Machine Learning Methods for Robust Artificial Intelligence Part 2: Rejection

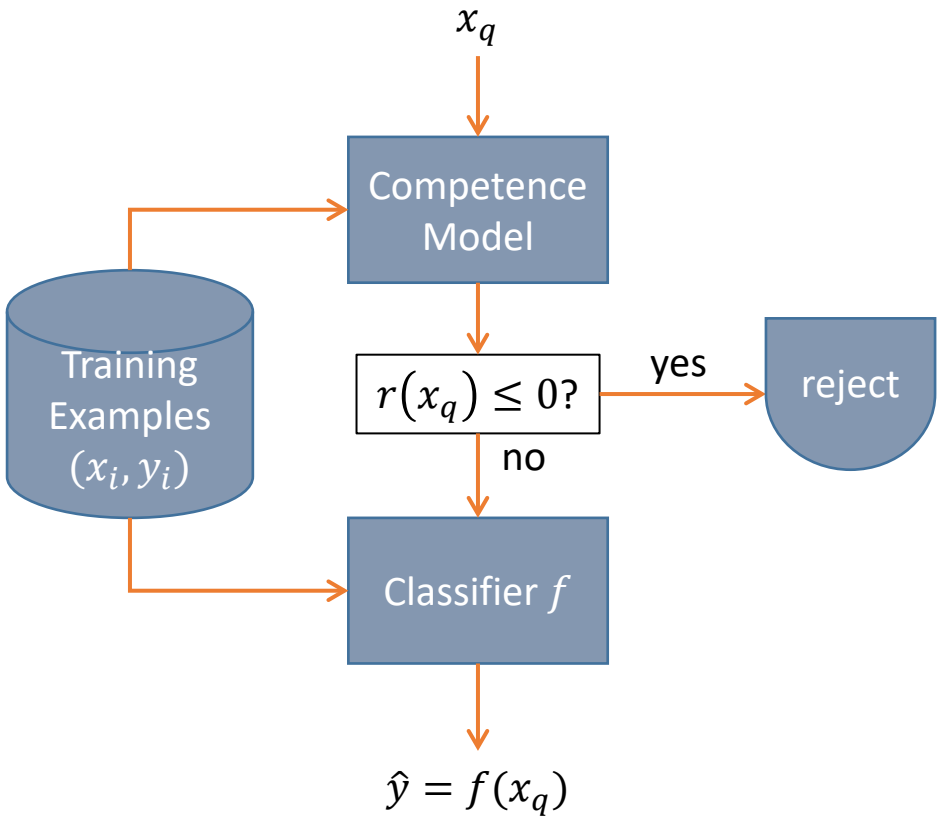
Thomas G. Dietterich, Oregon State University

tgd@cs.orst.edu

@tdietterich

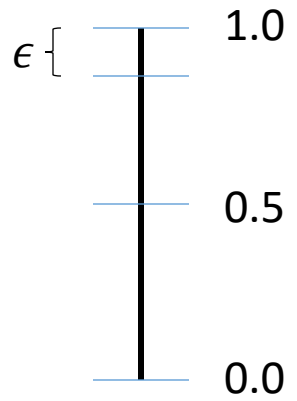
Rejection

- Given:
  - Training data $(x_1, y_1), \ldots, (x_N, y_N)$
  - Target accuracy level $1 - \epsilon$
  - Learn a classifier f and a rejection rule r
- At run time
  - Given query x_q
  - If $r(x_q) \leq 0$, REJECT
  - Else classify $f(x_q)$



Basic Theory

- Suppose $f^*(x, y) = P(y|x)$ is the optimal probabilistic classifier
- Best prediction is $\hat{y} = \arg\max_y f^*(x, y)$
- Then the optimal rejection rule is to REJECT if $f^*(x, \hat{y}) < 1 - \epsilon$
- (Chow 1970)



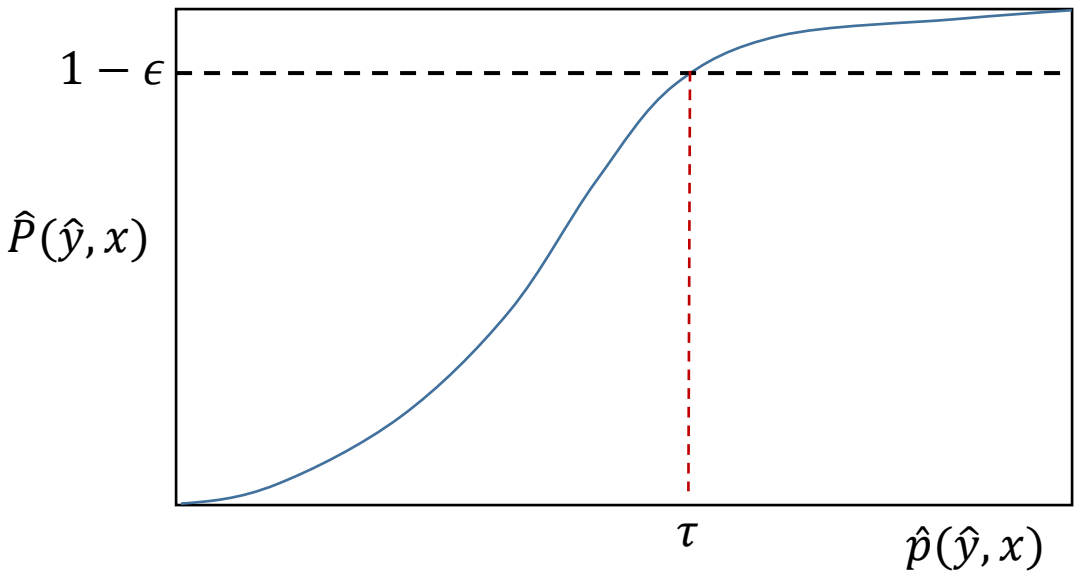
Two Paths Forward

- Path 1:
  - Calibrate the predicted probabilities of the classifier
  - Apply Chow's result
- Path 2:
  - Directly learn a rejection function $r(x_q)$

- Vapnik's principle: "When solving a problem of interest, do not solve a more general problem as an intermediate step."

Non-Optimal Case

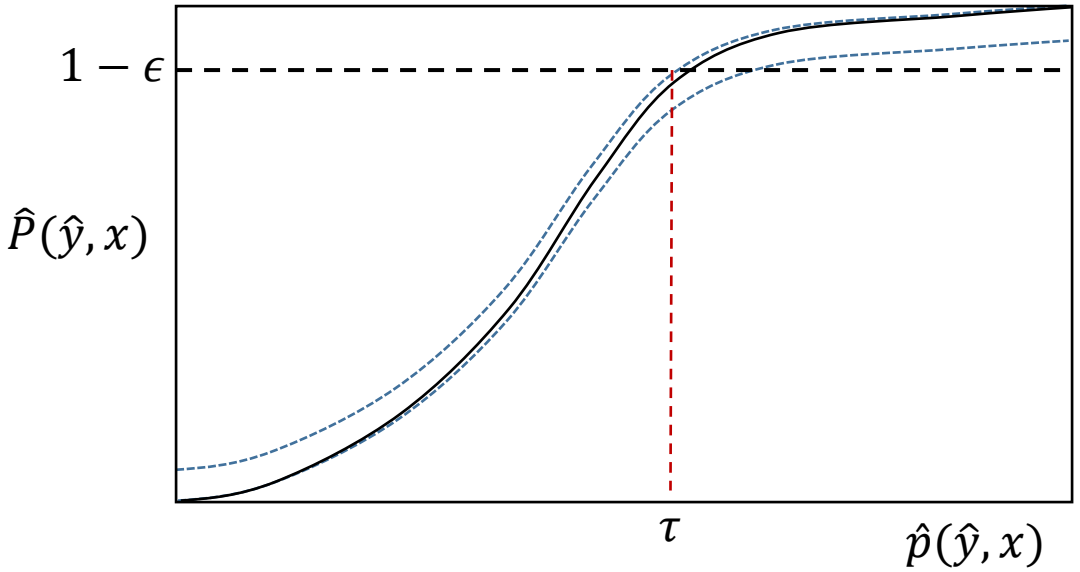
- Suppose the probabilities output by f are not optimal. We can still determine a threshold with performance guarantees
- Let $(f(x_i, \hat{y}_i), \mathbb{I}[\hat{y}_i = y_i])$ be a set of calibration data points $i = 1, \ldots, N$
- Sort them by $\hat{p}(\hat{y}_i | x_i) = f(x_i, \hat{y}_i)$
- Choose the smallest threshold τ such that if $f(x_i, \hat{y}_i) > \tau$ then the fraction of correct predictions is $1 - \epsilon$



- Theorem: If $N > \frac{1}{\eta^2} \log \frac{2}{\delta}$ then w.p. $1 - \delta$, the true error rate will be bounded by $1 - (\epsilon + \eta)$

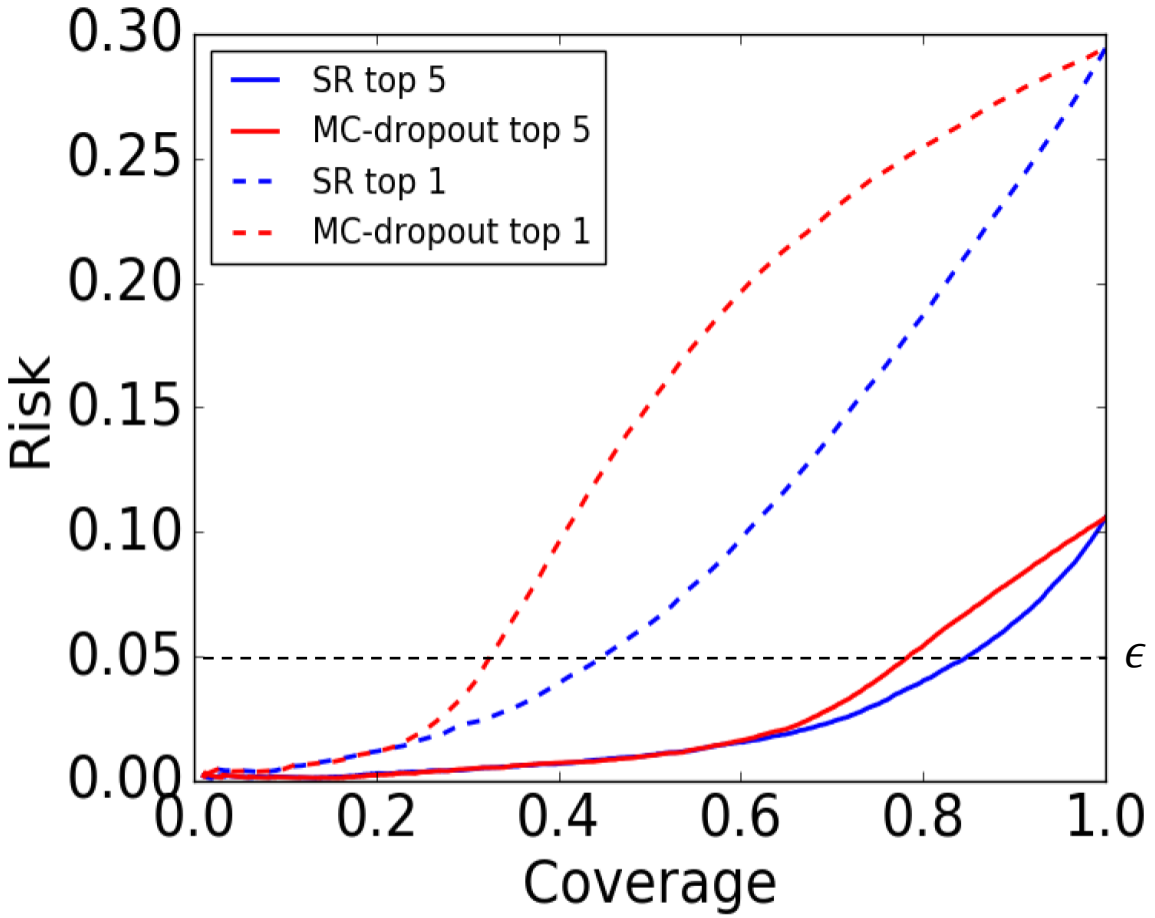
Finite Sample (PAC) Proof

- $P\left(\sqrt{n}\sup_x\left|\hat{F}_n(x) - F(x)\right| > \lambda\right) \leq 2\exp(-2\lambda^2)$ Massart (1990)
- Set $x \coloneqq \tau$
- $P\left(\eta > \frac{\lambda}{\sqrt{n}}\right) = 2\exp(-2\lambda^2)$
- Set $\frac{\lambda}{\sqrt{n}} = \eta$ and $\delta = 2\exp(-2\lambda^2)$; solve for n
- $\lambda = \eta\sqrt{n}$
- $\delta = 2\exp(-2\eta^2 n)$
- $\log\frac{\delta}{2} = -\eta^2 n$
- $n = \frac{1}{\eta^2}\log\frac{2}{\delta}$
- If $n > \frac{1}{\eta^2}\log\frac{2}{\delta}$ then w.p. $1-\delta$, the true error rate will be bounded by $1-(\epsilon+\eta)$



Related Work

- Geifman & El Yaniv (2017)
  - Develop confidence scores based on either the softmax (“SR”) or Monte Carlo dropout (“MC-dropout”)
  - Binary search for the threshold
  - Use an exact Binomial confidence interval instead of Massart’s bound
  - Union bound over the binary search queries



(c) Image-Net

Cost-Sensitive Rejection

- Cost Matrix
- Optimal Classifier
  - For $\hat{p}(y = 1|x) \geq \tau_1$, predict 1
  - For $\hat{p}(y = 2|x) \geq \tau_2$, predict 2
  - Else REJECT
- Search all pairs (τ_1, τ_2) to minimize expected cost
- Pietraszek (2005) provides a fast algorithm based on (a) isotonic regression and (b) computing the slopes on the ROC curve corresponding to τ_1 and τ_2

	Actions		
Probabilities	Predict 1	Predict 2	Reject
$P(y = 1\|x)$	0	c_{12}	c_{1r}
$P(y = 2\|x)$	c_{21}	0	c_{2r}

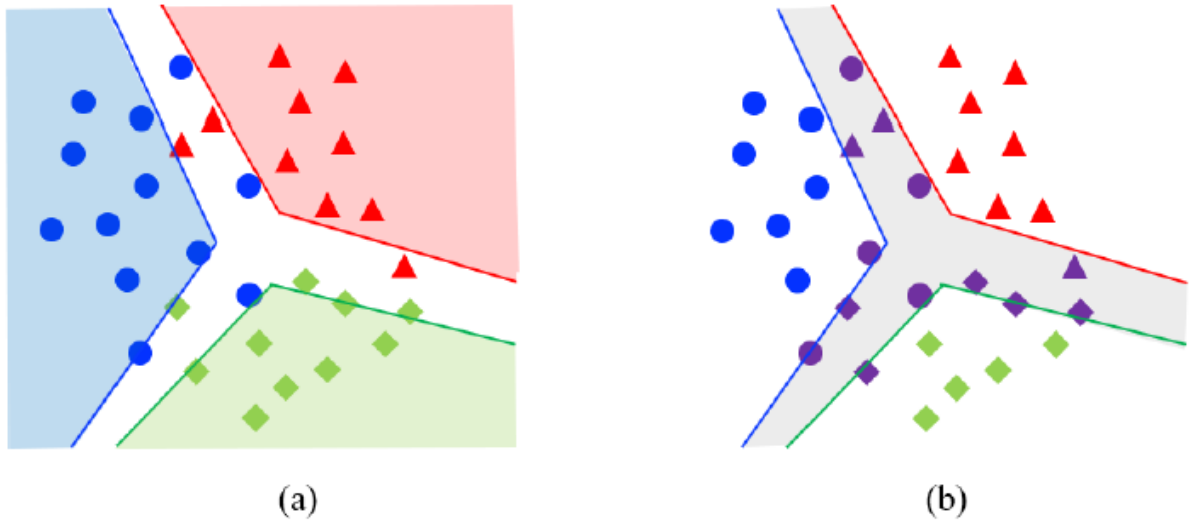
Note: If we have calibrated probabilities, we could easily compute the optimal decision, but our goal is to avoid calibration

Cost-Sensitive One-vs-Rest Classifiers

(Charoenphakdee, Cui, Zhang, Sugiyama, ICML 2021)

- Rescale costs to $[0,1]$ and assume constant cost of rejection
$$c < 0.5$$
- For each class k learn a cost-sensitive binary classifier
  - Predict 1 if $P(y = k|x) > 1 - c$
  - Else Predict 0
- Reject if
  - All classifiers predict 0
  - More than one classifier predicts 1
- We can choose a threshold to achieve $P(y = k|x) = 1 - c$

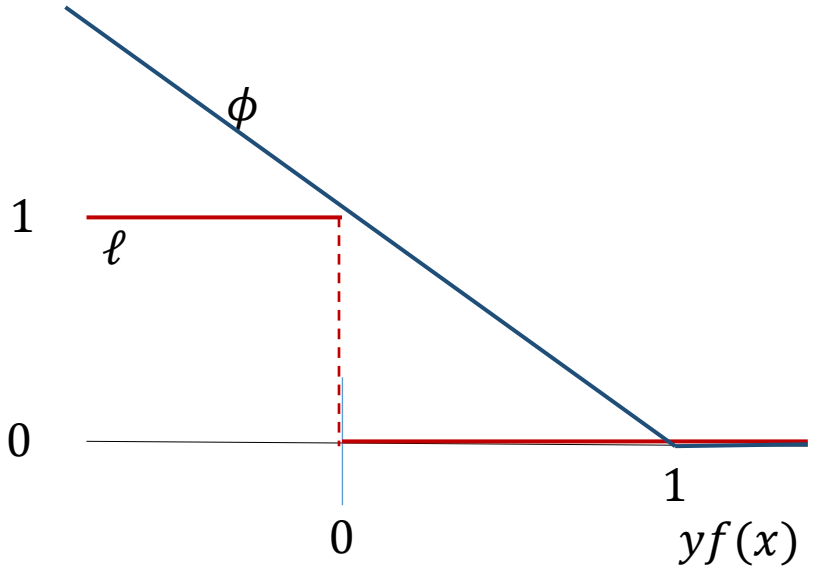
Costs	Actions			
Probabilities	Predict 1	Predict 2	Predict 3	Reject
$P(y = 1\|x)$	0	1	1	c
$P(y = 2\|x)$	1	0	1	c
$P(y = 3\|x)$	1	1	0	c



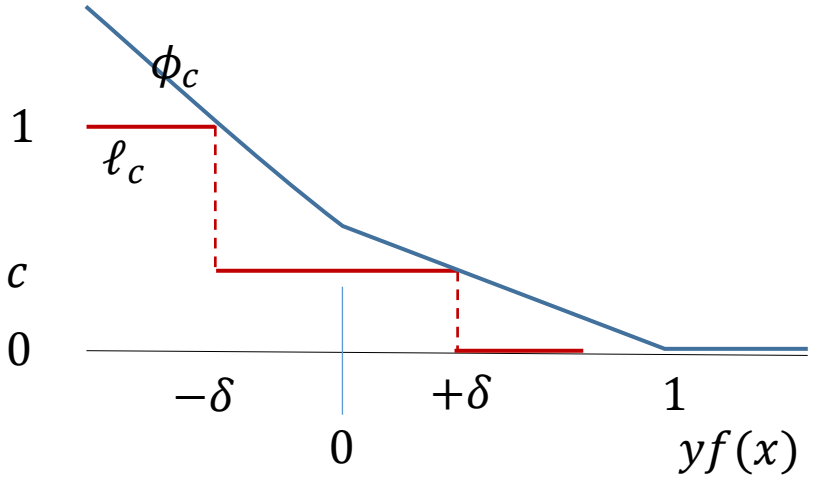
(a) (b)

Support Vector Machines

- Jointly learn the classifier f and the rejection function r
  - Method 1: Double Hinge Loss (DHL)
    - Herbei & Wegkamp, 2006; Bartlett & Wegkamp, 2008
  - Method 2: CHR Loss
    - Cortes, DeSalvo & Mohri, 2016
- Extend SVM Methodology
  - Define the loss function ℓ
  - Derive a convex upper bound on the loss ϕ
  - Apply convex optimization to minimize this upper bound



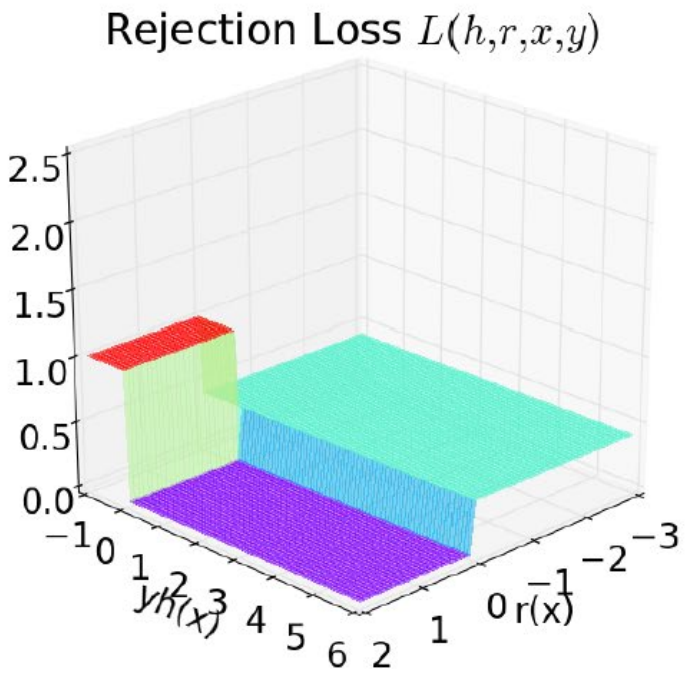
Double Hinge Loss

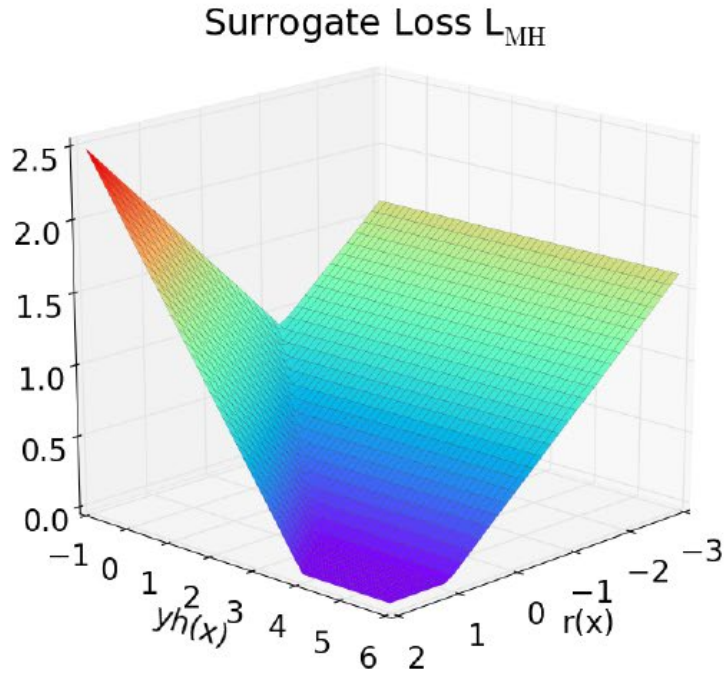


CHR Convex Upper Bound

Cortes, DeSalvo & Mohri, 2016

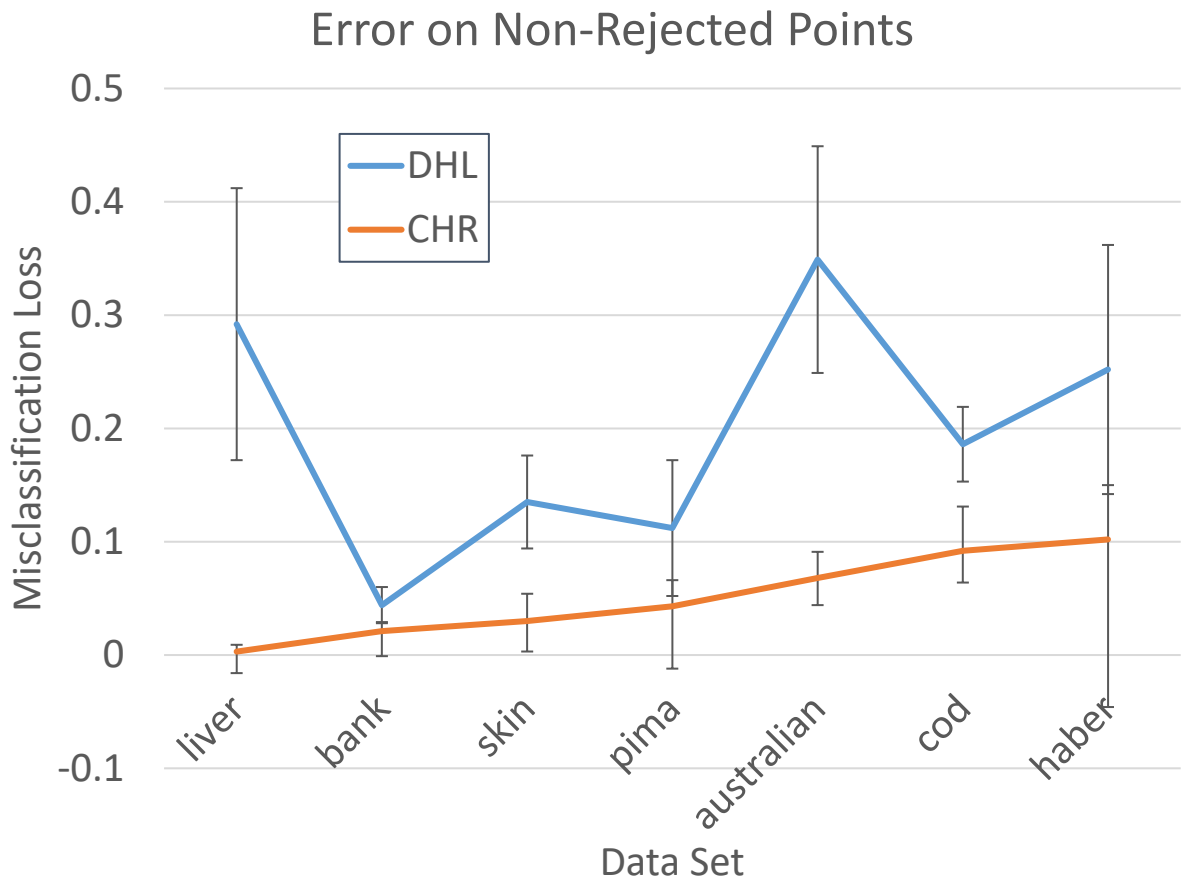
- $L_{MH}(r, f, x, y) = \max\left(1 + \frac{1}{2}\left(r(x) - yf(x)\right), c\left(1 - \frac{1}{1-2c}r(x)\right), 0\right)$





Comparison of SVM Methods

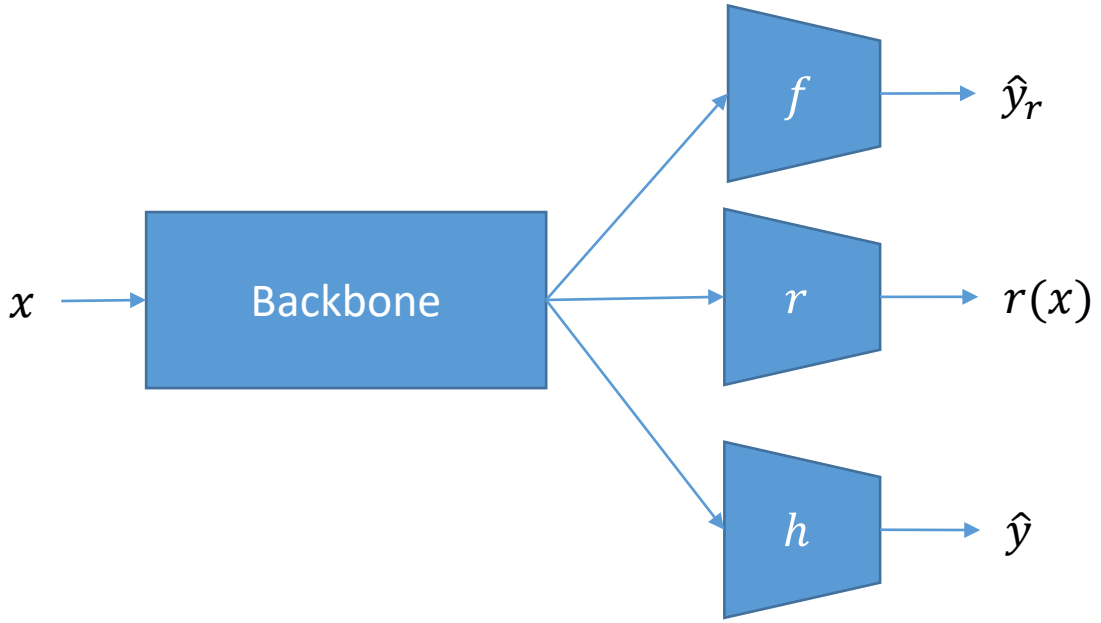
- Rejection cost $c = 0.25$
- Method
  - Fit CHR loss and measure fraction rejected
  - Tune DHL to reject the same number of points
  - Measure total loss
- CHR is strictly superior to double-hinge loss



Note: DHL modified to reject the same number of points as CHR

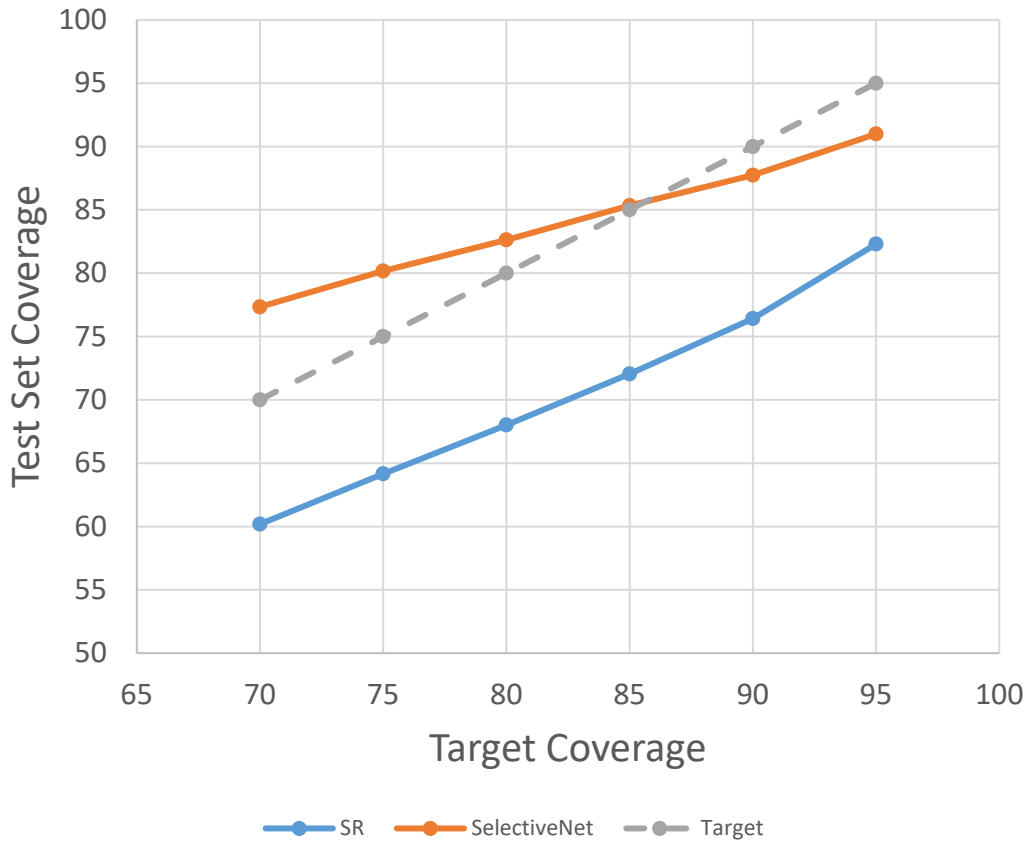
Deep Learning for Image Classification: SelectiveNet

- Geifman & El Yaniv (2019)
  - Minimize the error on the non-rejected images subject to a constraint on the coverage (fraction of images not rejected)
  - User must specify c, the target coverage, rather than ϵ, the target error rate
- Network has three "heads"
  - f and h are both classification heads trained with cross-entropy loss
  - r is the rejection classifier
  - h encourages the backbone to learn a latent representation that can classify *all* of the examples
- Loss function:
  - $\alpha\mathcal{L}_f + (1-\alpha)\mathcal{L}_h + \lambda([c-\phi(r)]_+)^2$
    - $\mathcal{L}_f$ classification loss on training examples for which $r(x) < 0.5$
    - $\mathcal{L}_h$ classification loss on all training examples
    - $\phi(r)$: fraction of training examples for which $r(x) < 0.5$; (i.e., not rejected)



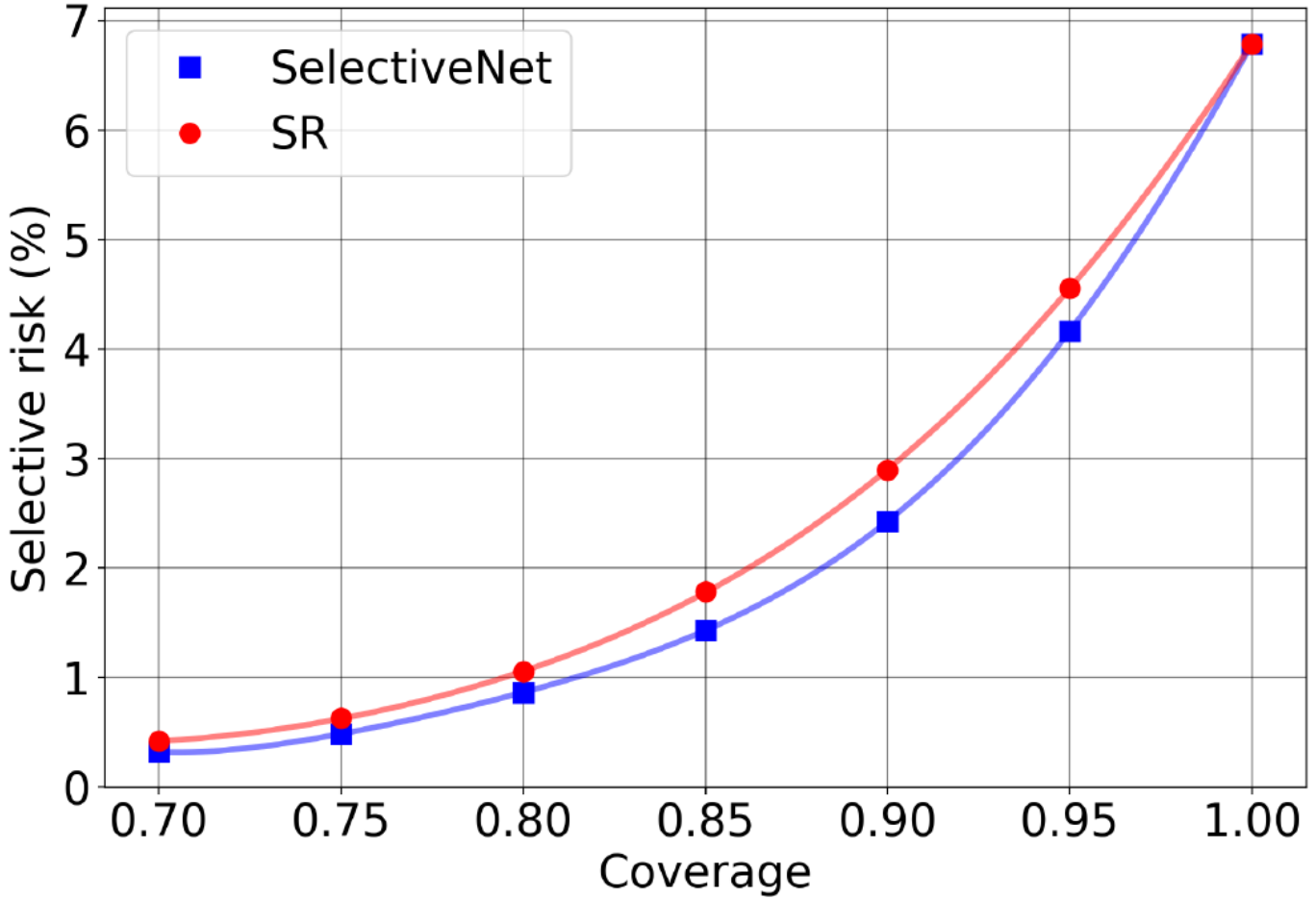
Results:

- SR: use $\max_k P(y = k|x)$ as the competence signal
- SelectiveNet: Use r as the rejection rule
- Using only the training data, the coverage achieved on the test set does not match the target
  - SelectiveNet is closer
- Solution: Use a calibration set to select a threshold τ and reject if $r(x) > \tau$



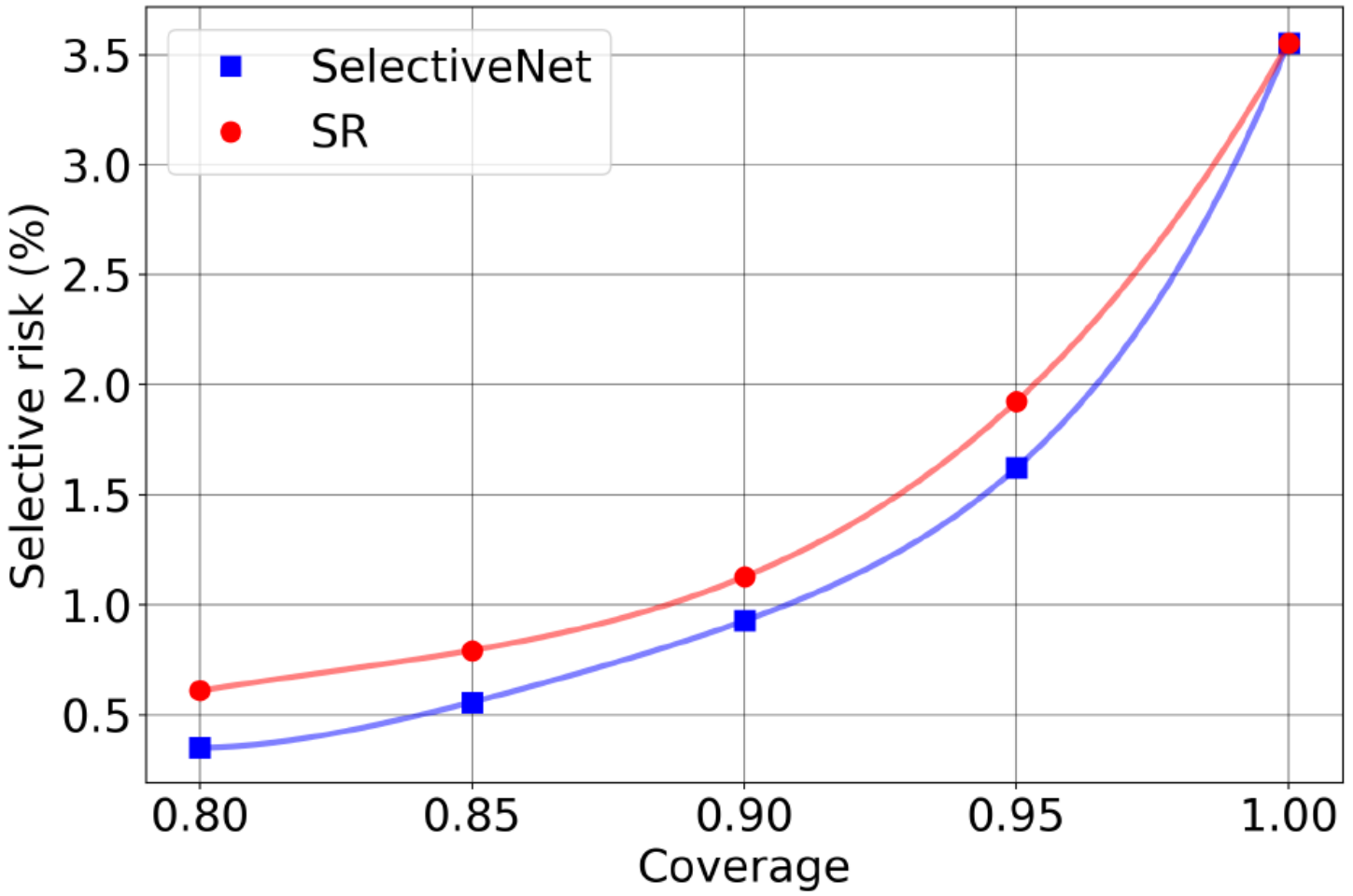
Results: CIFAR10

- Selective Risk = error rate on the test images that are classified (not rejected)
- Coverage = fraction of test images that are not rejected

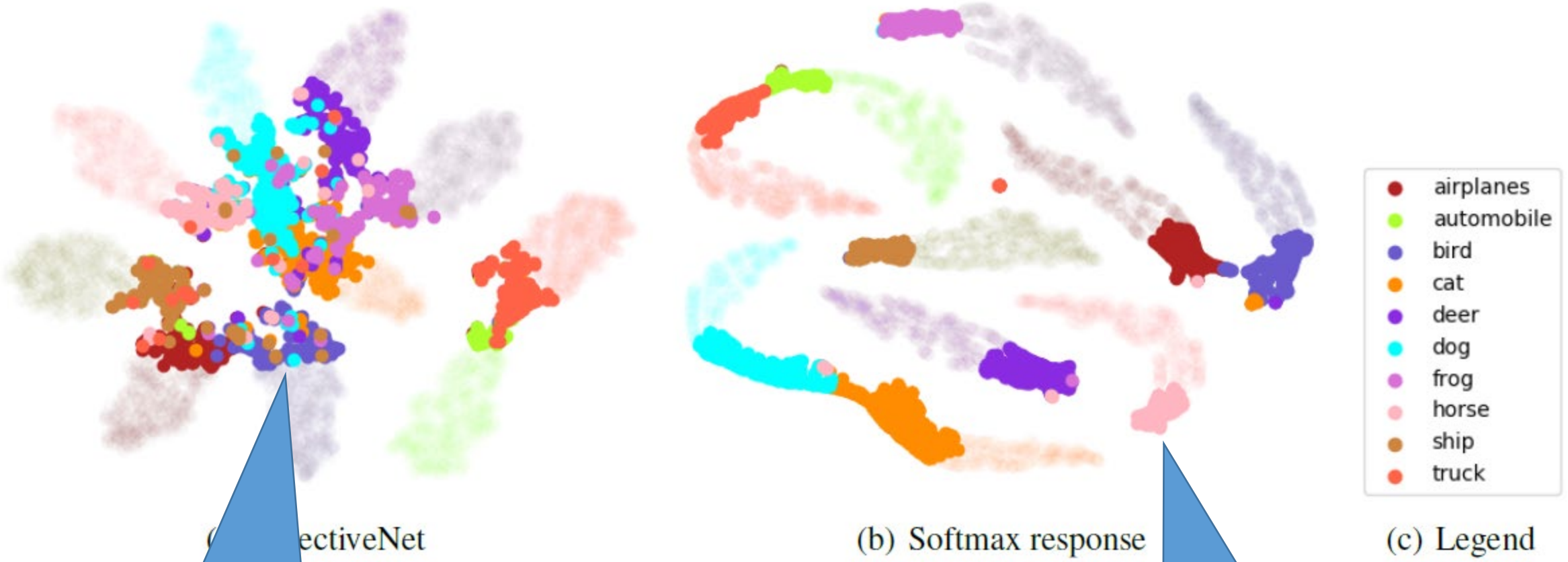


Results: Dogs vs Cats

- Selective Risk = error rate on the test images that are classified (not rejected)
- Coverage = fraction of test images that are not rejected



CIFAR-10 Visualization of Learned Representation



(a) ectiveNet (b) Softmax response (c) Legend

visualization of the embedding representation of (a) Se

SelectiveNet does not try to discriminate among images that will be rejected

Softmax tries to separate all of the classes (of course)

Discussion

- Jointly learning a classifier and a rejection function gives better results than learning a classifier and thresholding the class probabilities
- The classifier focuses attention on separating the instances that will not be rejected, and therefore learns a different latent representation
- Geifman & El Yaniv did not compare against calibrated class probabilities, but post-hoc calibration would not have any effect on the learned representation

Reject Option Summary

- Basic thresholding is easy and gives PAC guarantees
- 2-class thresholding with differential costs is easy
- K-class thresholding with constant rejection cost is easy
- K-class thresholding with differential costs: open problem
- Jointly training classifier and rejection function
  - Good solution for SVMs
  - Initial methods for DNNs

Prediction Sets

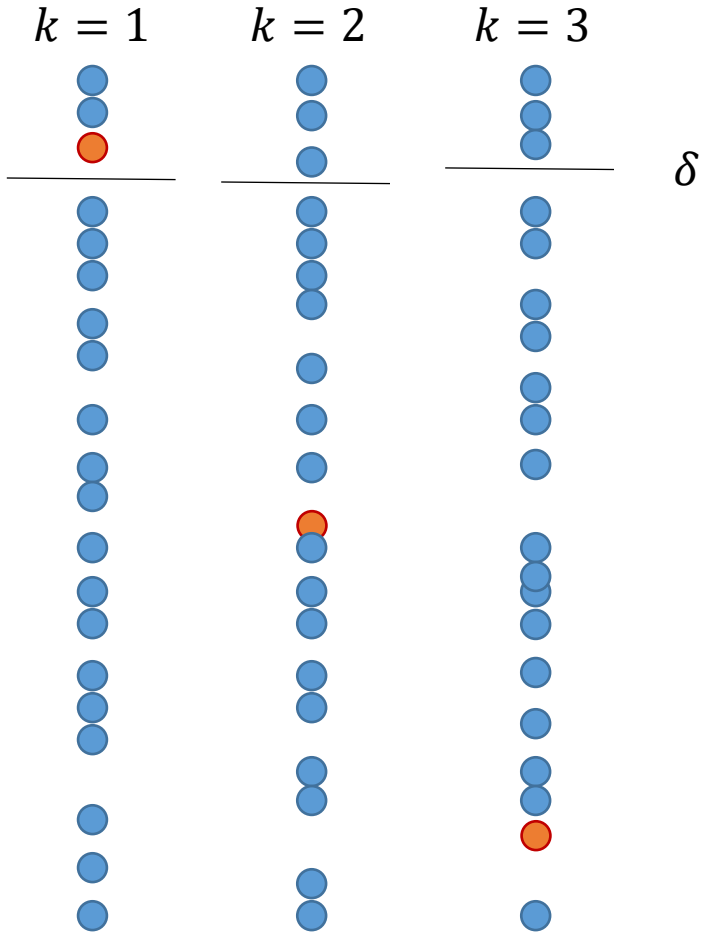
- Given:
  - Training data $[\![z_1, \ldots, z_{n-1}]\!]$ where $z_i = (x_i, y_i)$
  - Classifier f trained on the training data
  - Query x_q
  - Accuracy level δ
- Find:
  - Smallest set $C(x_q) \subseteq \{1, \ldots, K\}$ such that $y_q \in C(x_q)$ with probability $1 - \delta$

Conformal Prediction

- Conformal Prediction is a general method for obtaining prediction intervals with coverage guarantees. See Vovk, Gammerman, & Shafer (2005)
- Nonconformity measure $A_n: \mathcal{Z}^{n-1} \times \mathcal{Z} \mapsto \mathbb{R}$ indicates how different the last item is from the first $n-1$ items
- Method:
  - For each class k, let $z_n^k = (x_q, k)$
    - $\forall i\ \alpha_i^k := A\left(\left[\!\left[z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n^k\right]\!\right], z_i\right)$ "how different is z_i from the rest of the z values if we label x_q as class k?
    - Let $p^k =$ fraction of $\left[\!\left[\alpha_1^k, \ldots, \alpha_n^k\right]\!\right]$ that are $\geq \alpha_n^k$. "how many points are stranger than z_n^k?"
  - $C\left(x_q\right) = \left\{k \middle| p^k \geq \delta\right\}$ "include every class label that we can't reject with $p < \delta$"
  - Output $C\left(x_q\right)$
- Theorem: With probability $1-\delta$, $C\left(x_q\right)$ contains the correct class

Conformal Prediction

- Red points
  - x_q provisionally labeled by class k
  - measure non-conformity score α_q^k of the query
  - measure fraction of data points p^k with non-conformity score greater than α_q^k
  - if $p^k < \delta$, then we can reject the null hypothesis that $k \in C(x_q)$
- In this case, we can reject $k = 1$, but we must include $k = 2$ and $k = 3$
  - $C(x_q) = \{2,3\}$



Examples of Nonconformity Measures

- Conditional probability method:
  - Train a probabilistic classifier f on $[\![z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n]\!]$
  - Then compute

    $$A([\![z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n]\!], z_i) = -\log f(z_i) = -\log \hat{p}\left(y = k \middle| x_q\right)$$

- Nearest neighbor nonconformity
  - $A(B, z) = \dfrac{\text{distance to nearest } z' \in B \text{ in same class}}{\text{distance to nearest } z' \in B \text{ in different class}}$

Additional Information

- In addition to outputting $C(x_q)$, we can output
  - $\hat{y}_q = \arg\max_k p^k$ (the best prediction)
  - $p_q = \max_k p^k$ (the p-value of the best prediction)
  - $1 - \max_{k;k\neq\hat{y}_q} p^k$ (the "confidence". We have more confidence if the second-best p-value is small)

Batch (“inductive”) Conformal Prediction

- Divide data into training set and validate set
- Train f on the training data
- Let $[\![z_1, \dots, z_n]\!]$ be the validation data
- Let $\alpha_1, \dots, \alpha_n$ be the non-conformity scores of the validation data
  - $\alpha_i \coloneqq A([\![z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_n]\!], z_i)$
- Given query x_q
  - For $k = 1, \dots, K$
    - Let $z_q^k = (x_q, k)$
    - let $\alpha_q^k = A\left([\![z_1, \dots, z_n]\!], z_q^k\right)$
    - Let $p^k =$ fraction of $[\![\alpha_1, \dots, \alpha_n, \alpha_q^k]\!]$ that are $\geq \alpha_q^k$
  - $C(x_q) = \{k \mid p^k \geq \delta\}$

- Key difference: z_q^k does not affect the other non-conformity scores

Almost Equivalent to Learning a Threshold

- Let $\tau =$ the δ quantile of $[\![\alpha_1, \dots, \alpha_n]\!]$
- Given query x_q
  - For $k = 1, \dots, K$
    - Let $z_q^k = (x_q, k)$
    - let $\alpha_q^k = A\left([\![z_1, \dots, z_n]\!], z_q^k\right)$
  - $C(x_q) = \left\{k \middle| \alpha_q^k \geq \tau\right\}$

- Additional difference: τ is computed without considering α_q^k
- If n is large enough, this does not matter

Experimental Results

	Satellite	Shuttle	Segment
Hidden Units	23	12	11
Hidden Learning Rate	0.002	0.002	0.002
Output Learning Rate	0.001	0.001	0.001
Momentum Rate	0.1	0	0.1

- Non-conformity Measures
  - Resubstitution:
    - Train f on all data
    - Let $\hat{y}_i = f(x_i)$
    - $A\left([\![z_1, \ldots, z_{i-1}, z_i, \ldots, z_N]\!], (x_i, k)\right) = \mathbb{I}[\hat{y}_i = k]$
  - Leave One Out:
    - Train f on $[\![z_1, \ldots, z_{i-1}, z_i, \ldots, z_N]\!]$
    - Let $\hat{y}_i = f(x_i)$
    - $A\left([\![z_1, \ldots, z_{i-1}, z_i, \ldots, z_N]\!], (x_i, k)\right) = \mathbb{I}[\hat{y}_i = k]$

Satellite Data Set

Error:
$y_i \notin C(x_i)$

Nonconformity Measure	Confidence Level	Only one Label (%)	More than one label (%)	No Label (%)	Errors (%)
Resubstitution	99%	60.72	39.28	0.00	1.11
	95%	84.42	15.58	0.00	4.67
	90%	96.16	3.02	0.82	9.59
Leave one out	99%	61.69	38.31	0.00	1.10
	95%	85.70	14.30	0.00	4.86
	90%	96.11	3.10	0.79	9.43

Table 3. Results of the second mode of the Neural Networks ICP for the Satellite data set.

Shuttle Data Set

Nonconformity Measure	Confidence Level	Only one Label (%)	More than one label (%)	No Label (%)	Errors (%)
Resubstitution	99%	99.23	0.00	0.77	0.77
	95%	93.52	0.00	6.48	6.48
	90%	89.08	0.00	10.92	10.92
Leave one out	99%	99.30	0.00	0.70	0.70
	95%	93.86	0.00	6.14	6.14
	90%	88.72	0.00	11.28	11.28

Table 4. Results of the second mode of the Neural Networks ICP for the Shuttle data set.

Segmentation Data Set

Nonconformity Measure	Confidence Level	Only one Label (%)	More than one label (%)	No Label (%)	Errors (%)
Resubstitution	99%	90.69	9.31	0.00	0.95
	95%	97.71	1.25	1.04	3.68
	90%	94.68	0.00	5.32	6.71
Leave one out	99%	91.73	8.27	0.00	1.04
	95%	97.79	1.21	1.00	3.55
	90%	94.76	0.00	5.24	6.67

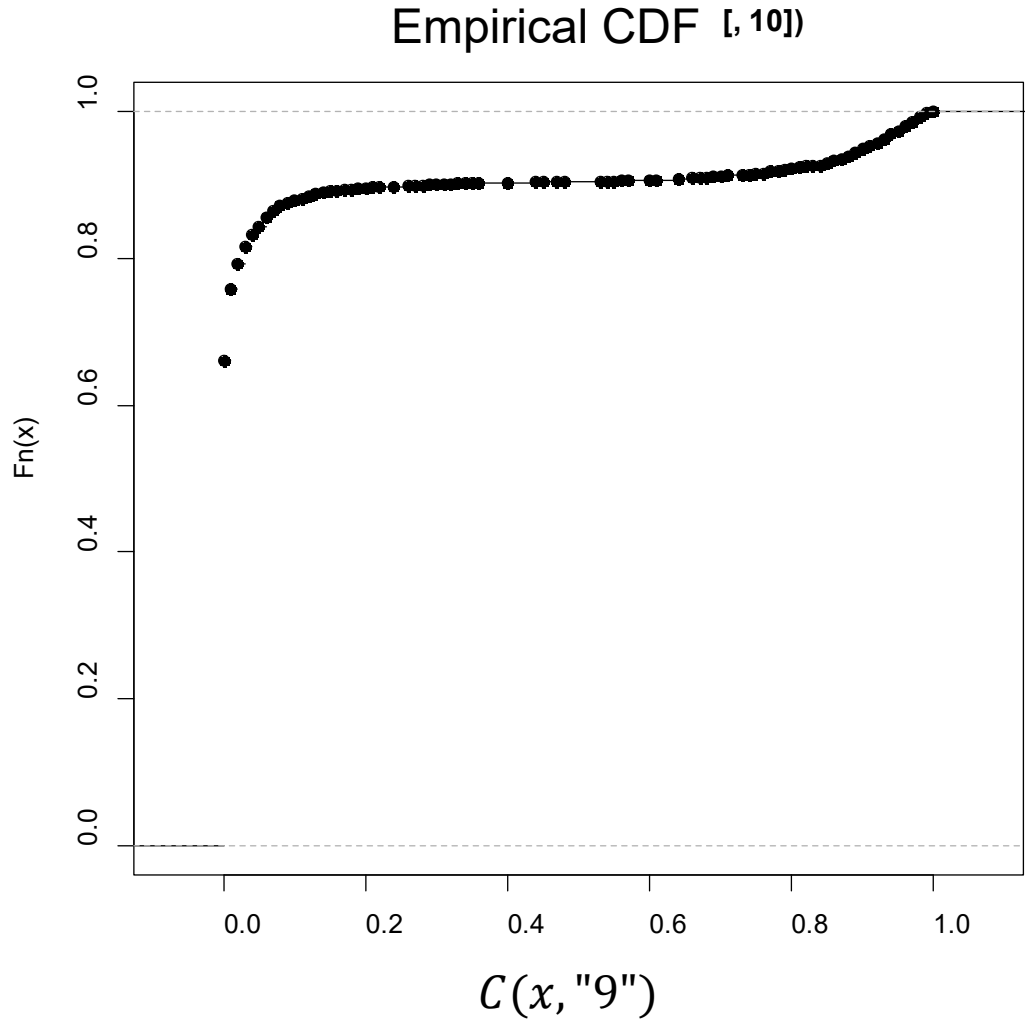
Table 5. Results of the second mode of the Neural Networks ICP for the Segment data set.

Pendigits + Random Forest

(Dietterich, unpublished)

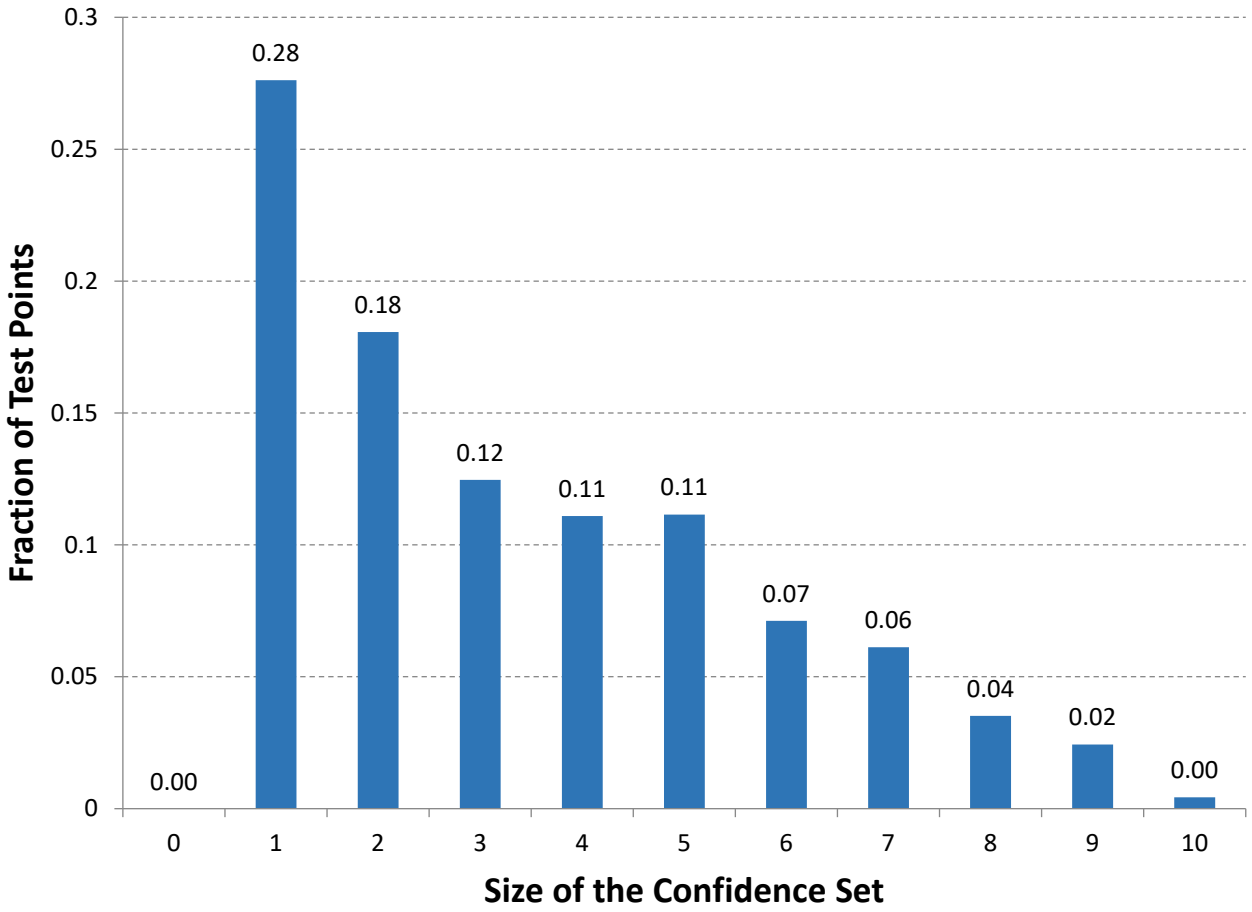
- Train a random forest on half of UCI Training Set
- Use the predicted class probability $P(y = k|x)$ as the (non)conformity score
- Compute τ values using other half of Training Set
- Compute C on the Test Set

Cumulative Distribution Function for Class “9”

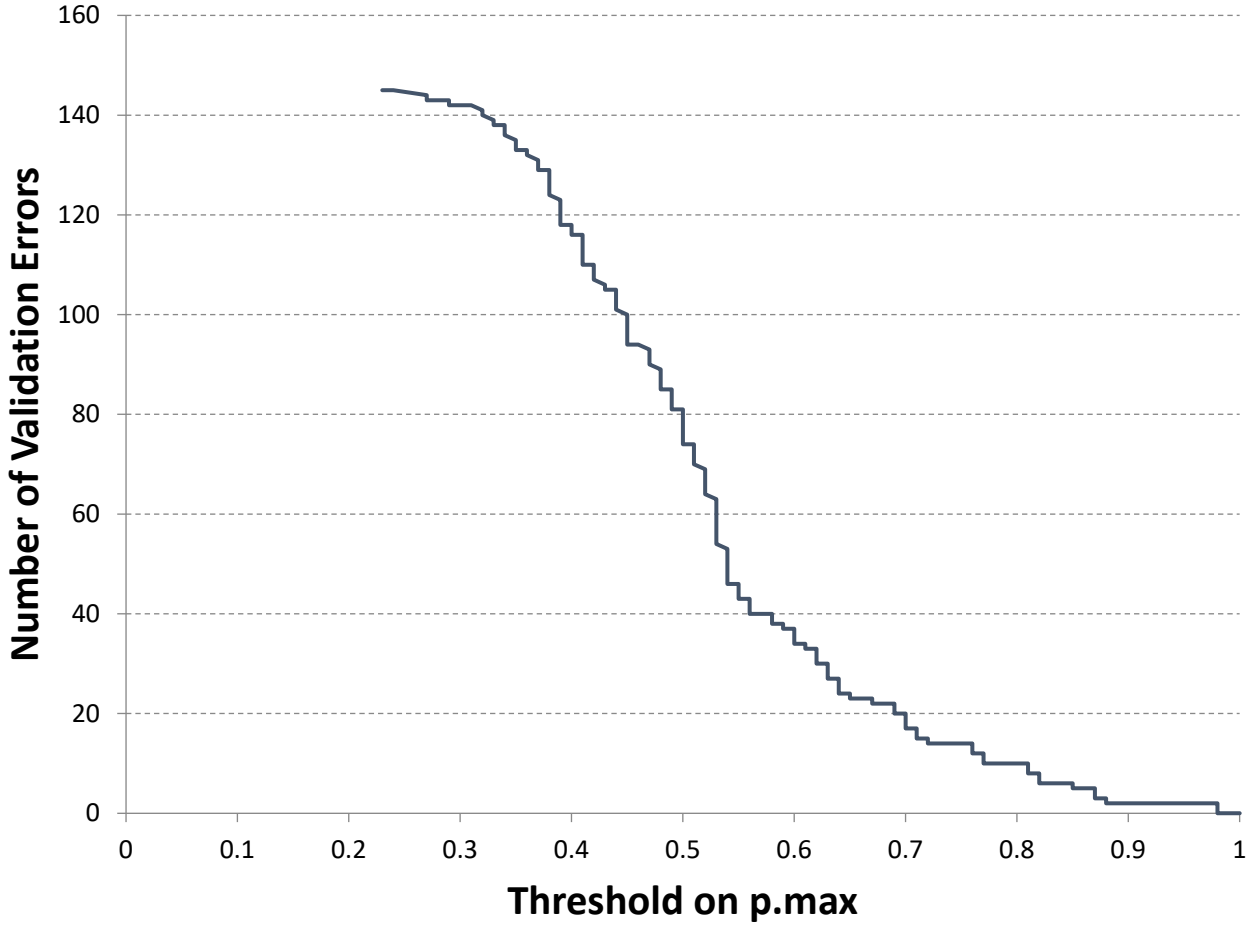


Pendigits Results

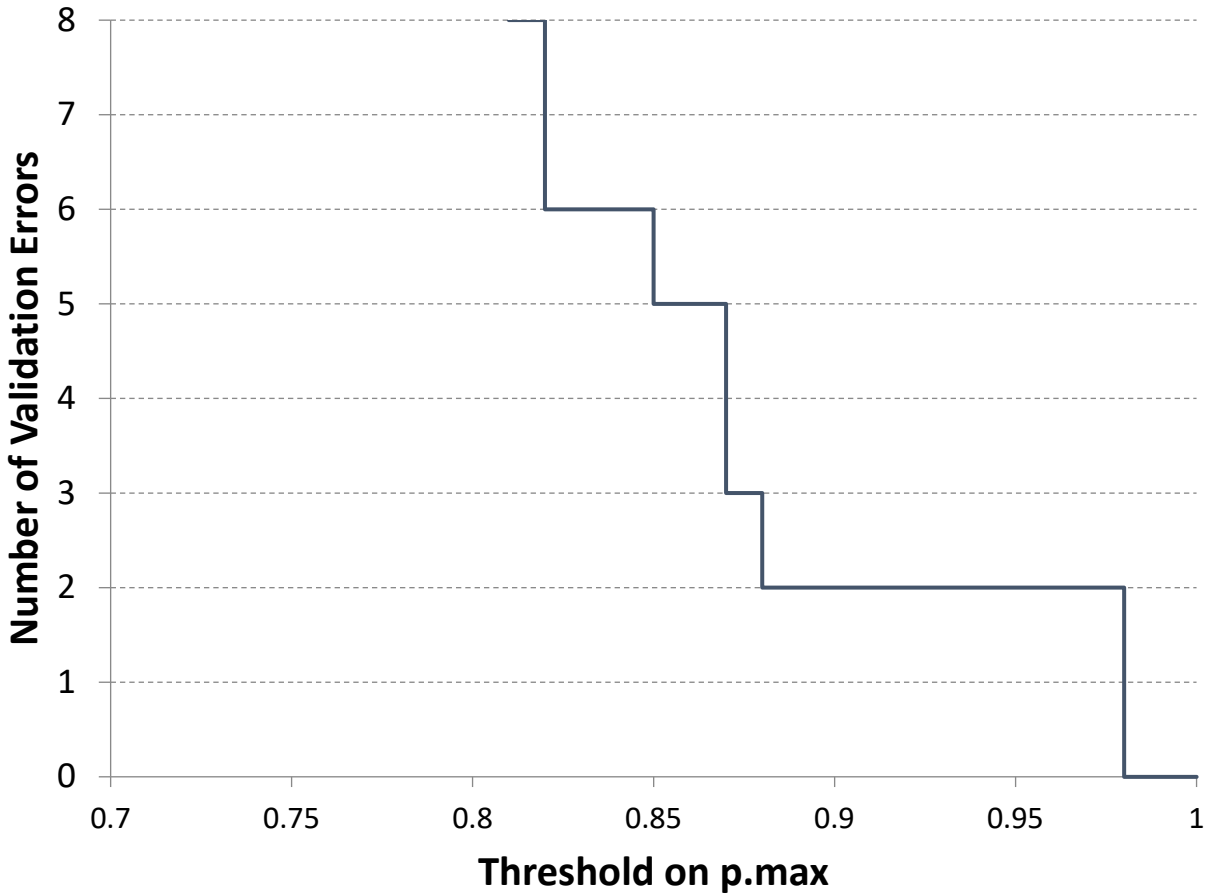
- All τ values were 0 (for $\epsilon = 0.001$)
- Probability $y \in \mathrm{C}(x) = 0.9997$
- Abstention rate = 0.72
- Sizes of prediction sets C:



Simple Thresholding of $\max_k \hat{p}(y = k|x)$



Zoomed In: $\tau = 0.87$ for $\delta = 0.05$



Test Set Results

- Probability of correct classification: 0.9987
- Rejection rate: 33.4%
  - [Conformal prediction was 72%]

Another Use Case: Vocabulary Reduction

- US Postal Service Address Reading Task
  - (Madhvanath, Kleinberg, Govindaraju, 1997)
- Two classifiers
  - Method 1: Fast but not always accurate
  - Method 2: Slower but more accurate
    - Can only afford to run on 1/3 of envelopes
    - Faster if it can be focused on a subset of the classes
- Apply conformal prediction using Method 1
  - Eliminate as many classes as possible
  - Apply Method 2 if $|C(x)| > 1$

Summary

- Rejection
  - Method 1: Threshold f with single or multiple thresholds
    - Multiple thresholds requires a change in the SVM methodology
  - Method 2: Learn a separate rejection function and threshold it
  - Method 3: Conformal: Use thresholding to construct a *confidence set*
    - Reject if $\left|C\left(x_q\right)\right| \neq 1$
    - Can perform "vocabulary reduction"
  - In my experience, Conformal Prediction is not good for Rejection, but more experiments are needed

Citations

- Bartlett, P., Wegkamp, M. (2008). Classification with a reject option using a hinge loss. JMLR, 2008.
- Chow (1970). On optimum recognition error and reject trade-off. IEEE Transactions on Computing.
- Cortes, C., DeSalvo, G., & Mohri, M. (2016). Learning with rejection. *Lecture Notes in Artificial Intelligence*, *9925 LNAI*, 67–82. http://doi.org/10.1007/978-3-319-46379-7_5
- Geifman, Y., El-Yaniv, R. (2017) Selective Classification for Deep Neural Networks. NIPS 2017. arXiv: 1705.08500
- Geifman, Y., & El-Yaniv, R. (2019). SelectiveNet: A deep neural network with an integrated reject option. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 3768–3776.
- Herbei, R., Wegkamp, M. (2005). Classification with reject option. Canadian Journal of Statistics.
- Madhvanath, S., Kleinberg, E., Govindaraju, V. (1997). Empirical Design of a Multi-Classifier Thresholding/Control Strategy for Recognition of Handwritten Street Names. *International Journal of Pattern Recognition and Artificial Intelligence, 11(6):*933-946. https://doi.org/10.1142/S0218001497000421
- Papadopoulos, H. (2008). Inductive Conformal Prediction: Theory and Application to Neural Networks. Book chapter. https://www.researchgate.net/publication/221787122_Inductive_Conformal_Prediction_Theory_and_Application_to_Neural_Networks
- Pietraszek, T. (2005). Optimizing abstaining classifiers using ROC analysis. In ICML, 2005
- Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. Journal of Machine Learning Research, 9, 371–421. Retrieved from http://arxiv.org/abs/0706.3188
- Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer.