NDD October 7th, 2021

1. Data experiment / OpenML
2. CNN – KDN → GNN?
3. New other data
4. High dimensional problem → sparse parity
5. Robust to contaminating distributions.
6. Run the benchmarks.

* The accuracy is better than the chance accuracy.

* FTE → no theoretically guarentees.
  can't say for sure since study's are empirical.
  Theory!!
  adversarial task! → hard to measure / task similarity

– recruiting gaussian distributions relevant to task.
– pruning the polytope
– KDF
– CNN??

– double counting → increase sample size.

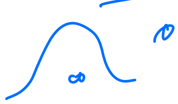
KDG / Jayanta Oct 11th 2021

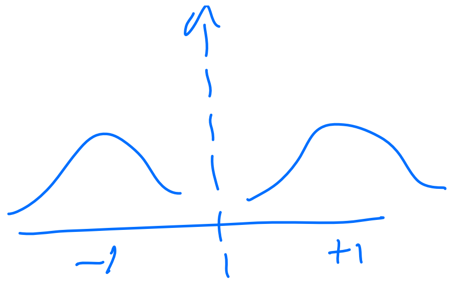
possible → possibilities.

2^5 possible polytopes. → 5

* penultimate layer →
* followed same path → same partition.
* CIFAR-10 – simple data
* covariance matrix ~ dimensionality ne
  → high dimensions
* KDN → feature selection.
* Deep Boltzmann Machines.
* High Dimensionality → KDN
* MNIST →
  CIFAR-10 →
    - test the KD-CNN on a lower dim dataset
    - penultimate layer — 1
    - all the fully-connected layers — 2
    - Deep Boltzmann machines
    - Dealing with high-dimensionality
    - Histogram of polytopes

Weighted Maximum Likelihood

 → weighting using the # paths.



weighting!

Dim-reduction!

sim_kdn

sim_kdf

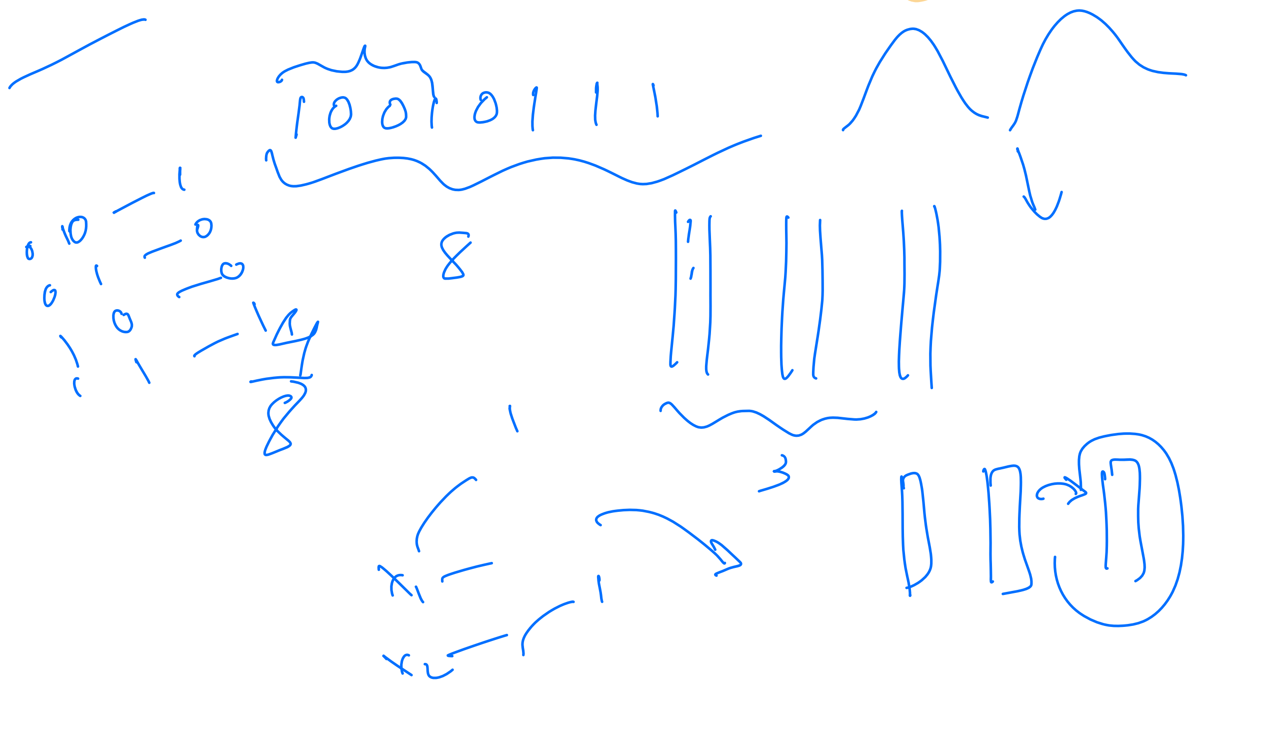
high-dim.

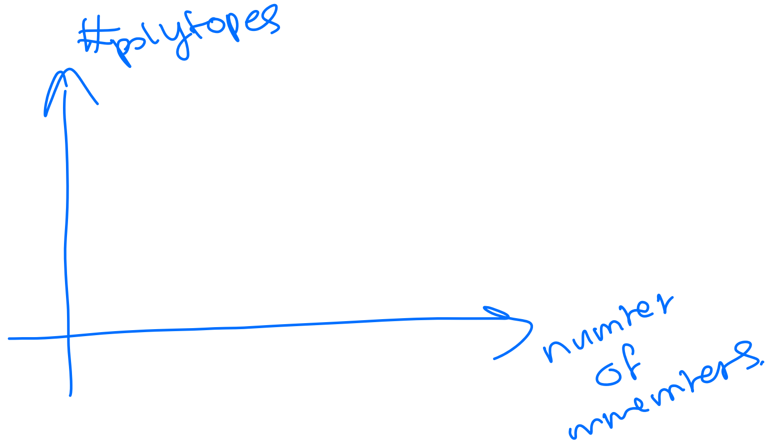
1 0 0 1 0 = 2

0 1 0 0 1 = 2

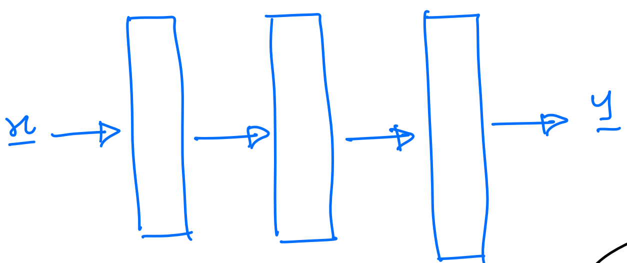
all the layers

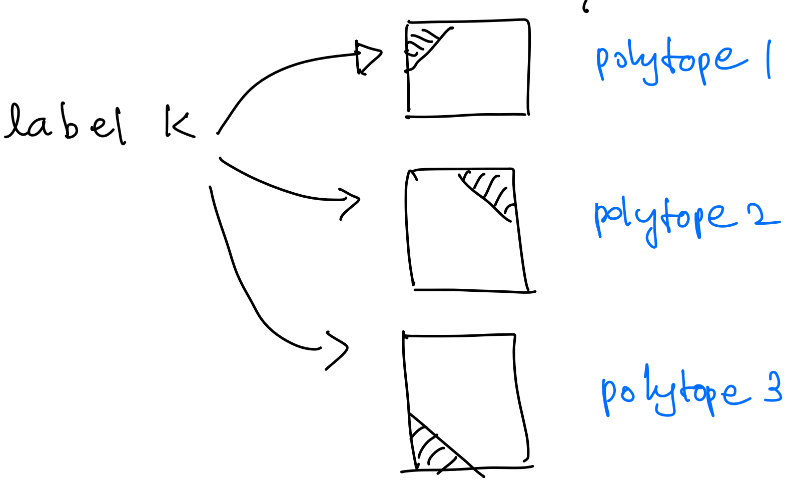
penultimate layers.





Kernel Density Network



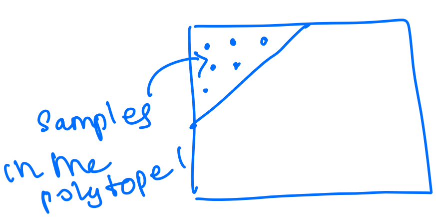


some polytope members has weighting of 1

other members would have a weighting based on their degree of match

polytope 2 and 3 would have less weighting/contribution

consider polytope 1 ⇒



samples ⇒ $(x_1, x_2, x_3, \dots, x_m)$

where $x_i \in \mathbb{R}^d$ (d-feature dimension)

* we are interested in fitting a Gaussian over the samples (aka polytope members)

We need to estimate parameters μ, Σ $\leftarrow$ $\phi(x|\mu,\Sigma)$

d-variate Gaussian

The joint distribution of the observed m polytope members. $\Rightarrow$

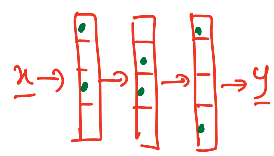
$$f(x_1, x_2, \dots, x_m | \mu', \Sigma')$$

If x_i $(i = 1, \dots, m)$ are iid,

$$\underbrace{f(x_1, x_2, \dots, x_m | \mu', \Sigma')} = \prod_{i=1}^{m} \phi(x_i | \mu, \Sigma)$$

$$\ell(\mu, \Sigma) = \prod_{i=1}^{m} \phi(x_i | \mu, \Sigma)$$

$$\hat{\mu}, \hat{\Sigma} = \underset{\mu, \Sigma}{\operatorname{argmax}} \prod_{i=1}^{m} \phi(x_i | \mu, \Sigma)$$



* In the KDN, all the sample points in a given polytope have the same activation pattern

eg:- $x_1 \rightarrow$ [1010 0110 1001]
$x_2 \rightarrow$ [1010 0110 1001]
$x_3 \rightarrow$ [1010 0110 1001]
$\vdots$
$x_m \rightarrow$ [1010 0110 1001]

* when $d \gg m$, it's not feasible to estimate Σ ($d \times d$ matrix) from just the m observations

* However, there might be other samples in the same class (but in different polytopes) that are approximately-matched with the activation patterns of the samples in the given polytope.

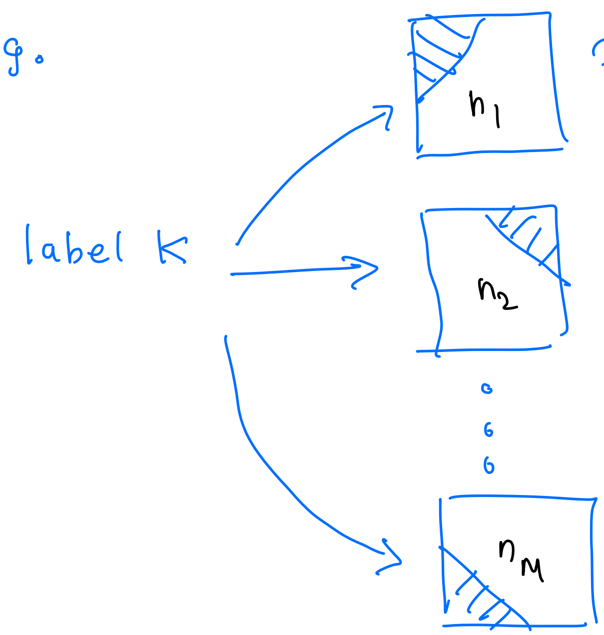
eg:- $x_1 \rightarrow [1010\ 0110\ 1001]$
$x^* \rightarrow [1010\ 1010\ 1001]$ $\Big\}$ $\underline{\underline{x_1 \approx x^*}}$

---

* How does Jayanta's weighting help with the incorporation of approximately-matched samples?

* we incorporate the samples from other polytopes of the same class to enrich the estimate we get for covariance & mean of underlying Gaussian.

e.g.



activation pattern

$P_k^{(1)}$ 1010 1001 1111

$P_k^{(2)}$ 1100 1001 1111

⋮

$P_k^{(M)}$ 1010 00101110

Consider fitting a Gaussian to $P_k^{(1)}$.

$P_k^{(1)}$ has n_1 samples but $n_1 << d$. Therefore we need more samples. $P_k^{(\ell)}$ where $\ell \neq 1$ are neighboring polytopes of $P_k^{(1)}$ that belong to the same class. We can leverage these neighbors to increase the sample size.

However the contributions from other polytopes would not be same as that in $P_k^{(1)}$. To correct for this contribution mismatch we can weight the samples from other polytopes.

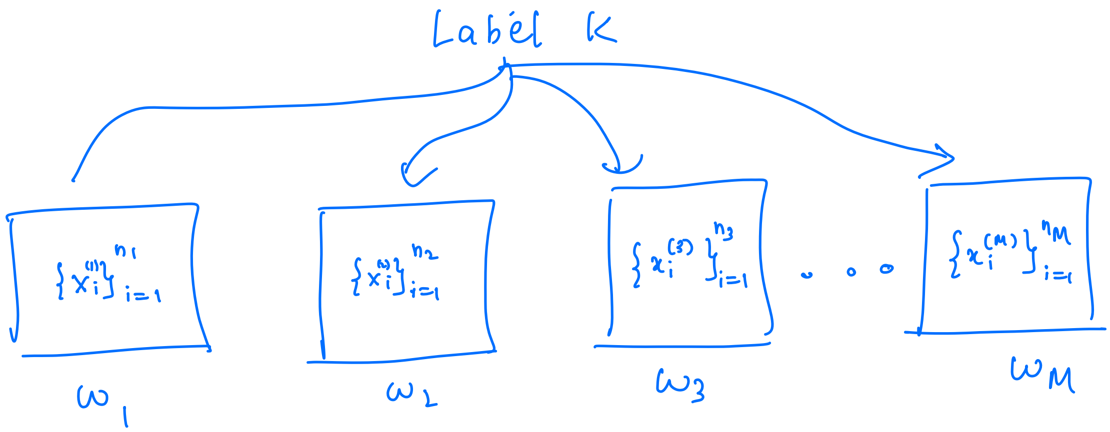
We can leverage the degree of activation pattern mismatch (match) to compute the weights for each polytope.

eg:- 1010 1001 1111 ← $P_k^{(1)}$'s activation pattern

101 1 0 001 0111 ← $P_k^{(\ell)}$ " "

∴ w_ℓ (weight assigned to $P_k^{(\ell)}$ samples) should reflect this degree of similarity / disagreement / match.

contd...



$x_1 \quad x_2 \quad x_3 \quad \cdots \quad x_m$

$\omega_1 \quad \omega_2 \quad \omega_2 \quad \cdots \quad \omega_m$

iid

$$f(x_1, x_2, \dots, x_m \mid \mu, \Sigma) = \prod_{i=1}^{m} \phi(x_i \mid \mu, \Sigma)$$

$$\ell(\mu, \Sigma \mid x_1, \dots, x_m) = \prod_{i=1}^{m} \phi(x_i \mid \mu, \Sigma)$$

since the contributions of each sample point to the likelihood function is different, we apply the weighting to each $\phi(x_i \mid \mu, \Sigma)$

$$\ell_{\text{weighted}}(\mu, \Sigma \mid x_1, \dots, x_m) = \prod_{i=1}^{m} \phi(x_i \mid \mu, \Sigma)^{w_i}$$

$\Big\downarrow$ using MLE

$$\hat{\mu} = \frac{\sum_{i=1}^{m} w_i x_i}{\sum_{i=1}^{m} w_i} \quad \& \quad \hat{\Sigma} = \frac{\sum_{i=1}^{m} w_i (x_i - \hat{\mu})(x_i - \hat{\mu})^T}{\sum_{i=1}^{m} w_i}$$

* Through this weighting scheme we can incorporate the data from the neighboring polytope to estimate the gaussian kernel parameters. Since more data is being used for the estimation, the $\hat{\mu}$, $\hat{\Sigma}$ would be better estimates.