CS484/684 Computational Vision

Deep Clustering

UNIVERSITY OF WATERLOO

unsupervised classification ≡ clustering

Outline

UNIVERSITY OF WATERLOO

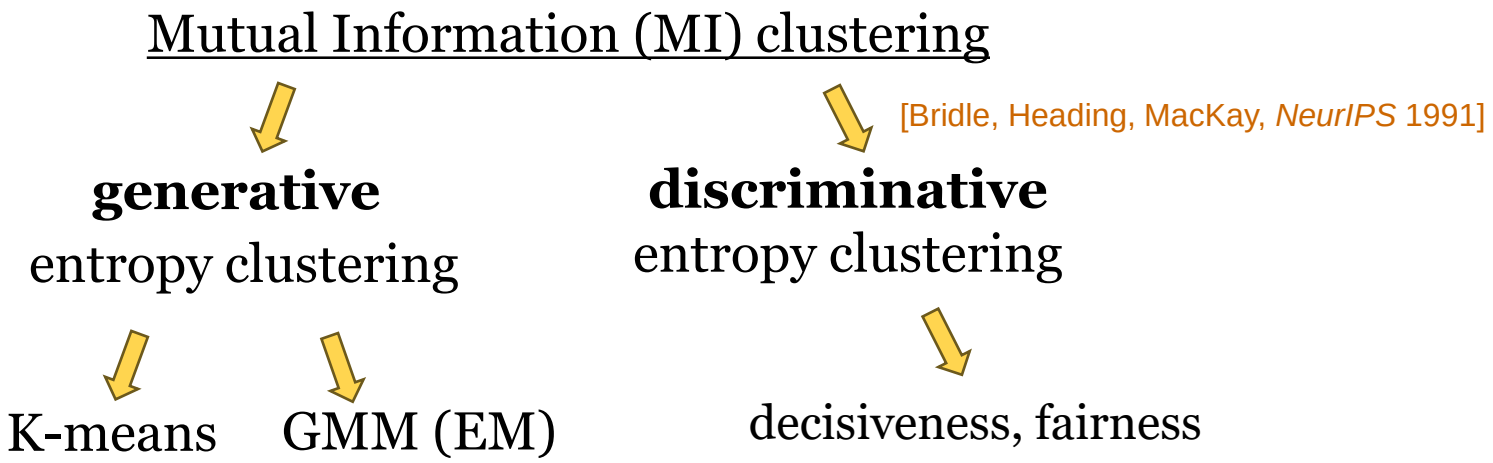
I. Intro: unified view on common unsupervised losses

- variance, entropy, mutual information
- **generative** (K-means, GMM, …) & **discriminative** (*decisiveness*, *fairness*, …)

II. Clustering with *softmax* models

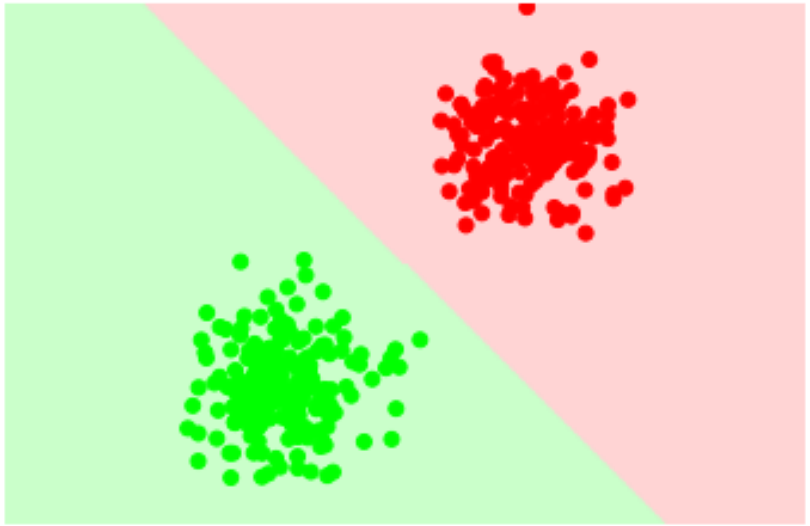
- relation to K-means and SVM clustering
- **max-margin property** for generalized decisiveness (*Renyi* entropy)
- optimization: self-labeling, **soft** pseudo-labels
- **collision cross-entropy**

I. Intro: unified view on common unsupervised losses



K-means

$$\min_{S,\boldsymbol{\mu}} E(S,\boldsymbol{\mu}) \quad = \quad \sum_{k=1}^{K} \sum_{p \in S^k} \|f_p - \mu_k\|^2$$

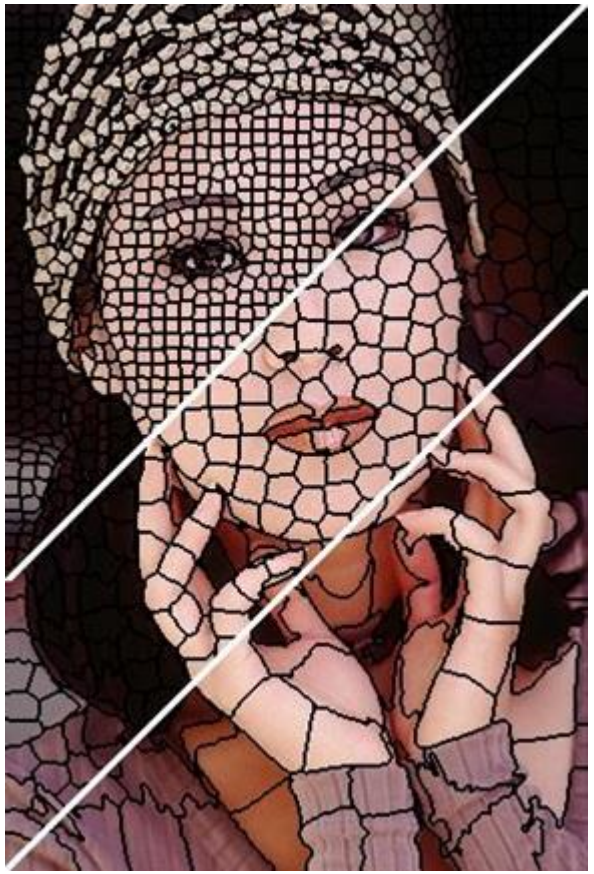


$$\Leftrightarrow \quad \min_{S} \sum_{k=1} |S^k| \, \mathbf{var}(S^k)$$

variance clustering

color quantization
- RGB feat ures *f*

superpixels
- RGBXY features *f*



[Achanta et al., PAMI 2011]

K-means +

kernelized formulation

$$\min_{S,\boldsymbol{\mu}} E(S,\boldsymbol{\mu}) = \sum_{k=1}^{K} \sum_{p \in S^k} \|f_p - \mu_k\|^2$$

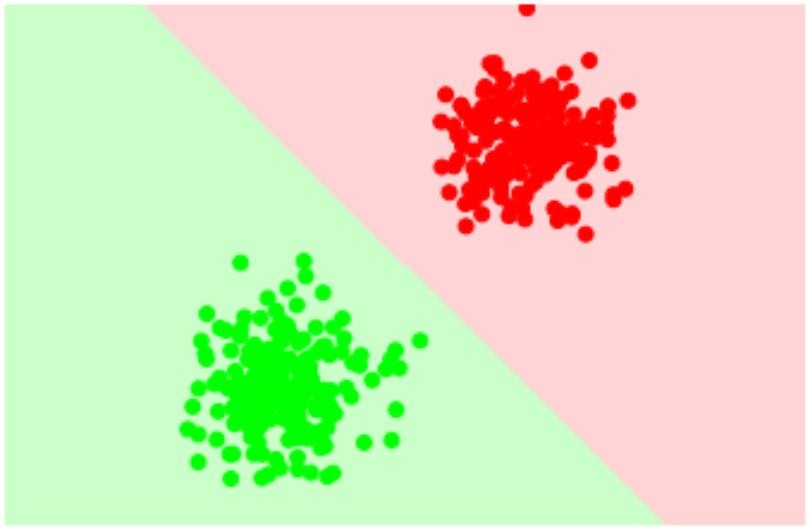
$$\min_{S} E(S) = \sum_{k} \sum_{pq \in S^k} \frac{\|f_p - f_q\|^2}{2|S^k|}$$

$$= -\sum_{k} \sum_{pq \in S^k} \frac{\langle f_p, f_q \rangle}{|S^k|} + const$$

with Gaussian kernel $\langle\ ,\ \rangle_G$

unsupervised segmentation

- implicit high-dimensional features f



$$\Leftrightarrow \quad \min_{S} \sum_{k=1} |S^k|\ \mathbf{var}(S^k)$$

variance clustering

Normalized Cuts [Shi&Malik 2000]

K-means +

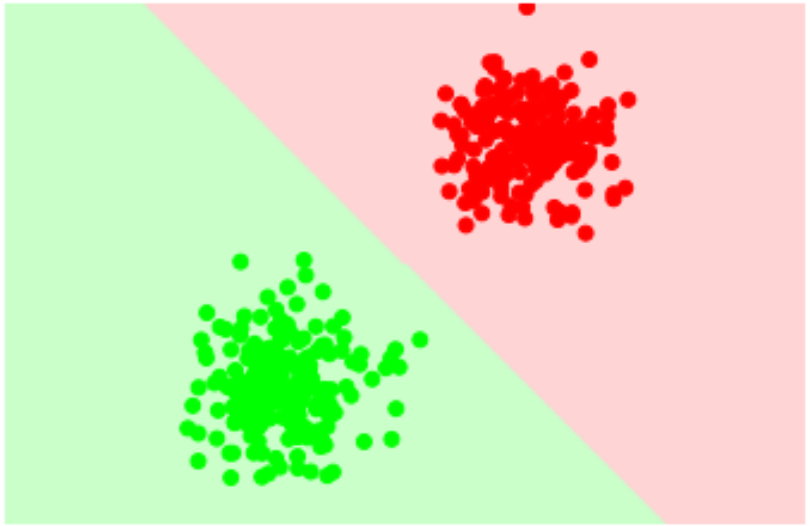
probabilistic formulation

$$\min_{S,\boldsymbol{\mu}} E(S,\boldsymbol{\mu}) = \sum_{k=1}^{K} \sum_{p \in S^k} \|f_p - \mu_k\|^2$$

$$-\sum_{k} \sum_{p \in S^k} \ln e^{-\|f_p - \mu_k\|^2}$$

Gaussian density

negative log-likelihoods

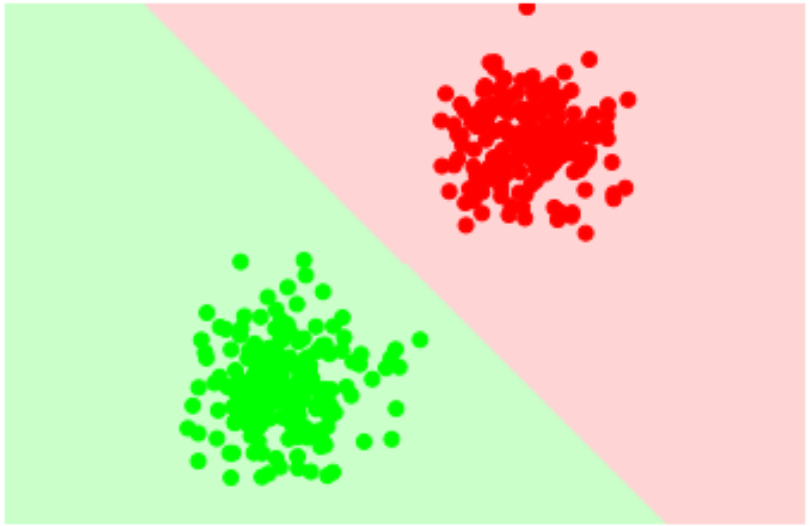


$$\Leftrightarrow \quad \min_{S} \sum_{k=1} |S^k| \, \mathbf{var}(S^k)$$

variance clustering

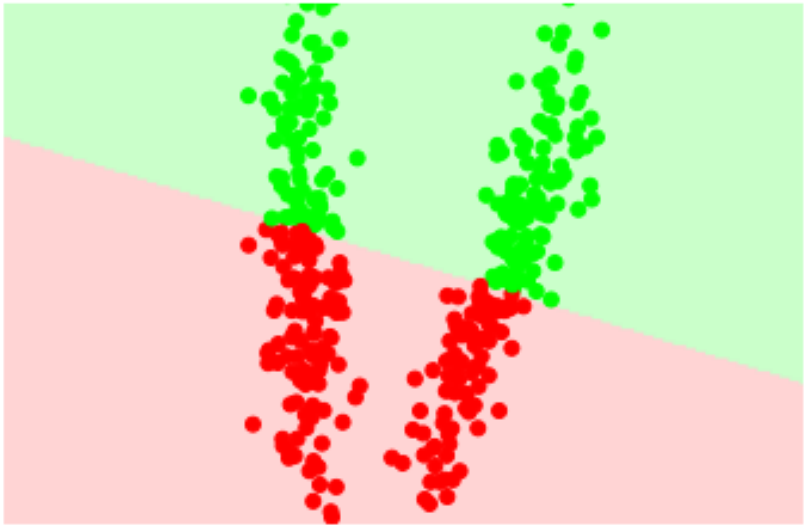
Probabilistic K-means

$$\min_{S,P} E(S,P) \quad = \quad -\sum_{k=1}^{K} \sum_{p \in S^k} \ln P_k(f_p)$$



isotropic Gaussian densities

$$P_k \in \mathcal{N}(\mu_k, I)$$

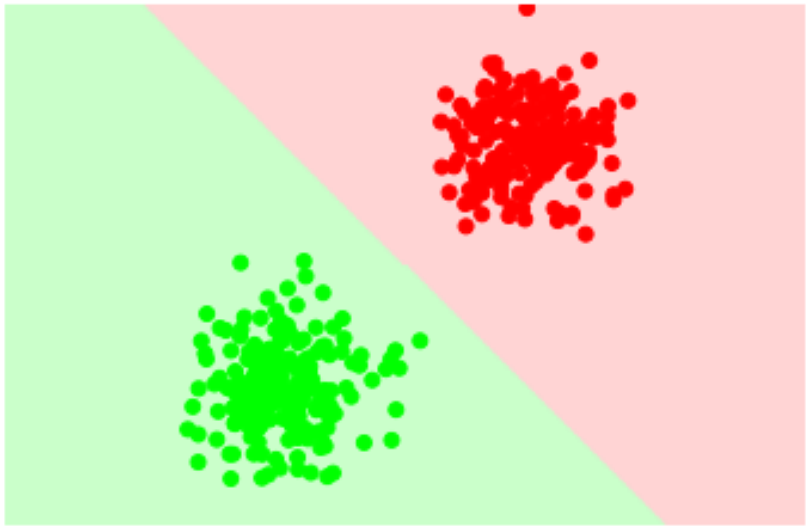


isotropic Gaussian densities

$$P_k \in \mathcal{N}(\mu_k, I)$$

Probabilistic K-means

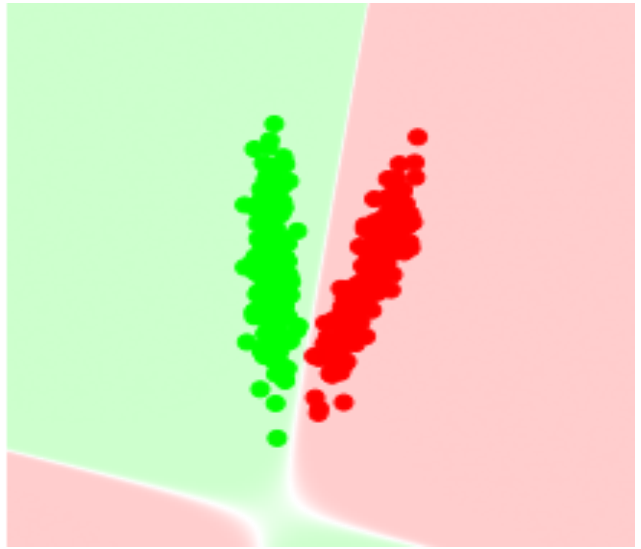
$$\min_{S,P} E(S,P) \quad = \quad -\sum_{k=1}^{K} \sum_{p \in S^k} \ln P_k(f_p)$$



isotropic Gaussian densities

$$P_k \in \mathcal{N}(\mu_k, I)$$

elliptic K-means or GMM



general Gaussian densities

$$P_k \in \mathcal{N}(\mu_k, \Sigma_k)$$

clustering by fitting probability density models

Probabilistic K-means

$$\min_{S,P} E(S,P) \;=\; -\sum_{k=1}^{K}\sum_{p\in S^k} \ln P_k(f_p)$$

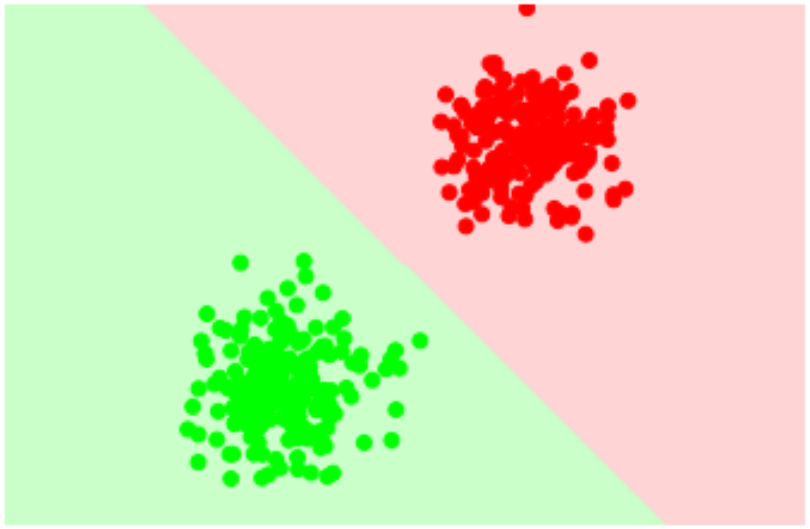
Monte Carlo approximation $\approx$

$$-\int \ln P_k(f)\, dS_k(f)$$

$$\sum_k |S_k| \; H(S_k, P_k)$$

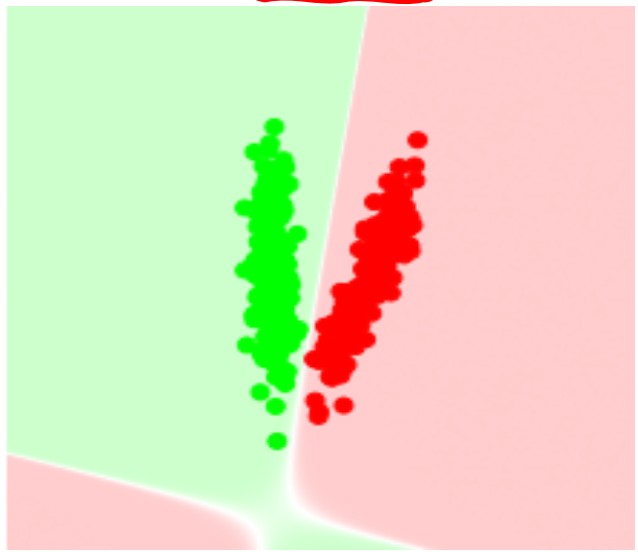
cross-entropy between true data density in S_k and density model P_k

$$H(S_k) \;+\; KL(S_k \,\|\, P_k)$$



isotropic Gaussian densities

$$P_k \in \mathcal{N}(\mu_k, I)$$



general Gaussian densities

$$P_k \in \mathcal{N}(\mu_k, \Sigma_k)$$

clustering by fitting probability density models

Entropy clustering (generative)

$$\min_{S,P} \sum_k |S_k| \underbrace{H(S_k, P_k)}_{H(S_k) \;+\; KL(S_k \,\|\, P_k)}$$

$$\min_{S} \sum_k |S_k| \; H(S_k)$$

entropy of data density in S_k

uni-modal objects

isotropic Gaussian densities

$$P_k \in \mathcal{N}(\mu_k, I)$$

fitting Gaussian densities

multi-modal objects

general density models

$$P_k \in \mathcal{P}(S_k)$$

fitting general density models

(histograms, kernel densities, mixtures)

variance clustering

$$\min_S \sum_k |S_k| \; var(S_k)$$

entropy clustering

$$\min_S \sum_k |S_k| \; H(S_k)$$

uni-modal objects

isotropic Gaussian densities

$$P_k \in \mathcal{N}(\mu_k, I)$$

multi-mode object

general density models

$$P_k \in \mathcal{P}(S_k)$$

fitting Gaussian densities

fitting general density models

(histograms, kernel densities, mixtures)

"complex" data $\Longrightarrow$ "complex" densities

UNIVERSITY OF WATERLOO

elliptic K-means or GMM

quadr
decisi
bounda

general Gaussian densities

$$P_k \in \mathcal{N}(\mu_k, \Sigma_k)$$

fitting Gaussian densities

Bayesian posterior for estimated Gaussian densities:

$$P(k|x) = \frac{P(x|k)P(k)}{\sum_j P(x|j)\,P(j)} \sim \frac{e^{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)}}{\sum_j e^{-\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)}}$$

quadratic Gaussian classifier

model complexity

	basic K-means	GMM
number of parameters:	$K \times N$	$K \times N + K \times N^2$

Data is *linearly separable*
can we stick to a linear classifier?

Towards discriminative models

Bayesian posterior implied by density models:

$$P(k|x) = \frac{P(x|k)P(k)}{\sum_j P(x|j)\,P(j)} \sim \frac{e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}}{\sum_j e^{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)}}$$

replace by a **posterior model**

e.g. softmax + linear classifier $\sigma^k(\boldsymbol{w}x) = \frac{e^{w_k \cdot x}}{\sum_j e^{w_j \cdot x}}$

number of parameters $K \times N$

$$\sigma_{\mathbf{w}} := \sigma(\mathbf{w}x)$$

Clustering with *softmax* models? Losses?

Entropy clustering with *softmax* models

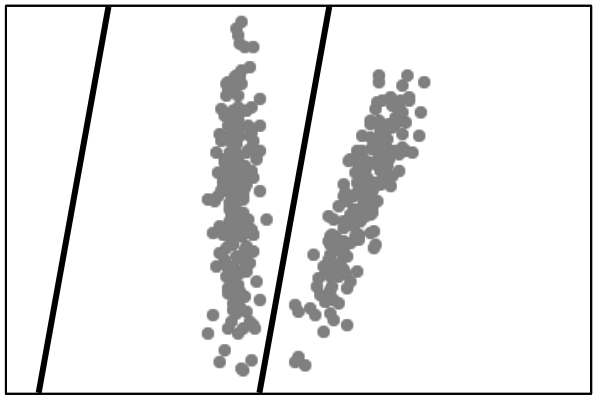
UNIVERSITY OF WATERLOO

average entropy of prediction $\overline{H(\sigma)} := \frac{\sum_p H(\sigma_p)}{|\Omega|}$

entropy of average prediction $-H(\overline{\sigma}) := -H\left(\frac{\sum_p \sigma_p}{|\Omega|}\right)$

Discriminative properties:

1. **Decisiveness**: avoid data points near the boundary
   - clustering [Bridle, Heading, MacKay, *NeurIPS* 1991]
   - semi-supervised learning [Grandvalet & Bengio, NeurIPS'04]
2. **Fairness**: similar cluster cardinalities



assuming linear classifier $\sigma_{\mathbf{w}} := \sigma(\mathbf{w}x)$

Mutual Information for Clustering

UNIVERSITY OF WATERLOO

$$\frac{1}{|\Omega|}\sum_k |S_k|\, H(S_k)$$

(as in earlier slides)

$$H\left(\frac{1}{|\Omega|}\sum_p \sigma_p\right) \qquad \frac{1}{|\Omega|}\sum_p H(\sigma_p)$$

"fairness" **"decisiveness"**

[Bridle, Heading, MacKay, *NeurIPS* 1991]

needed only if representation is not fixed

$$I(X, C) = H(X) - H(X|C) \equiv H(C) - H(C|X)$$

data, class predictions

entropy of all data density

(average) entropy of data density in each class

entropy of average predictions

average entropy of prediction

generative algorithms
(optimizing density models)

discriminative algorithms
(optimizing prediction models)

Q: equivalence of generative and discriminative algorithms for clustering ???

A: practical clustering algorithms **may have different hypothesis spaces**
(density models, classifier models)

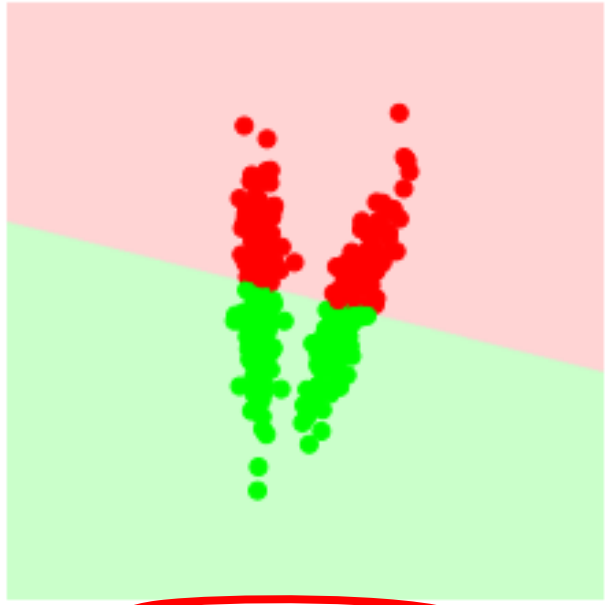
II. Discriminative clustering with *softmax* models

- relation to K-means and SVM clustering
- **max-margin property** for generalized decisiveness (*Renyi* entropy)
- optimization: self-labeling, pseudo-labels
- **collision cross-entropy**

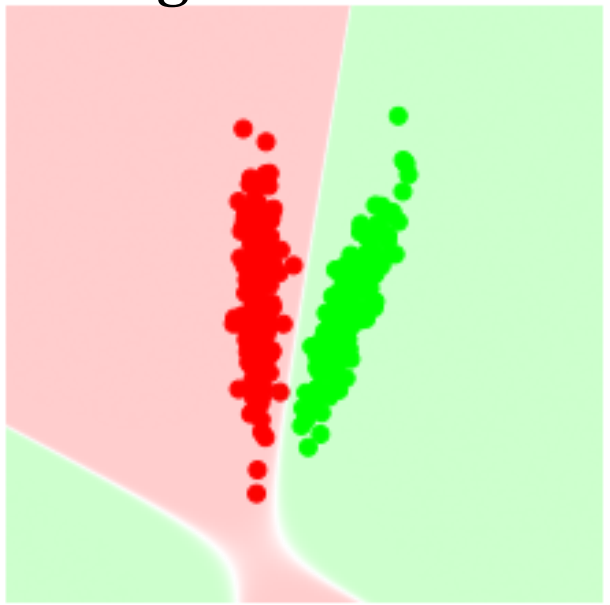
Generative vs. Discriminative

Generative clustering

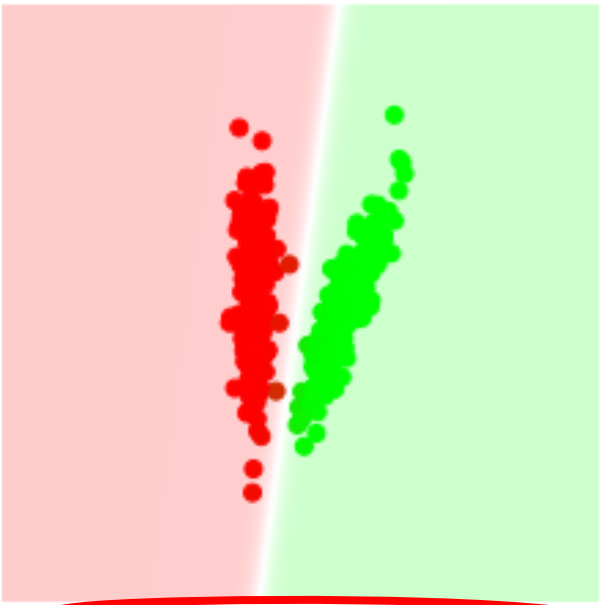
Discriminative clustering



K-means
$\Theta(K \times N)$



GMM
$\Theta(K \times N^2)$



decisive & fair $\sigma_{\mathbf{w}}$
$\Theta(K \times N)$

Theorem [Jabi et.al. PAMI'21] : Decisive & fair linear classifier $\equiv$ K-means

Our Theorem [tbs] : **Decisiveness** has **margin maximizing property**

Global and Local Minima

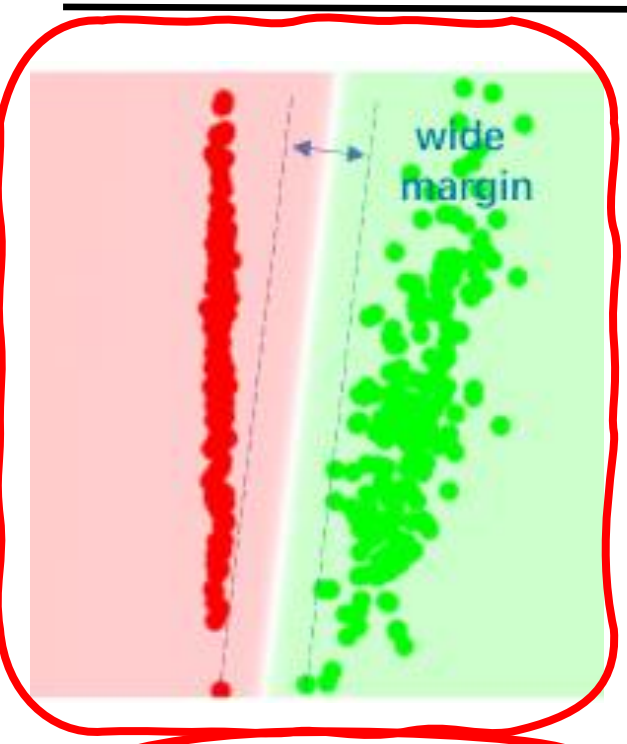
UNIVERSITY OF WATERLOO

discriminative entropy clustering

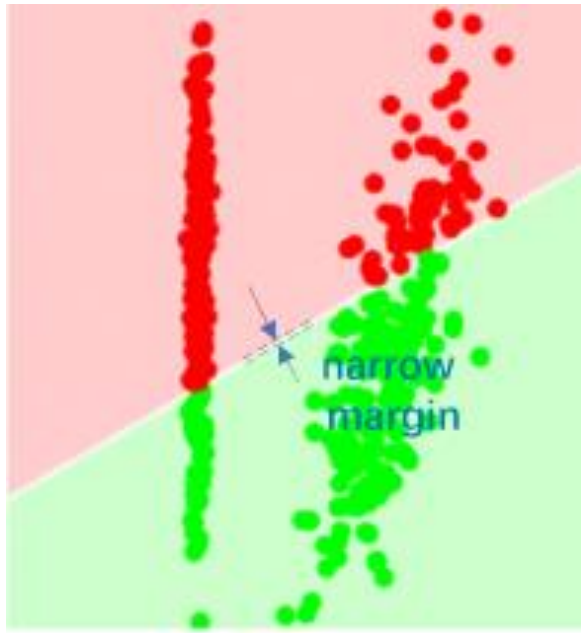
$$CE(\mathbf{w}) = \overline{H(\sigma_{\mathbf{w}})} - H(\overline{\sigma_{\mathbf{w}}}) + \gamma \|\mathbf{w}\|^2$$

"regularized" **soft K-means** [Jabi et.al. PAMI'21]

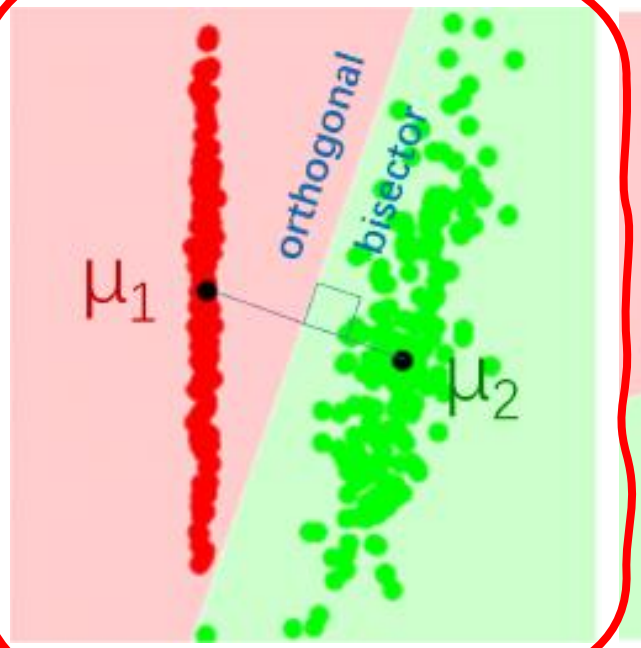
$$sKM(S, \boldsymbol{\mu}) = \sum_k \sum_p S_p^k \|x_p - \mu_k\|^2 - \gamma \overline{H(S)} - \overline{\|x\|^2}$$



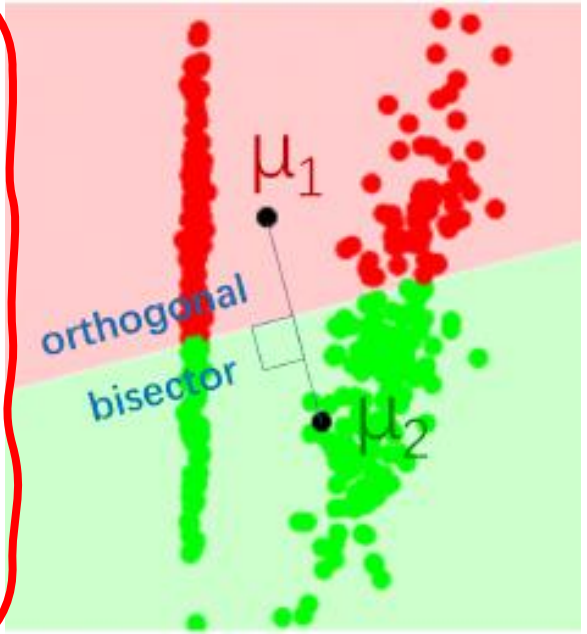
CE loss = -0.6910



CE loss = -0.6610



sKM loss = 419.60



sKM loss = 401.45

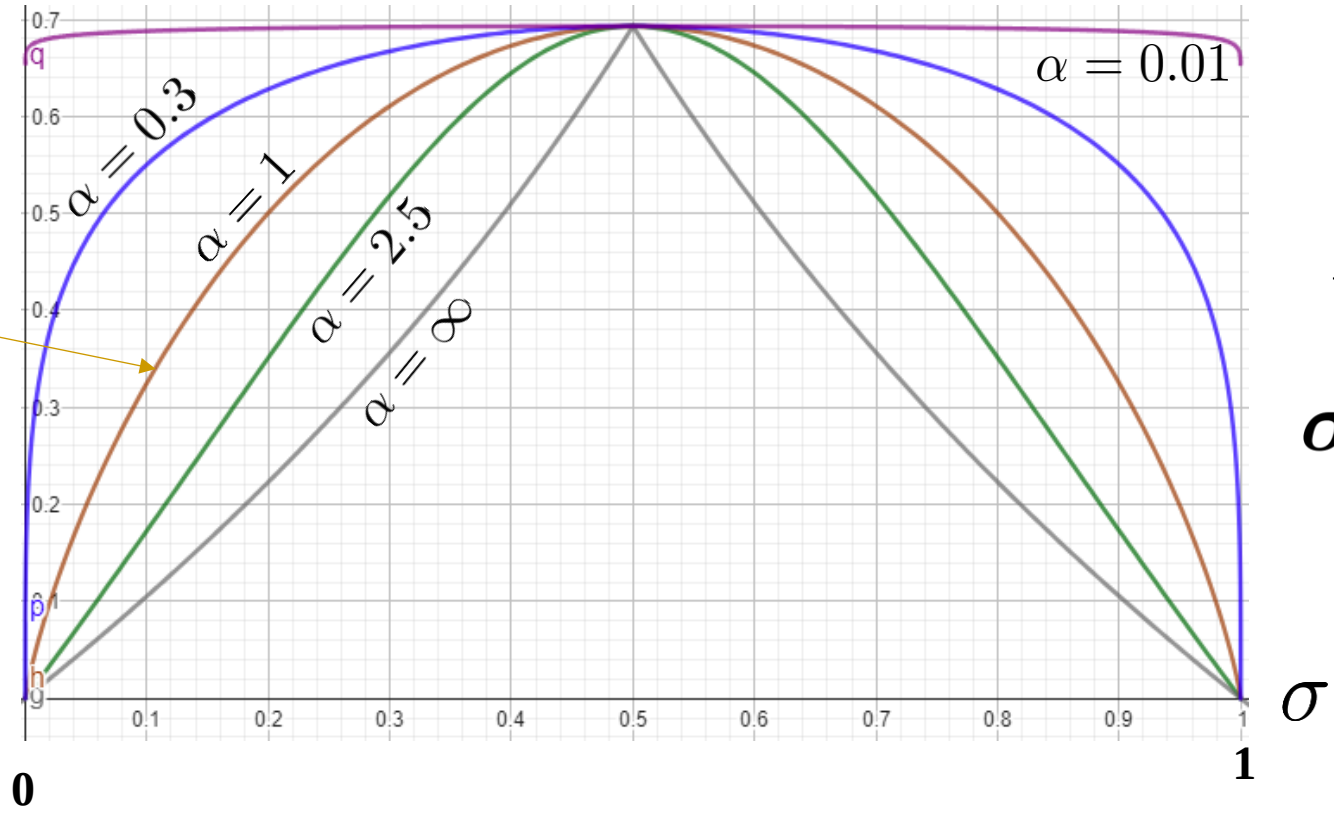
Generalized Decisiveness: Renyi Entropy (R_α)

$$\overline{H(\sigma)} := \frac{\sum_p H(\sigma_p)}{|\Omega|}$$

[A. Renyi, 1961]

Shannon

$H(\boldsymbol{\sigma}) = R_1(\boldsymbol{\sigma})$



binary case, K=2

$$\boldsymbol{\sigma} := (\sigma, 1-\sigma)$$

We prove a margin maximizing property for Renyi decisiveness of any order $\alpha > 0$.

Margin Maximization: Renyi Decisiveness (R_α)

UNIVERSITY OF WATERLOO

Theorem [Our generalization to clustering]
Consider any set *FL* of feasible (fair) linearly separable binary labelings/clusterings.
Assuming **w**(γ) minimizes **regularized decisiveness R_α** over liner classifiers $\mathbf{W}^{FL}$ consistent with *FL*

unsupervised

clustering loss

$$\arg \min_{\mathbf{w} \in \mathbf{W}^{FL}} \gamma \|\mathbf{w}\|^2 + \overline{R_\alpha(\boldsymbol{\sigma}_{\mathbf{w}})}$$

then $\dfrac{\mathbf{w}(\gamma)}{\|\mathbf{w}(\gamma)\|} \xrightarrow{\gamma \to 0} \mathbf{u}^{\hat{\mathbf{y}}}$ corresponding to the max-margin clustering $\hat{\mathbf{y}} \in FL$.

- unsupervised generalization of max-margin property of logistic regression [Rosset, Zhu, Hastie, 2003]
- relates to SVM clustering [Xu,Neufeld,Larson,Schuurmans, 2004]

Global minima for various γ

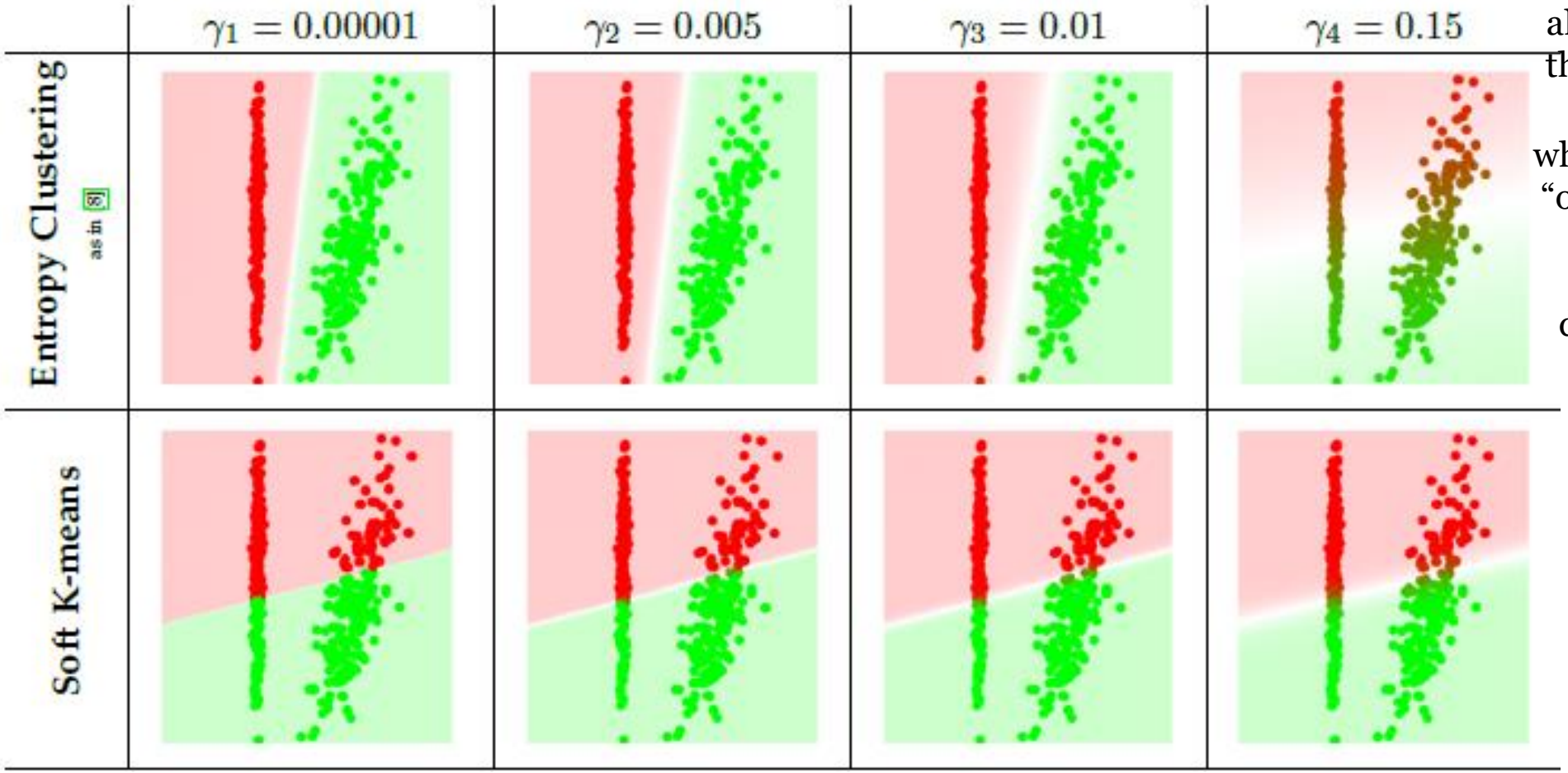
UNIVERSITY OF WATERLOO

$\ldots\gamma\,\|\mathbf{w}\|^2\ldots$

γ —› 0 maximizes margin

$\ldots-\gamma\,\overline{H(S)}\ldots$

γ controls "softness" only



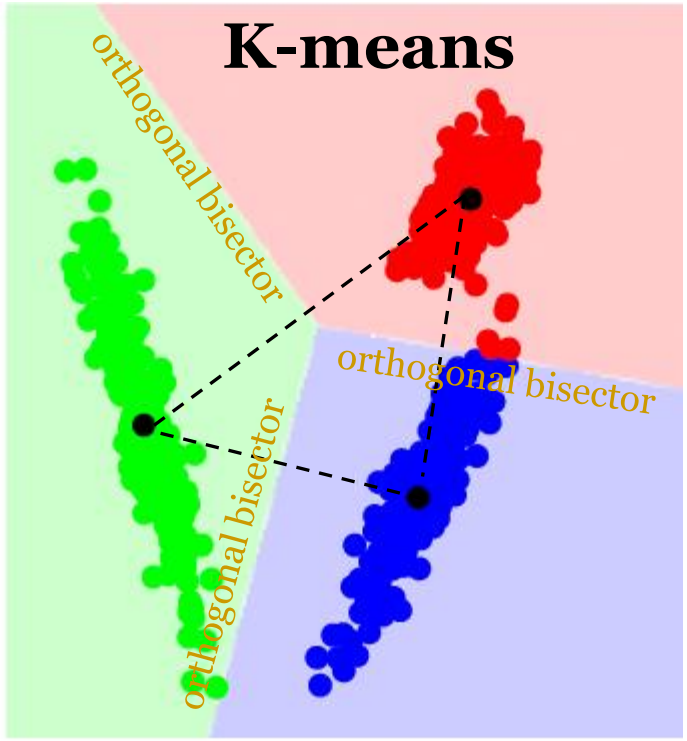
Larger γ reduces ||w|| also softening the boundary,

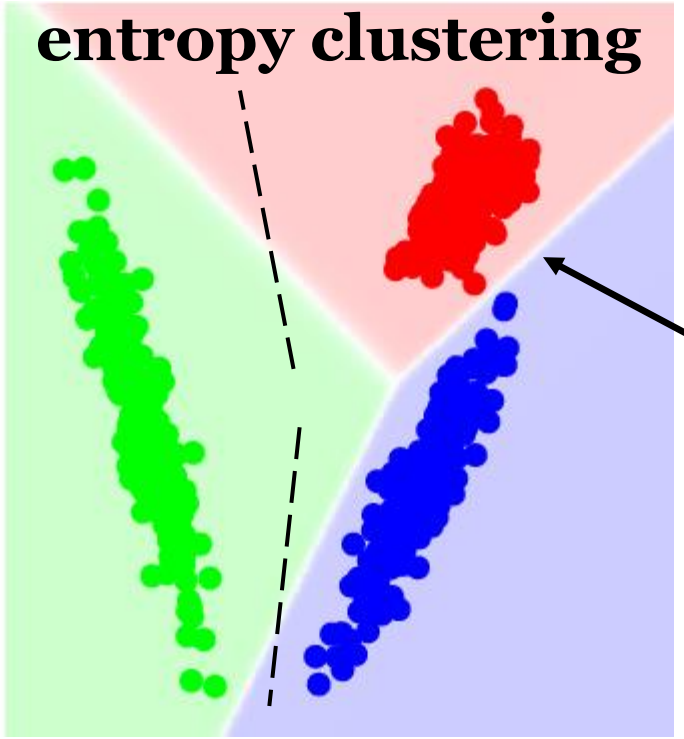
which orients it "orthogonally" to data to improve decisiveness

Multi-class example (K=3)

would be max-margin
decision boundary
for green and red

does not care about margins





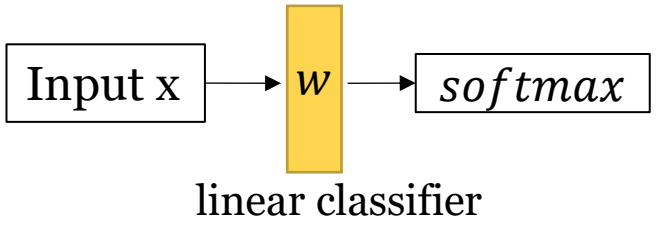
maximizes only
minimum pairwise margin

(red–blue in this case)

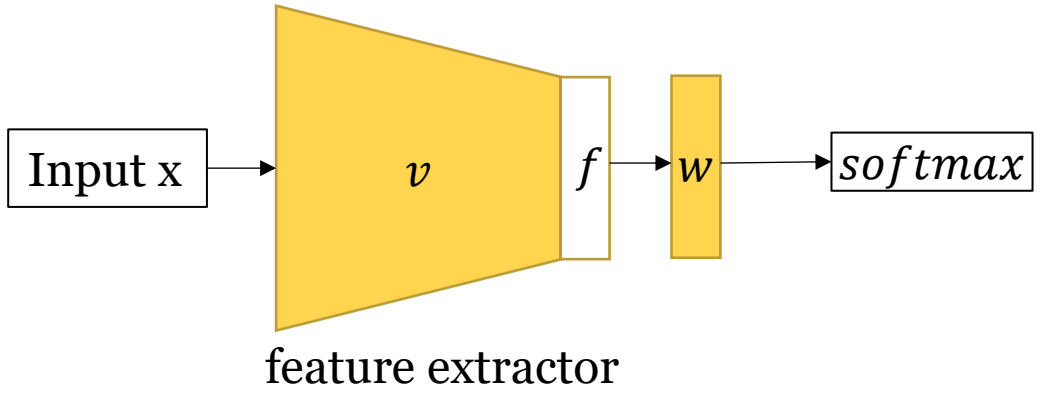
would be max-margin
decision boundary
for green and blue

Non-linear (deep) clustering

linear posterior model



deep posterior model



$$\sigma(\boldsymbol{w}x)$$

$$\sigma(\boldsymbol{w}f_{\boldsymbol{v}}(x))$$

ANY backbone

$$H(\bar{\sigma}) - \overline{H(\sigma)}$$

applies to **any** softmax model

MNIST clustering: linear classifier + low-level features

Linear classifier + low-level features (**raw image intensities**)

loss:	K-means ($\boldsymbol{\mu}$)	MI ($\boldsymbol{w}$)
predictor:	$\|\boldsymbol{\mu}^k - x\|$	$\sigma^k(\boldsymbol{w}x)$
accuracy:	53.2%	60.2%

Note: Hungarian matching is used
to match with ground truth classification

MNIST clustering: linear classifier + deep features

UNIVERSITY OF WATERLOO

fixed pretrained features

loss:	K-means ($\boldsymbol{\mu}$)	MI ($\boldsymbol{w}$)
predictor:	$\|\boldsymbol{\mu}^k - f(x)\|$	$\sigma^k(\boldsymbol{w}f(x))$
accuracy: *(pretrained features)*	50.46% - resnet18	52.77% - resnet18

w/ feature finetuning

loss:	MI ($\boldsymbol{v}, \boldsymbol{\mu}$)	MI ($\boldsymbol{v}, \boldsymbol{w}$)
predictor:	$\|\boldsymbol{\mu}^k - f_{\boldsymbol{v}}(x)\|$	$\sigma^k(\boldsymbol{w}f_{\boldsymbol{v}}(x))$
accuracy: *(pretrained features)*	65.01% - resnet18	73.22% - resnet18

MNIST clustering: linear classifier + deep features

UNIVERSITY OF WATERLOO

fixed pretrained features

loss:	K-means ($\boldsymbol{\mu}$)	MI ($\boldsymbol{w}$)
predictor:	$\|\boldsymbol{\mu}^k - f(x)\|$	$\sigma^k(\boldsymbol{w}f(x))$
accuracy: *(pretrained features)*	50.46% - resnet18	52.77% - resnet18

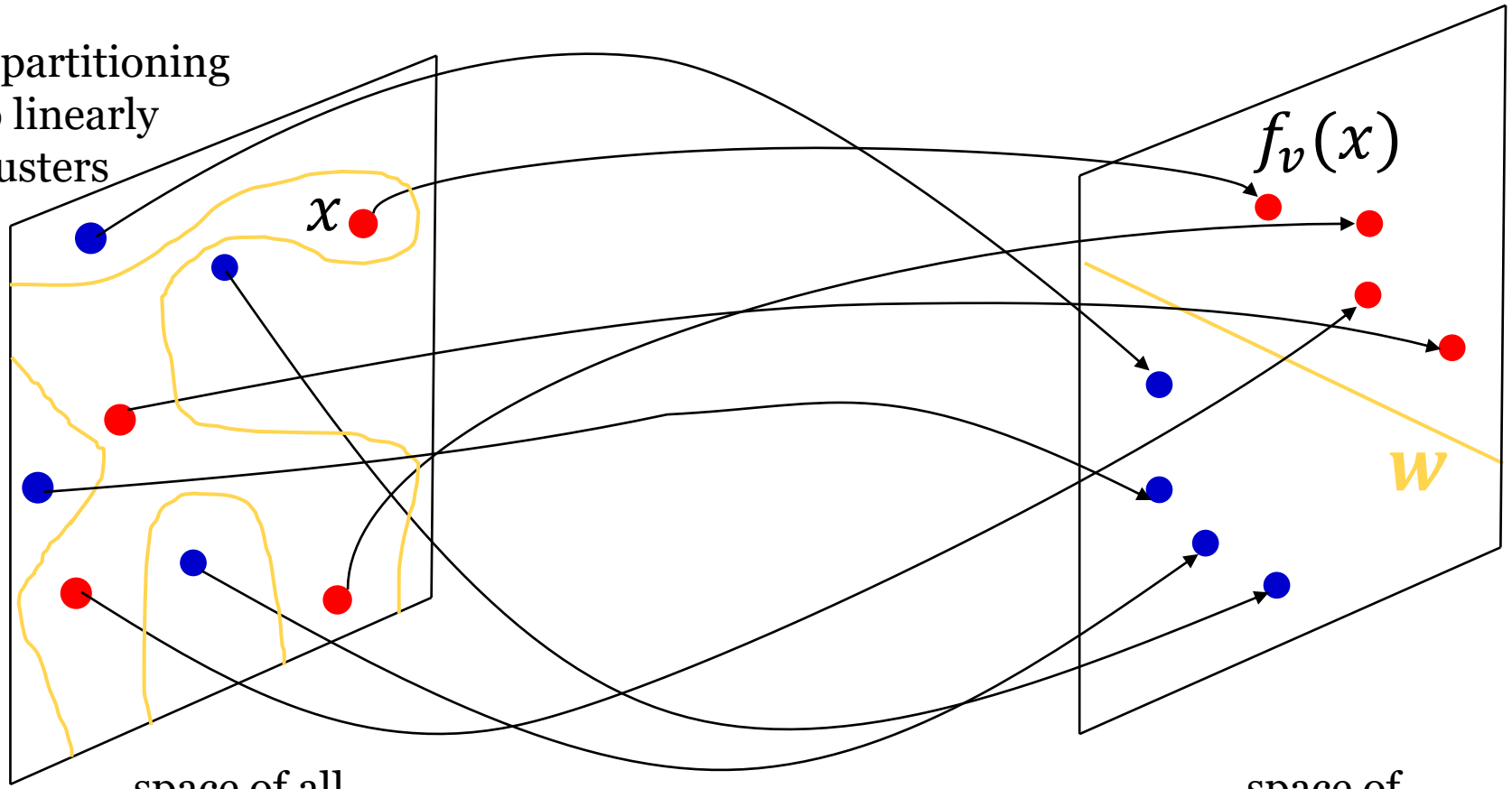
w/ feature finetuning

loss:	MI ($\boldsymbol{v}, \boldsymbol{\mu}$)	MI ($\boldsymbol{v}, \boldsymbol{w}$)
predictor:	$\|\boldsymbol{\mu}^k - f_{\boldsymbol{v}}(x)\|$	$\sigma^k(\boldsymbol{w}f_{\boldsymbol{v}}(x))$
accuracy: *(pretrained features)*	65.01% - resnet18	73.22% - resnet18
	+self-augmentation	**+self-augmentation**
	89.01%	97.37%

perhaps margin maximization matters in practice

Why self-augmentation?

Arbitrary input partitioning can be mapped to linearly separable deep clusters



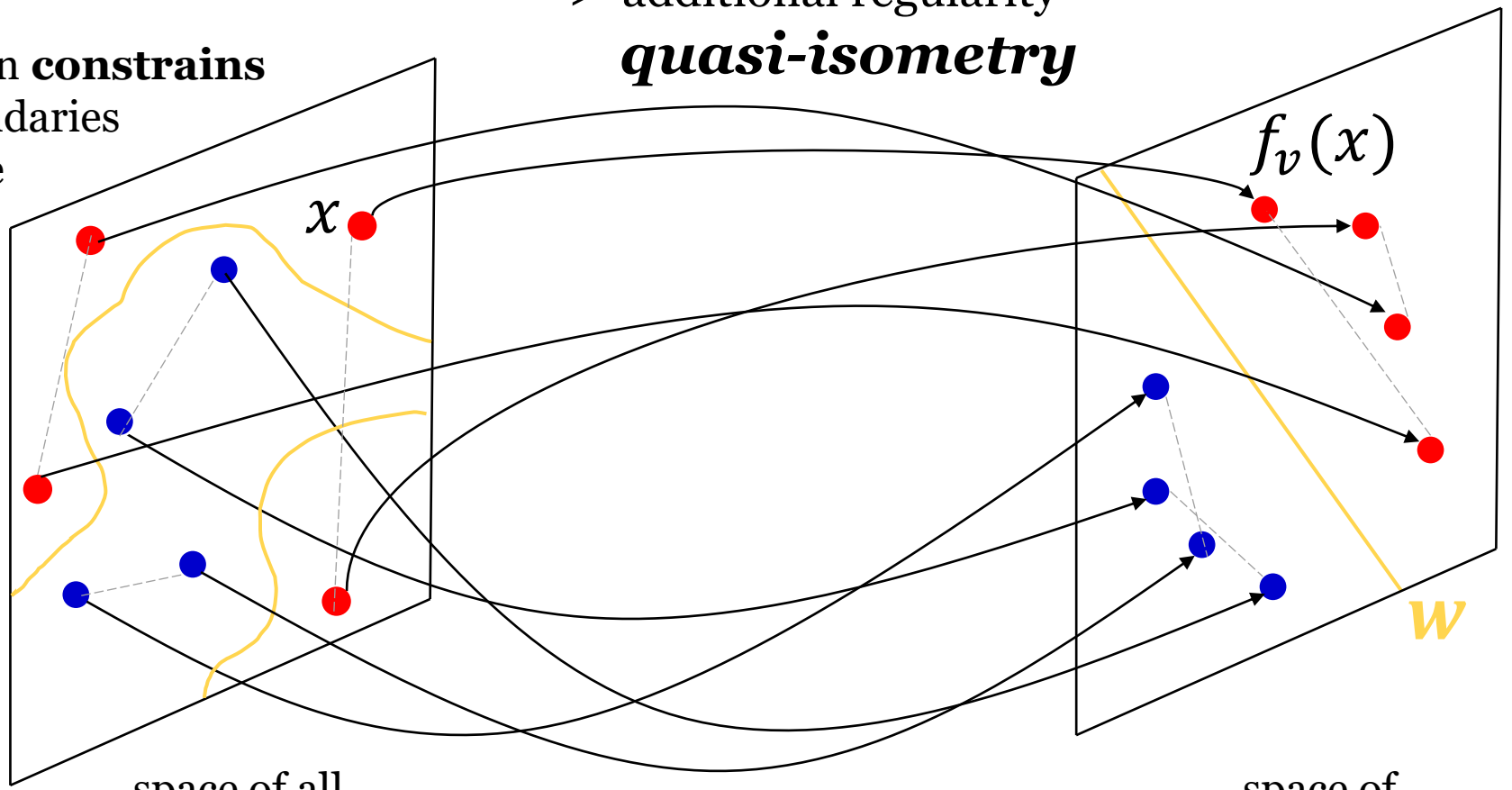
Why self-augmentation?

UNIVERSITY OF WATERLOO

=> additional regularity

quasi-isometry

Self-augmentation **constrains** the decision boundaries in the input space



space of all possible inputs

mapping f_v

under-constrained

space of deep features

II. Algorithms for discriminative clustering with *softmax* models

- gradient descent
- self-labeling, pseudo-labels
- **collision cross-entropy**

Optimization

[Bridle, Heading, MacKay, NeurIPS 1991]

$$L(\sigma) = \overline{H(\sigma)} - H(\overline{\sigma})$$

concave — convex

← **Gradient Descent**

"self-labeling" surrogate with *pseudo-labels* y

$y \approx \sigma$

$$L(\sigma, y) = \overline{H(y, \sigma)}$$

$$s.t. \;\; y \in \Delta_{0,1}^{K}$$

"hard" **pseudo-labels**

$$\bar{y} = u$$ ← uniform distribution

- integer programming for *y*

← **Optimal Transport**

- standard network training with CE

← **Gradient Descent**

[Asano, Rupprecht, Vedaldi, ICLR 2020]

Optimization

[Bridle, Heading, MacKay, NeurIPS 1991]

$$L(\sigma) = \overline{H(\sigma)} - H(\overline{\sigma})$$

← **Gradient Descent**

"self-labeling" surrogate with *pseudo-labels* y

$y \approx \sigma$

$$L(\sigma, y) = \overline{H(y, \sigma)} - H(\overline{y}) \qquad s.t. \; y \in \Delta^K$$

linear w.r.t. y — convex

- soft pseudo-labels

- convex w.r.t. *y*

- standard network training with CE

← **approximate solver**

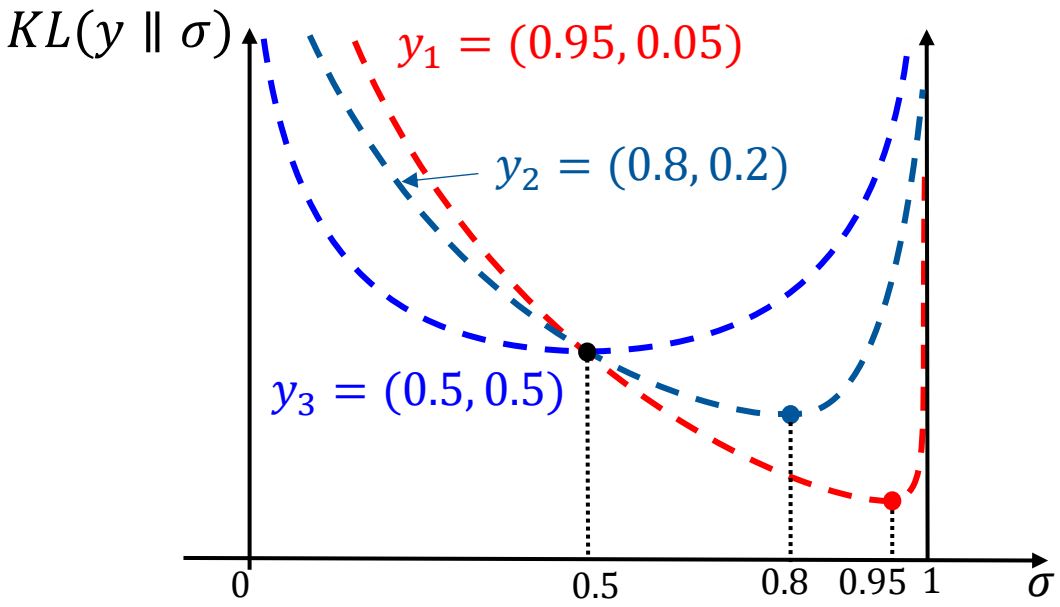
← **Gradient Descent**

[Jabi et.al. PAMI 2021]

Standard CE and soft pseudo-labels

$$H(y, \sigma) \quad = \quad KL(y \parallel \sigma) + H(y)$$

this loss minimizes the distance/divergence between two distributions



$$\sigma_k \approx y_k \quad \forall k$$

$\Pr(C = k|x)$ $\Pr(T = k|x)$

predicted class C true class T

$$\mathbf{Pr(\ C\ =\ T\ )}$$

How about maximizing the **probability of "collision" between C and T** ?

Collision Cross Entropy $H_C(\sigma, y)$ vs. ~~$H(y, \sigma)$~~

UNIVERSITY OF WATERLOO

target distribution

estimated distribution

$$-\ln \Pr(C = T) = -\ln \sum_k \sigma_k \cdot y_k$$

symmetric loss maximizing probability of equality/collision between two random variables

~~$\sigma_k \approx y_k$~~ $\forall k$

$\Pr(C = k|x)$ $\Pr(T = k|x)$

predicted class C true class T

$$\sum_k \sigma_k \cdot y_k = \sum_k \Pr(C = k, T = k) = \Pr(C = T)$$

How about maximizing the **probability of "collision" between C and T** ?

Collision Cross Entropy $H_C(\sigma, y)$ vs. ~~$H(y, \sigma)$~~

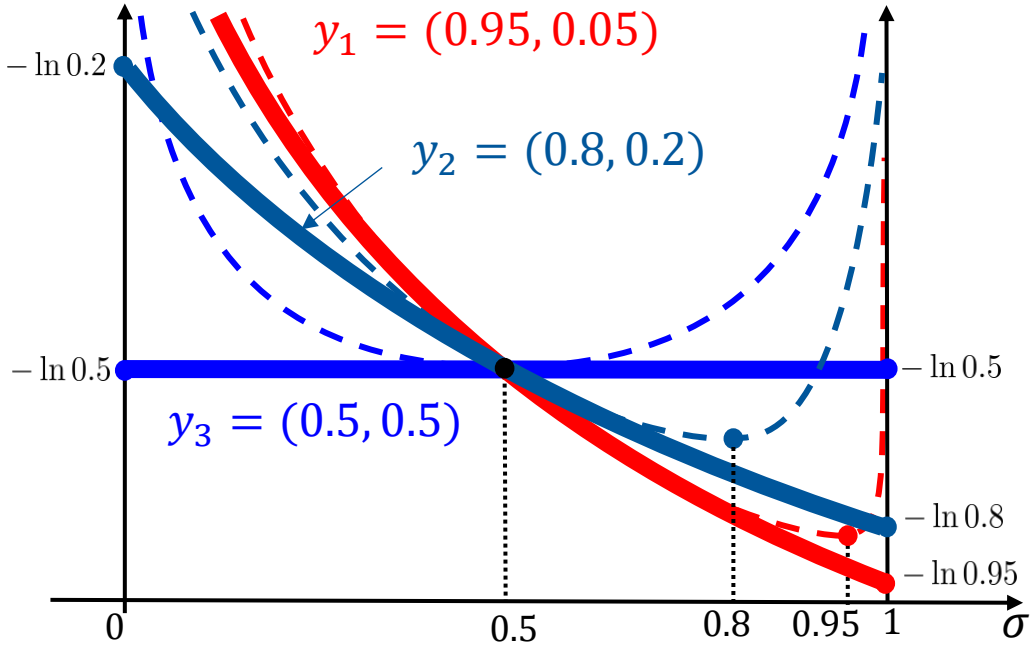
UNIVERSITY OF WATERLOO

target distribution

estimated distribution

$$-\ln \Pr(C = T) = -\ln \sum_k \sigma_k \cdot y_k$$

symmetric loss maximizing probability of equality/collision between two random variables



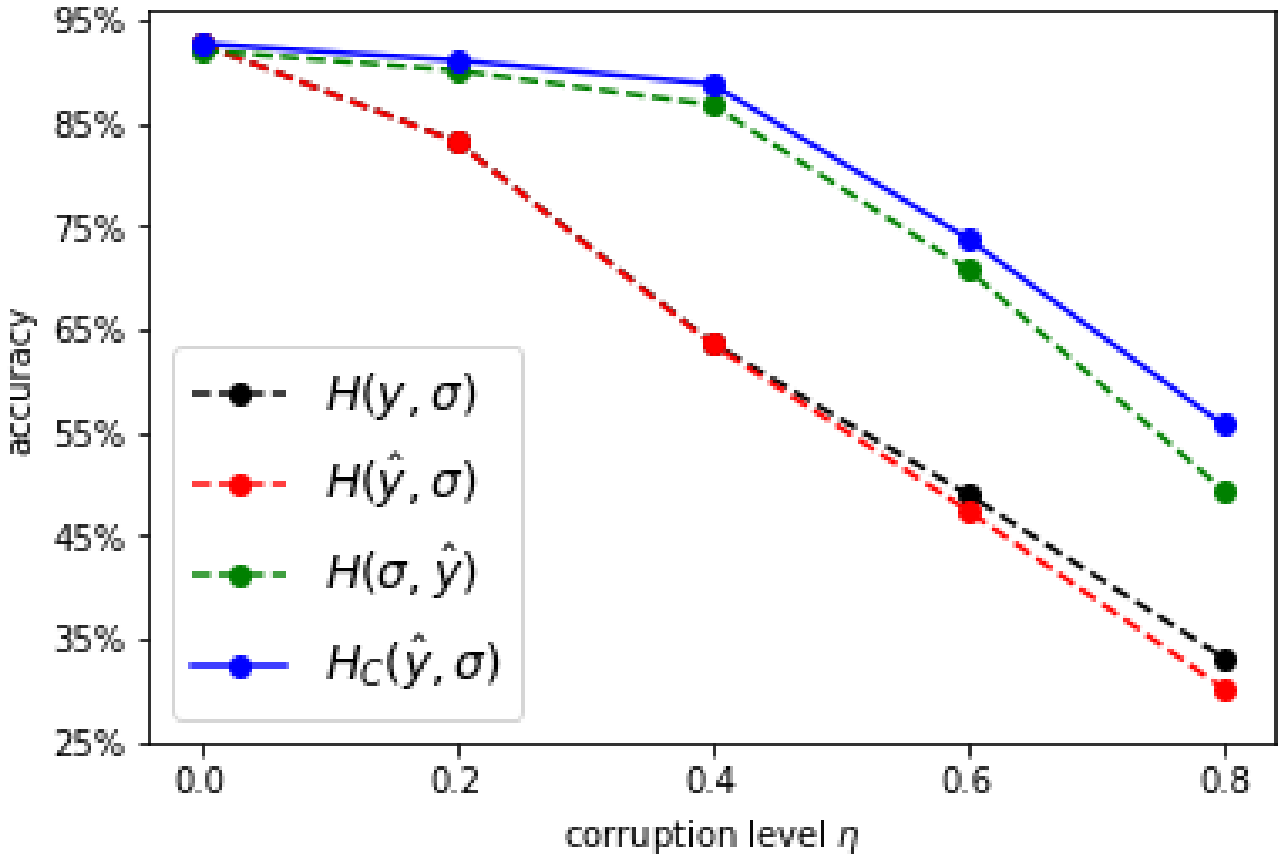
Robustness of predictions σ to uncertainty of labels y

labels after corruption

y : hard

$\hat{y}$: soft

$$\hat{y} = (1-\eta)\cdot y + \eta \cdot u$$



Optimization

[Bridle, Heading, MacKay, NeurIPS 1991]

$$L(\sigma) = \overline{H(\sigma)} - H(\overline{\sigma})$$

← **Gradient Descent**

"self-labeling" surrogate with *pseudo-labels* y

$y \approx \sigma$

our algorithm

$$L(\sigma, y) = \overline{H_C(\sigma, y)} - H(\overline{y}) \quad s.t. \;\; y \in \Delta^K$$

- **soft pseudo-labels**

convex w.r.t. y convex w.r.t. y

Collision CE

- convex w.r.t. y ← **EM**

- standard network training with collision CE ← **Gradient Descent**

Experiments (joint clustering with feature learning)

pretrained features - self-supervised representation learning
(SimSLR contrastive learning, Hinton 2020)

(resnet18)

	STL10	CIFAR10	CIFAR100-20
SCAN [44]	75.5% (2.0)	81.8% (0.3)	42.2% (3.0)
IMSAT [14]	70.23% (2.0)	77.64% (1.3)	43.68% (0.4)
MIADM [15]	67.84% (0.2)	74.76% (0.3)	43.47% (0.5)
Our	**78.12% (0.1)**	**83.27% (0.2)**	**47.01% (0.2)**

trained from the scratch

(vgg4)

	STL10	CIFAR10	CIFAR100-20	MNIST
IMSAT [14]	25.28%(0.5)	21.4%(0.5)	14.39%(0.7)	92.90%(6.3)
IIC [16]	24.12%(1.7)	21.3%(1.4)	12.58%(0.6)	82.51%(2.3)
SeLa [1]	23.99%(0.9)	24.16%(1.5)	**15.34%(0.3)**	52.86%(1.9)
MIADM [15]	23.37%(0.9)	23.26%(0.6)	14.02%(0.5)	78.88%(3.3)
Our	**25.98%(1.1)**	**24.26%(0.8)**	15.14%(0.5)	**95.11%(4.3)**