Selective Classification for Deep Neural Networks

Yonatan Geifman, Ran El-Yaniv

Motivation

- Would you let a 95% accuracy CNN examine your MRI scans?
- Would you let a 54% accuracy classifier invest your funds?
- Would you fall a sleep in a 99% accurate autonomous car?

Knowledge

Knowns

Unknowns

Knowledge

Known knowns

Known unknowns

Unknown unknowns

Statistical Learning

- Underlying unknown distribution $P(X, Y)$
- A labeled set $S_m = \{(x, y)\}^m \sim P$
- Our goal is to find $f \in \mathcal{F}$ that minimizes the risk:

$$R(f) \triangleq E_P[\ell(f(x), y)]$$

Selective Classification

- Selective Classifier is a pair (f, g)

$$(f,g)(x) = \begin{cases} f(x), & \text{if } g(x) = 1; \\ \text{don't know}, & \text{if } g(x) = 0. \end{cases}$$

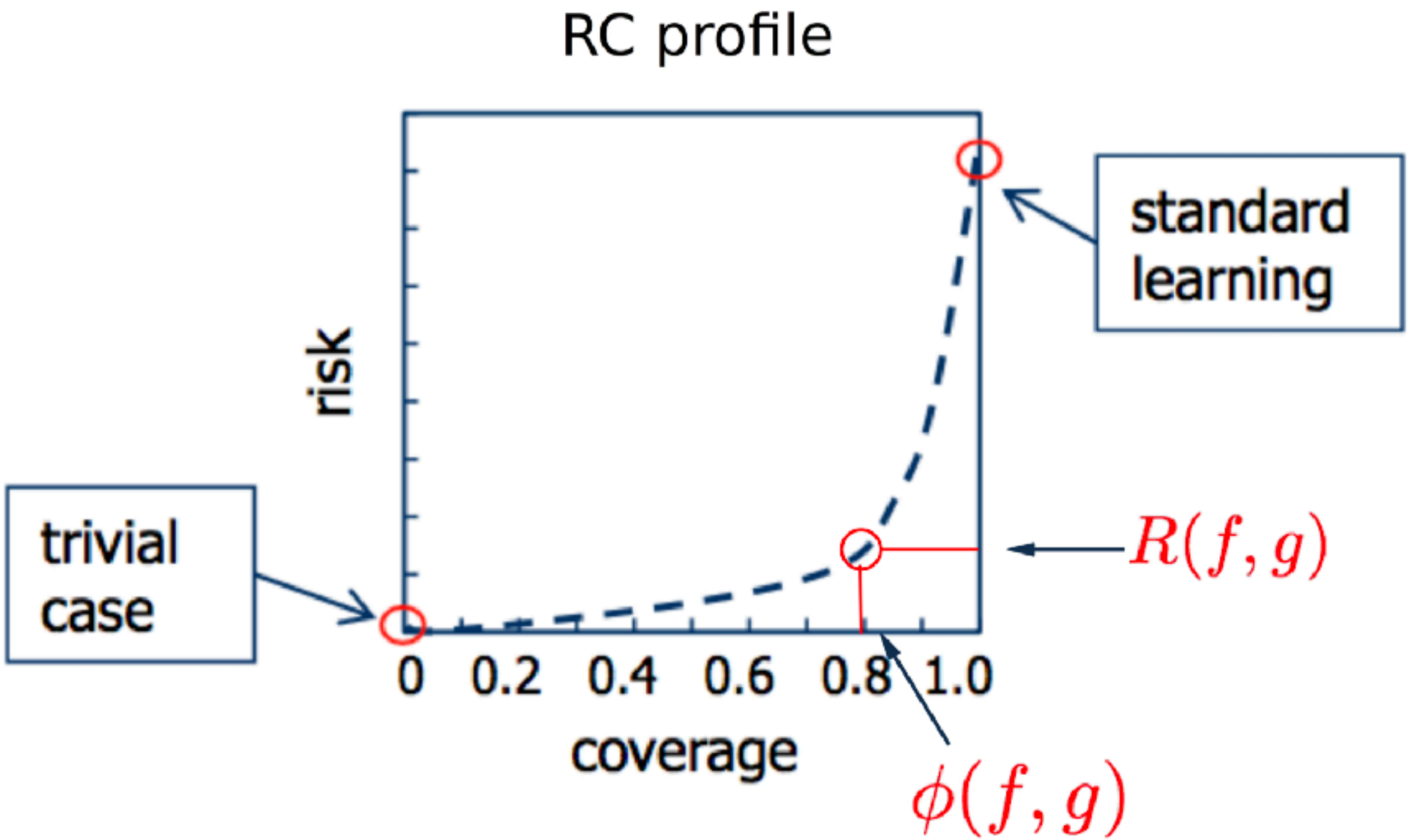
- Coverage:

$$\phi(f,g) \triangleq E_P[g(x)]$$

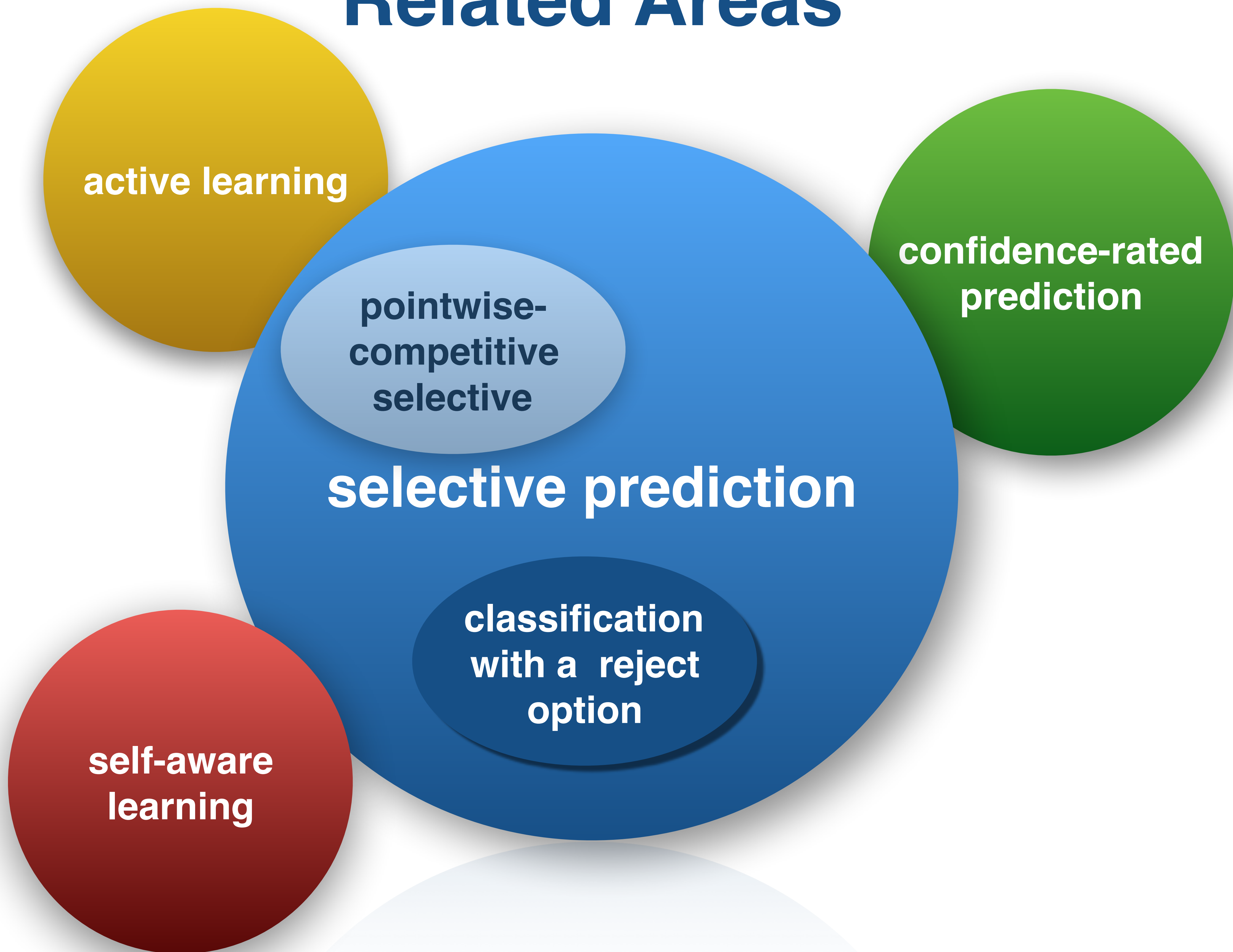
- Risk: $$R(f,g) \triangleq \frac{E_P[\ell(f(x),y)g(x)]}{\phi(f,g)}.$$

Ran El-Yaniv, and Yair Wiener. "On the foundations of noise-free selective classification."

Selective Classification

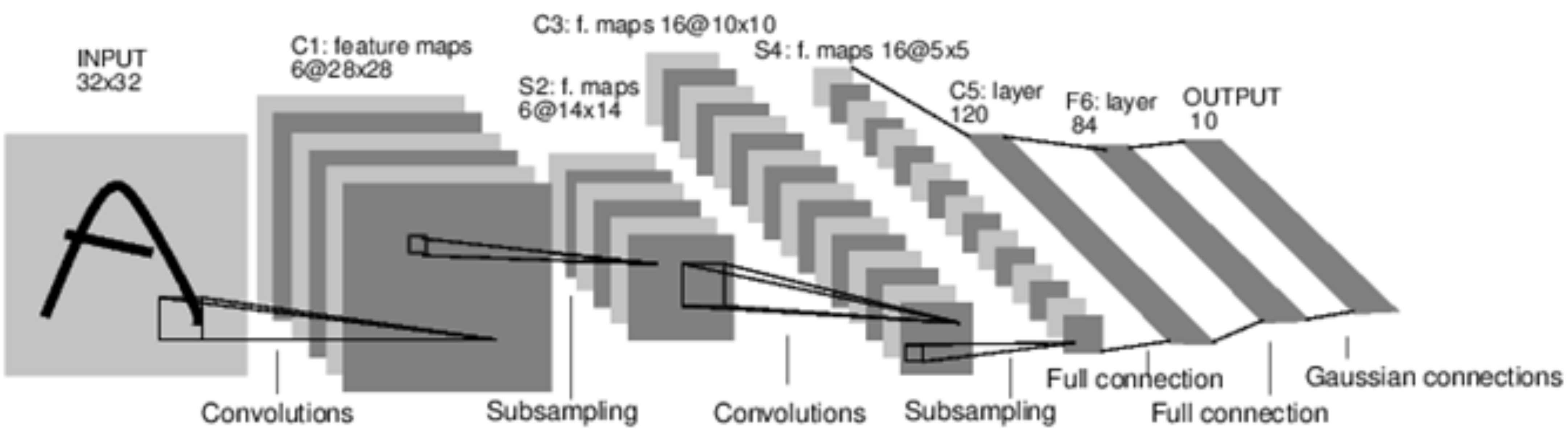


Related Areas



Deep Neural Networks

- Multiple layers of processing units
- Feature representations learned at each layer
- Low level features to high level
- In this work we focus on convolutional neural networks



A Full Convolutional Neural Network (LeNet)

Confidence Rate Functions

- For a classifier f ,We seek for a confidence rate function κ_f that reflects loss monotonicity

$$\kappa_f(x_1) \leq \kappa_f(x_2) \quad \Longleftrightarrow \quad \ell(f(x_1), y_1) \geq \ell(f(x_2), y_2)$$

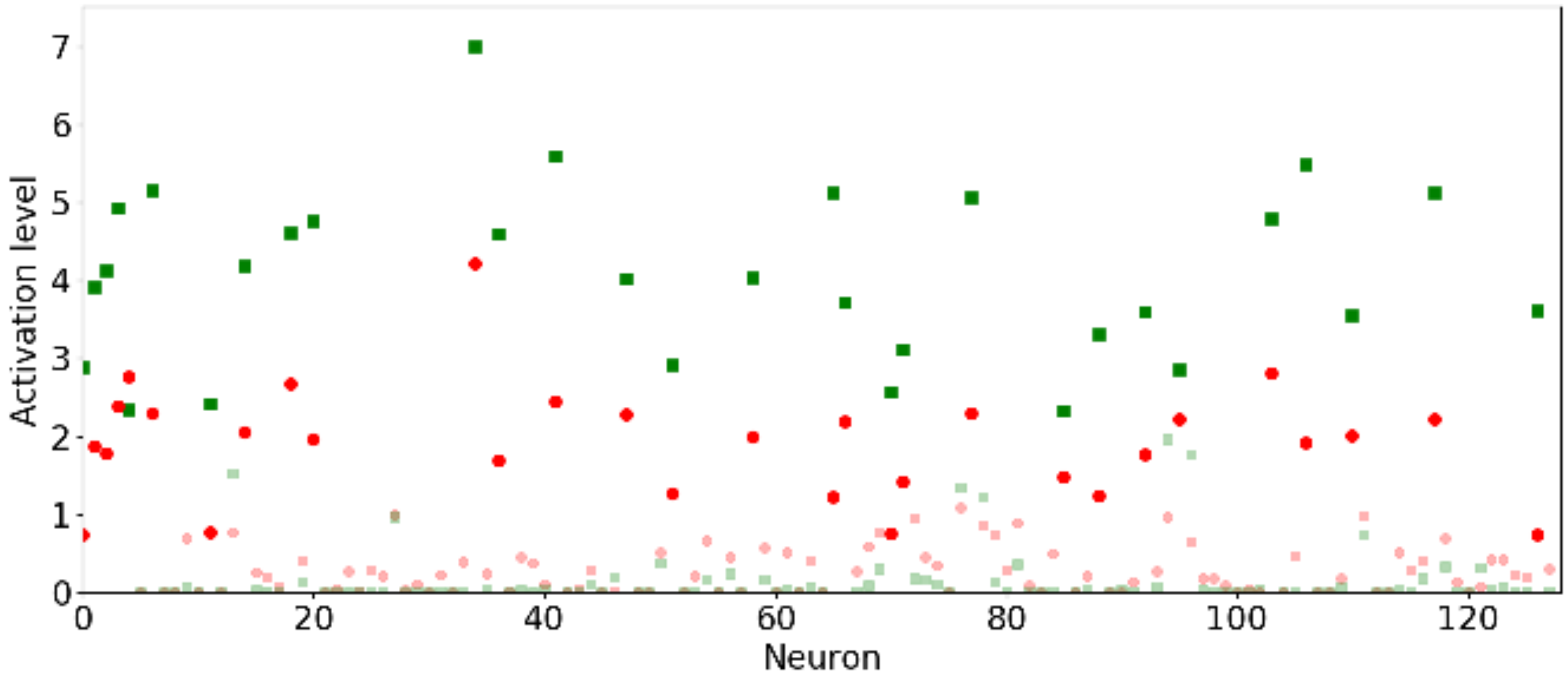
- We discuss two candidates:
  - SOFTMAX response
  - MC-Dropout

Confidence - Softmax Response

- Simply take κ to be the Softmax output

$$\kappa_f \triangleq \max_{j \in \mathcal{Y}} (f(x|j))$$

- Motivation - MNIST activations:



Confidence - MC-Dropout

- Apply dropout at inference
- Estimate prediction variance over numerous (100) forward passes with dropout (p=0.5)
- Intuition - kind of ensemble variance

Gal, Yarin, and Zoubin Ghahramani. "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning."

Selection with Guaranteed Risk (SGR)

- A selective classifier obtained by thresholding the confidence rate function

$$g_\theta(x) \triangleq \begin{cases} 1, & \text{if } \kappa_f(x) \geq \theta; \\ 0, & \text{otherwise.} \end{cases}$$

- Given a training set S_m, a desired risk r^*, and a confidence parameter δ, our goal is to learn a selective classifier such that:

$$Pr_{S_m}\{R(f,g) > r^*\} < \delta$$

Lemma 1 - Binomial Tail

- Let $B^*(\hat{r}_i, \delta, S_m)$ be the solution b of the following equation

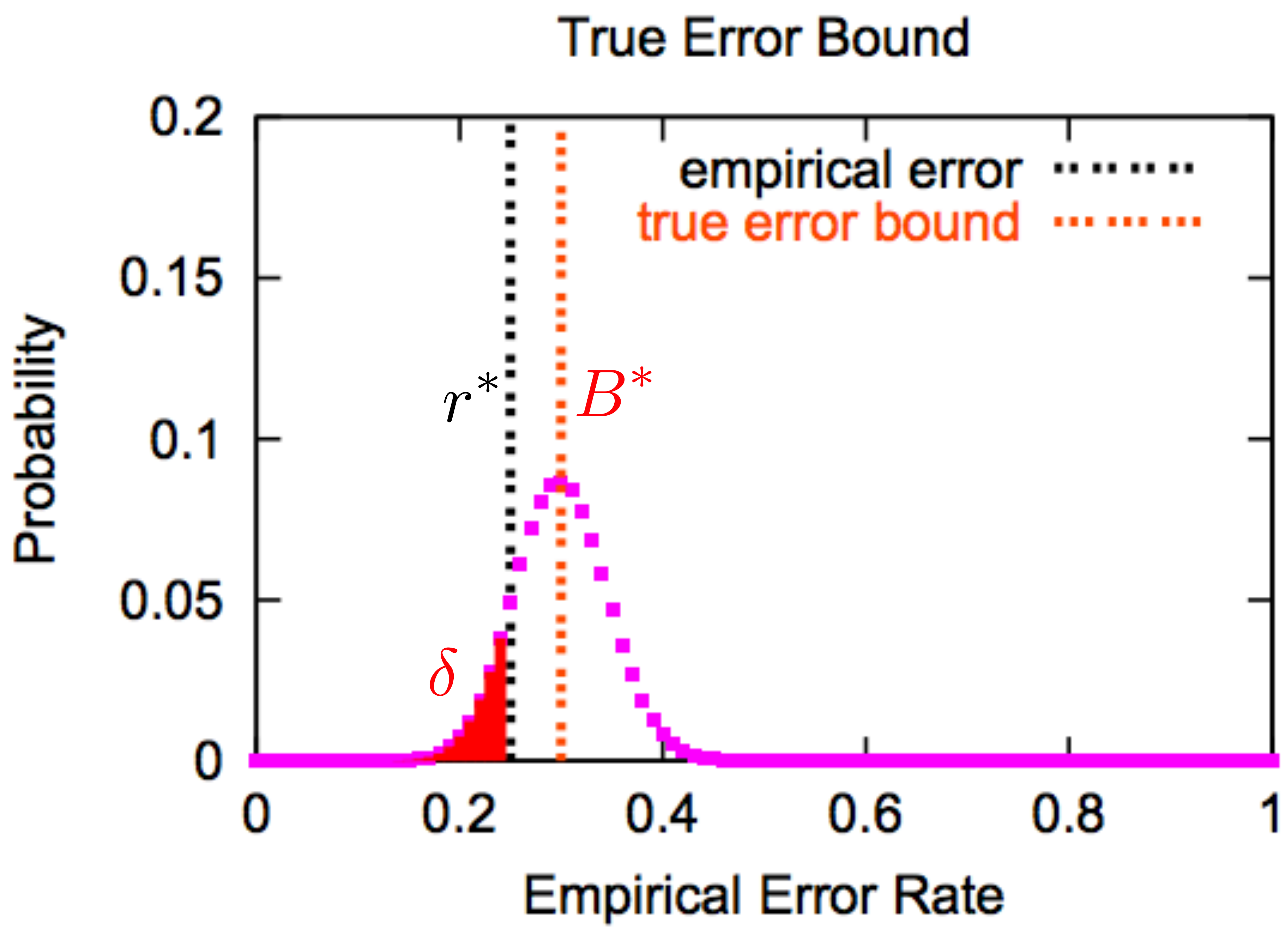
$$\sum_{j=0}^{m \cdot \hat{r}(f|S_m)} \binom{m}{j} b^j (1-b)^{m-j} = \delta.$$

Then

$$Pr_{S_m}\{R(f|P) > B^*(\hat{r}_i, \delta, S_m)\} < \delta$$

O. Gascuel and G. Caraux. Distribution-free performance bounds with the resubstitution error estimate.

Lemma 1 - Binomial Tail



Langford, John. "Tutorial on practical prediction theory for classification."

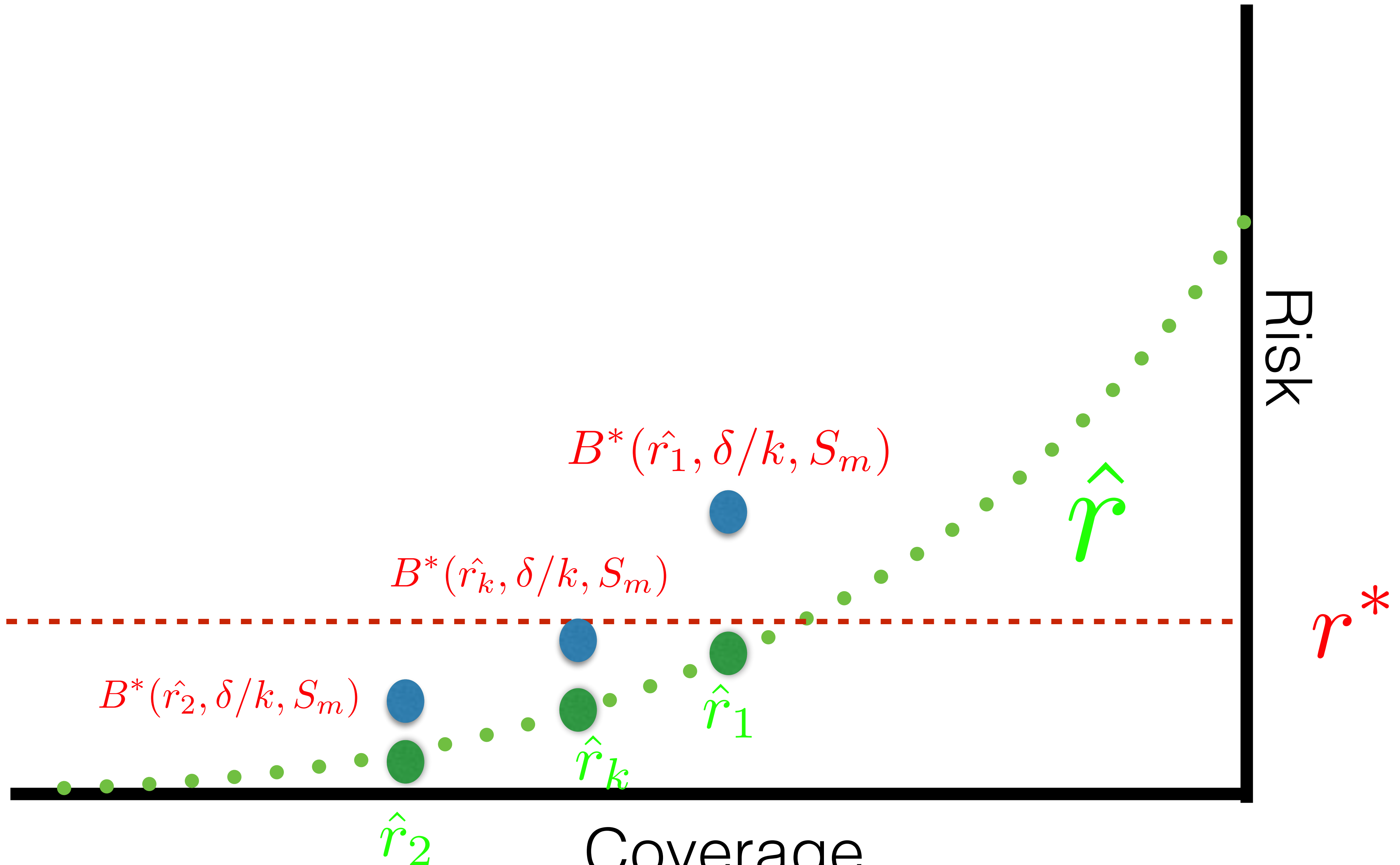
SGR Algorithm

- For a given training set $S_m \sim P(X,Y)$, a desired risk r^* and a confidence parameter δ

- set $k = \lceil \log(m) \rceil$

- Use binary search to find $\hat{\theta} \in \{\kappa(x) : x \in S_m\}$ such that $B^*(\hat{r}_\theta, \delta/k, S_m) \leq r^*$

SGR Algorithm



SGR Algorithm

Algorithm 1 *Selection with Guaranteed Risk* (SGR)

SGR(f,κ_f,δ,r^*,S_m)
Sort S_m according to $\kappa_f(x_i)$, $x_i \in S_m$ (and now assume w.l.o.g. that indices reflect this ordering).
$z_{\min} = 1$; $z_{\max} = m$
for $i = 1$ **to** $k \triangleq \lceil \log_2 m \rceil$ **do**
  $z = \lceil (z_{\min} + z_{\max})/2 \rceil$
  $\theta = \kappa_f(x_z)$
  $g_i = g_\theta$ {(see (3))}
  $\hat{r}_i = \hat{r}(f, g_i | S_m)$
  $b_i^* = B^*(\hat{r}_i, \delta / \lceil \log_2 m \rceil, g_i(S_m))$ {see Lemma 3.1 }
  if $b_i^* < r^*$ **then**
    $z_{\max} = z$
  else
    $z_{\min} = z$
  end if
end for
Output- (f, g_k) and the bound b_k^*.

Theorem 1 - SGR Generalization bound

Theorem: For an application of SGR on $S_m \sim P(X,Y)$ with a given r^* and δ, the output (f, g_k) Satisfies

$$Pr_{S_m}\{R(f,g) > r^*\} < \delta$$

Theorem 1 - SGR Generalization bound - Proof Sketch

- On each iteration

$$Pr_{S_m}\{R(f, g_i) > B^*(\hat{r}_i, \delta, S_m)\} < \delta/k$$

- Due to the binary search

$$\exists i : B^*(\hat{r}_i, \delta, S_m) \leq r^*$$

- An application of the union bound among iterations complete the proof

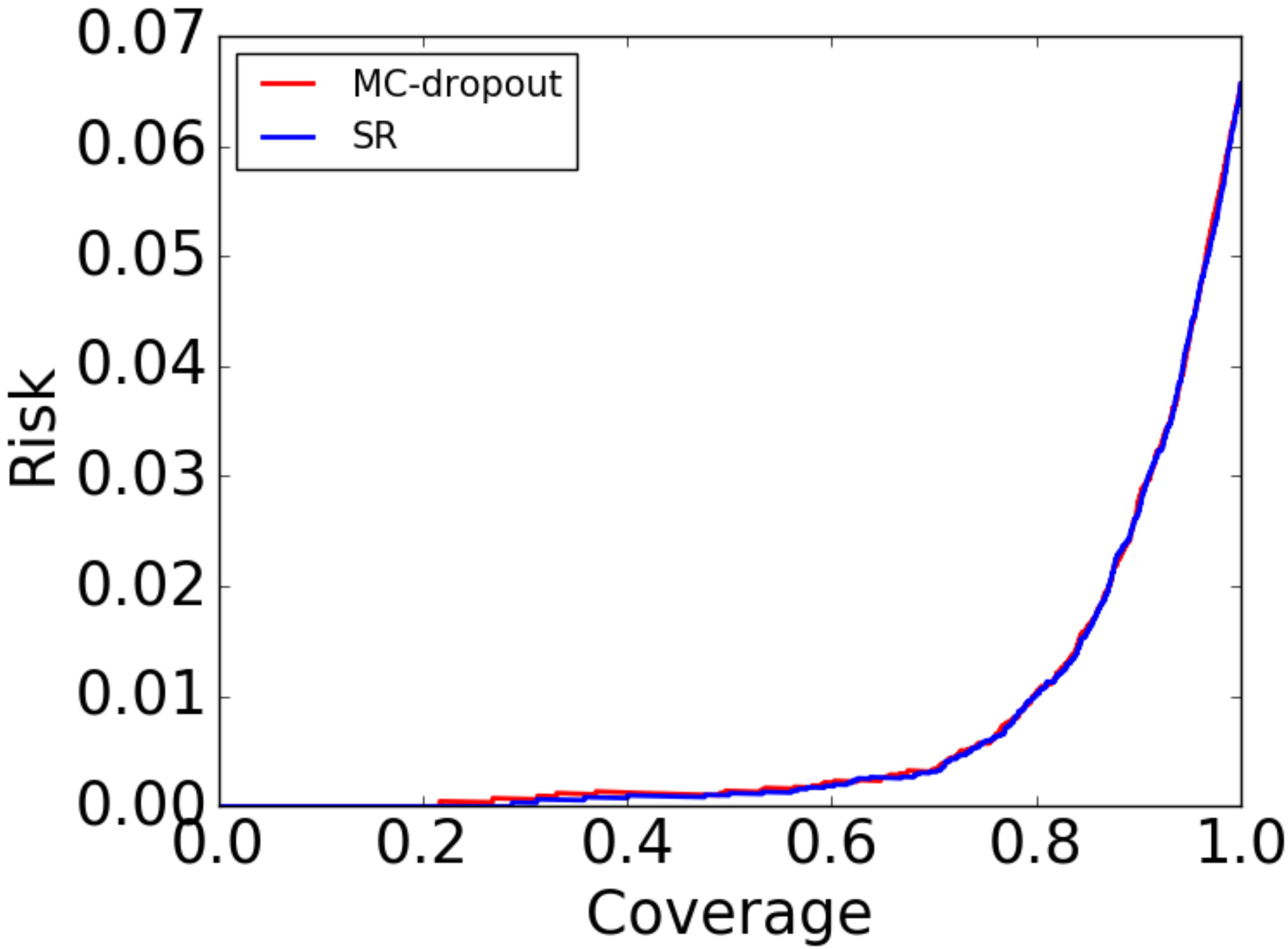
SGR Algorithm

- A generalization bound for DNNs
- The tightest bound possible
- Can work on a pre-trained network
-

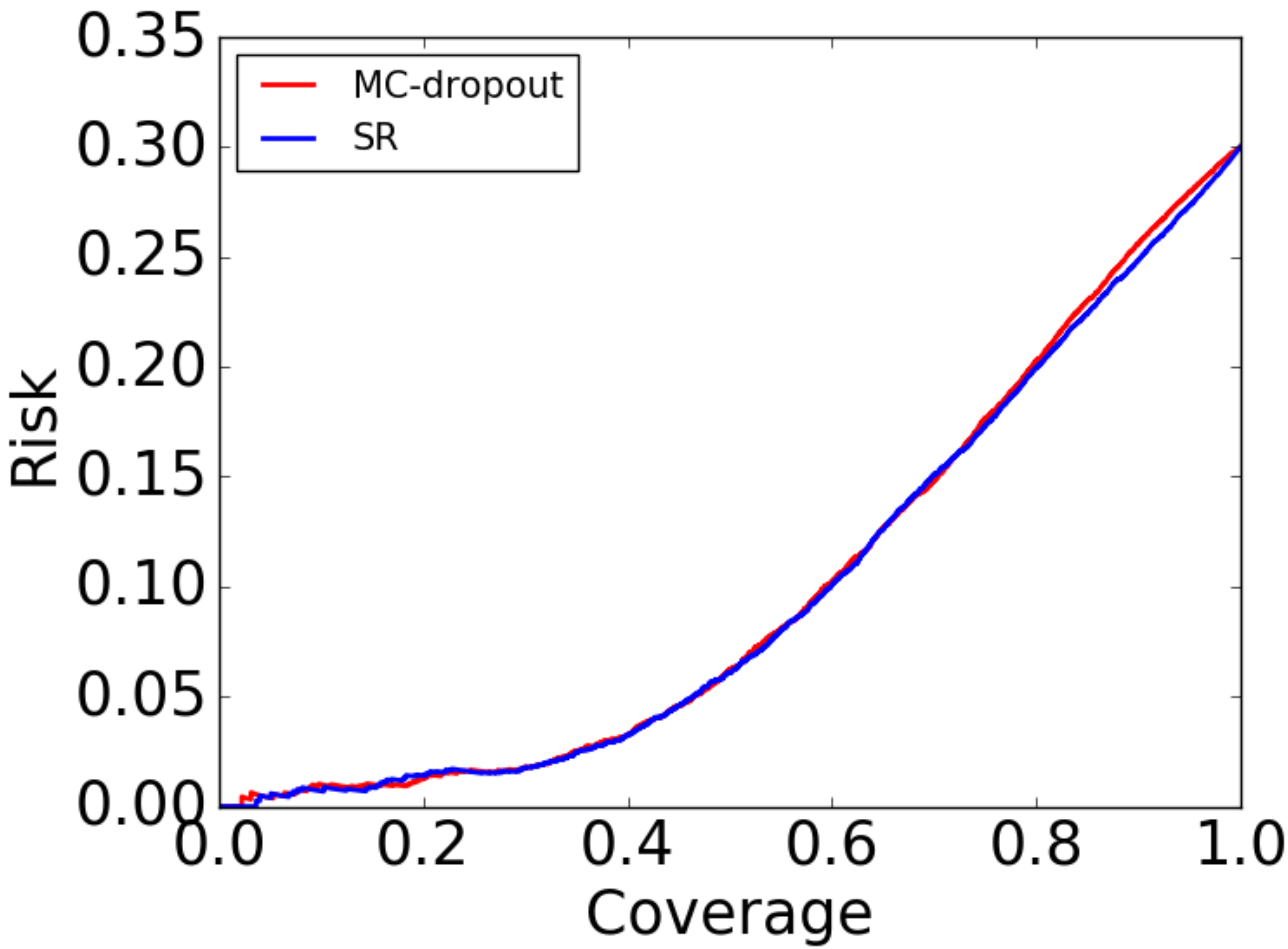
Experimental Setting

- Datasets:
    - CIFAR-10 - VGG-16
    - CIFAR-100 - VGG-16
    - IMAGENET - VGG-16 + Resnet-50 (top1 and top 5)

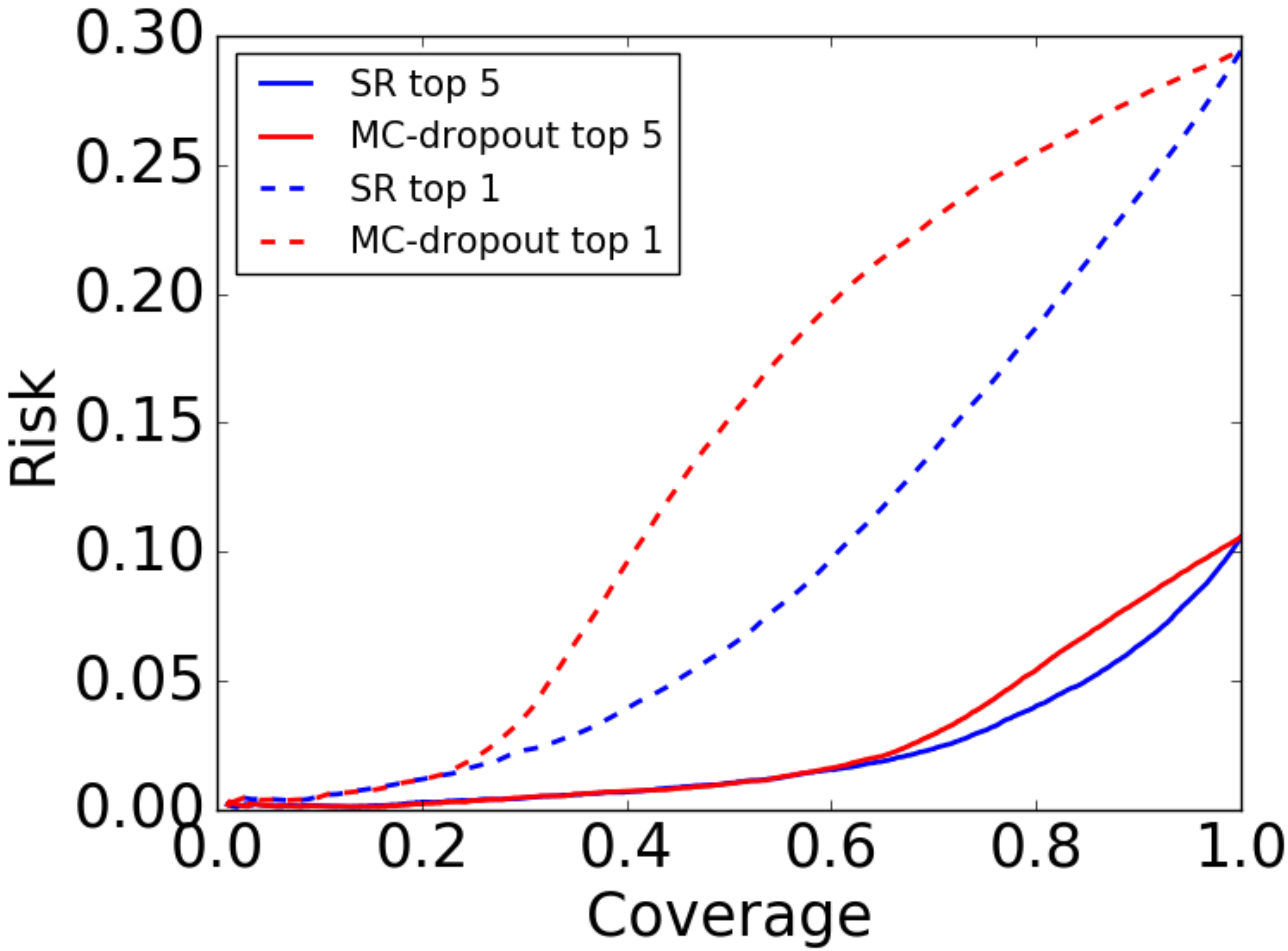
Experiments - RC-curve - CIFAR-10



Experiments - RC-curve - Cifar-100



Experiments - RC-curve Imagenet



Experiments - SGR

- CIFAR-10 - VGG-16

Desired risk (r^*)	Train risk	Train coverage	Test risk	Test coverage	Risk bound (b^*)
0.01	0.0079	0.7822	0.0092	0.7856	0.0099
0.02	0.0160	0.8482	0.0149	0.8466	0.0199
0.03	0.0260	0.8988	0.0261	0.8966	0.0298
0.04	0.0362	0.9348	0.0380	0.9318	0.0399
0.05	0.0454	0.9610	0.0486	0.9596	0.0491
0.06	0.0526	0.9778	0.0572	0.9784	0.0600

- IMAGENET - top 5 with Resnet-50

Desired risk (r^*)	Train risk	Train coverage	Test risk	Test coverage	Risk bound(b^*)
0.01	0.0080	0.3796	0.0085	0.3807	0.0099
0.02	0.0181	0.5938	0.0189	0.5935	0.0200
0.03	0.0281	0.7122	0.0273	0.7096	0.0300
0.04	0.0381	0.8180	0.0358	0.8158	0.0400
0.05	0.0481	0.8856	0.0464	0.8846	0.0500
0.06	0.0581	0.9256	0.0552	0.9231	0.0600
0.07	0.0663	0.9508	0.0629	0.9484	0.0700

Future Work

- Optimal confidence rate function
- Selection with Guaranteed coverage
- Neyman-Pearson selective classification
- Learning the pair (f, g) together

Questions?