# Learning to benchmark

**Alfred Hero**

**University of Michigan - Ann Arbor**
Dept of Electrical Engineering and Computer Science (EECS)
Dept of Biomedical Engineering (BME)
Dept of Statistics
Program in Applied and Interdisciplinary Mathematics
Program in Applied Physics
Program in Computational Medicine and Bioinformatics

July 14, 2021

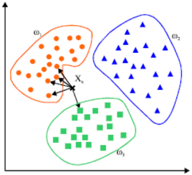
BayesErrorEstimator.jpynb

# Acknowledgements

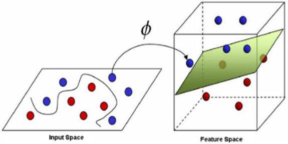
Students and former students who helped develop this framework

- Kristjan Greenewald - IBM AI
- Kevin Moon - Utah State University
- Morteza Noshad - Stanford University
- Brandon Oselio - University of Michigan
- Kumar Sricharan - Intuit, Inc
- Salimeh Sekeh - University of Maine
- Dennis Wei - IBM AI
- Easton Xu - Chinese Academy of Science

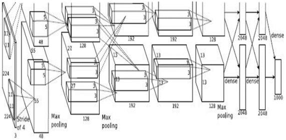
# Benchmarks in Machine Learning



kNN classifier

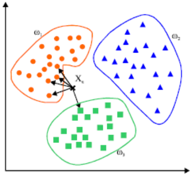



SVM classifier

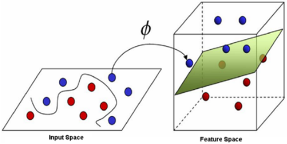



CNN classifier

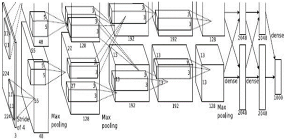
# Benchmarks in Machine Learning



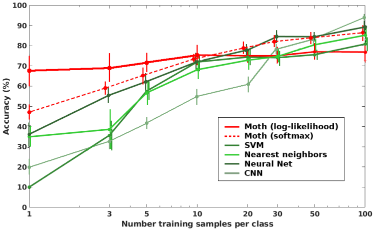
kNN classifier




SVM classifier

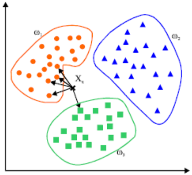


CNN classifier

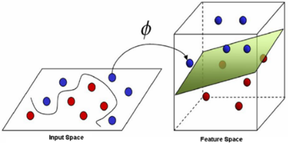



Accuracy for MNIST (Delahunt *et al* 2019)

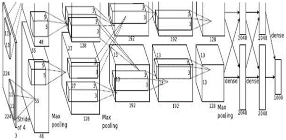
# Benchmarks in Machine Learning



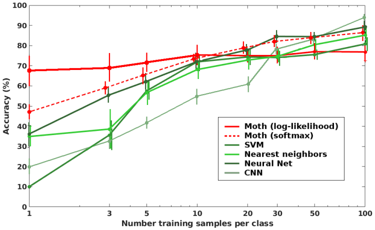

kNN classifier



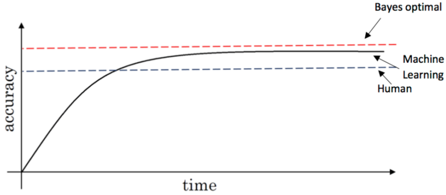

SVM classifier




CNN classifier




Accuracy for MNIST (Delahunt *et al* 2019)




Bayes-optimal benchmark
(Andrew Ng's Blog, Dec. 22, 2018)

# Early results relevant to benchmark learning

- k-NN useful for learning upper and lower bounds on Bayes probability of error

### Theorem (Cover and Hart (1967))

*Let $\hat{\epsilon}_n^{kNN}$ be the empirical error rate of the k-NN binary classifier applied to a training set $\{X_i, Y_i\}_{i=1}^n$ drawn i.i.d. from distribution $f(X, Y)$. Then, as $n \to \infty$,*

$$\frac{1}{2}\left(1 - \sqrt{1 - 2\hat{\epsilon}_n^{kNN}}\right) \leq \epsilon^* \leq \hat{\epsilon}_n^{kNN}, \qquad (a.s)$$

*where $\epsilon^*$ is Bayes error probability.*

# Early results relevant to benchmark learning

- k-NN useful for learning upper and lower bounds on Bayes probability of error

**Theorem (Cover and Hart (1967))**

*Let $\hat{\epsilon}_n^{kNN}$ be the empirical error rate of the k-NN binary classifier applied to a training set $\{X_i, Y_i\}_{i=1}^n$ drawn i.i.d. from distribution $f(X, Y)$. Then, as $n \to \infty$,*

$$\frac{1}{2}\left(1 - \sqrt{1 - 2\hat{\epsilon}_n^{kNN}}\right) \leq \epsilon^* \leq \hat{\epsilon}_n^{kNN}, \qquad (a.s)$$

*where $\epsilon^*$ is Bayes error probability.*

- If family $\mathcal{F}$ of distributions completely unconstrained, there will exist $f(X, Y) \in \mathcal{F}$ for which Bayes probability of error is not learnable.

**Theorem (Thm. 8.5 Devroye, Györfi, Lugosi (1996) )**

*For every n, for any estimate $\hat{\epsilon}_n$ of the Bayes error probability $\epsilon^*$ and for every $\delta > 0$, there exists a distribution of $(X, Y)$ such that*

$$\mathbb{E}\left\{|\hat{\epsilon}_n - \epsilon^*|\right\} \geq \frac{1}{4} - \delta.$$

# Learning to bound Bayes error: MST

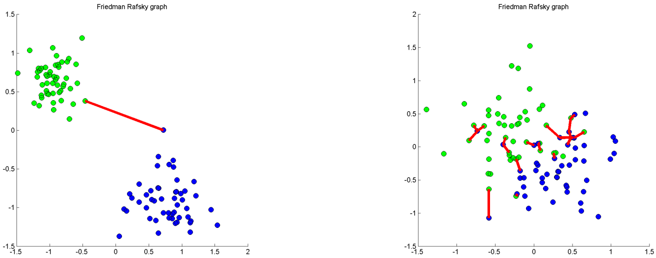



**Figure:** Friedman-Rafsky statistic converges to bound on Bayes classification error.

Friedman-Rafky (FR) statistic[1] $=$ #dichotomous edges btwn $M + N$ features

[1] J. Friedman and L. Rafsky (1979), Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. The Annals of Statistics.

[2] N. Henze and M. D. Penrose (1999). On the multivariate runs test. Annals of Statsistics.

[3] V. Berisha, A. Wisler, A.O. Hero, and A. Spanias (2016), "Empirically Estimable Classification Bounds Based on a Nonparametric Divergence Measure" IEEE Transactions on Signal Processing.

# Learning to bound Bayes error: MST

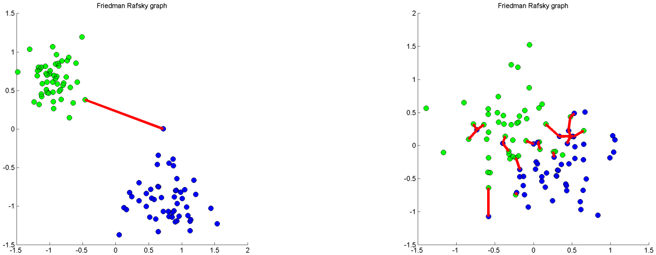



**Figure:** Friedman-Rafsky statistic converges to bound on Bayes classification error.

Friedman-Rafky (FR) statistic[1] $= \#$dichotomous edges btwn $M + N$ features

- If class distributions are continuous, $\mathrm{FR}/(M+N)$ approximates an information divergence measure[2].

---

[1] J. Friedman and L. Rafsky (1979), Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. The Annals of Statistics.

[2] N. Henze and M. D. Penrose (1999). On the multivariate runs test. Annals of Statsistics.

[3] V. Berisha, A. Wisler, A.O. Hero, and A. Spanias (2016), “Empirically Estimable Classification Bounds Based on a Nonparametric Divergence Measure” IEEE Transactions on Signal Processing.

# Learning to bound Bayes error: MST

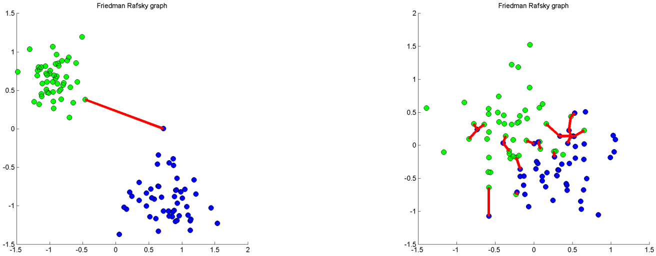



**Figure:** Friedman-Rafsky statistic converges to bound on Bayes classification error.

Friedman-Rafky (FR) statistic[1] = #dichotomous edges btwn $M + N$ features

- If class distributions are continuous, $\mathrm{FR}/(M + N)$ approximates an information divergence measure[2].
- This measure specifies upper and lower bounds on Bayes error[3]

[1] J. Friedman and L. Rafsky (1979), Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. The Annals of Statistics.

[2] N. Henze and M. D. Penrose (1999). On the multivariate runs test. Annals of Statsistics.

[3] V. Berisha, A. Wisler, A.O. Hero, and A. Spanias (2016), "Empirically Estimable Classification Bounds Based on a Nonparametric Divergence Measure" IEEE Transactions on Signal Processing.

# Benchmarking performance of Bayes classifier

Consider classification problem

- $Y \in \{0, 1\}$ an unknown label with priors $\{q, p\}$, $p + q = 1$.

$$P(Y = k) = p^k q^{1-k}, \quad k = 0, 1$$

- $X$ an observed random variable with conditional distribution

$$f(x|Y = k) = [f_1(x)]^k [f_0(x)]^{1-k}, \quad k = 0, 1$$

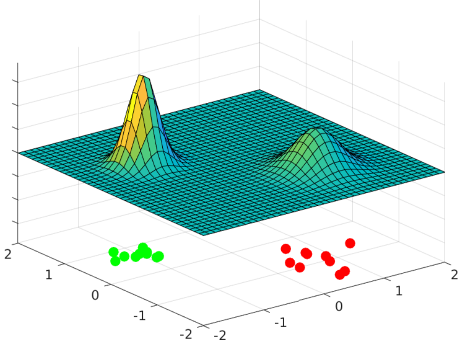



**Figure:** Density and realizations over two dimensional feature space.

# Bayes error rate: best achievable misclassification error probability

Bayes error rate $\epsilon_p$ is avg missclassification error probability of Bayes classifier

$$\epsilon_p(f_0, f_1) = P(C(X) \neq Y), \qquad C(x) = \mathrm{argmax}_{k \in \{0,1\}}\{P(Y = k|X = x)\}$$

[1] Sec. 2.4, Devroye, Görfi, Lugosi, A Probabilistic Theory of Pattern Recognition 1996

# Bayes error rate: best achievable misclassification error probability

Bayes error rate $\epsilon_p$ is avg missclassification error probability of Bayes classifier

$$\epsilon_p(f_0, f_1) = P(C(X) \neq Y), \qquad C(x) = \mathrm{argmax}_{k\in\{0,1\}}\{P(Y=k|X=x)\}$$

Bayes error has integral representation[1]

$$\epsilon_p(f_0, f_1) = \frac{1}{2} - \frac{1}{2}\int |qf_0(x) - pf_1(x)|dx,$$

[1] Sec. 2.4, Devroye, Görfi, Lugosi, A Probabilistic Theory of Pattern Recognition 1996

# Bayes error rate: best achievable misclassification error probability

Bayes error rate $\epsilon_p$ is avg missclassification error probability of Bayes classifier

$$\epsilon_p(f_0, f_1) = P(C(X) \neq Y), \qquad C(x) = \mathrm{argmax}_{k\in\{0,1\}}\{P(Y=k|X=x)\}$$

Bayes error has integral representation[1]

$$\epsilon_p(f_0, f_1) = \frac{1}{2} - \frac{1}{2}\int |qf_0(x) - pf_1(x)|dx,$$

Alternative representation as an $f$-divergence btwn distributions

$$\epsilon_p(f_0, f_1) = \frac{1+|p-q|}{2} - \frac{1}{2}\int g(f_1(x)/f_0(x))f_0(x)dx,$$

where $g(u)$ is the convex non-smooth function

$$g(u) = |pu - q| - |p - q|.$$

[1] Sec. 2.4, Devroye, Görfi, Lugosi, A Probabilistic Theory of Pattern Recognition 1996

# The $f$-divergence between a pair of distributions

The f-divergence (Csiszár)[1], (Ali-Silvey)[2]:

$$D_g(f_1 \| f_0) = \int g\left(\frac{f_1(x)}{f_0(x)}\right) f_0(x) dx$$

where $g(u)$ is a convex function on $\mathbb{R}^+$ and $g(1) = 0$.

[1] I. Csiszár (1963), Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizitat von Markoffschen Ketten. Magyar. Tud. Akad. Mat. Kutato Int. Kozl. 8:85–108.

[2] S. M. Ali and S. D. Silvey (1966), A general class of coefficients of divergence of one distribution from another, J. Royal Stat. Soc., Ser.B , 28:131-142.

# The $f$-divergence between a pair of distributions

The f-divergence (Csiszár)[1], (Ali-Silvey)[2]:

$$D_g(f_1 \| f_0) = \int g\left(\frac{f_1(x)}{f_0(x)}\right) f_0(x)dx$$

where $g(u)$ is a convex function on $\mathbb{R}^+$ and $g(1) = 0$.

Properties of $f$-divergence: if $g$ is strictly convex then $D_g(f_1 \| f_0)$ is

- non-negative reflexive : $D_g(f_1 \| f_0) \geq 0$ with equality iff $f_1 = f_0$
- monotone: $D_g(f_1 \| f_0)$ non-increasing under transformations $x \to T(x)$
- jointly convex: $D_g(f_1 \| f_0)$ is convex in $(f_0, f_1)$

[1] I. Csiszár (1963), Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizitat von Markoffschen Ketten. Magyar. Tud. Akad. Mat. Kutato Int. Kozl. 8:85–108.

[2] S. M. Ali and S. D. Silvey (1966), A general class of coefficients of divergence of one distribution from another, J. Royal Stat. Soc., Ser.B , 28:131-142.

# The $f$-divergence between a pair of distributions

The f-divergence (Csiszár)[1], (Ali-Silvey)[2]:

$$D_g(f_1 \| f_0) = \int g\left(\frac{f_1(x)}{f_0(x)}\right) f_0(x) dx$$

where $g(u)$ is a convex function on $\mathbb{R}^+$ and $g(1) = 0$.

Properties of $f$-divergence: if $g$ is strictly convex then $D_g(f_1 \| f_0)$ is

- non-negative reflexive : $D_g(f_1 \| f_0) \geq 0$ with equality iff $f_1 = f_0$
- monotone: $D_g(f_1 \| f_0)$ non-increasing under transformations $x \to T(x)$
- jointly convex: $D_g(f_1 \| f_0)$ is convex in $(f_0, f_1)$

Examples: $g(u) = u\log(u)$ (KL); $g(u) = (1 - u^\alpha)\frac{1}{1-\alpha}$ (Rényi-$\alpha$).

[1] I. Csiszár (1963), Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizitat von Markoffschen Ketten. Magyar. Tud. Akad. Mat. Kutato Int. Kozl. 8:85–108.

[2] S. M. Ali and S. D. Silvey (1966), A general class of coefficients of divergence of one distribution from another, J. Royal Stat. Soc., Ser.B , 28:131-142.

# Instances of $f$-divergences[1]

- Total variation distance $g(u) = \frac{1}{2}|u-1|$

$$D^{TV}(f_1\|f_0) = \frac{1}{2}\int |f_1(x) - f_0(x)|dx$$

[1] Csiszár, I., and Shields, P. C. (2004). Information theory and statistics: A tutorial. Foundations and Trends in Communications and Information Theory, 1(4), 417-528.

# Instances of $f$-divergences[1]

- Total variation distance $g(u) = \frac{1}{2}|u-1|$

$$D^{TV}(f_1 \| f_0) = \frac{1}{2}\int |f_1(x) - f_0(x)| dx$$

- $\alpha$-divergence: $g(u) = (1-u^{\alpha})\frac{1}{1-\alpha}$

$$D^{R}(f_1 \| f_0) = \left(1 - \int f_1^{\alpha}(x) f_0^{1-\alpha}(x) dx\right) \frac{1}{1-\alpha}$$

[1] Csiszár, I., and Shields, P. C. (2004). Information theory and statistics: A tutorial. Foundations and Trends in Communications and Information Theory, 1(4), 417-528.

# Instances of $f$-divergences[1]

- Total variation distance $g(u) = \frac{1}{2}|u-1|$

$$D^{TV}(f_1\|f_0) = \frac{1}{2}\int |f_1(x) - f_0(x)|dx$$

- $\alpha$-divergence: $g(u) = (1-u^{\alpha})\frac{1}{1-\alpha}$

$$D^{R}(f_1\|f_0) = \left(1 - \int f_1^{\alpha}(x) f_0^{1-\alpha}(x)dx\right)\frac{1}{1-\alpha}$$

- Kullback-Liebler divergence: $g(u) = u\log u$:

$$D^{KL}(f_1\|f_0) = \int f_1(x)\log\left(\frac{f_1(x)}{f_0(x)}\right)dx$$

[1] Csiszár, I., and Shields, P. C. (2004). Information theory and statistics: A tutorial. Foundations and Trends in Communications and Information Theory, 1(4), 417-528.

# Instances of $f$-divergences[1]

- Total variation distance $g(u) = \frac{1}{2}|u-1|$

$$D^{TV}(f_1 \| f_0) = \frac{1}{2}\int |f_1(x) - f_0(x)| dx$$

- $\alpha$-divergence: $g(u) = (1 - u^{\alpha})\frac{1}{1-\alpha}$

$$D^{R}(f_1 \| f_0) = \left(1 - \int f_1^{\alpha}(x) f_0^{1-\alpha}(x) dx\right) \frac{1}{1-\alpha}$$

- Kullback-Liebler divergence: $g(u) = u \log u$:

$$D^{KL}(f_1 \| f_0) = \int f_1(x) \log\left(\frac{f_1(x)}{f_0(x)}\right) dx$$

- Hellinger-Bhattacharyya divergence $g(u) = (\sqrt{u} - 1)^2$

$$D^{H}(f_1 \| f_0) = \int \left(\sqrt{f_1(x)} - \sqrt{f_0(x)}\right)^2 dx$$

[1] Csiszár, I., and Shields, P. C. (2004). Information theory and statistics: A tutorial. Foundations and Trends in Communications and Information Theory, 1(4), 417-528.

# Other instances of $f$-divergences

- Generalized total variation distance[1]: $g(u) = |pu - q|/2 - |p - q|/2$

$$D_p^{GTV} = \frac{1}{2}\int |pf_1(x) - qf_0(x)|dx + |p - q|/2$$

- Henze-Penrose divergence[2]: $g(u) = \frac{1}{4pq}\left[\frac{(pt-q)^2}{pt+q} - (p-q)^2\right]$

$$D_p^{HP} = \frac{1}{4pq}\left[\int \frac{(pf_1(x) - qf_0(x))^2}{pf_1(x) + qf_0(x)} dx - (p-q)^2\right].$$

[1] T. Kailath (1967), The divergence and Bhattacharyya distance measures in signal selection, IEEE T. Communication Technology, 15:1:52–60

[2] N. Henze and M. D. Penrose (1999). On the multivariate runs test. Annals of Stats, 290-298.

# $f$-divergences and Bayes error rate

These divergences can each be related to minimum probability of error

- Exact $f$-divergence representation

$$\epsilon_p(f_1, f_0) = \frac{1 + |p - q|}{2} - D_p^{GTV}(f_1(x) \| f_0(x))$$

[1] T. Kailath (1967), The divergence and Bhattacharyya distance measures in signal selection, IEEE T. Communication Technology, 15:1:52–60

# $f$-divergences and Bayes error rate

These divergences can each be related to minimum probability of error

- Exact $f$-divergence representation

$$\epsilon_p(f_1, f_0) = \frac{1 + |p - q|}{2} - D_p^{GTV}(f_1(x) \| f_0(x))$$

- Bhattacharyya bound[1]

$$\tfrac{1}{2} - \tfrac{1}{2}\sqrt{1 - (BC_p)^2} \leq \epsilon_p \leq \tfrac{1}{2} BC_p,$$

where $BC_p = \frac{\sqrt{pq}}{2}(1 - D_p^H)$ is the Bhattacharyya coefficient BC.

[1] T. Kailath (1967), The divergence and Bhattacharyya distance measures in signal selection, IEEE T. Communication Technology, 15:1:52–60

# $f$-divergences and Bayes error rate

These divergences can each be related to minimum probability of error

- Exact $f$-divergence representation

$$\epsilon_p(f_1, f_0) = \frac{1+|p-q|}{2} - D_p^{GTV}(f_1(x)\|f_0(x))$$

- Bhattacharyya bound[1]

$$\tfrac{1}{2} - \tfrac{1}{2}\sqrt{1-(BC_p)^2} \leq \epsilon_p \leq \tfrac{1}{2}BC_p,$$

where $BC_p = \frac{\sqrt{pq}}{2}(1 - D_p^H)$ is the Bhattacharyya coefficient BC.

- *Learning to benchmark* can be reduced to $f$-divergence estimation.

[1] T. Kailath (1967), The divergence and Bhattacharyya distance measures in signal selection, IEEE T. Communication Technology, 15:1:52–60

# Extension: learning mutual information

The Mutual Information is also an $f$-divergence

$$MI(X_1; X_2) = \int g\left(\frac{f(X_1, X_2)}{f(X_1)f(X_2)}\right) f(X_1)f(X_2)$$

where Shannon MI is obtained for the case that

$$g(u) = ulog(u)$$

Such divergences can be learned from training data[1] $\{(X_1(k), X_2(k))\}_{k=1}^n$

[1] K. Moon, K. Sricharan, A. Hero, "Ensemble Estimation of Generalized Mutual Information with Applications to Genomics," IEEE Transactions on Information Theory, to appear 2021.

# Extension: multiclass classifier benchmarking

Bayes error for Multiclass classification has representation ($K$ classes)

$$\epsilon_p(f_1, \ldots, f_K) = 1 - p_1 - \sum_{k=2}^{K} \int g_k\left(\frac{f_1(x)}{f_k(x)}, \ldots, \frac{f_{k-1}(x)}{f_k(x)}\right) f_k(x)dx$$

where

$$g_k(u_1, \ldots, u_{k-1}) = \max\left(0, p_k - \max_{1 \le i \le k-1} \{p_i u_i\}\right)$$

This representation can be used for learning Bayes error[1]

Simpler multiclass divergences can also be learned to bound Bayes error[2]

[1] M. Noshad, L. Xu, and A. Hero, "Learning to Benchmark: Estimating Best Achievable Misclassification Error from Training Data," arXiv:1909.07192, Sept. 2019.

[2] S. Sekeh, B. Oselio and A. Hero, "Learning to Bound the Multi-class Bayes Error," IEEE Trans. on Signal Processing, vol. 68, pp. 3793 – 3807, May 2020.

# Learning $f$-Divergence

- **Goal**: Accurate and computationally fast estimation of $f$-divergence
- **Assumption**: Strictly bounded and continuous class distributions $f_1$, $f_0$.
- **Density plug-in estimator of $f$-divergence:**

$$\widehat{D_g}(f_1 \| f_0) = \int g\left(\frac{\widehat{f_1}(x)}{\widehat{f_0}(x)}\right)\widehat{f_0}(x)dx$$

  where
  - $\widehat{f_0}, \widehat{f_1}$ are density estimates, e.g., with kernel bandwidth parameter $\epsilon$
  - Gabor kernel, histogram, k-NN kernel[1] (Devroye 2012)

- **Root mean squared error (RMSE)** decreases slowly in $n$=#samples

$$\mathrm{RMSE} = \sqrt{\mathrm{Bias}^2 + \mathrm{Variance}} = cn^{-1/2d}$$

[1] L. Devroye, G. Lugosi, "Combinatorial methods in density estimation," Springer 2012.

# Learning $f$-Divergence

- **Goal**: Accurate and computationally fast estimation of $f$-divergence
- **Assumption**: Strictly bounded and continuous class distributions $f_1$, $f_0$.
- **Density plug-in estimator of $f$-divergence:**

$$\widehat{D_g}(f_1 \| f_0) = \int g\left(\frac{\widehat{f_1}(x)}{\widehat{f_0}(x)}\right)\widehat{f_0}(x)dx$$

  where
  - $\widehat{f_0}, \widehat{f_1}$ are density estimates, e.g., with kernel bandwidth parameter $\epsilon$
  - Gabor kernel, histogram, k-NN kernel[1] (Devroye 2012)

- **Root mean squared error (RMSE)** decreases slowly in $n$=#samples

$$\mathrm{RMSE} = \sqrt{\mathrm{Bias}^2 + \mathrm{Variance}} = cn^{-1/2d}$$

$\Rightarrow$ Compare to optimal *parametric* RMSE rate:

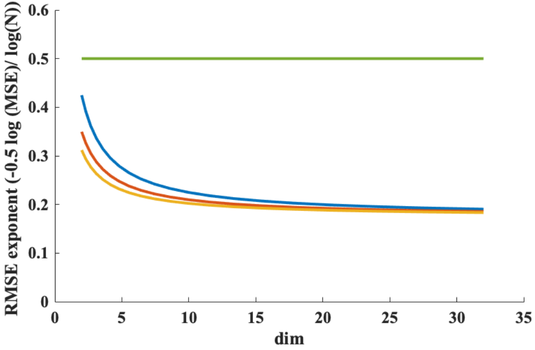
$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = cn^{-1/2}$$

[1] L. Devroye, G. Lugosi, "Combinatorial methods in density estimation," Springer 2012.

# Learning to benchmark via ensemble learning: preview

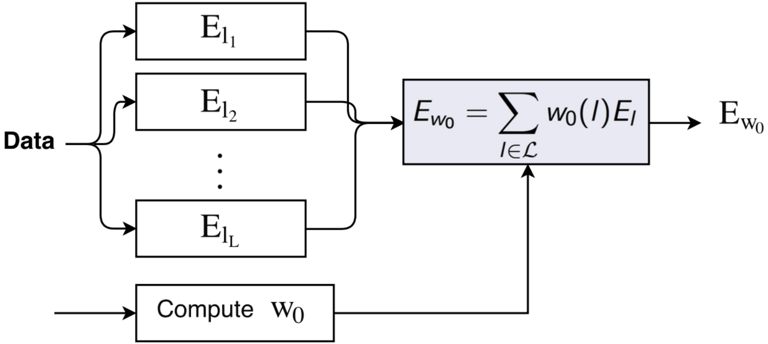
- We combine an ensemble of base plug-in estimators of $f$-divergence
- The ensemble weights are derived under a smoothness assumption:

  *The class densities $f_0$, $f_1$ are d-times continuously differentiable.*
- Resulting ensemble estimator achieves *parametric* rates of convergence




- K.R. Moon, K. Sricharan, K. Greenewald, and A.O. Hero, "Ensemble Estimation of Information Divergence," Entropy 2018
- M. Noshad, L. Xu and A. Hero, "Learning to Benchmark: Estimating Best Achievable Misclassification Error from Training Data," arXiv:1909.07192 Sept. 2019

# Ensemble learners




- $\{E_{l_i}\}_{i=1}^{L}$ ensemble of base estimators (weak learners)
- $\mathrm{w}_0 = (w_0(l))_{l=1}^{L}$ a vector of boosting weights
- $E_{\mathrm{w}_0}$: combined base estimators (boosted learner)

## Ensemble learners

Most ensemble learning approaches use *data-dependent* weights:

- Boosting classifiers with Adaboost[1] and other objective functions.

[1] Y. Freund and R. E. Schapire (1996). Experiments with a new boosting algorithm. Intl Conf on Machine Learning. pp. 148-156.

[2] Bickel, P. J., Ritov, Y. A., and Zakai, A. (2006). Some theory for generalized boosting algorithms. J. of Machine Learning Research, 705-732.

[3] Moon, Sricharan, Greenewald, Hero. "Ensemble estimation of information divergence." Entropy 20, no. 8 (2018): 560.

## Ensemble learners

Most ensemble learning approaches use *data-dependent* weights:

- Boosting classifiers with Adaboost[1] and other objective functions.

Under some conditions such methods achieve Bayes optimal performance[2]

[1] Y. Freund and R. E. Schapire (1996). Experiments with a new boosting algorithm. Intl Conf on Machine Learning. pp. 148-156.

[2] Bickel, P. J., Ritov, Y. A., and Zakai, A. (2006). Some theory for generalized boosting algorithms. J. of Machine Learning Research, 705-732.

[3] Moon, Sricharan, Greenewald, Hero. "Ensemble estimation of information divergence." Entropy 20, no. 8 (2018): 560.

# Ensemble learners

Most ensemble learning approaches use *data-dependent* weights:

- Boosting classifiers with Adaboost[1] and other objective functions.

Under some conditions such methods achieve Bayes optimal performance[2]

**Alternative**: we solve an *offline* inverse problem for rate-optimal weights[3]

[1] Y. Freund and R. E. Schapire (1996). Experiments with a new boosting algorithm. Intl Conf on Machine Learning. pp. 148-156.

[2] Bickel, P. J., Ritov, Y. A., and Zakai, A. (2006). Some theory for generalized boosting algorithms. J. of Machine Learning Research, 705-732.

[3] Moon, Sricharan, Greenewald, Hero. "Ensemble estimation of information divergence." Entropy 20, no. 8 (2018): 560.

# Ensemble learners

Most ensemble learning approaches use *data-dependent* weights:

- Boosting classifiers with Adaboost[1] and other objective functions.

Under some conditions such methods achieve Bayes optimal performance[2]

**Alternative**: we solve an *offline* inverse problem for rate-optimal weights[3]

This can be applied to different base divergence estimators:

- Kernel density estimates (KDE)
- k-NN density estimates
- NN ratio estimates
- Locality sensitive hashing (LSH) density estimates

[1] Y. Freund and R. E. Schapire (1996). Experiments with a new boosting algorithm. Intl Conf on Machine Learning. pp. 148-156.

[2] Bickel, P. J., Ritov, Y. A., and Zakai, A. (2006). Some theory for generalized boosting algorithms. J. of Machine Learning Research, 705-732.

[3] Moon, Sricharan, Greenewald, Hero. "Ensemble estimation of information divergence." Entropy 20, no. 8 (2018): 560.

# Locality sensitive hashing (LSH) plug-in estimator

$$\widehat{D}_g(f_1 \| f_0) := \sum_{i:M_i>0} g\left(\frac{N_i/N}{M_i/M}\right) M_i/M$$

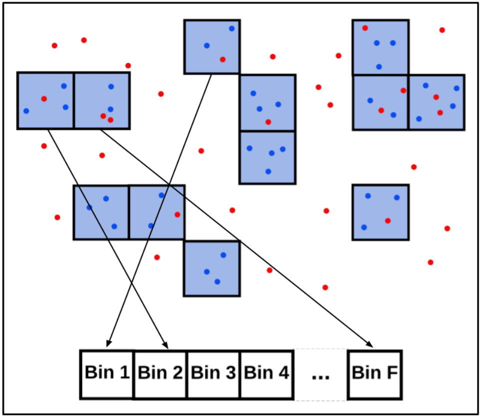



**Figure:** LSH quantizes X data with cell resolution $\epsilon$ and random displacement $b$

# Locality sensitive hashing plug-in estimator: bias and variance

**Theorem (Bias Expansion)**

*If $f_0$ and $f_1$ are $d$-times differentiable, the mean of $\widehat{D}_g$ has representation*

$$\mathbb{E}[\widehat{D}_g] = D(f_1 \| f_0) + \mathbb{B}(\widehat{D}_g)$$

$$\mathbb{B}(\widehat{D}_g) = \sum_{i=1}^{d} C_i \epsilon^i + O\left(\frac{1}{n\epsilon^d}\right).$$

# Locality sensitive hashing plug-in estimator: bias and variance

**Theorem (Bias Expansion)**

*If $f_0$ and $f_1$ are $d$-times differentiable, the mean of $\widehat{D}_g$ has representation*

$$\mathbb{E}[\widehat{D}_g] = D(f_1 \| f_0) + \mathbb{B}(\widehat{D}_g)$$

$$\mathbb{B}(\widehat{D}_g) = \sum_{i=1}^{d} C_i \epsilon^i + O\left(\frac{1}{n\epsilon^d}\right).$$

**Theorem (Variance)**

*The variance of the hash-based estimator decreases at least as fast as $1/n$*

$$\mathbb{V}(\widehat{D}_g) \leq O\left(\frac{1}{n}\right).$$

# Locality sensitive hashing plug-in estimator: bias and variance

**Theorem (Bias Expansion)**

*If $f_0$ and $f_1$ are $d$-times differentiable, the mean of $\widehat{D}_g$ has representation*

$$\mathbb{E}[\widehat{D}_g] = D(f_1 \| f_0) + \mathbb{B}(\widehat{D}_g)$$

$$\mathbb{B}(\widehat{D}_g) = \sum_{i=1}^{d} C_i \epsilon^i + O\left(\frac{1}{n\epsilon^d}\right).$$

**Theorem (Variance)**

*The variance of the hash-based estimator decreases at least as fast as $1/n$*

$$\mathbb{V}(\widehat{D}_g) \leq O\left(\frac{1}{n}\right).$$

$\Rightarrow$ Choosing $\epsilon = O\left(n^{-1/2d}\right)$ forces bias remainder to $O\left(\frac{1}{n\epsilon^d}\right) = O(1/\sqrt{n})$

# Locality sensitive hashing plug-in estimator: bias and variance

**Theorem (Bias Expansion)**

*If $f_0$ and $f_1$ are d-times differentiable, the mean of $\widehat{D}_g$ has representation*

$$\mathbb{E}[\widehat{D}_g] = D(f_1 \| f_0) + \mathbb{B}(\widehat{D}_g)$$

$$\mathbb{B}(\widehat{D}_g) = \sum_{i=1}^{d} C_i \epsilon^i + O\left(\frac{1}{n\epsilon^d}\right).$$

**Theorem (Variance)**

*The variance of the hash-based estimator decreases at least as fast as $1/n$*

$$\mathbb{V}(\widehat{D}_g) \leq O\left(\frac{1}{n}\right).$$

$\Rightarrow$ Choosing $\epsilon = O\left(n^{-1/2d}\right)$ forces bias remainder to $O\left(\frac{1}{n\epsilon^d}\right) = O(1/\sqrt{n})$

$\Rightarrow$ This makes the slowest term in the bias decay as $\mathbb{B}(\widehat{D}_g) = O(n^{-1/2d})$

# Ensemble learning to reduce bias solves an inverse problem

- Let $\{\widehat{D}_g^{\epsilon(t)}\}_{t\in\mathcal{L}}$ be a set of $L = |\mathcal{L}|$ base learners.
- $\epsilon(t) = tn^{-1/2d}$ is a set of bandwidth parameters.
- $\mathcal{L} := \{t_1, ..., t_L\}$ is a set of scale factors.

**Define**: Ensemble divergence estimator $L \geq d$: $\widehat{D}_{\mathrm{w}} := \sum_{j=1}^{L} w_j \widehat{D}_{\epsilon(t_j)} = \mathrm{w}^T \hat{\mathrm{D}}_\epsilon$

## Ensemble learning to reduce bias solves an inverse problem

- Let $\{\widehat{D}_g^{\epsilon(t)}\}_{t \in \mathcal{L}}$ be a set of $L = |\mathcal{L}|$ base learners.
- $\epsilon(t) = tn^{-1/2d}$ is a set of bandwidth parameters.
- $\mathcal{L} := \{t_1, ..., t_L\}$ is a set of scale factors.

**Define**: Ensemble divergence estimator $L \geq d$: $\widehat{D}_{\mathrm{w}} := \sum_{j=1}^{L} w_j \widehat{D}_{\epsilon(t_j)} = \mathrm{w}^T \hat{\mathrm{D}}_\epsilon$

Bias of ensemble divergence estimator:

$$\mathbb{B}\left[\widehat{D}_{\mathrm{w}}\right] = \sum_{i=1}^{d} C_i n^{-i/2d} \sum_{j=1}^{L} w_j t_j^i + O\left(\frac{1}{\sqrt{n}}\right)$$

# Ensemble learning to reduce bias solves an inverse problem

- Let $\{\widehat{D}_g^{\epsilon(t)}\}_{t\in\mathcal{L}}$ be a set of $L = |\mathcal{L}|$ base learners.
- $\epsilon(t) = tn^{-1/2d}$ is a set of bandwidth parameters.
- $\mathcal{L} := \{t_1, ..., t_L\}$ is a set of scale factors.

**Define**: Ensemble divergence estimator $L \geq d$: $\widehat{D}_{\mathrm{w}} := \sum_{j=1}^{L} w_j \widehat{D}_{\epsilon(t_j)} = \mathrm{w}^T \hat{\mathrm{D}}_\epsilon$

Bias of ensemble divergence estimator:

$$\mathbb{B}\left[\widehat{D}_{\mathrm{w}}\right] = \sum_{i=1}^{d} C_i n^{-i/2d} \sum_{j=1}^{L} w_j t_j^i + O\left(\frac{1}{\sqrt{n}}\right)$$

Bias reduced to $O\left(\frac{1}{\sqrt{n}}\right)$ if $\{w_j\}_{j=1}^{L}$ selected to solve linear system $\mathrm{Aw} = 0$:

$$\begin{bmatrix} t_1 & \dots & \dots & t_L \\ t_1^2 & \ddots & \ddots & t_L^2 \\ \vdots & \ddots & \ddots & \vdots \\ t_1^d & \dots & \dots & t_L^d \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ \vdots \\ w_L \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

# Ensemble learning to reduce bias solves an inverse problem

- Let $\{\widehat{D}_g^{\epsilon(t)}\}_{t\in\mathcal{L}}$ be a set of $L = |\mathcal{L}|$ base learners.
- $\epsilon(t) = tn^{-1/2d}$ is a set of bandwidth parameters.
- $\mathcal{L} := \{t_1, ..., t_L\}$ is a set of scale factors.

**Define**: Ensemble divergence estimator $L \geq d$: $\widehat{D}_{\mathrm{w}} := \sum_{j=1}^{L} w_j \widehat{D}_{\epsilon(t_j)} = \mathrm{w}^T \hat{\mathrm{D}}_\epsilon$

Bias of ensemble divergence estimator:

$$\mathbb{B}\left[\widehat{D}_{\mathrm{w}}\right] = \sum_{i=1}^{d} C_i n^{-i/2d} \sum_{j=1}^{L} w_j t_j^i + O\left(\frac{1}{\sqrt{n}}\right)$$

Bias reduced to $O\left(\frac{1}{\sqrt{n}}\right)$ if $\{w_j\}_{j=1}^{L}$ selected to solve linear system $\mathrm{A}\mathrm{w} = 0$:

$$\begin{bmatrix} t_1 & \dots & \dots & t_L \\ t_1^2 & \ddots & \ddots & t_L^2 \\ \vdots & \ddots & \ddots & \vdots \\ t_1^d & \dots & \dots & t_L^d \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ \vdots \\ w_L \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

$\Rightarrow$ For large $d$, Chebychev methods used to stabilize solution (Noshad '19)

## Controlling ensemble estimator variance

Variance of ensemble divergence estimator is quadratic in w

$$\mathbb{V}(\widehat{D}_{\mathrm{w}}) = \mathbb{V}(\mathrm{w}^T \hat{\mathrm{D}}_\epsilon) = \mathrm{w}^T \mathrm{cov}(\hat{\mathrm{D}}_\epsilon)\mathrm{w} \le \|\mathrm{w}\|^2 \lambda_{max}.$$

# Controlling ensemble estimator variance

Variance of ensemble divergence estimator is quadratic in w

$$\mathbb{V}(\widehat{D}_{\mathrm{w}}) = \mathbb{V}(\mathrm{w}^T \hat{\mathrm{D}}_\epsilon) = \mathrm{w}^T \mathrm{cov}(\hat{\mathrm{D}}_\epsilon)\mathrm{w} \le \|\mathrm{w}\|^2 \lambda_{max}.$$

$\Rightarrow$ Select w as solution to linearly constrained quadratic program

$$\begin{aligned} \min_{\mathrm{w}} \quad & \|\mathrm{w}\|_2, & \textbf{[OPT1]} \\ \textit{subject to} \quad & \sum_{j=1}^{L} w_j = 1, \\ & \sum_{j=1}^{L} w_j t_j^i = 0, \quad i \in [d] \end{aligned}$$

## Controlling ensemble estimator variance

Variance of ensemble divergence estimator is quadratic in w

$$\mathbb{V}(\widehat{D}_{\mathrm{w}}) = \mathbb{V}(\mathrm{w}^T \hat{\mathrm{D}}_\epsilon) = \mathrm{w}^T \mathrm{cov}(\hat{\mathrm{D}}_\epsilon)\mathrm{w} \leq \|\mathrm{w}\|^2 \lambda_{max}.$$

$\Rightarrow$ Select w as solution to linearly constrained quadratic program

$$\begin{aligned} \min_{\mathrm{w}} \quad & \|\mathrm{w}\|_2, \qquad \textbf{[OPT1]} \\ \textit{subject to} \quad & \sum_{j=1}^{L} w_j = 1, \\ & \sum_{j=1}^{L} w_j t_j^i = 0, \quad i \in [d] \end{aligned}$$

- If $L > d$, the solution $\mathrm{w}^*$ to **[OPT1]** ensures MSE of $O(1/n)$.
- Weights are computed offline, not dependent on data or data's distribution
- For large $d$, $\{t_j\}$ can be selected as Chebyshev nodes

# Solution of [**OPT1**]

In matrix form the constraints in [**OPT1**] are $\mathrm{A}\mathrm{w} = \mathrm{b}$ and min-norm solution is

$$\mathrm{w}^* = (\mathrm{A}^T\mathrm{A})^\dagger \mathrm{A}^T \mathrm{b}$$

# Solution of [**OPT1**]

In matrix form the constraints in [**OPT1**] are $\mathrm{A}\mathrm{w} = \mathrm{b}$ and min-norm solution is

$$\mathrm{w}^* = (\mathrm{A}^T\mathrm{A})^\dagger \mathrm{A}^T \mathrm{b}$$

Q. How to select $t_i$'s in order to simplify the solution $\mathrm{w}^*$?

# Solution of [**OPT1**]

In matrix form the constraints in [**OPT1**] are $\mathrm{A}\mathrm{w} = \mathrm{b}$ and min-norm solution is

$$\mathrm{w}^* = (\mathrm{A}^T\mathrm{A})^{\dagger}\mathrm{A}^T\mathrm{b}$$

Q. How to select $t_i$'s in order to simplify the solution $\mathrm{w}^*$?
A. Cast [**OPT1**] as a min-norm polynomial approximation problem

# Solution of [**OPT1**]

In matrix form the constraints in [**OPT1**] are $\mathrm{A}\mathrm{w} = \mathrm{b}$ and min-norm solution is

$$\mathrm{w}^* = (\mathrm{A}^T\mathrm{A})^\dagger \mathrm{A}^T \mathrm{b}$$

Q. How to select $t_i$'s in order to simplify the solution $\mathrm{w}^*$?
A. Cast [**OPT1**] as a min-norm polynomial approximation problem

$$\begin{aligned} &\min_{\mathrm{w}} && \|\mathrm{w}\|_2, && \textbf{[OPT1]} \\ &\textit{subject to} && \sum_{j=1}^{L} w_j p_i(t_j) = p_i(0), \quad i \in [d] && \end{aligned}$$

where, for $\alpha > \max\{t_i\}$, $p_i : [0, \alpha] \to \mathbb{R}$ are degree $d$ polynomials with coefficients $\boldsymbol{\beta}_i = [\beta_{i,d}, \ldots, \beta_{i,0}]$:

$$p_i(t) = \beta_{i,d} t^d + \ldots + \beta_{i,1} t + \beta_{i,0}, \qquad i = 1, \ldots, d+1$$

# Solution of [OPT1]

Shifted Chebyshev polynomials (SCP) $T_n^\alpha : [0, \alpha] \to \mathbb{R}$,

$$T_n^\alpha(t) = T_n(2t/\alpha - 1), \quad n = 0, 1, \dots,$$

where $T_n : [-1, 1] \to \mathbb{R}$ is a Chebyshev polynomial of the first kind of degree $n$.

# Solution of [**OPT1**]

Shifted Chebyshev polynomials (SCP) $T_n^\alpha : [0, \alpha] \to \mathbb{R}$,

$$T_n^\alpha(t) = T_n(2t/\alpha - 1), \quad n = 0, 1, \dots,$$

where $T_n : [-1, 1] \to \mathbb{R}$ is a Chebyshev polynomial of the first kind of degree $n$.

- The roots $\{s_i\}_{i=1}^n$ of $T_n^\alpha$ have the form

$$s_k = \frac{\alpha}{2} \cos\left(\left(k + \frac{1}{2}\right)\frac{\pi}{L}\right)$$

- If $\{s_i\}_{i=1}^n$ are roots of $T_n^\alpha$, a discrete orthogonality property holds

$$\sum_{i=0}^{n-1} T_l^\alpha(s_i) T_m^\alpha(s_i) = 0, \qquad l \neq m, \;\; l, m < n$$

# Solution of [**OPT1**]

Shifted Chebyshev polynomials (SCP) $T_n^\alpha : [0, \alpha] \to \mathbb{R}$,

$$T_n^\alpha(t) = T_n(2t/\alpha - 1), \quad n = 0, 1, \dots,$$

where $T_n : [-1, 1] \to \mathbb{R}$ is a Chebyshev polynomial of the first kind of degree $n$.

- The roots $\{s_i\}_{i=1}^n$ of $T_n^\alpha$ have the form

$$s_k = \frac{\alpha}{2} \cos\left(\left(k + \frac{1}{2}\right)\frac{\pi}{L}\right)$$

- If $\{s_i\}_{i=1}^n$ are roots of $T_n^\alpha$, a discrete orthogonality property holds

$$\sum_{i=0}^{n-1} T_l^\alpha(s_i) T_m^\alpha(s_i) = 0, \qquad l \neq m, \ \ l, m < n$$

**Theorem (Chebyshev solution)**

*When the parameters $\{t_i\}_{i=1}^L$ are selected as the roots $\{s_i\}_{i=1}^L$, of $T_L^\alpha(t)$, then the solution of [**OPT1**] is*

$$w_i^* = \frac{2}{L} \sum_{k=0}^{d} T_k^\alpha(0) T_k^\alpha(s_i) - \frac{1}{L}, \qquad i = 1, \dots, L,$$

# Chebyshev stabilization of ensemble wieghts ($L = 10$)

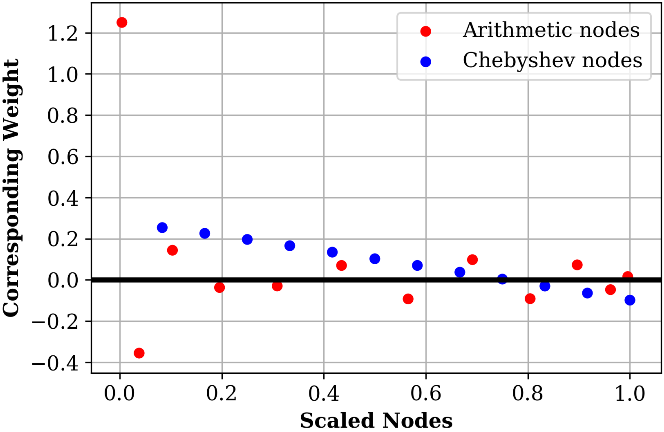



**Figure:** For $L = 10$ the arithmetic nodes (bandwidth scaled by $k, k+1, \ldots$) give weights with higher dynamic range than the proposed Chebyshev node approach.[1]

[1] M. Noshad, L. Xu and A. Hero, "Learning to Benchmark: Estimating Best Achievable Misclassification Error from Training Data," arXiv:1909.07192, Sept. 2019.

# Chebyshev stabilization of ensemble wieghts ($L = 100$)

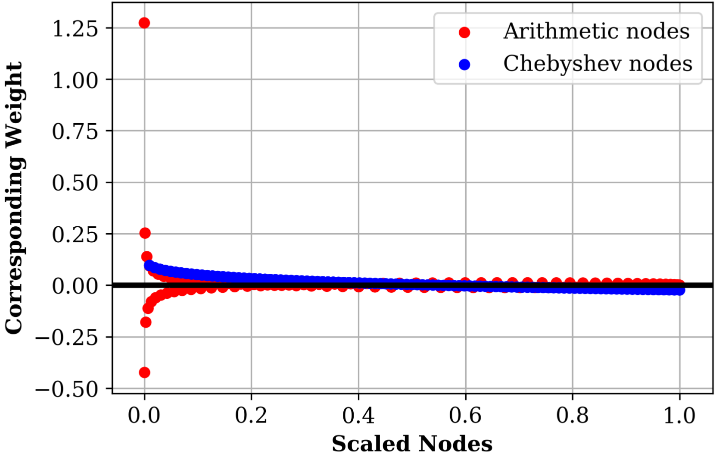



**Figure:** For $L = 100$ the arithmetic nodes (bandwidth scaled by $k, k+1, \ldots$) give weights with much higher dynamic range than the proposed Chebyshev node approach.

# Chebyshev wieghts improve MSE of benchmark learner

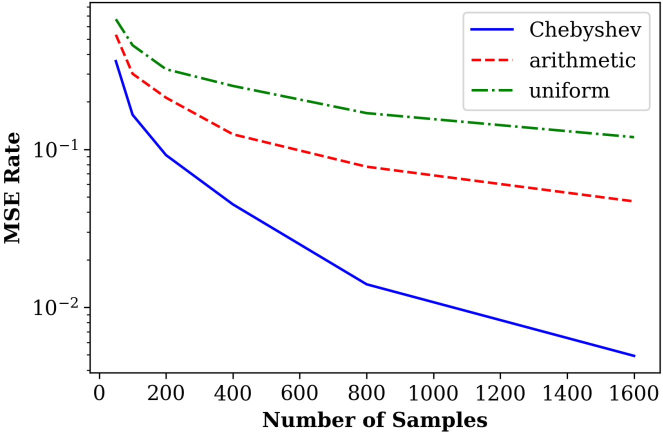
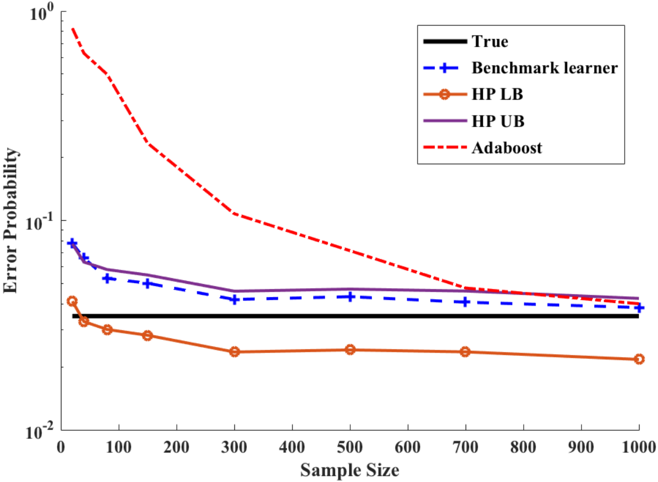



**Figure:** For a binary classification problem (mean of Gaussian isotropic dsn in dim $d = 100$) the proposed Chebyshev node approach provides significant improvement of MSE in Bayes estimation error rate.

# Benchmark learner as a minibatch stopping rule

Simulation: classification of 2 mean shifted 10 dim Gaussian densities




Ref: Noshad and Hero, AISTAT 2018

# Benchmark learner for assessing multiclass classification

Simulation: $K = 4$ classes in concentric sphere regions over $d = 20$ dimensions

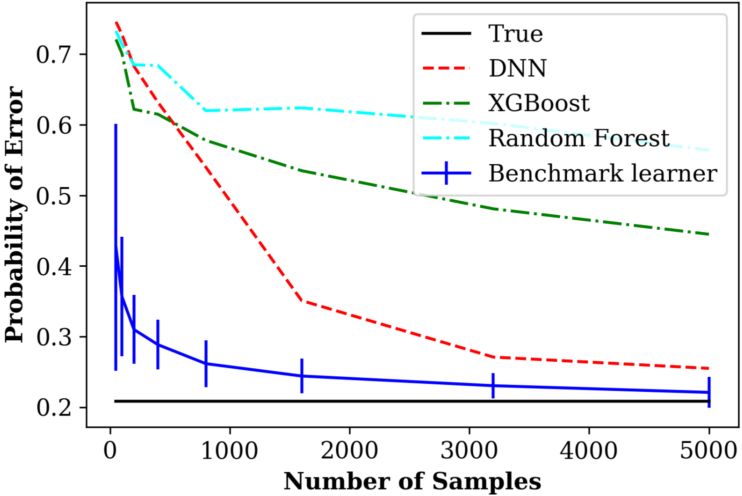
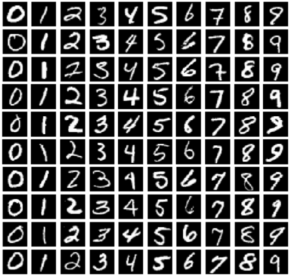



**Figure:** Benchmark learner suggests small margin for improvement. DNN: 5 hidden layers with [20,64,65,10,40] RELU neurons trained with ADAM and 10% dropout.

# Benchmarking MNIST digit classification

MNIST handwritten digit corpus:

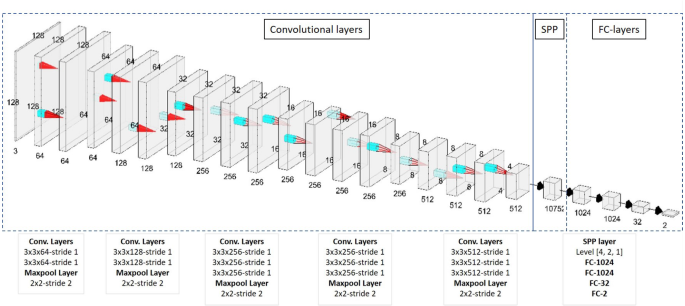
- $K = 10$ classes
- $d = 784$ dimensions
- $n = 60,000$ samples



| Papers | Method | Error rate |
|---|---|---|
| (Cireşan et al., 2010) | Single 6-layer DNN | 0.35% |
| (Ciresan et al., 2011) | Ensemble of 7 CNNs and training data expansion | 0.27% |
| (Cireşan et al., 2012) | Ensemble of 35 CNNs | 0.23% |
| (Wan et al., 2013) | Ensemble of 5 CNNs and DropConnect regularization | 0.21% |
| Benchmark learner | Ensemble $\epsilon$-ball estimator | 0.14% |

Table 1: Comparison of error probabilities of several the state of the art deep models with the benchmark learner, for the MNIST handwriting image classification dataset

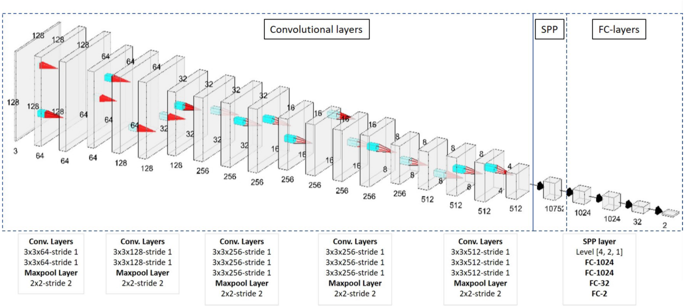
# Mutual information estimation: application to DNN information bottleneck




Convolutional neural network (CNN) for image classification[1]

- DNNs have remarkable empirical performance,

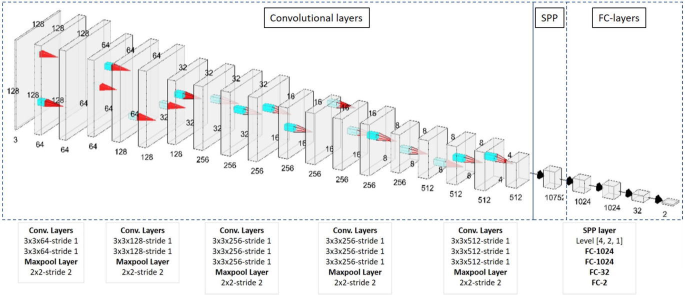
# Mutual information estimation: application to DNN information bottleneck




Convolutional neural network (CNN) for image classification[1]

- DNNs have remarkable empirical performance, but there is limited understanding of why DNN perform so well

# Mutual information estimation: application to DNN information bottleneck




Convolutional neural network (CNN) for image classification[1]

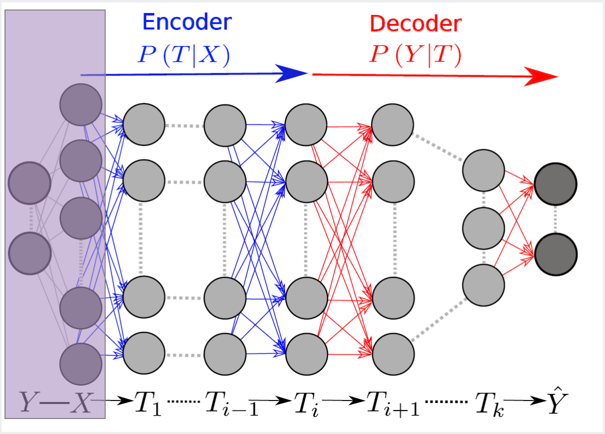
- DNNs have remarkable empirical performance, but there is limited understanding of why DNN perform so well

*The compositional learning hypothesis*: (A. Yuille, CVPR 2010)
DNN's learn in two phases:

- Phase 1: learn the easy cases (**memorize**)
- Phase 2: generalize to the hard cases (**compress**)

[1] B. DuFumier. A new deep learning approach to solar flare prediction. ENSTA internship report, Sept. 2018

# Tishby's framework: encoder/decoder information bottleneck



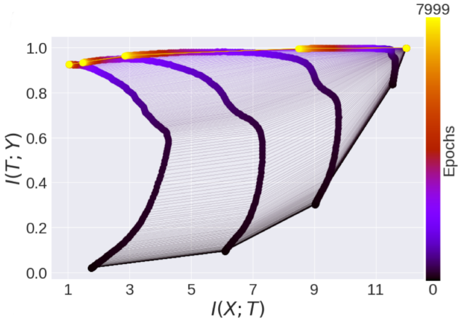

- Encoder I/O: input $X$, ouptut $T$ (features)
- Decoder I/O: input $T$, output $Y$ (labels)

[1] R Schwartz-Ziv and N Tishby. "Opening the black box of deep neural networks via information." arXiv 2017

[2] AM Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, BD Tracey, DD. Cox, "On the information bottleneck theory of deep learning," ICLR 2018

# Information plane: a layer-by-layer plot of discrimination vs. compression

Compression $\longleftrightarrow$ Memorization

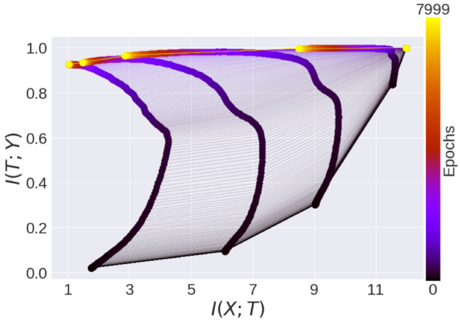



- Plot of training-trajectories of $[I(X;T_i), I(T_i;Y)]$ for different layers $T_i$

$$I(X;T) = \int f_{XT} \log\left(\frac{f_{XT}}{f_X f_T}\right), \quad I(T;Y) = \int f_{TY} \log\left(\frac{f_{TY}}{f_T f_Y}\right)$$

# Information plane: a layer-by-layer plot of discrimination vs. compression

Compression $\longleftrightarrow$ Memorization




- Plot of training-trajectories of $[I(X;T_i), I(T_i;Y)]$ for different layers $T_i$

$$I(X;T) = \int f_{XT} \log\left(\frac{f_{XT}}{f_X f_T}\right), \quad I(T;Y) = \int f_{TY} \log\left(\frac{f_{TY}}{f_T f_Y}\right)$$

- Schwartz-Ziv&Tishby[1] observed **memorization→compression** for *tanh* activation (MLP 10-8-6-4-2 and classification of 10D Gaussian)

[1] R Schwartz-Ziv and N Tishby. Opening the black box of deep neural networks via information. arXiv 2017

# Does memorization→compression depend on activation function?

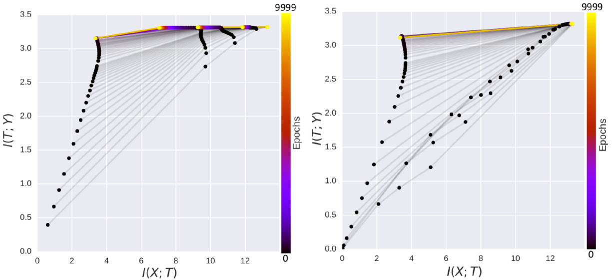



**Figure:** Figure 1.C (*tanh*) and 1.D (*ReLU*) from Saxe *et al*[1]

- 784-1024-20-20-20-10 MLP trained on MNIST dataset
- Output layer: *sigmoid*. Hidden layers: *tanh* at left and *ReLu* at right.
- Trained using SGD on cross–entropy loss with minibatch size 128
- Learning rate= 0.001

[1] Saxe, Bansal, Dapello, Advani, Kolchinsky, Tracey, and Cox, "On the information bottleneck theory of deep learning," ICLR, 2018.

# Does memorization→compression depend on activation function?

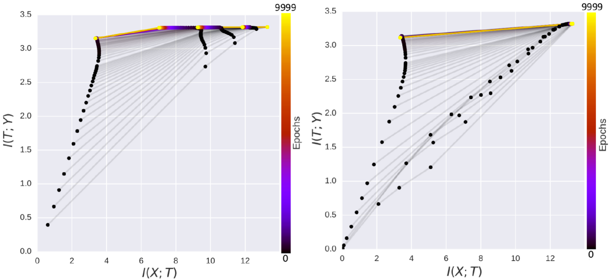
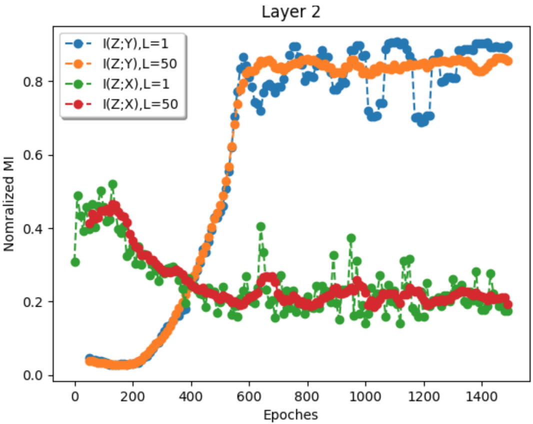



**Figure:** Figure 1.C (*tanh*) and 1.D (*ReLU*) from Saxe *et al*[1]

- 784-1024-20-20-20-10 MLP trained on MNIST dataset
- Output layer: *sigmoid*. Hidden layers: *tanh* at left and *ReLu* at right.
- Trained using SGD on cross–entropy loss with minibatch size 128
- Learning rate= 0.001
- Saxe *et al* claim that *ReLU* inner layers **exhibit no compression**

[1] Saxe, Bansal, Dapello, Advani, Kolchinsky, Tracey, and Cox, "On the information bottleneck theory of deep learning," ICLR, 2018.

# Information plane for MLP/*ReLU* using ensemble MI estimation




- 10-8-6-4-2 MLP/*ReLU* trained on 10,000 samples of 10D Gaussian
- MI with $L = 1$ (green&blue) is the Schwartz-Ziv&Tishby MI estimate
- Proposed ensemble MI implementation[1] (red&orange) is more stable

[1] Noshad, Yu, Hero, "Scalable MI estimation using dependence graphs," ICASSP 2019.

# Ensemble estimation provides confirmatory evidence

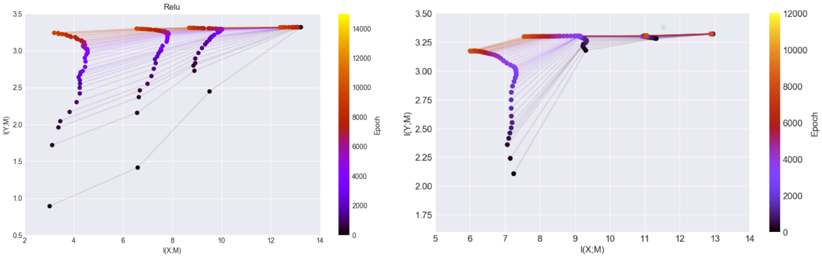



**Figure:** Left: **MLP/ReLU** 784-1024-20-20-20-10. Right: **CNN/ReLU** 784-4-8-16-10

- MLP and CNN trained on MNIST dataset[1]

$\Rightarrow$ Memorization$\to$Compression phenonomon occurs in both MLP and CNN

[1] Noshad, Yu, Hero, "Scalable MI estimation using dependence graphs," ICASSP 2019.

# Summary

Main takeaways

- Learning to benchmark involves 2 types of meta-learning
  - Meta-learning v0: Learning ensembles of weak base-learners (Freund&Schapire 1996)
  - Meta-learning v1: Learning the Bayes error rate (BER)[1,2,3]

[1] Moon, Sricharan, Greenewald, Hero. "Ensemble estimation of information divergence." *Entropy*, 20, no. 8, 2018.

[2] Noshad and Hero, "Scalable hash-based estimation of divergence measures," AISTATS 2018.

[3] Noshad, Zeng, Hero, "Scalable mutual information estimation using dependence graphs," IEEE ICASSP, 2019

## Summary

Main takeaways

- Learning to benchmark involves 2 types of meta-learning
  - Meta-learning v0: Learning ensembles of weak base-learners (Freund&Schapire 1996)
  - Meta-learning v1: Learning the Bayes error rate (BER)[1,2,3]
- Ensemble benchmark learner achieves rate optimal performance in both computational complexity and sample complexity

[1] Moon, Sricharan, Greenewald, Hero. "Ensemble estimation of information divergence." *Entropy*, 20, no. 8, 2018.

[2] Noshad and Hero, "Scalable hash-based estimation of divergence measures," AISTATS 2018.

[3] Noshad, Zeng, Hero, "Scalable mutual information estimation using dependence graphs," IEEE ICASSP, 2019

# Summary

Main takeaways

- Learning to benchmark involves 2 types of meta-learning
  - Meta-learning v0: Learning ensembles of weak base-learners (Freund&Schapire 1996)
  - Meta-learning v1: Learning the Bayes error rate (BER)[1][2][3]
- Ensemble benchmark learner achieves rate optimal performance in both computational complexity and sample complexity
- Benchmark learning applications:
  - Performance monitoring: learning sufficient sample size
  - Feature learning: performing data-driven feature selection
  - Interpretable learning: exploring DNN compositional learning hypothesis

[1] Moon, Sricharan, Greenewald, Hero. "Ensemble estimation of information divergence." *Entropy*, 20, no. 8, 2018.

[2] Noshad and Hero, "Scalable hash-based estimation of divergence measures," AISTATS 2018.

[3] Noshad, Zeng, Hero, "Scalable mutual information estimation using dependence graphs," IEEE ICASSP, 2019

# Selected references


1 K. Moon, K. Sricharan, A. Hero, "Ensemble Estimation of Generalized Mutual Information with Applications to Genomics," IEEE Transactions on Information Theory, to appear 2021.

2 S. Sekeh, B. Oselio and A. Hero, "Learning to Bound the Multi-class Bayes Error," IEEE Trans. on Signal Processing, vol. 68, pp. 3793 – 3807, May 2020.

3 M. Noshad, L. Xu and A. Hero, "Learning to Benchmark: Estimating Best Achievable Misclassification Error from Training Data," arXiv:1909.07192, Sept. 2019.

4 Moon, Sricharan, Greenewald, Hero. "Ensemble estimation of information divergence." *Entropy*, 20, no. 8, 2018.

5 Python script implementing the method of [3] is available on `Google colaboratory`:


BayesErrorEstimator.jpynb