Machine Learning Methods for Robust Artificial Intelligence

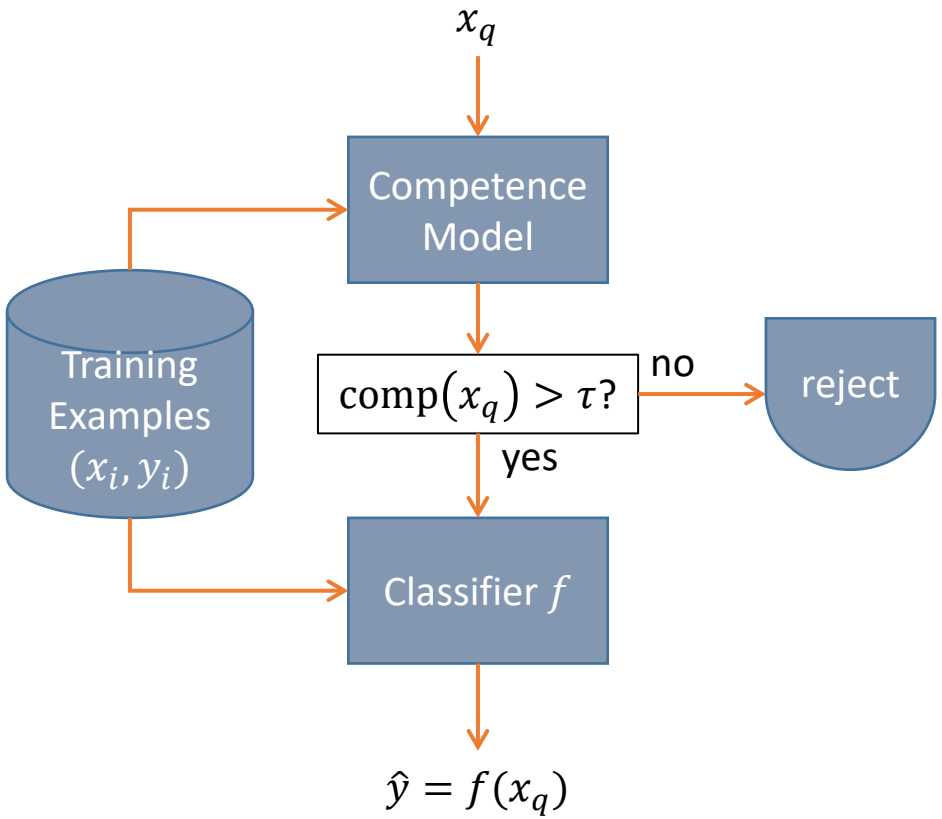
Thomas G. Dietterich

Distinguished Professor (Emeritus)

Oregon State University

Goal: Trustworthy Classifiers

- Every classifier should also have a model of its own competence

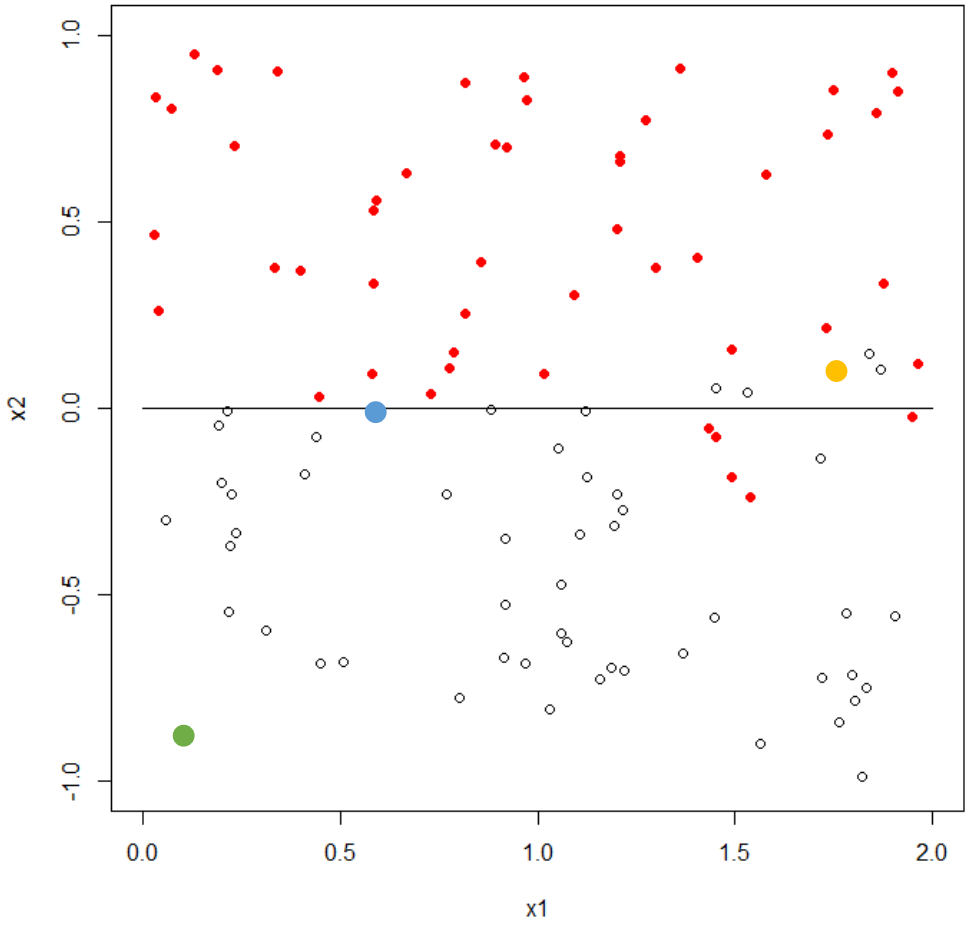


Threats to Competence

- x_q is near a decision boundary (the features of x_q are ambiguous)
- x_q is in a region with high labeling noise
- x_q is in a region with little training data

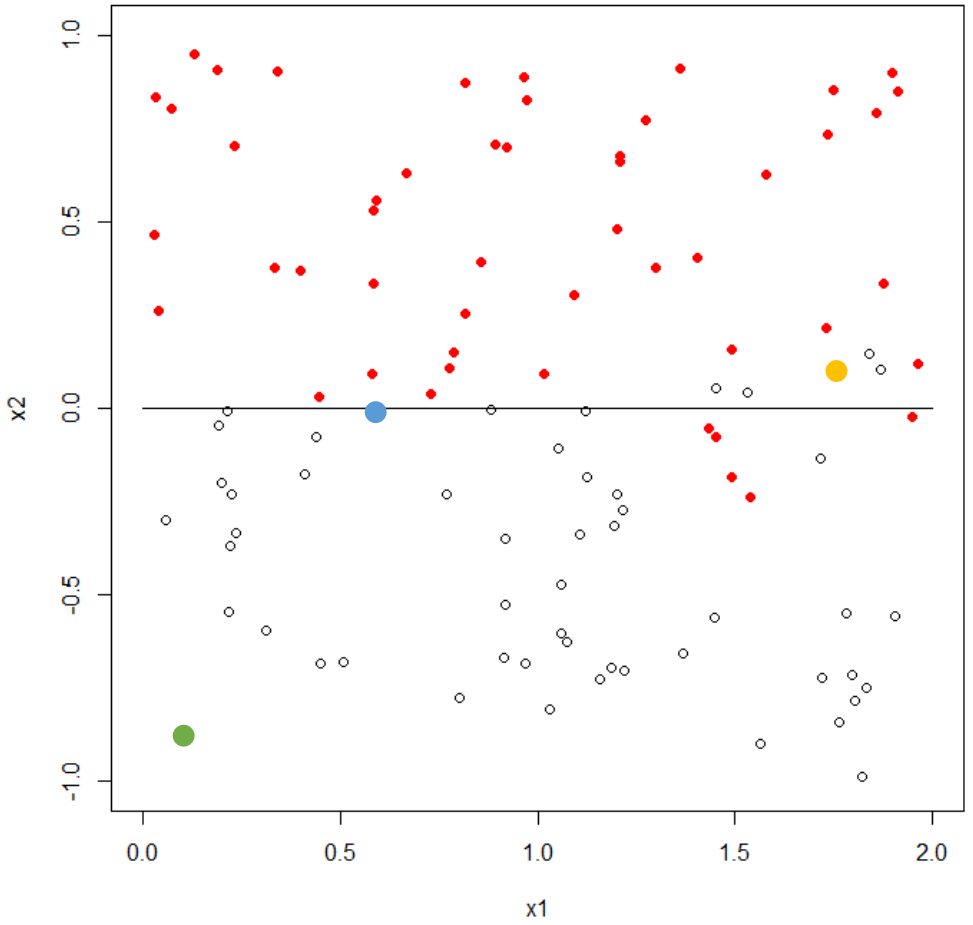
  x_q belongs to a class that was not present in the training data: “novel category problem”

Traditional classifiers will make predictions in *all* these cases



Approaches

- Calibrated probabilities
  - Well-calibrated estimates $\hat{p}(y|x)$ of $P(y|x)$
    - $\text{comp}(x) = \max_k \hat{p}(y = k|x)$
  - Should output 0.5 for the ● and ● cases
- Competence function (rejection function)
  - $\text{comp}(x) > \tau$
- Outlier/Anomaly/Out-of-Distribution detector
  - Should detect the ● case
- Prediction sets
  - Output $R(x_q)$ and a guarantee that $y \in R(x_q)$



Other threats not covered in this class

- Distribution shift / domain shift
  - We will assume independent and identically-distributed data, but with the possibility that runtime data includes novel classes
- Adversarial examples
  - We will not consider data points deliberately modified to fool the classifier

Related Approaches Not Covered

- Batch/group detection
  - We will focus on the setting where we have only one query x_q and we must decide whether the classifier is competent to classify x_q
  - In some applications, we are given a batch of queries $\{x_q^1, \ldots, x_q^B\}$, and we can combine evidence to decide whether the classifier is competent to classify the whole batch. This is particularly important when detecting distribution shifts.
- Structured data: time series, spatial data, text, graphs, etc.

Notation

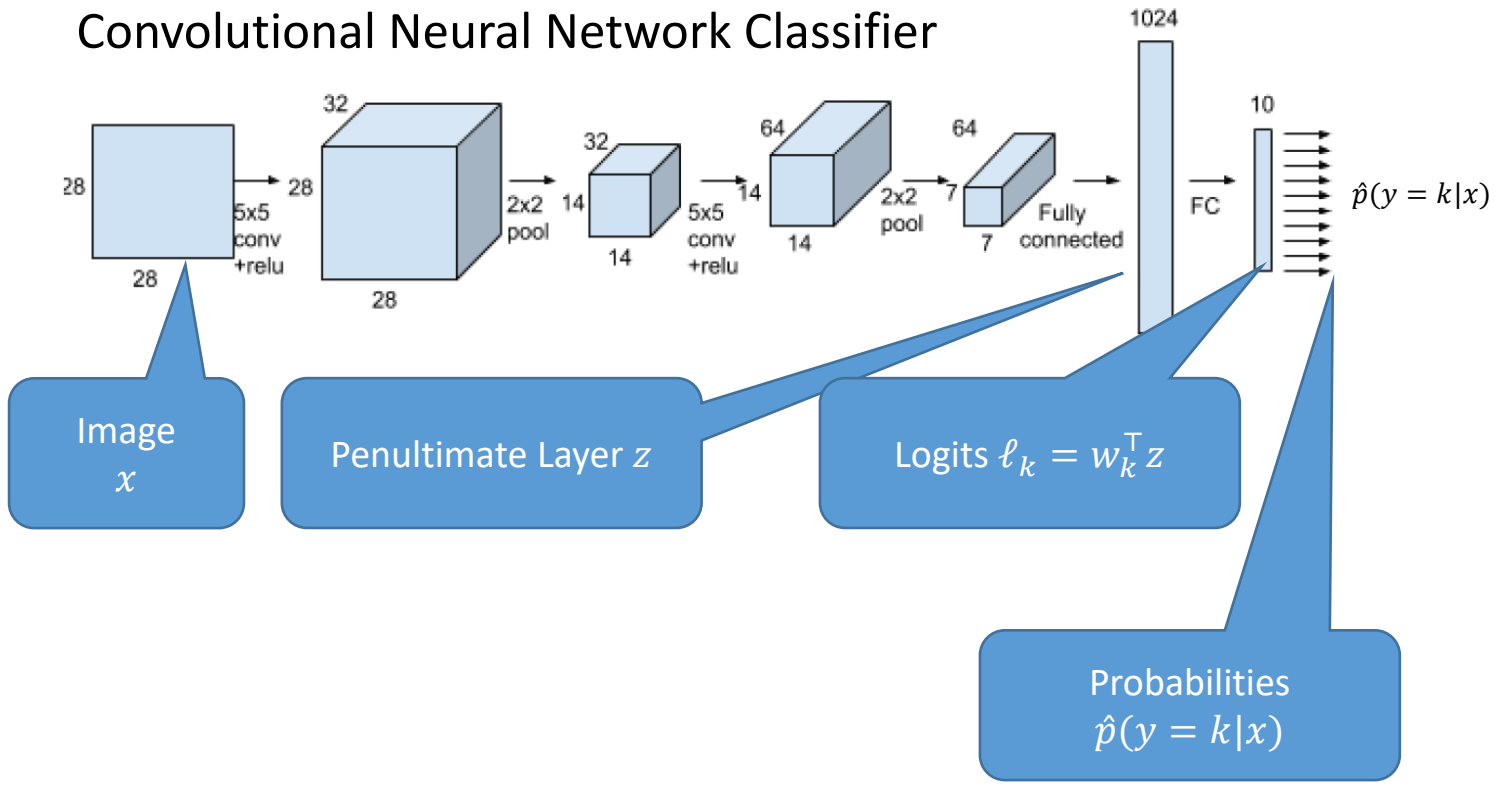
- Input space $\mathcal{X}$ of dimension d
- Output space $\mathcal{Y} = \{1, \ldots, K\}$ classes
- True joint distribution $P(x, y) = P(x)P(y|x)$
- Training data $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ drawn from $P(x, y)$
- Fitted function $f: \mathcal{X} \mapsto \Delta^{K-1}$ the K-dimensional probability simplex
- $f(x) = [\hat{p}(y = 1|x), \ldots, \hat{p}(y = K|x)]$ class probability vector
- $\hat{y} = \arg\max_k \hat{p}(y = k|x)$ predicted class

- $\mathbb{I}[u]$ is 1 if u is true and 0 otherwise
- Some classifiers do not output probabilities (e.g., SVMs), but we will ignore this in our notation

Notation for Deep Networks

- Input image x
- Network backbone, also called the "encoder": $z = E(x)$
- Latent representation z
- Logits $\ell_k = w_k^\top z$
- Predicted probabilities

$$\hat{p}(y = k|x) = \frac{\exp \ell_k(z)}{\sum_{k'} \exp \ell_{k'}(z)}$$



Calibrated Classifiers

- A classifier is well-calibrated if the output probability $\hat{p}(y = k|x_q)$ is equal to the true conditional probability $P\left(y = k\middle|x_q\right)$

Part 1: Calibrated Probabilities

- Reasons for Creating Calibrated Probabilities
- Reason 1: Rational Decision Making
  - If $L(k, k')$ is the loss received if $y = k$, then the expected loss of predicting k' is
    - $\sum_k P(y = k|x)L(k, k')$
  - We can choose k' to minimize this expected loss
    - $\hat{k} = \arg\max_{k'} \sum_k P(y = k|x)L(k, k')$
  - We can consider other decisions including abstention. Let $L(k, \text{abstain})$ be the cost of abstaining
    - E.g., Cost of asking a person to make the decision
    - $\hat{a} = \arg\max_{a \in \{1,\ldots,K,\text{abstain}\}} \sum_k P(y = k|x)L(k, a)$

Reason 2: Interpretability

- People can understand a probability statement like $\hat{p}(y = k|x) = 0.8$ better when the probability is well-calibrated

Reason 3: System Integration

- It is easier to integrate multiple AI subsystems if they all work with well-calibrated probabilities
- Examples:
  - Fusing multiple sensors
  - Combining evidence from multiple sources

Reason 4: Improved Accuracy

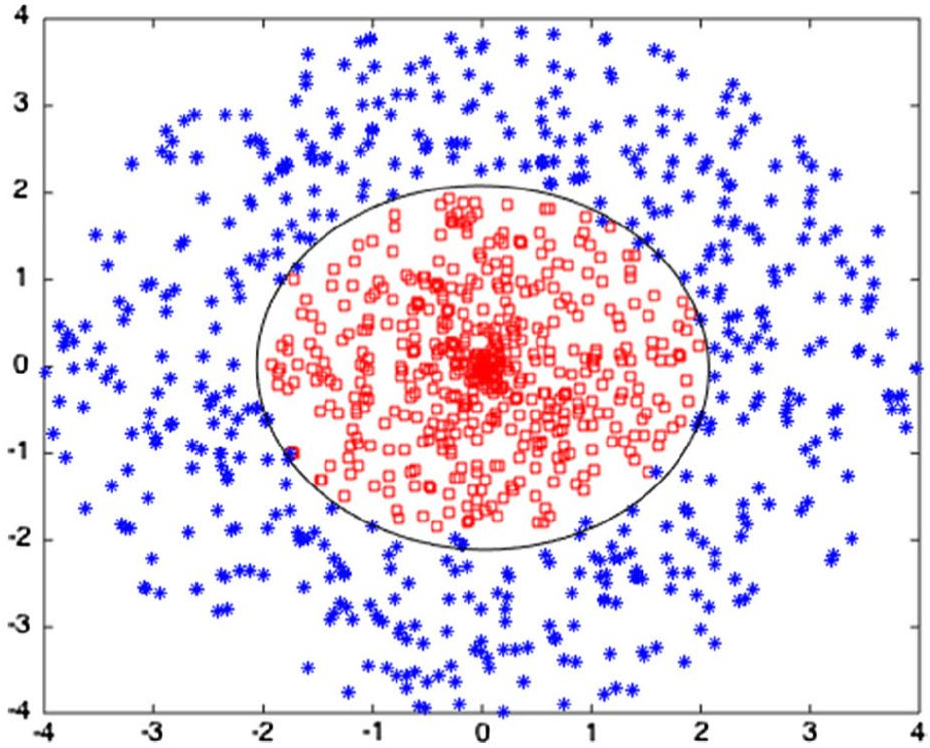


Fig. 3 Scatter plot of the simulated data. The two classes of the binary classification task are indicated by the red squares and blue stars. The black oval indicates the decision boundary found using SVM with a quadratic kernel (colour figure online)

Uncalibrated linear SVM achieves 0.64 accuracy
Calibrated linear SVM achieves 0.79 accuracy

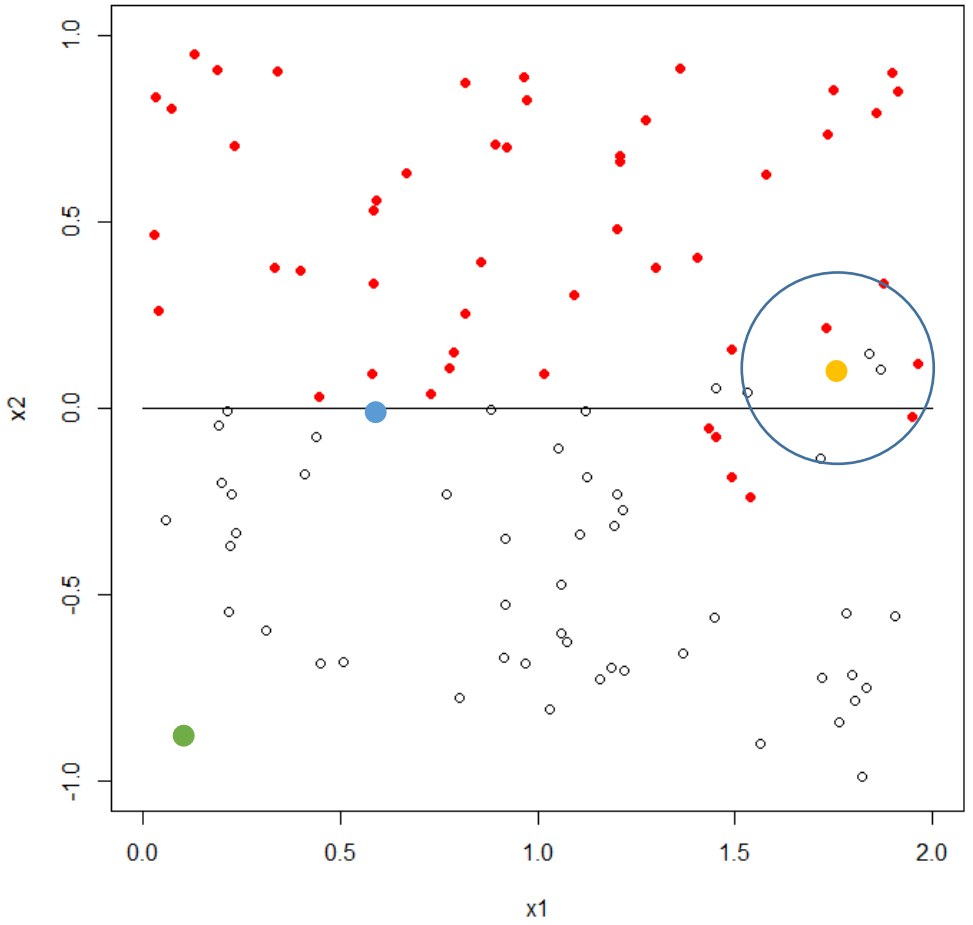
Of course, using a quadratic kernel
gives 0.99 accuracy

Measuring Calibration Error

- We can't measure the true conditional probability at a single point unless we have many training examples identically equal to x_q
- Instead, we must use some *set* of points to create an estimate $\hat{P}(y = k|x_q)$

Estimating $P(y = k|x_q)$

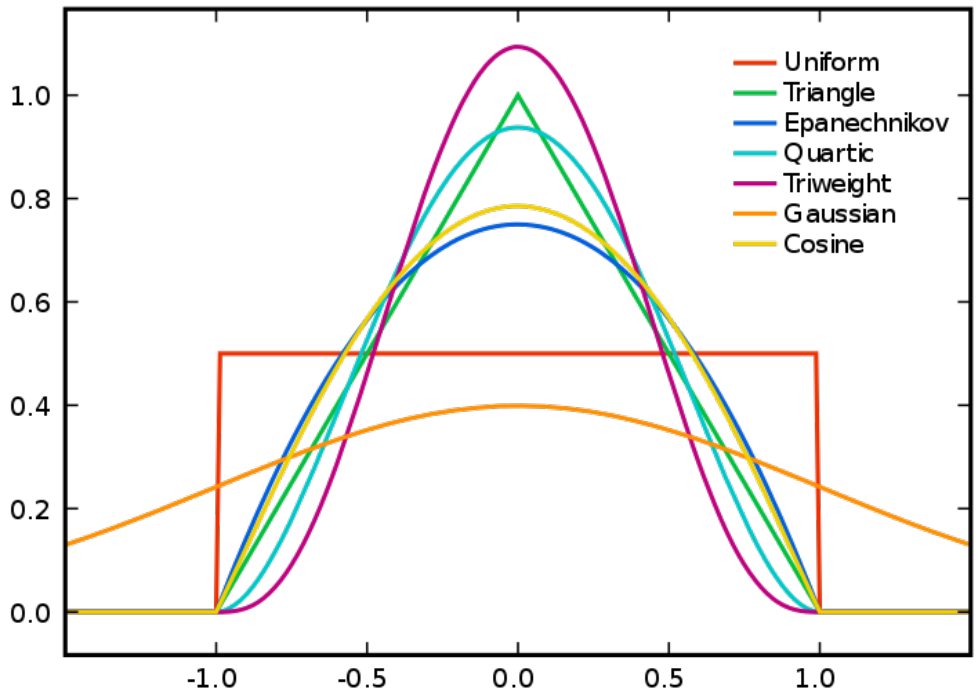
- Method 1: Neighborhood in the input space
  - let $\eta(x_q)$ be the H data points nearest to x_q.
  - $\hat{P}(y = k|x_q) = \frac{|\{(x,y): x \in \eta(x_q) \text{ and } y = k\}|}{|\eta|}$
  - Example: 3 out of $H = 7$ points are class 1 (red), so $\hat{P}(y = 1|x_q) = \frac{3}{7}$
  - Can generalize this to a similarity kernel $k(x, x')$
  - $\hat{P}(y = k|x_q) = \frac{\sum_i k(x_i, x_q)\mathbb{I}(y_i = k)}{\sum_i k(x_i, x_q)}$



Digression: Kernels

- A kernel $k(x, x')$ is a real-valued function that satisfies certain properties. Typical properties include
  - $0 \leq k(x, x') \leq 1 \;\; \forall x, x'$
  - $k(x, x) = 1$ self-similarity is maximum
  - $k(x, x') = k(x', x)$ symmetric
  - $k(x, x') \to 0$ as $\|x - x'\| \to \infty$
- Different kernels satisfy different properties
- Radial kernels
  - $k(x, x') = k(\|x - x'\|)$ for some distance $\| \cdot \|$

$$k(x, x') = k(x - x')$$



By Brian Amberg - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=5329895

Estimating $P(y = k|x_q)$

- Method 2: Neighborhood in the predicted probability space
- Let $f(x_q) = \left(\hat{p}(y = 1|x_q), \ldots, \hat{p}(y = K|x_q)\right)$ be the predicted class probabilities
- Let $\eta(x_q)$ be a set of data points for which $\hat{p}(y = k|x_i)$ is close to $\hat{p}(y = k|x_q)$:

$$\eta(x_q) \coloneqq \{x_i : |\hat{p}(y = k|x_i) - \hat{p}(y = |x_q)| \; is \; small\}$$

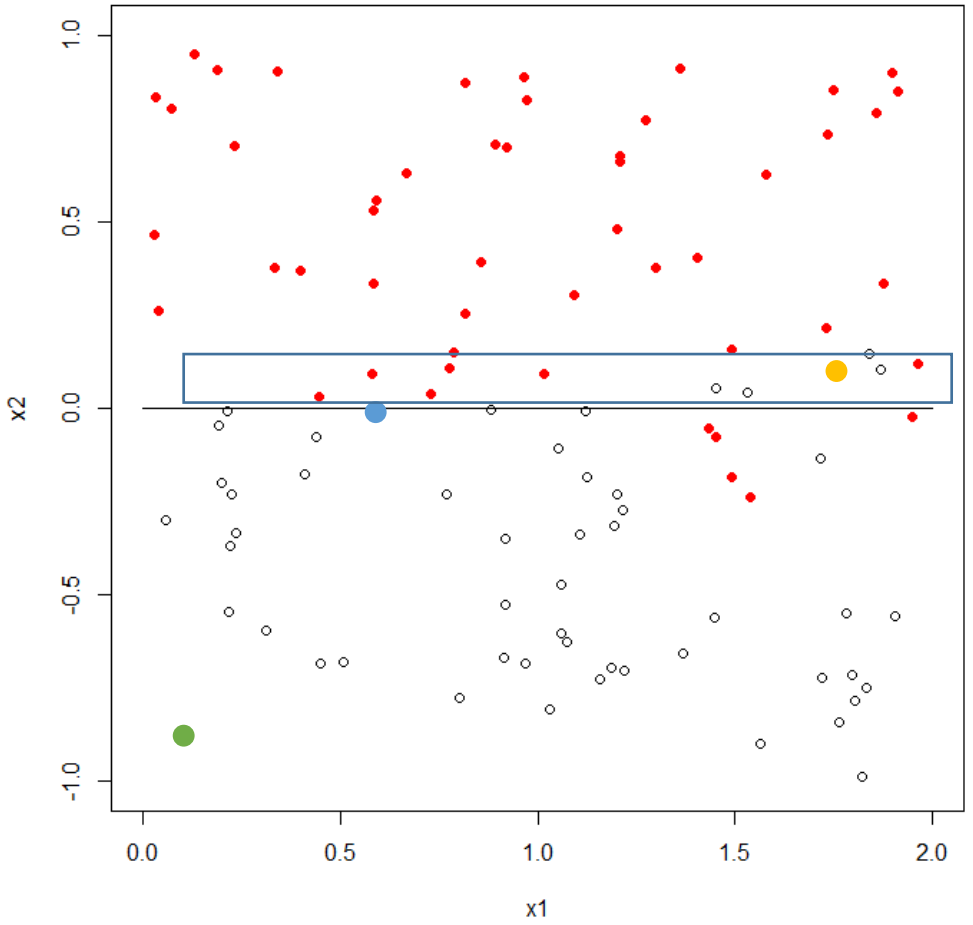
- The estimated $\hat{P}$ is the fraction of those points that belong to class k:

$$\hat{P}(y = k|x_q) = \frac{|\{(x_i, y_i) : x_i \in \eta(x_q) \text{ and } y_i = k\}|}{|\eta|}$$

- Example: 6 out of H =9 points are class 1 (red), so $\hat{P}(y = 1|x_q) = 6/9$

- Can generalize this to a similarity kernel in *predicted probability space:* $k(\hat{p}(y = k|x_q), \hat{p}(y - k|x_i))$

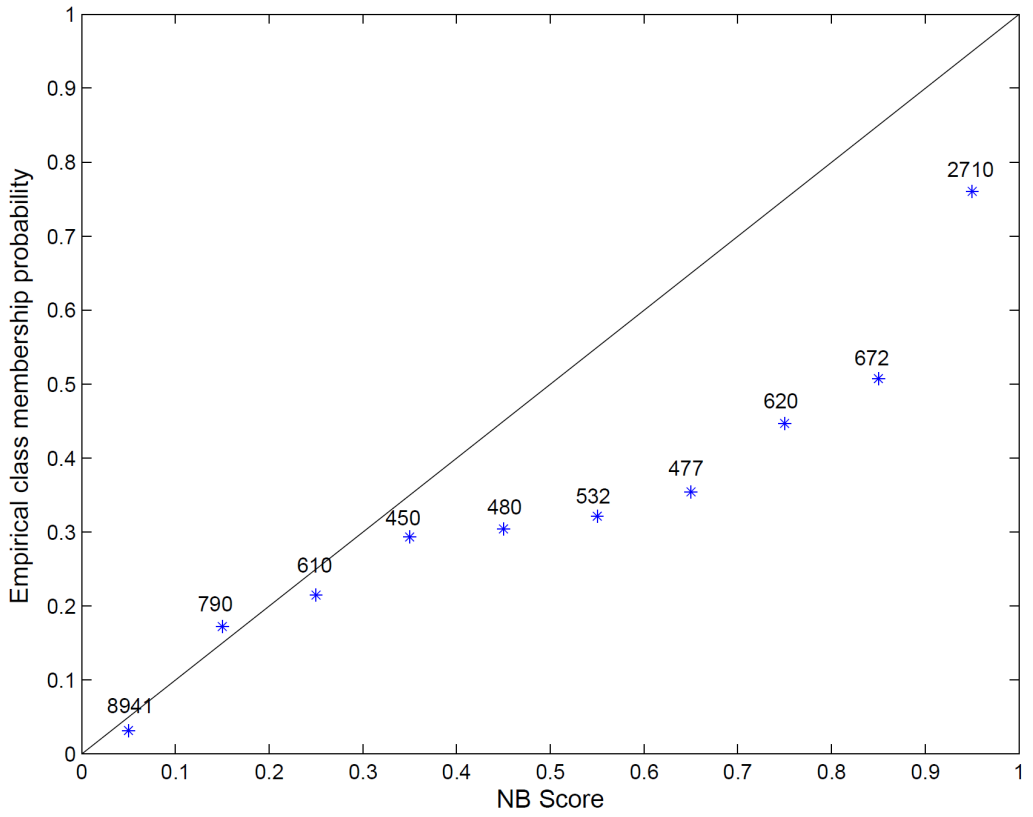
$$\hat{P}(y = k|x_q) = \frac{\sum_i k(\hat{p}(y = k|x_q), \hat{p}(y - k|x_i)) \mathbb{I}[y_i = k]}{\sum_i k(\hat{p}(y = k|x_q), \hat{p}(y - k|x_i))}$$



Measuring Calibration with a Reliability Diagram

- Given a "calibration set" of data points and a classifier, we can compute a reliability diagram (2-class case):
  - Divide $[0,1]$ into M bins (often $M = 10$). Bins may be of equal width or of equal quantiles according to $\hat{p}(y = 1|x)$
  - For bin $b \in \{1, \dots, M\}$, let B_b be the set of points whose probability scores $\hat{p}(y = 1|x)$ belong in bin B_b
  - $\hat{p}(B_b) = \frac{1}{|B_b|}\sum_{x \in B_b} \hat{p}(y = 1|x)$. This is the average predicted probability of the points in B_b
  - $\hat{P}(B_b) = \frac{1}{|B_b|}\sum_{x \in B_b} \mathbb{I}[y = 1]$. This is the fraction of predictions that are correct
  - Let $P_x(b) = |B_b|/N$ be the fraction of calibration points that fall into bin b
- Calibration Score
  - $\sum_{b=1}^{M} P_x(b)\left[\hat{p}(B_b) - \hat{P}(B_b)\right]^2$ expected squared calibration error
- Expected Calibration Error (ECE)
  - $\sum_{b=1}^{M} P_x(b)\left|\hat{p}(B_b) - \hat{P}(B_b)\right|$ expected absolute calibration error

Reliability Diagram (Naïve Bayes; ADULT)



Zadrozny & Elkan, 2002

Calibration Score and the Brier Score

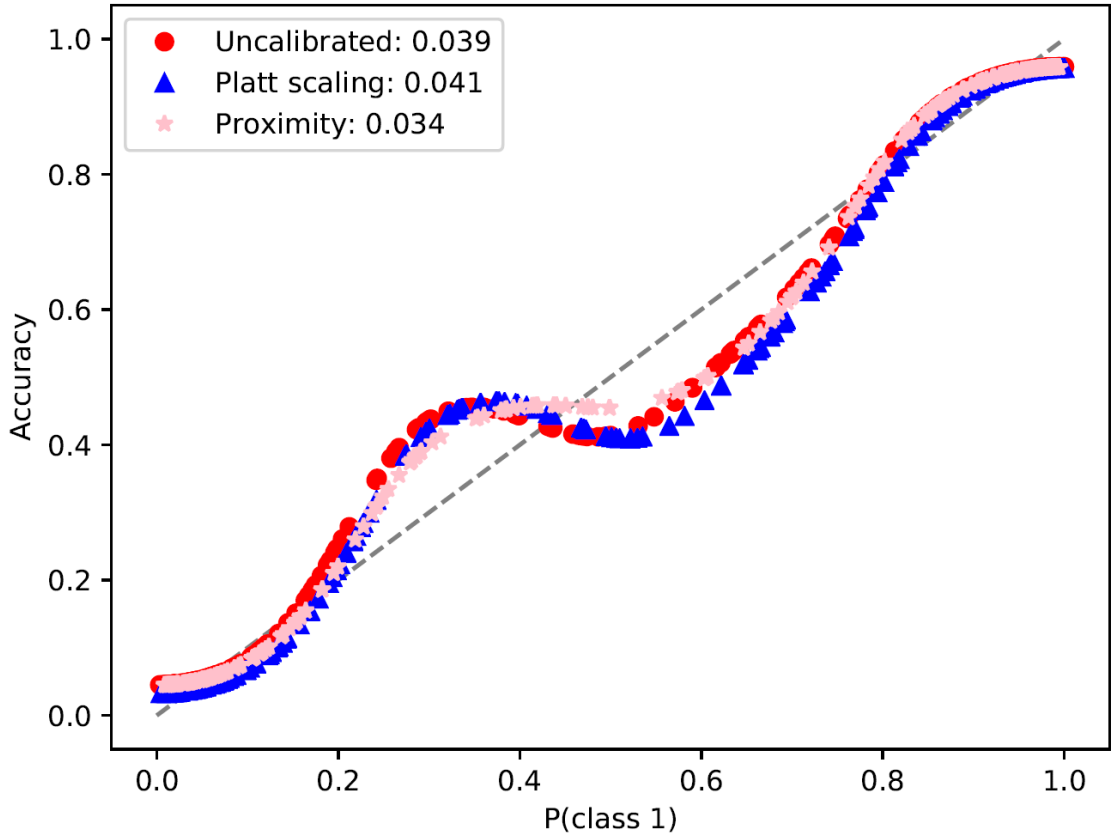
- The Brier Score is a proper scoring rule for probabilistic models
  - $BrierScore = \frac{1}{N}\sum_i(\hat{p}(\hat{y}_i|x_i) - \mathbb{I}[\hat{y}_i = y_i])^2$
- It can also be written in terms of the bins as
  - $BrierScore = \sum_b P_x(b)\left[\hat{p}(B_b) - \hat{P}(B_b)\right]^2 + \sum_b P_x(b)\left[\hat{P}(B_b)\left(1 - \hat{P}(B_b)\right)\right]$
  - Here The first term is the Calibration Score
- The second term is called the "Refinement Score". It is minimized when $\hat{P}(B_b)$ is near 0 or 1.
- A classifier that minimizes the Brier Score seeks to be well-calibrated *and* highly certain
- The Brier score is a proper scoring rule

Improving Calibration does not necessarily Improve Refinement

- A classifier can be well-calibrated but useless
  - Suppose 70% of the calibration data points belong to class 1
  - Then always predict $\hat{y} = 1$ with $\hat{p}(\hat{y}) = 0.7$
  - This is perfectly calibrated but useless
  - Note that the Refinement score will be large
    - $0.7 \times 0.3 = 0.21$

Kernel ECE

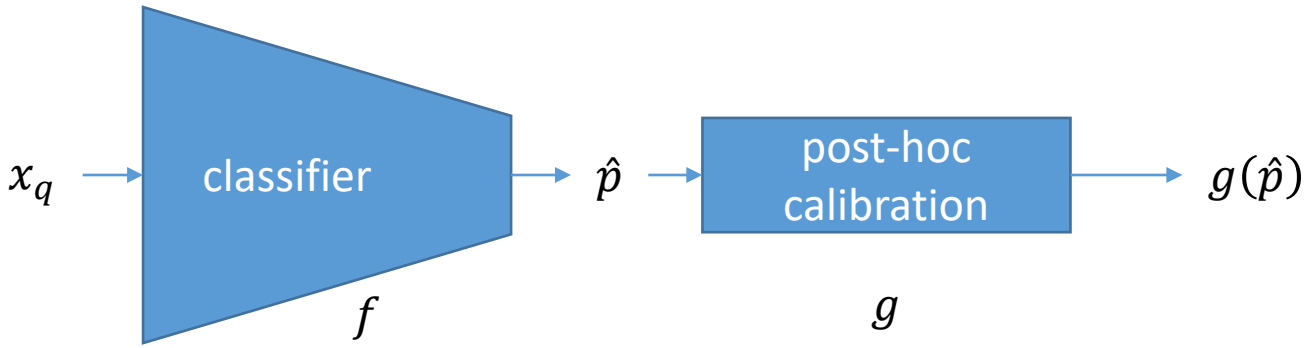
- Most papers use a fixed set of 10 or 100 equal-width bins
- This introduces biases near the bin boundaries
  - There are methods for reducing those biases
- Better method: Kernel ECE
  - Kumar, Sarawagi & Jain (2018)
  - Use a kernel in the predicted probability space



source: Kiri Wagstaff

Post-Hoc Calibration Methods

- Divide data into a training set and a calibration set
- Train the classifier as usual on the training set (e.g., to maximize accuracy, AUC, etc.)
- Learn a calibration function that transforms the classifier's output probabilities into well-calibrated probabilities



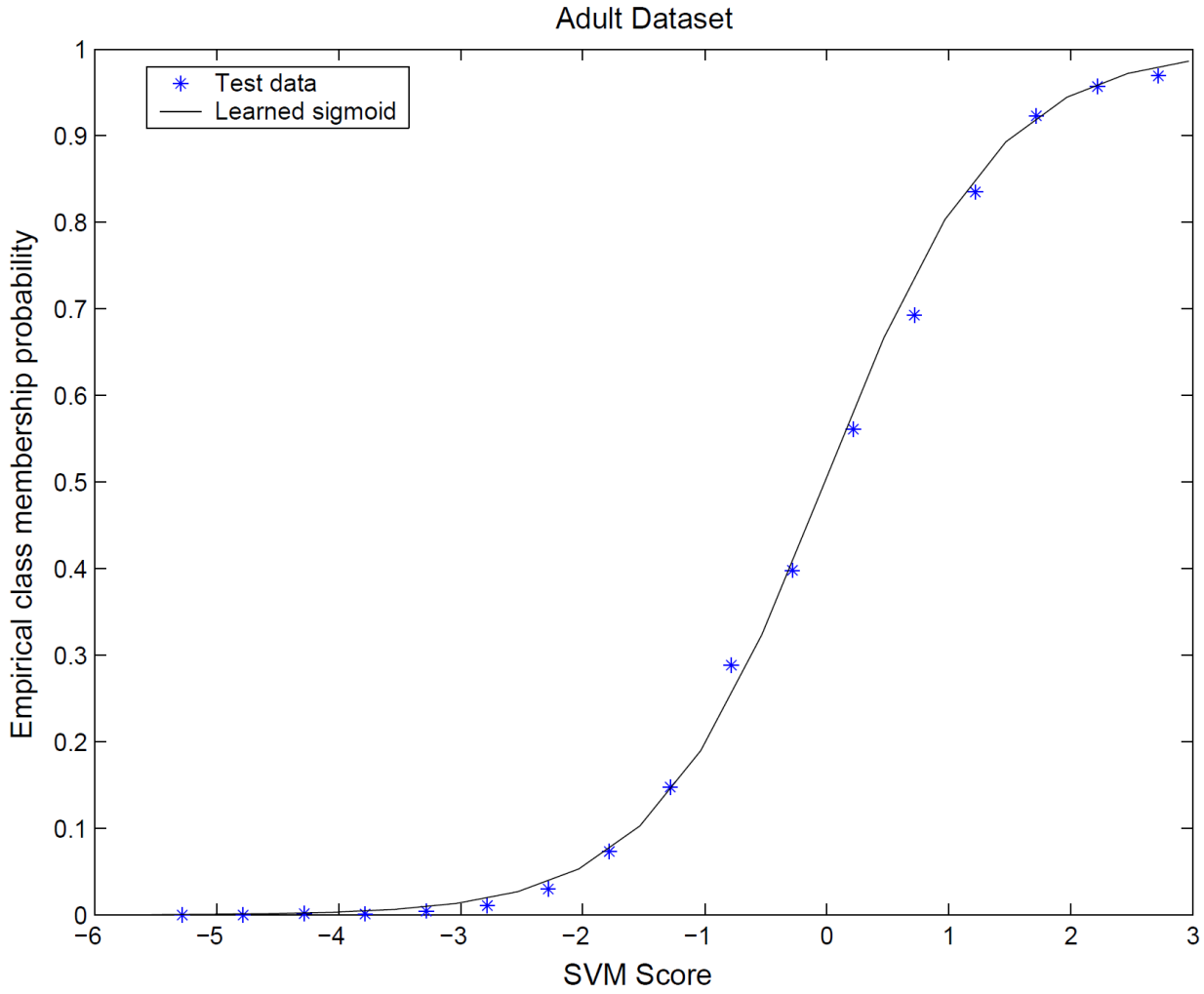
Calibration Method 1: Binning

- Fit a function g to map $\hat{p}$ to P and then replace $\hat{p}$ with $g(\hat{p})$
- "training data" consist of $(\hat{p}_i, \mathbb{I}[y_i = k])$ pairs
- Fixed-width Bins
  - Sort the data by $\hat{p}$
  - Let $B_1, \ldots, B_M$ each be of width $\frac{1}{M}$
  - Estimate $\hat{P}(B_b)$ for each bin
  - $g(\hat{p}) = \hat{P}(B_b)$ for the bin B_b containing $\hat{p}$
- Quantile Bins
  - Define the bins so that each bin contains $\frac{1}{M}$ of the training data

Calibration Method 2: Platt Scaling

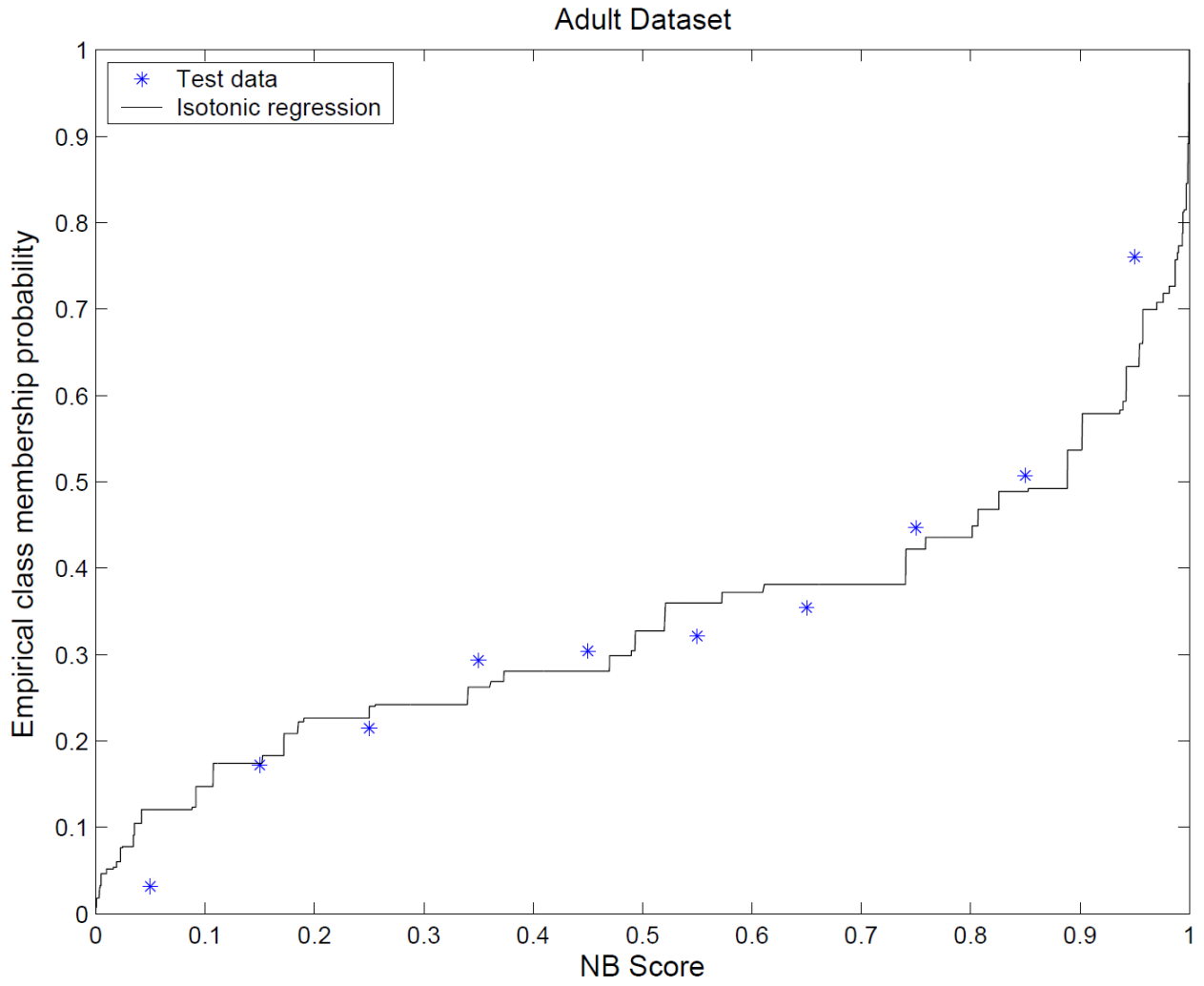
(Platt, 1999)

- $g(\hat{p}; a, b) = \frac{1}{1+e^{a+b\hat{p}}}$
- Logistic regression with a single "feature" ($\hat{p}$)



Method 3: Isotonic Regression

- Find the function g that is monotonically increasing from 0 to 1 and minimizes the Brier Score
- Pool-Adjacent Violators Algorithm
  - Ayer, et al. (1955)
  - Robertson, Wright, & Dykstra (1988)



PAV

Ayer, M., Brunk, H., Ewing, G., Reid, W., Silverman, E. (1955)

- Input: $(\hat{p}_i, \mathbb{I}[y_i = k])$ sorted in ascending order by $\hat{p}_i$
- Initialize $\hat{m}_{i,i} = \mathbb{I}[y_i = k]; w_{i,i} = 1$
- While $\exists i \; s.t. \hat{m}_{k,i-1} \geq \hat{m}_{k,i}$
  - $w_{k,l} \coloneqq w_{k,i-1} + w_{i,l}$
  - $\hat{m}_{k,l} \coloneqq \frac{w_{k,i-1}\hat{m}_{k,i-1} + w_{i,l}\hat{m}_{i,l}}{w_{k,l}}$
  - Insert $\hat{m}_{k,l}$ in place of $\hat{m}_{k,i-1}$ and $\hat{m}_{k,i}$
- Output the function
  - $\hat{m}(\hat{p}) = \hat{m}_{i,j}$ for $\hat{p} \in \left(\hat{p}_i, \hat{p}_j\right]$

Method 4: Regularized Isotonic Regression

- Isotonic Regression can be rewritten as the solution to the following problem
- Choose $\hat{P}_i$ to minimize
  - $\frac{1}{2}\sum_{i=1}^{N}\left(\hat{P}_i - \hat{p}_i\right)^2 + \lambda \sum_{i=1}^{N-1}\left(\hat{P}_i - \hat{P}_{i+1}\right)\mathbb{I}\left[\hat{P}_i > \hat{P}_{i+1}\right]$ subject to $\lambda = +\infty$
- Tibshirani, Hastie & Tibshirani (2011) developed mPAVA, which constructs the complete regularization path from $\lambda = 0$ to $\lambda = \infty$
  - Efficient algorithm that produces a sequence of "near isotonic" regression models $g_1, \ldots, g_t, \ldots$
- ENIR (Ensemble of Near Isotonic Regressions; Naeini & Cooper, 2018) computes the BIC score of each g_t, normalizes these scores, and then computes the weighted average of the models to obtain g

Method 5: Other Flexible Models

- Splines (Lucena, 2018 arxiv 1809.07751)
- Piecewise linear functions via a tree-based decomposition (Leathart, Frank, Holmes, Pfahringer, 2017)
- Gaussian Processes (Song, Kull, Flach, 2018)

Methods for Multiclass Classifiers

- Method 1: Normalized one-vs-rest calibration
  - For each class k, learn a binary calibration function g_k based on a one-vs-rest classifier
  - Define $g\big(\hat{p}(y=1|x), \ldots, \hat{p}(y=K|x)\big)$ as follows
    - Let the predicted probability for class k be

$$\frac{g_k\big(\hat{p}(y=k|x)\big)}{\sum_{k'} g_{k'}\big(\hat{p}(y=k'|x)\big)}$$

Multiclass Method 2: Softmax Temperature Tuning (Guo et al, 2017)

- Let $\ell = \ell_1, \ldots, \ell_K$ be the logits of a DNN
- Scale the logits by dividing by a temperature T:

$$\hat{p}(y = k|x) = \frac{\exp \frac{\ell_k}{T}}{\sum_{k'} \exp \frac{\ell_{k'}}{T}} = \sigma_{SM}(\ell)$$

- Adjust T to fit the calibration data

Multiclass Methods 3 and 4: Generalized Platt Scaling

- Matrix Scaling
  - Learn a matrix $\mathbf{W}$ and vector $\mathbf{b}$ to fit $\sigma_{SM}(\mathbf{W}\ell + \mathbf{b})$ to a 1-hot encoding of y_i
- Vector Scaling
  - Matrix scaling with $\mathbf{W} = \mathrm{diag}(\mathbf{w})$

Experiments 1: Niculescu-Mizil & Caruana (2005)

- Insights
  - Max-margin methods push $\hat{p}$ toward 0.5
  - Naïve Bayes pushes $\hat{p}$ toward 1.0
  - Calibration flattens out this distribution
  - Max-margin methods are fit well by logistic regression (Platt scaling), which also needs relatively little data
  - Isotonic Regression works well with Naïve Bayes but usually requires more calibration data

Boosted Trees

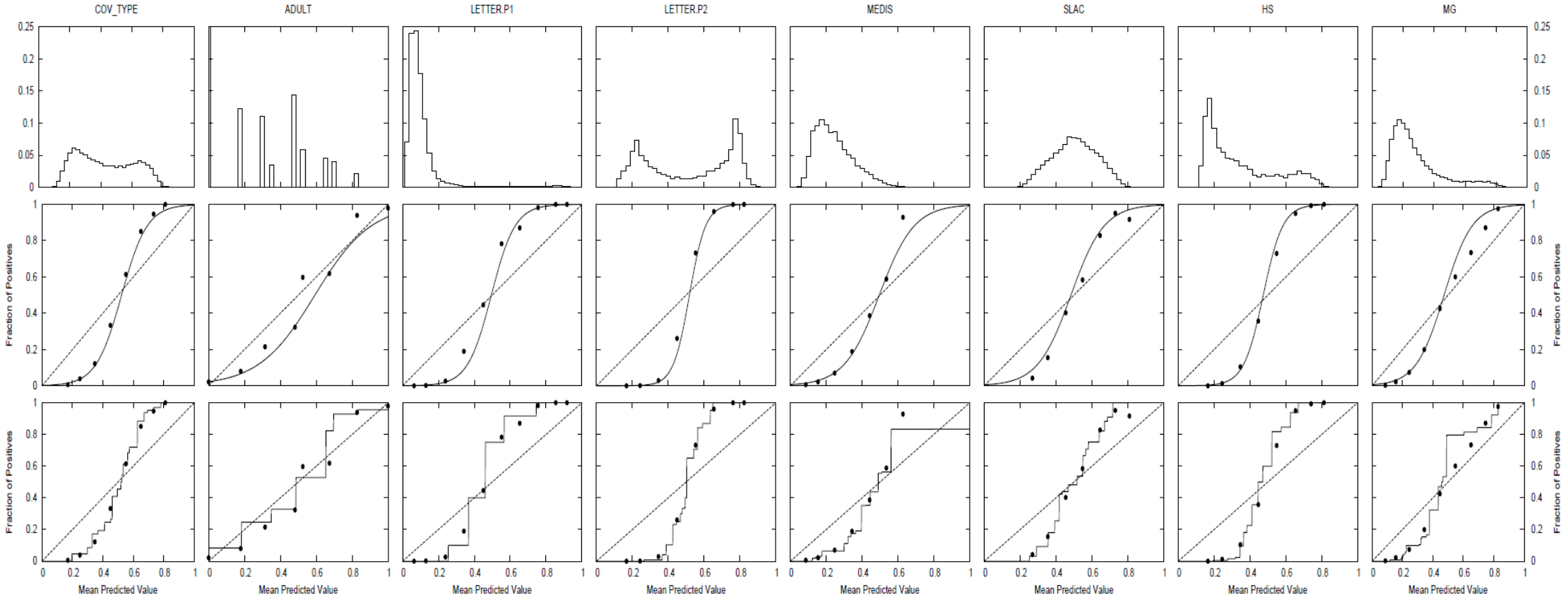


Figure 1. Histograms of predicted values and reliability diagrams for boosted decision trees.

Boosted Trees after Platt Calibration

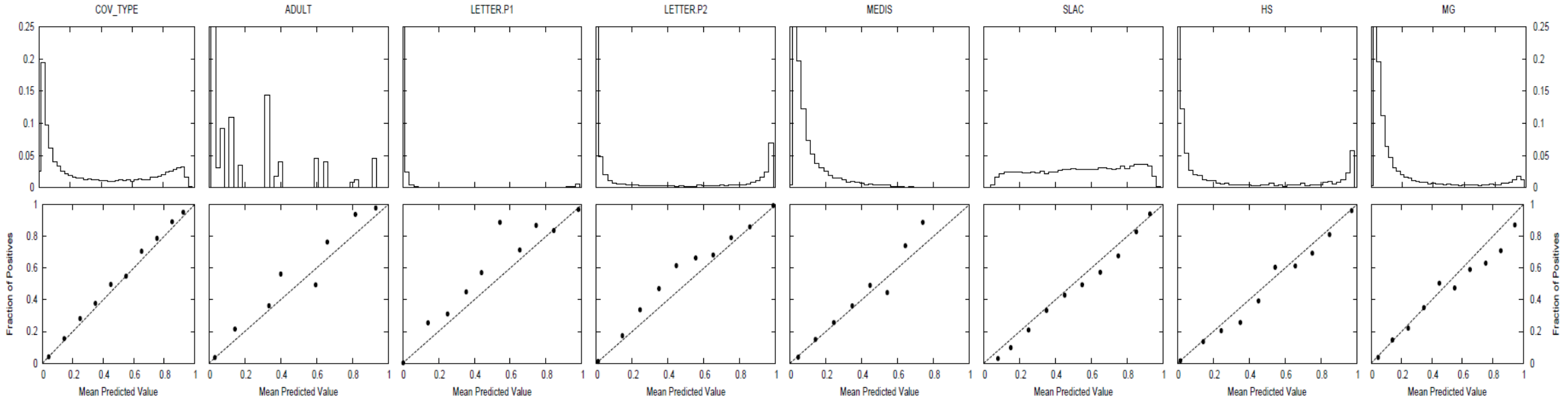


Figure 2. Histograms of predicted values and reliability diagrams for boosted trees calibrated with Platt's method.

Boosted Trees after Isotonic Regression Calibration

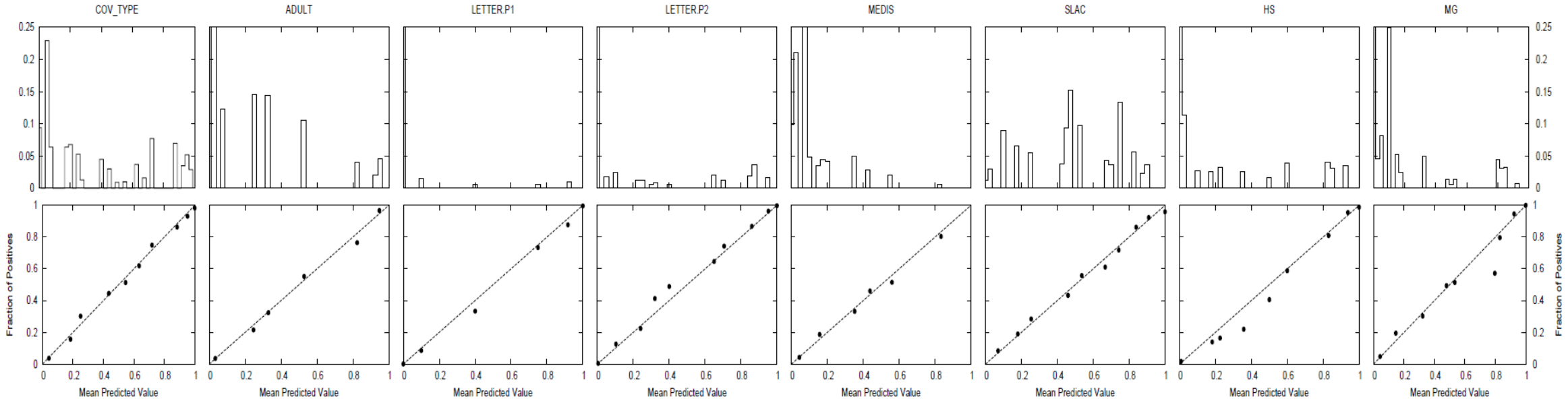
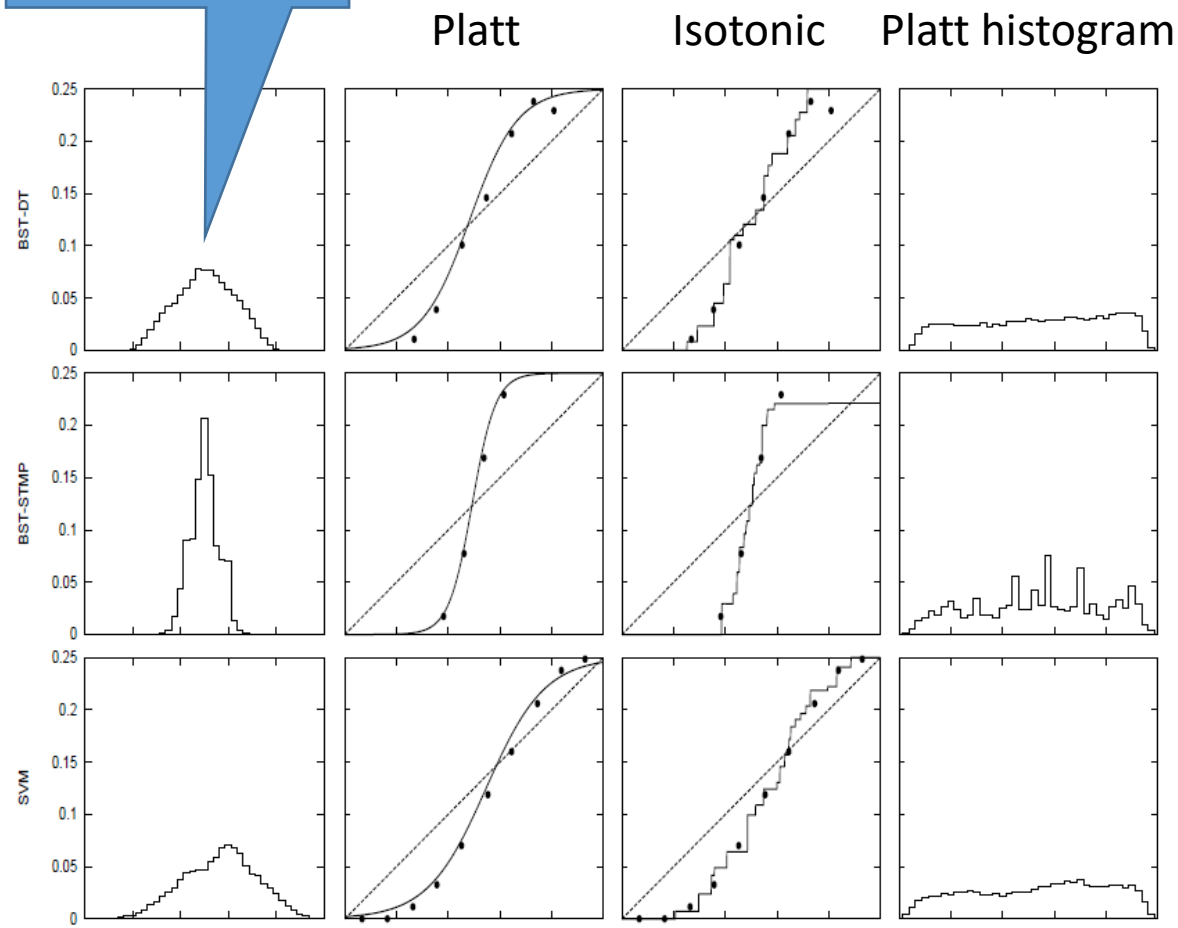


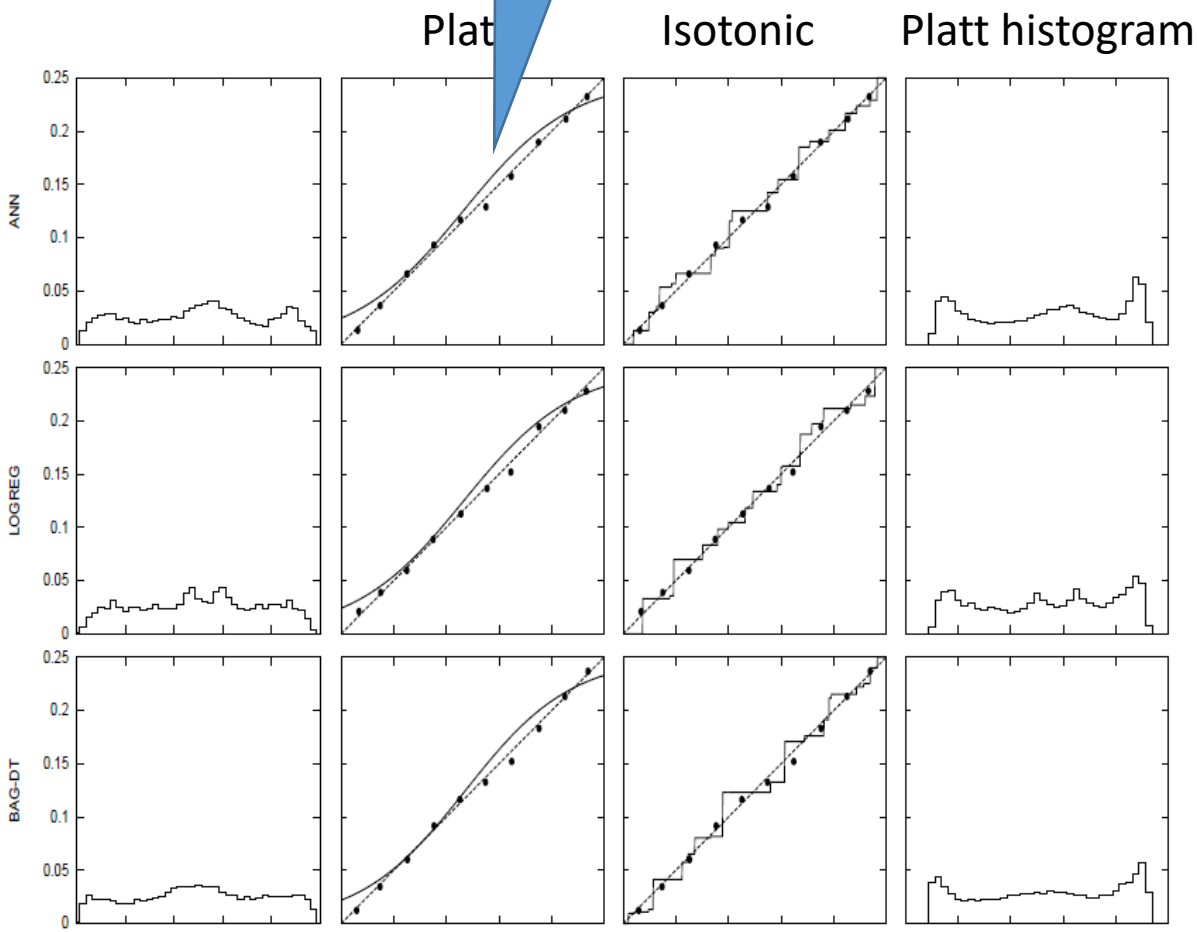
Figure 3. Histograms of predicted values and reliability diagrams for boosted trees calibrated with Isotonic Regression.

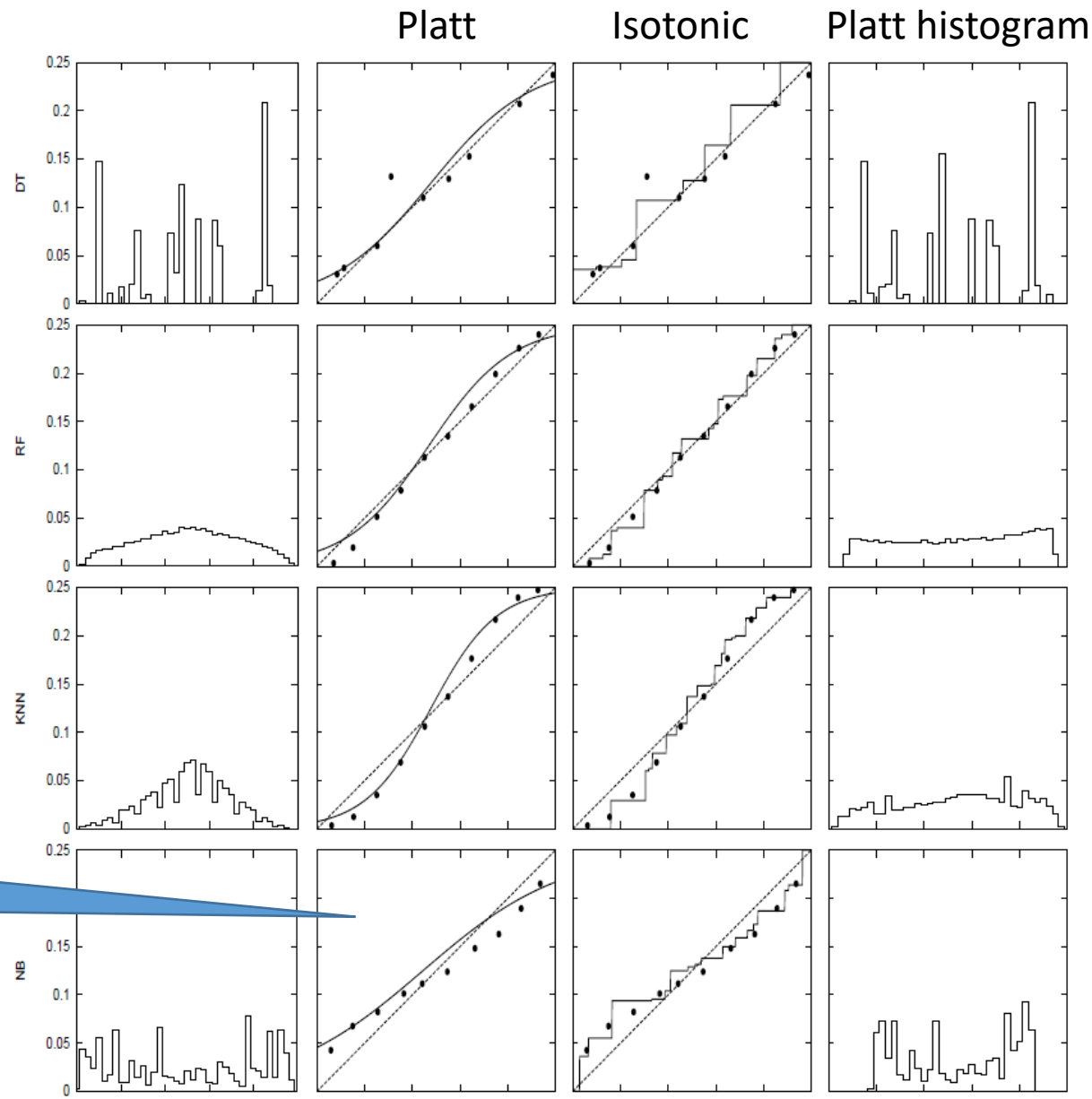
10 Different Learning Algorithms (on SLAC dataset)

$\hat{p}$ concentrated in the middle

Already well-calibrated



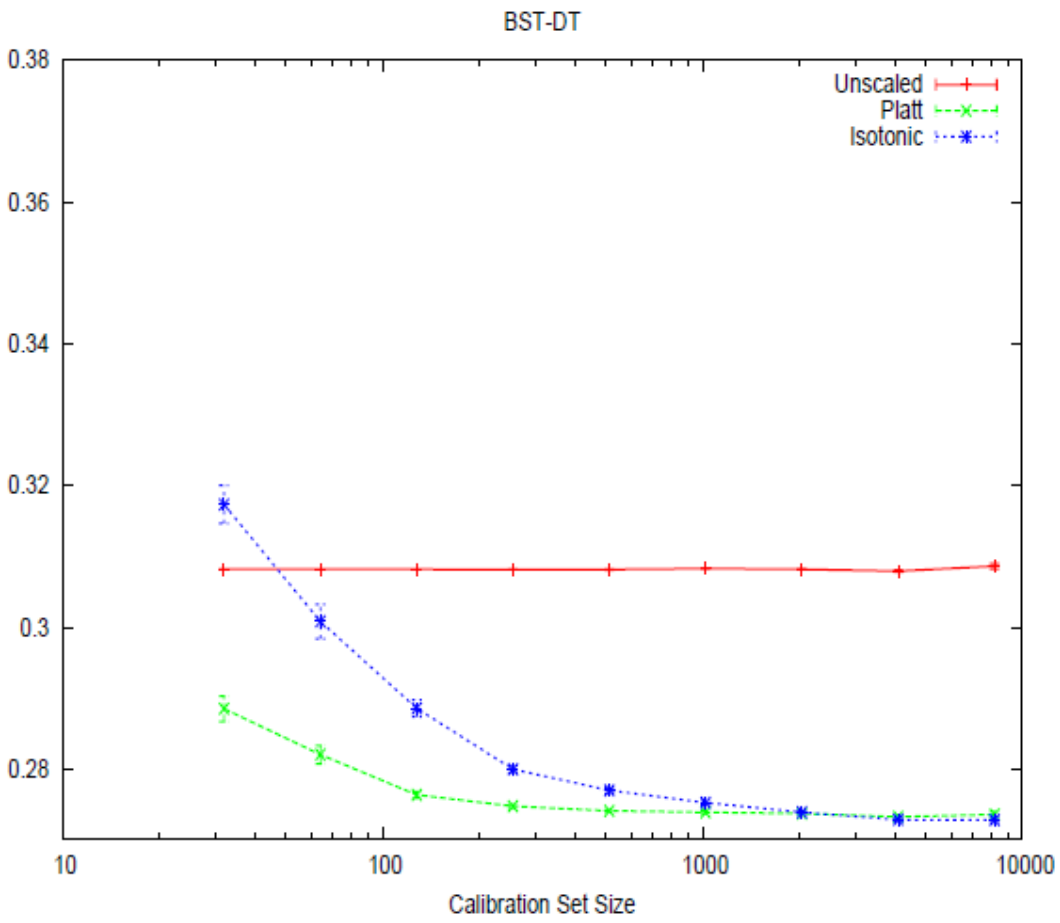




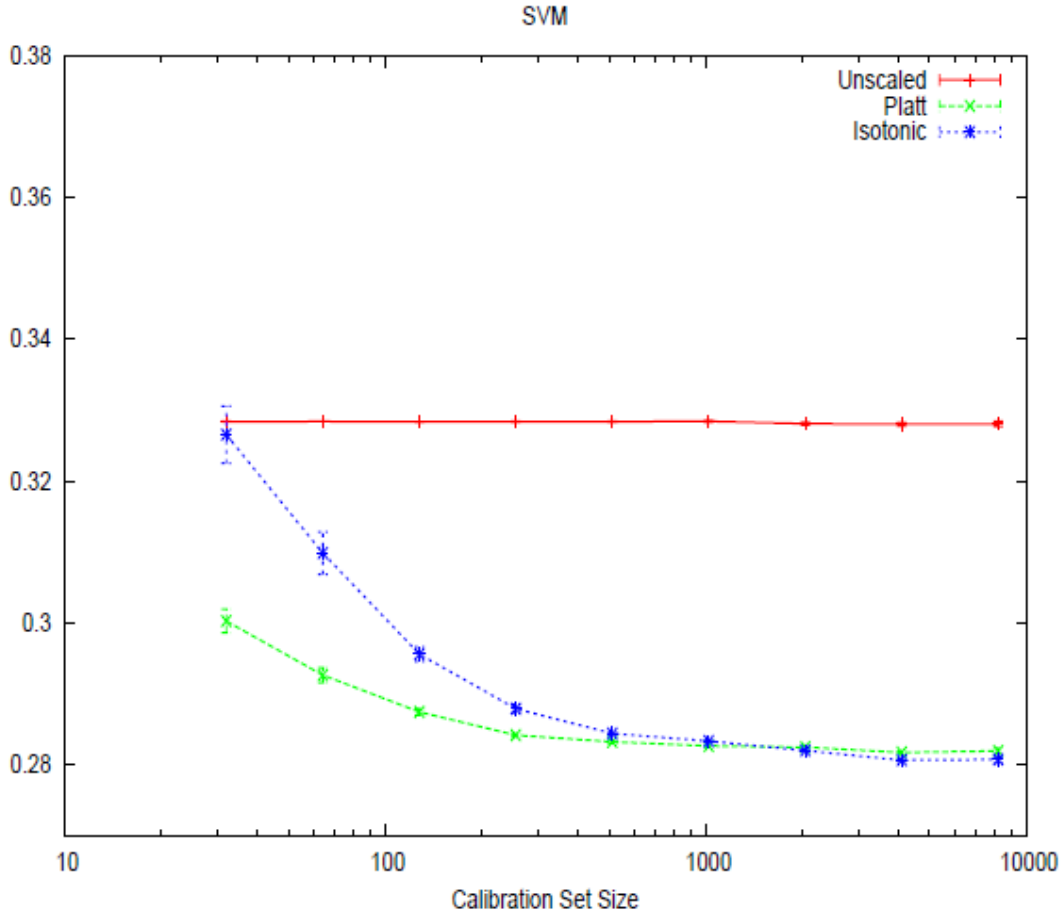
Sigmoid is not a good model for NB

Figure 6. Histograms and reliability diagrams for SLAC.

How big does the calibration set need to be?

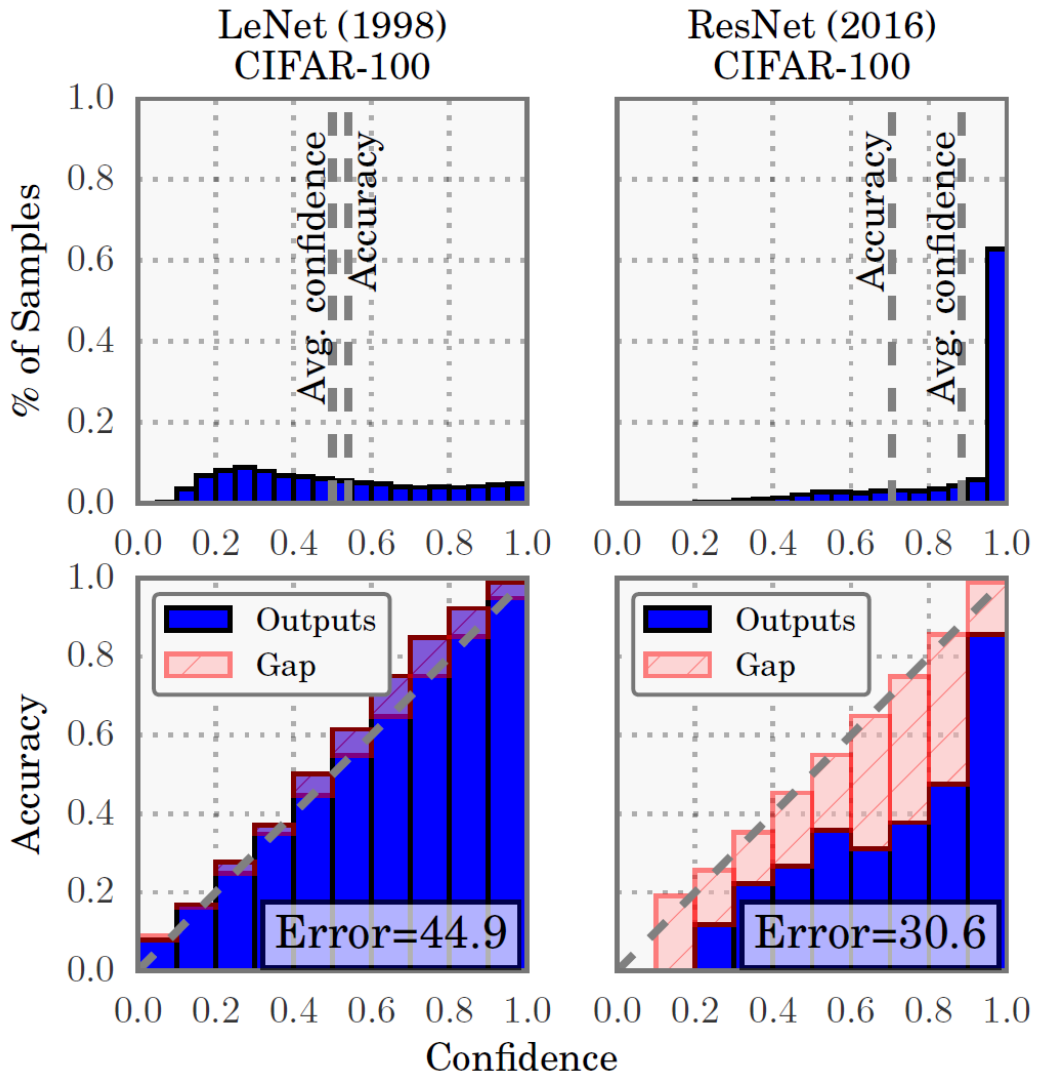


Platt: 500; Isotonic: 8000?



Platt: 500; Isotonic: 4000

Experiments 2: Guo, Pleiss, Sun & Weinberger



ResNet is much more confident

ResNet over-confident!

What are the causes of bad calibration?

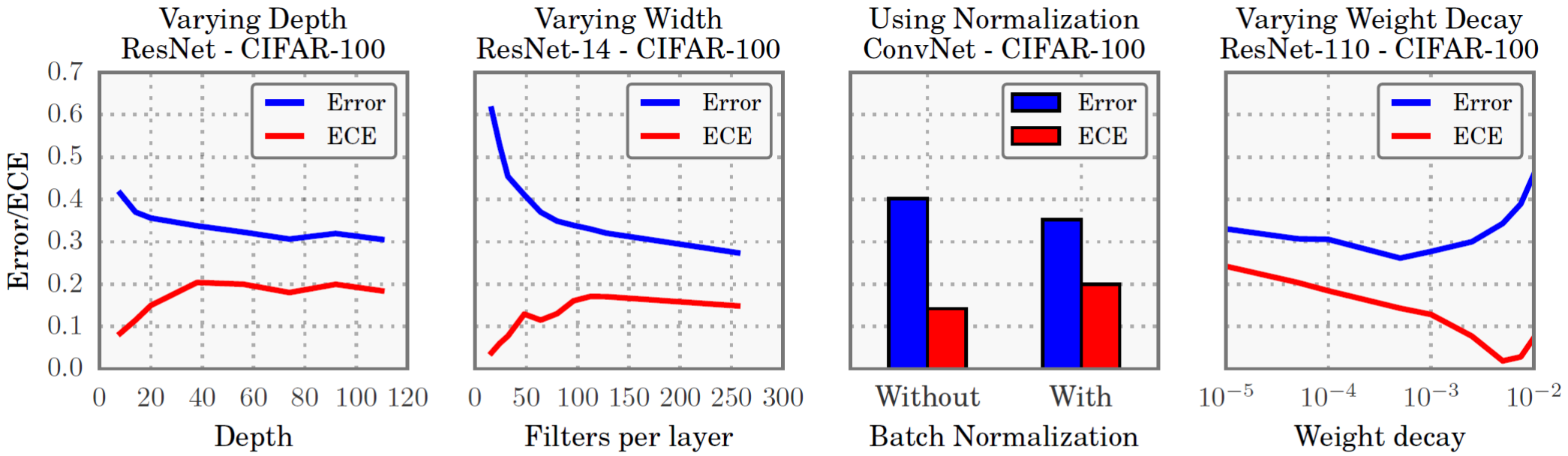
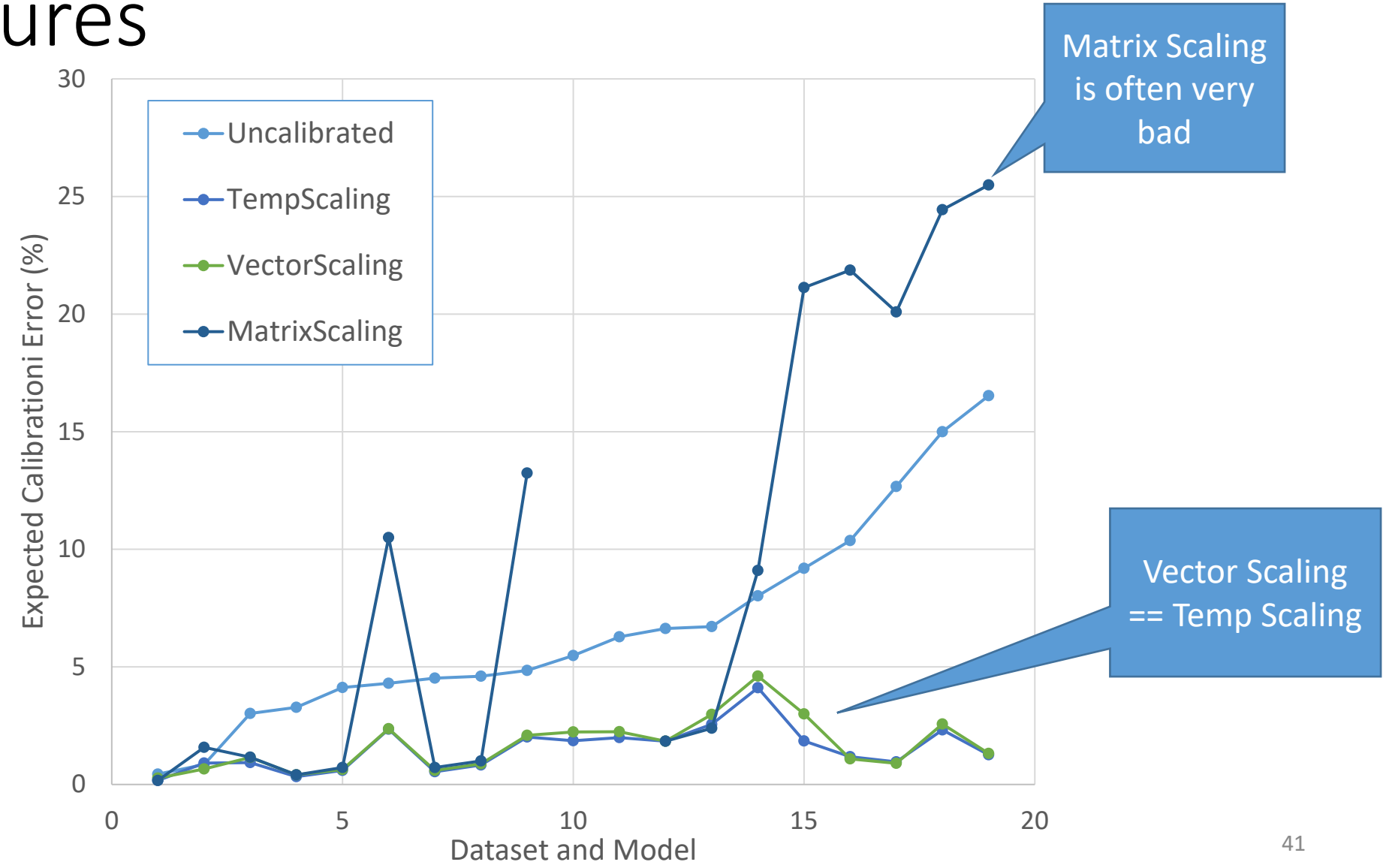


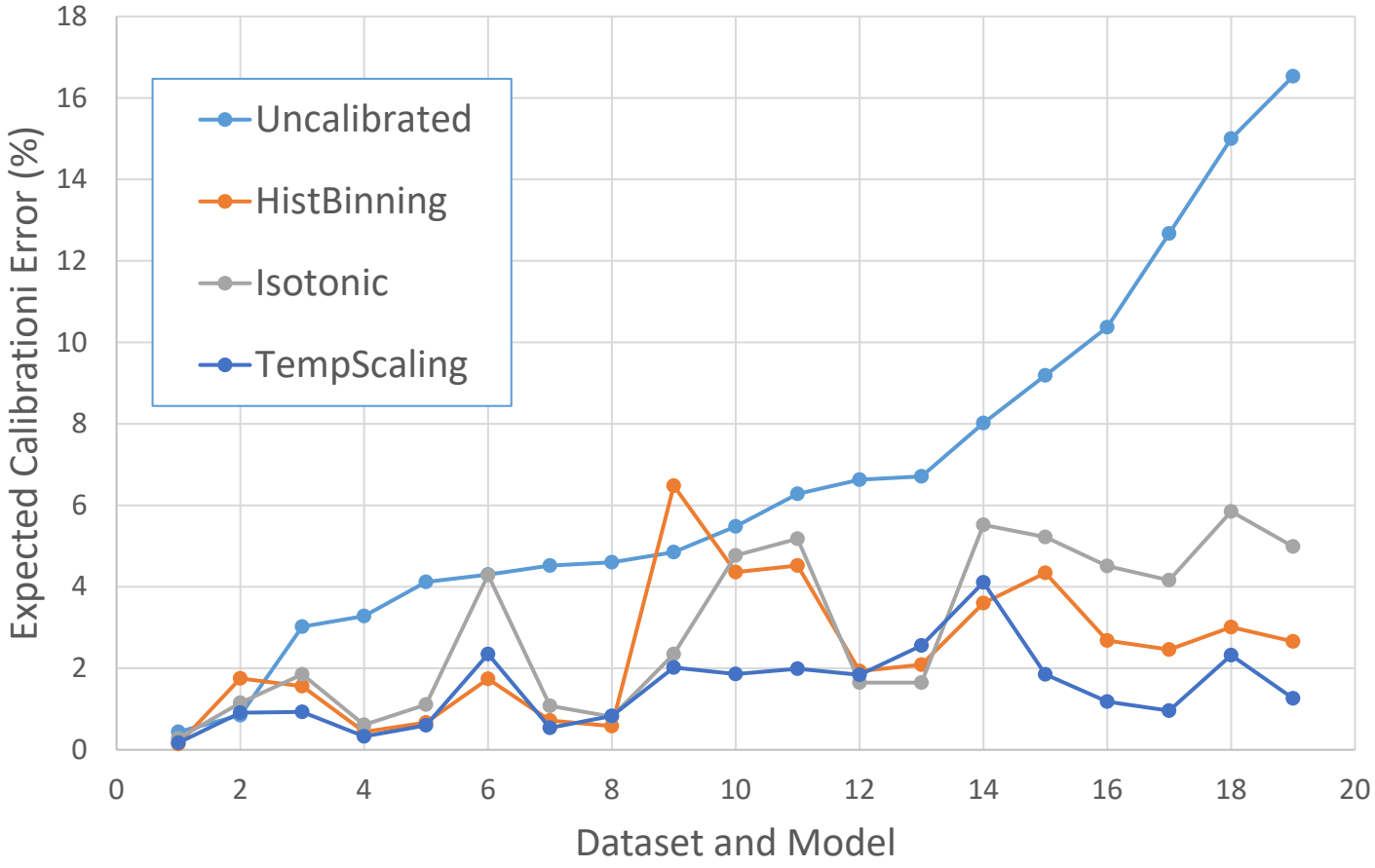
Figure 2. The effect of network depth (far left), width (middle left), Batch Normalization (middle right), and weight decay (far right) on miscalibration, as measured by ECE (lower is better).

Note: ECE = mean absolute calibration error $\sum_b \frac{|B_b|}{N} \left|\hat{P}(B_b) - \hat{p}(B_b)\right|$

Comparison on Multiple Tasks and Architectures



Comparison against other methods



Insights and Questions

- The simple Temperature Calibration model works well and works better than more complex generalizations of Platt Scaling
- Temperature Calibration can be derived as the solution to a maximum entropy optimization problem
  - Maximize entropy of $\hat{P}$ subject to (a) $\hat{P}$ is a probability and (b) the sum of true class logits == mean value of all logits weighted by $\hat{P}$
  - Not clear why (b) makes sense
- Need to compare against Platt Scaling each class separately and then normalizing

Experiment 3: Naeini & Cooper (ENIR, 2018)

- 40 UCI and LibSVM benchmark datasets
- Classifiers: Naïve Bayes, Logistic Regression, SVM
- Hyperparameters tuned via 10x10-fold cross-validation
- Calibration Algorithms:
  - Isotonic Regression (IsoRegC)
  - BBQ: Bayesian Quantile Binning (ensemble of quantile bin models)
  - ENIR: Ensemble of Near Isotonic Regressions
- Calibration reuses the training data
- No comparison against Platt scaling or other model-based approaches
- Metrics:
  - AUC = area under ROC curve
  - ACC = accuracy
  - RMSE = square root of the Calibration score
  - ECE = expected absolute calibration error
  - MCE = maximum absolute calibration error

Percentage Change using ENIR

Table 3 The 95% confidence interval for the average percentage of improvement over the base classifiers (LR, SVM, NB) by using the ENIR method for post-processing

	LR	SVM	NB
AUC	[−0.008, 0.003]	[−0.010, 0.003]	[−0.010, 0.000]
ACC	[0.002, 0.016]	[−0.001, 0.010]	[0.012, 0.068]
RMSE	[−0.124, −0.016]	[−0.310, −0.176]	[−0.196, −0.100]
ECE	[−0.389, −0.153]	[−0.768, −0.591]	[−0.514, −0.274]
MCE	[−0.313, −0.064]	[−0.591, −0.340]	[−0.552, −0.305]

Positive entries for AUC and ACC mean ENIR is on average providing better discrimination than the base classifiers. Negative entries for RMSE, ECE, and MCE mean that ENIR is on average performing better calibration than the base classifiers

Naïve Bayes & LR ACC always improves

Calibration Metrics always improve

Accuracy improvements probably result from better thresholding

Insights and Questions

- Using a regularized version of Isotonic Regression (ENIR) does not improve accuracy or AUC compared to regular Isotonic Regression
- But it does improve measures of calibration

- The main advantage of regularizing should be to reduce the amount of calibration data that is needed, but the authors did not study this question

Summary of Miscalibration Behaviors

Based on these Papers

- Max Margin Methods (SVM, boosted trees):
  - $\hat{p}$ concentrates near 0.5
  - Sigmoid-shaped Reliability Diagram
  - Platt (logistic regression) model fits well, learns quickly
- Naïve Bayes and Deep Nets
  - $\hat{p}$ concentrates near 0 and 1; systematically optimistic
  - Sigmoid model fits NB poorly; Isotonic regression is better
  - Temperature Calibration worked better for Deep Nets
- Random Forests, Bagging, MLPs
  - Naturally well-calibrated except at extreme probabilities
    - We have counter-examples for random forests
  - Sigmoid model fits poorly
  - Need lots of calibration data to obtain any improvements

Open Questions for Calibration

- Do we care equally about all parts of the $\hat{p}$ space?
  - For high-confidence predictions
    - We only care about large values of $\hat{p}$
  - For anomaly detection
    - We only care about very small values of $\hat{p}$
  - For stock market trading
    - We care about values of $\hat{p} = 0.5 + \epsilon$
- To estimate $P(y|x)$, we use a combined neighborhood η over $\mathcal{X} \times \mathcal{Y}$?
- Can we address the causes of miscalibration?

References

- Ayer, M., Brunk, H., Ewing, G., Reid, W., Silverman, E. (1955). An empirical distribution function for sampling with incomplete information. *Annals of Mathematical Statistics*, 26(4):641–647, 1955. (PAV Algorithm)
- Charoenphakdee, N., Cui, Z., Zhang, Y., & Sugiyama, M. (2020). Classification with Rejection Based on Cost-sensitive Classification. *ArXiv*, *2010.11748*. http://arxiv.org/abs/2010.11748 also ICML 2021
- Cohen, I., Goldszmidt, M. (2004). Properties and benefits of calibrated classifiers. HP Labs Report HPL-2004-22.
- Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. *ArXiv*. Retrieved from http://arxiv.org/abs/1706.04599
- Kumar, A., Sarawagi, S., & Jain, U. (2018). Trainable calibration measures for neural networks from kernel mean embeddings. *35th International Conference on Machine Learning, ICML 2018*, *6*, 4378–4389.
- Lucena, B. (2018) Spline-Based Probability Calibration. https://export.arxiv.org/abs/1809.07751
- Leathart, T., Frank. E., Holmes, G., Pfahringer, B. (2017). Probability Calibration Trees. *Asian Conference on Machine Learning (2017).* JMLR: Workshop and Conference Proceedings 77:145-160.
- Naeini, Cooper, G. (2018). Binary classifier calibration using an ensemble of piecewise linear regression models. *Knowledge and Information Systems, 54(1)*: 151-170. https://arxiv.org/pdf/1511.05191.pdf

References (2)

- Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. *Proceedings of the 22nd International Conference on Machine Learning ICML '05*, (2005), 625–632. http://doi.org/10.1145/1102351.1102430
- Platt, J. (1999). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers* (pp. 61–74).
- Robertson, T., Wright, F., Dykstra, R. (1988). *Order Restricted Statistical Inference*, chapter 1. John Wiley & Sons, 1988.
- Song, H., Kull, M., Flach, P. (2018). Non-Parametric Calibration of Probabilistic Regression. https://arxiv.org/abs/1806.07690
- Tishirani, R. J., Hoefling, H., Tibshirani, R., (2011). Nearly isotonic regression. *Technometrics, 53(1)*:54–61, 2011
- Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naïve Bayesian classifiers. *ICML* (pp. 609–616).
- Zadrozny, B., Elkan, C. (2002). Transforming Classifier Scores into Accurate Multiclass Probability Estimates. *KDD* (pp. 694–699).
- Zhang, J., Kailkhura, B., & Han, T. Y.-J. (2020). Mix-n-Match: Ensemble and Compositional Methods for Uncertainty Calibration in Deep Learning. *ICML 2020*. http://arxiv.org/abs/2003.07329