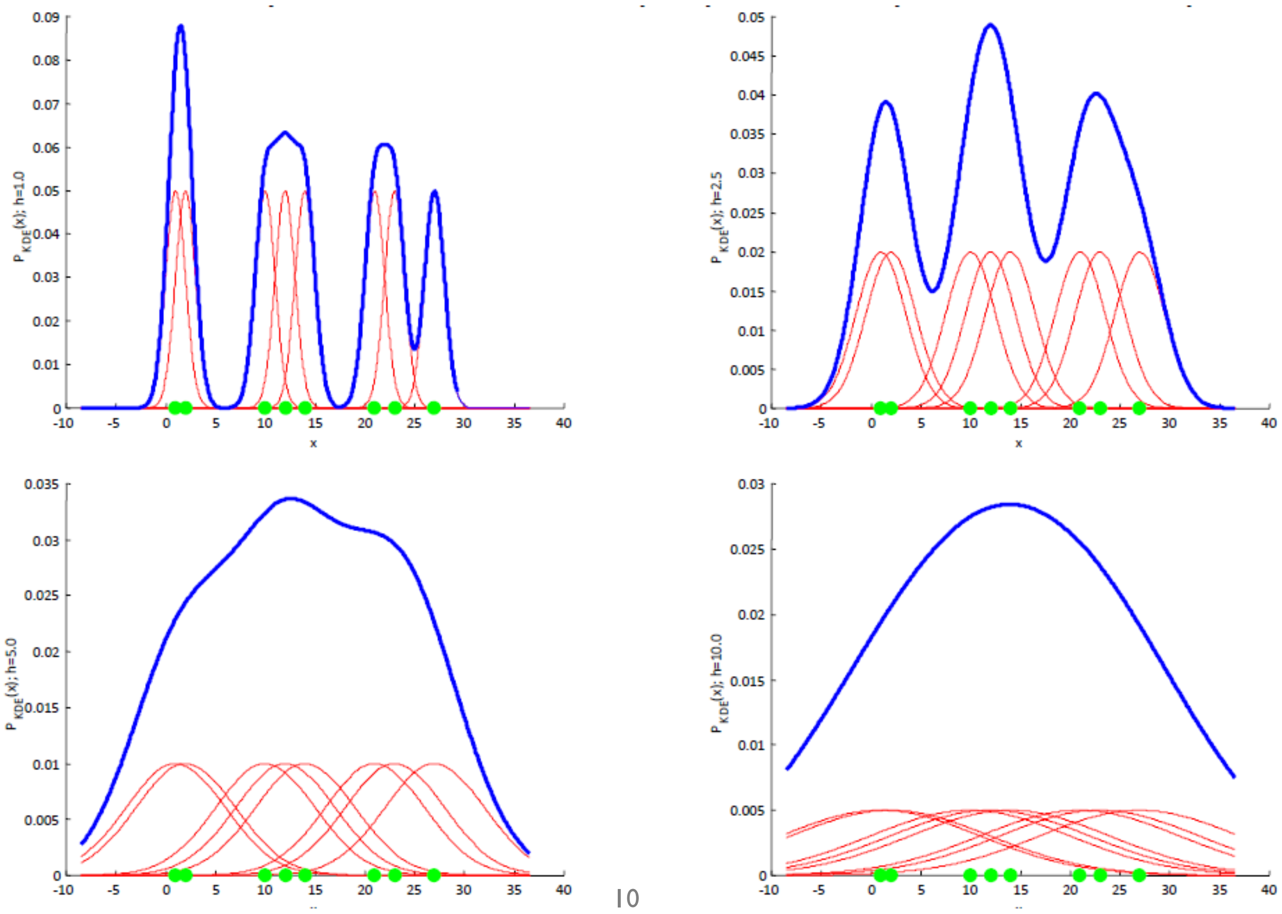
- Example of smooth kernels

Uniform	$K(u) = \frac{1}{2} \mathbf{1}_{\{\lvert u\rvert \le 1\}}$	
Triangular	$K(u) = (1 - \lvert u\rvert) \mathbf{1}_{\{\lvert u\rvert \le 1\}}$	
Epanechnikov	$K(u) = \frac{3}{4}(1 - u^2) \mathbf{1}_{\{\lvert u\rvert \le 1\}}$	
Quartic (biweight)	$K(u) = \frac{15}{16}(1 - u^2)^2 \mathbf{1}_{\{\lvert u\rvert \le 1\}}$	
Triweight	$K(u) = \frac{35}{32}(1 - u^2)^3 \mathbf{1}_{\{\lvert u\rvert \le 1\}}$	
Tricube	$K(u) = \frac{70}{81}(1 - \lvert u\rvert^3)^3 \mathbf{1}_{\{\lvert u\rvert \le 1\}}$	
Gaussian	$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$	
Cosine	$K(u) = \frac{\pi}{4} \cos\left(\frac{\pi}{2}u\right) \mathbf{1}_{\{\lvert u\rvert \le 1\}}$	
Logistic	$K(u) = \frac{1}{e^u + 2 + e^{-u}}$	
Silverman kernel[4]	$K(u) = \frac{1}{2} e^{-\frac{\lvert u\rvert}{\sqrt{2}}} \cdot \sin\left(\frac{\lvert u\rvert}{\sqrt{2}} + \frac{\pi}{4}\right)$	

https://en.wikipedia.org/wiki/Kernel_(statistics)

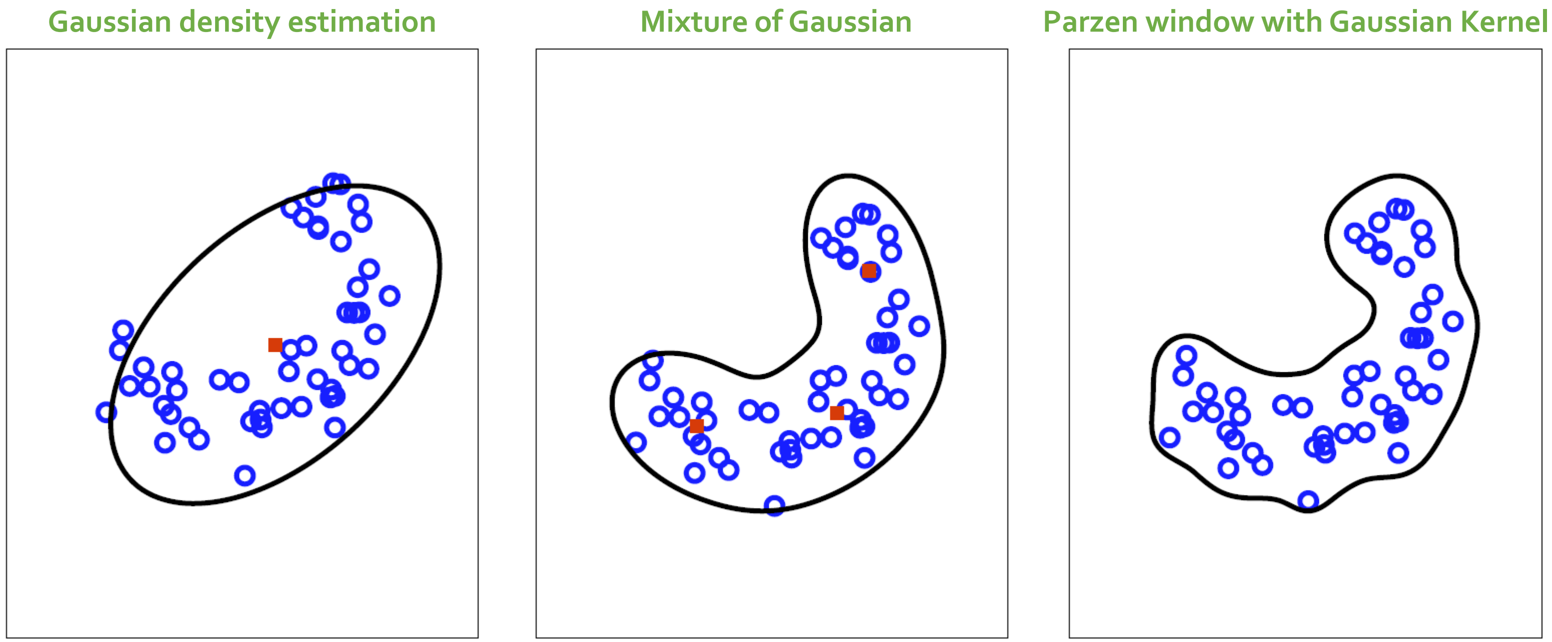
Parzen Window Density Estimation

- Smoothing parameter (bandwidth) h
  - ✓ A large h will over-smooth the density distribution
  - ✓ A small h will result in a spiky density distribution



Parzen Window Density Estimation

- Kernel Density Estimation
  - ✓ The smoothing parameter h can be optimized through EM algorithm



고려대학교 KOREA UNIVERSITY

DSBA Data Science & Business Analytics


Q & A

References

Research Papers

- Breunig, M.M., Kriegel, H.-P., Ng, R.T., and Sander, J. (2000). LOF: Identifying density-based local outliers. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data: 93-104.
- Chalapathy, R., Menon, A. K., & Chawla, S. (2018). Anomaly detection using one-class neural networks. arXiv preprint arXiv:1802.06360.
- Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys 41(3): 15.
- Harmeling, S. Dornhege, G., Tax, D. Meinecke, F., and Muller, K-R. (2006). From outliers to prototype: Ordering data. Neurocomputing 69(13-15): 1608-1618.
- Hariri, S., Kind, M. C., & Brunner, R. J. (2018). Extended Isolation Forest. arXiv preprint arXiv:1811.02141.
- Kang, P. and Cho, S. (2009). A hybrid novelty score and its use in keystroke dynamics-based user authentication. Pattern Recognition 42(11): 3115-3127.
- Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on (pp. 413-422). IEEE.
- Liu, F. T., Ting, K. M., & Zhou, Z. H. (2012). Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD), 6(1), 3.
- Oza, P., & Patel, V. M. (2018). One-class convolutional neural network. IEEE Signal Processing Letters, 26(2), 277-281.
- Perera, P., Nallapati, R., & Xiang, B. (2019). Ocgan: One-class novelty detection using gans with constrained latent representations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2898-2906).

References

Research Papers

- Pimentel, M.A.F., Clifton, D.A., Clifton, L., and Tarassenko, L. (2014). A review of novelty detection, Signal Processing 99: 215-249.
- Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., & Langs, G. (2017, June). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In International Conference on Information Processing in Medical Imaging (pp. 146-157). Springer, Cham.
- Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., and Williamson, R. C.(2001). Estimating the support of a high-dimensional distribution. Neural computation 13(7): 1443-1471.
- Tax, D.M. (2001). One-class classification, Ph.D. Thesis, Delft University of Technology, Netherlands.
- Tax, D.M. and Duin, R.P. (2004). Support vector data description. Machine learning 54(1): 45-66.

Other materials

- Pages 28-33 & 36: http://research.cs.tamu.edu/prism/lectures/pr/pr_l7.pdf
- Figures in Auto-encoder section: https://dl.dropboxusercontent.com/u/19557502/6_01_definition.pdf
- Gramfort, A. (2016). Anomaly/Novelty detection with scikit-learn: https://www.slideshare.net/agramfort/anomalynovelty-detection-with-scikitlearn