How effective is your classifier?
Revisiting the role of metrics in machine learning

SANMI KOYEJO
CS @ ILLINOIS

Joint work with Ran Li, Xiaoyan Wang, Gaurush Hiranandani, Shant Boodaghians, and Ruta Mehta

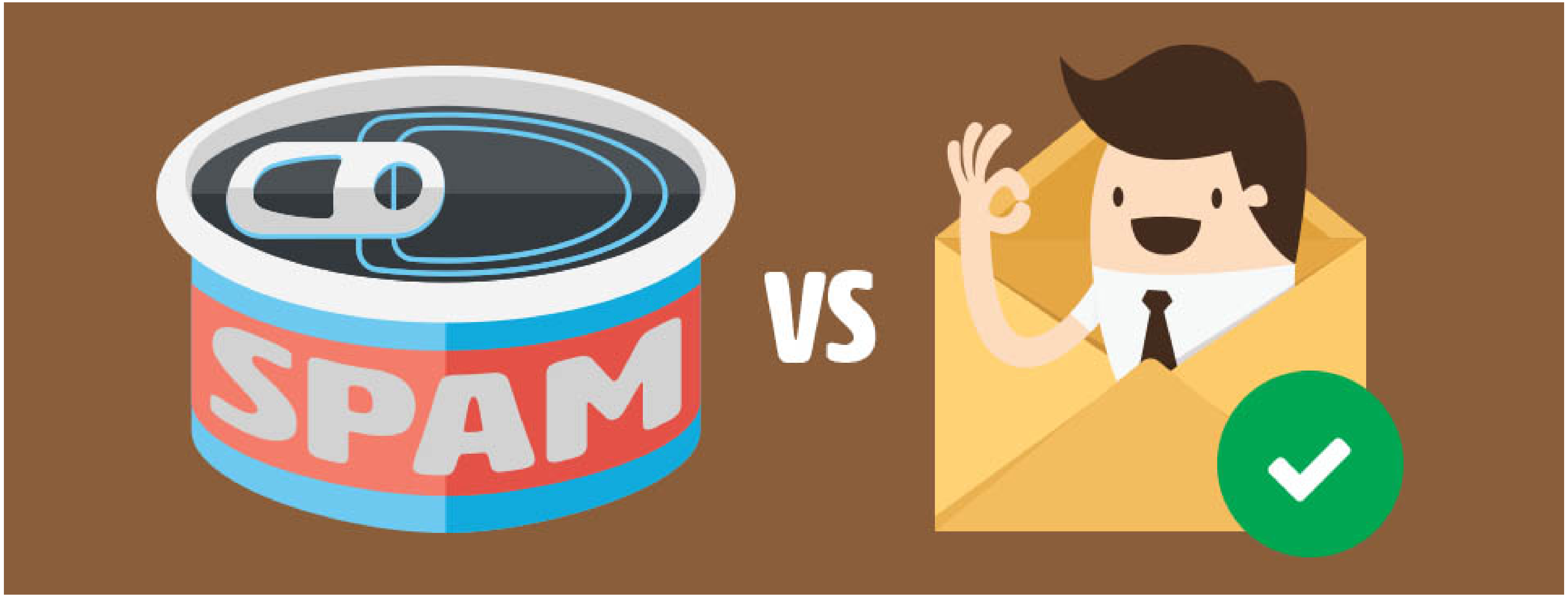
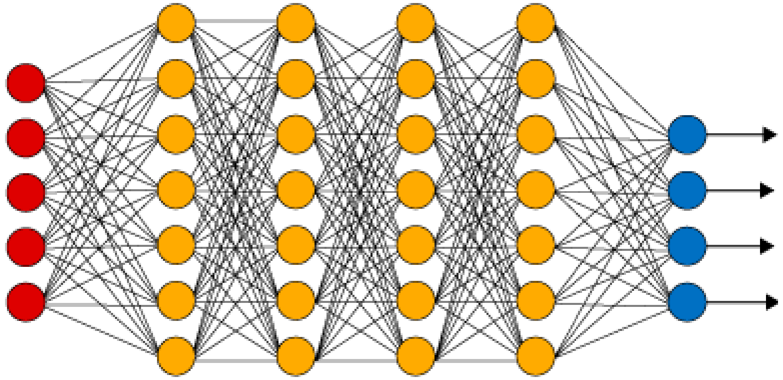


Image Source: https://davepannell.com/public/2016/03/Email-marketing-vs-spam.jpg



- Accuracy = 95%
- $$$

- Users complain that most real emails are labelled spam
- ~90% of all email is spam*
- Suggests that accuracy is the wrong metric as it gives equal weight to all errors

Error analysis

		Ground truth	
		Spam	Not Spam
Predicted	Spam	TP	FP
	Not Spam	FN	TN

- Accuracy $= \mathrm{TP} + \mathrm{TN} = 1 - \mathrm{FP} - \mathrm{FN}$
- To improve user calibration, try evaluating and/or optimizing weighted accuracy e.g.

$$\phi(h) = 1 - 0.1\,\mathrm{FP} - \mathrm{FN}$$

The confusion matrix

		Ground truth	
		Y = 1	Y = 0
Predicted	h(x) = 1	TP	FP
	h(x) = 0	FN	TN

$$= \mathbf{C}(h)$$

Beyond Accuracy, more general metrics are nested functions

$$\phi(h) = \psi\big(\mathbf{C}(h)\big)$$

- Metrics are used to compare classifiers, or can be optimized directly
- The classifier performance metric can be approximated from data.

Lots of real world examples

$$\phi(h) = a_1\mathrm{TP} + a_2\mathrm{FP} + a_3\mathrm{FN} + a_4\mathrm{TN}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},\ \mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{FP}+\mathrm{TN}},\ \mathrm{Prec} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}},\ \mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TP}},\ \mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FN}}.$$

$$\mathrm{AM} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\pi} + \frac{\mathrm{TN}}{1-\pi}\right) = \frac{(1-\pi)\mathrm{TP}+\pi\mathrm{TN}}{2\pi(1-\pi)},\ F_\beta = \frac{(1+\beta^2)\mathrm{TP}}{(1+\beta^2)\mathrm{TP}+\beta^2\mathrm{FN}+\mathrm{FP}} = \frac{(1+\beta^2)\mathrm{TP}}{\beta^2\pi+\gamma},$$

$$\mathrm{JAC} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}+\mathrm{FP}} = \frac{\mathrm{TP}}{\pi+\mathrm{FP}} = \frac{\mathrm{TP}}{\gamma+\mathrm{FN}},\quad \mathrm{WA} = \frac{w_1\mathrm{TP}+w_2\mathrm{TN}}{w_1\mathrm{TP}+w_2\mathrm{TN}+w_3\mathrm{FP}+w_4\mathrm{FN}}.$$

$$\pi = \mathrm{TP}+\mathrm{FN},\ \gamma = \mathrm{TP}+\mathrm{FP}$$

Metrics in ranking and recommendation

"Results show that improvements in RMSE often do not translate into [top-N ranking] accuracy improvements. In particular, a naive non-personalized algorithm can outperform some common recommendation approaches and almost match the accuracy of sophisticated algorithms"

P. Cremonesi, Y. Koren, and R. Turrin. "Performance of recommender algorithms on top-n recommendation tasks." Recsys, 2010.

Metric choice has a large impact on real-world machine learning performance.

1

Given a complex metric, how can we efficiently construct classifiers that (approximately) optimize it?

2

Given a new classification problem, which metric should you use to measure performance?

One simple trick…

A RE-WEIGHTING STRATEGY

Multiclass classification

		Ground truth			
		Y = 1	Y = 2	…	Y=K
Predicted	h(x) = 1	c_{11}	c_{12}		c_{1K}
	h(x) = 2	c_{21}	c_{22}		c_{2K}
	⋮				…
	h(x) = K	c_{K1}	c_{K2}		c_{KK}

$$C(h)$$

Standard metric is Accuracy

$$\phi(h) = c_{11} + c_{22} + \ldots + c_{KK}$$

$$= \langle I, C(h) \rangle$$

$$s_i(x) \approx p(y = i|x)$$

e.g. logistic regression, RF, DNN, …

$$h(x) = \underset{i \in [K]}{\operatorname{argmax}}\, s_i$$

Standard Prediction Strategy

$$\max_h \langle A, C(h) \rangle$$

$$s_i(x) \approx p(y = i|x)$$

e.g. logistic regression, RF, DNN, …

$$h(x) = \underset{i \in [K]}{\operatorname{argmax}}\ \mathbf{a}_i^\top \mathbf{s}(x)$$

Proposed Postprocessing Strategy

Narasimhan, H., et al. "Consistent multiclass algorithms for complex performance measures." ICML. 2015.

A small experiment

$$\eta_k(x) \propto e^{\mathbf{w}_k^\top \mathbf{x}}$$

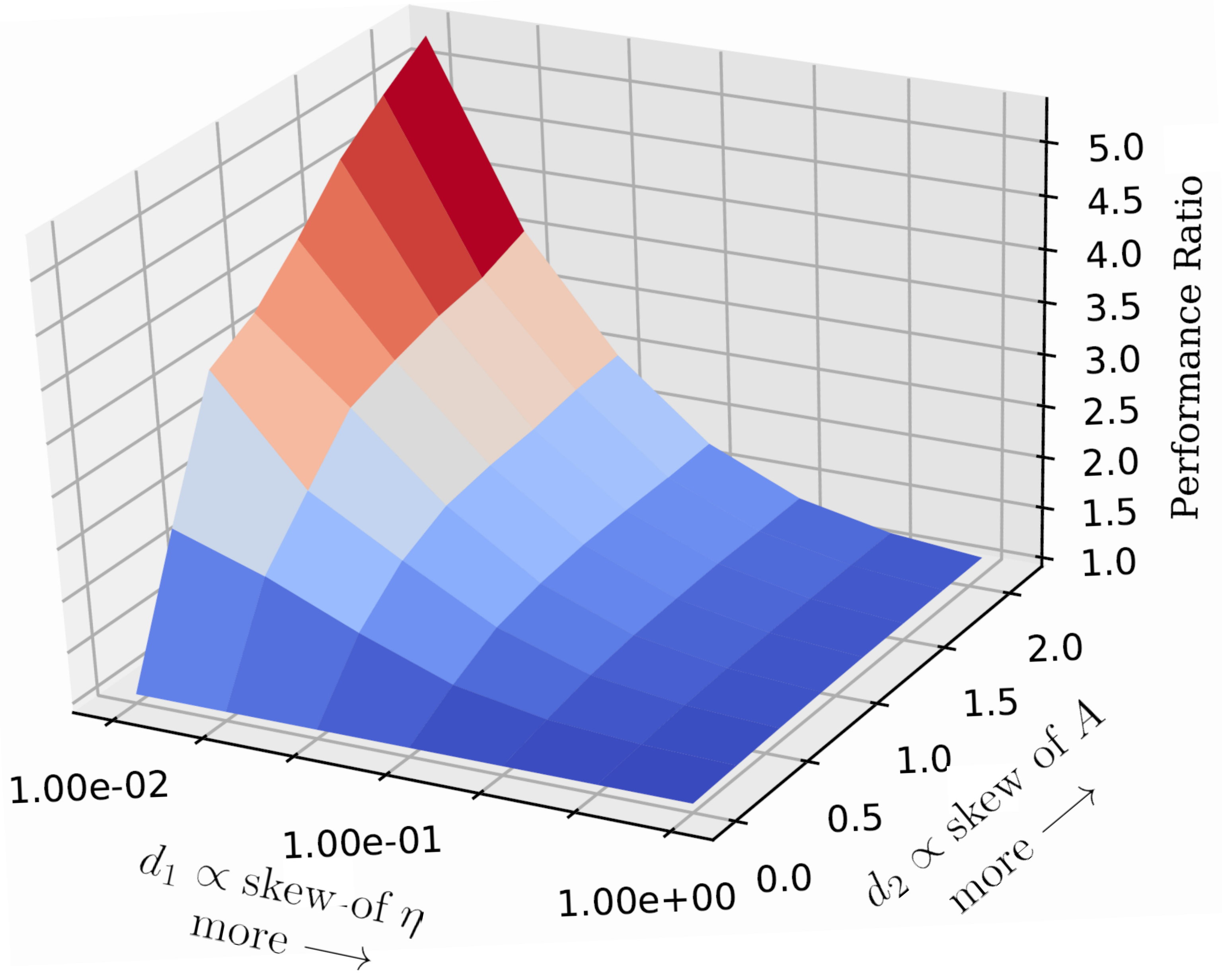
$$\mathbf{x} \sim \mathcal{N}(0, \mathbf{I});\ \ \mathbf{w}_k = d_1|k - K|\mathbf{1}$$

$$A_{j,j} = e^{-d_2 j}$$

1. Generate random data from model
2. Fit a logistic regression model
3. Post-process predictions

$$\text{Performance Ratio} = \frac{\text{Perf. of weighted postprocess}}{\text{Perf. of std. prediction}}$$

Simple re-weighting can have a huge effect!



$$\max_h \psi\big(C(h)\big)$$

$$s_i \approx p(y = i|x)$$

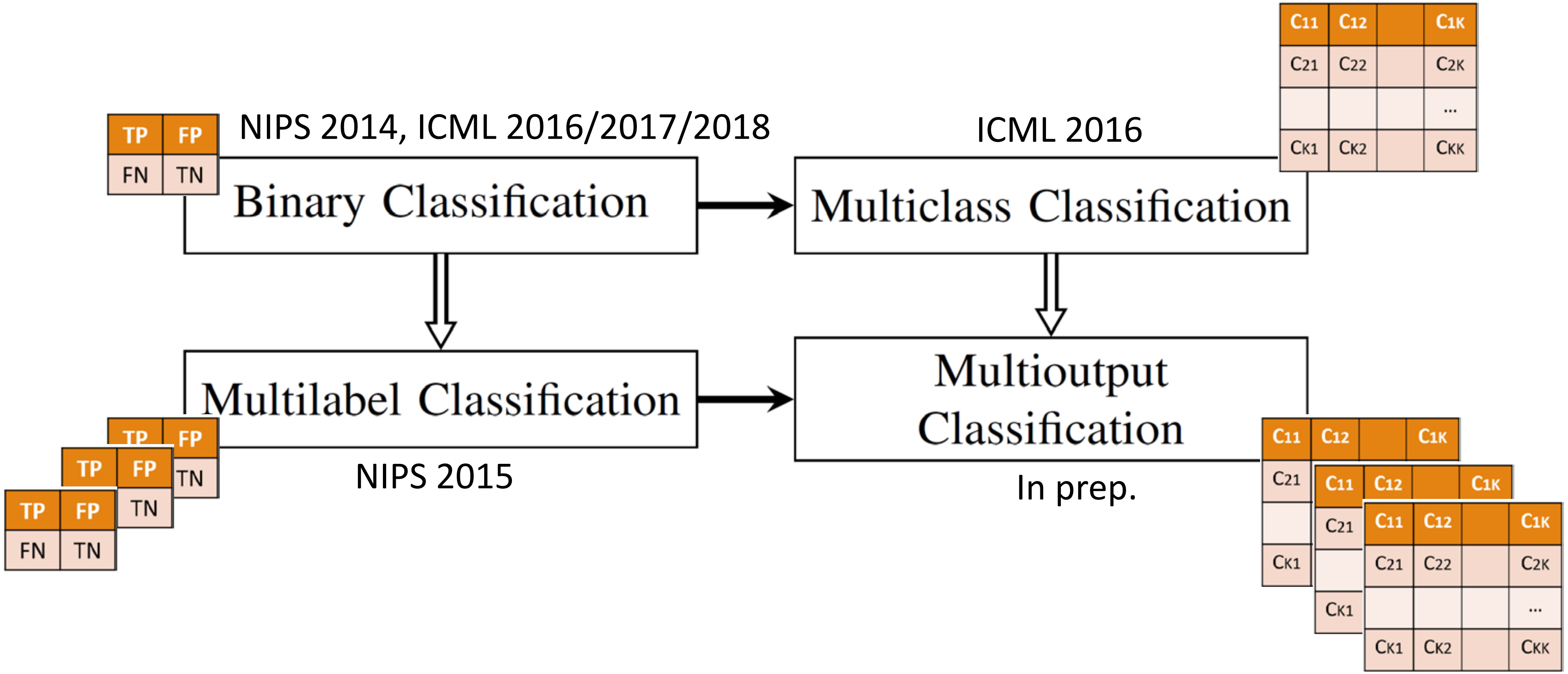
any calibrated classifier

$$h(x) = \operatorname*{argmax}_{i \in [K]} \mathbf{b}_i^\top \mathbf{s}$$

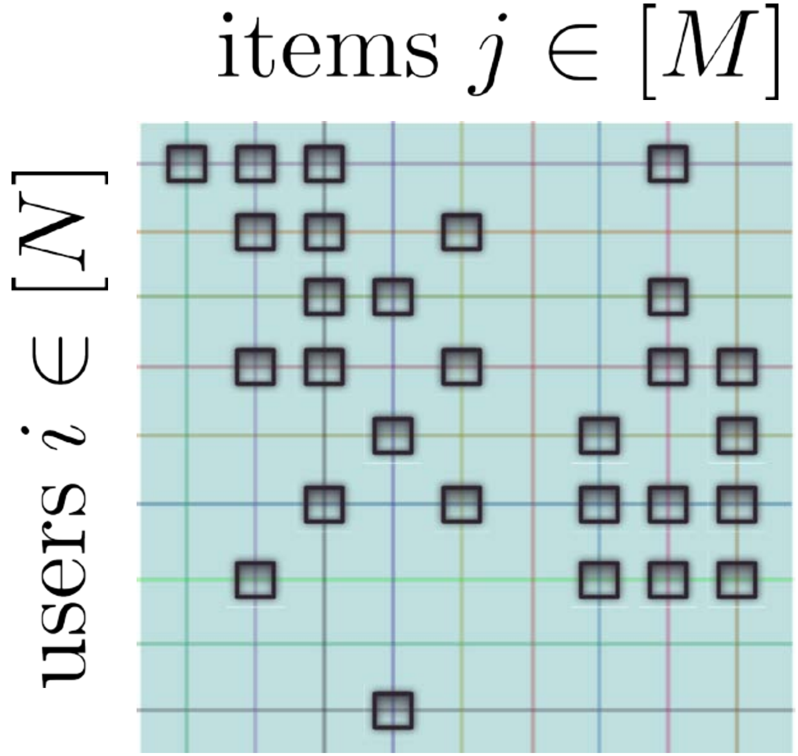
$$B = \nabla\psi|_{C=C^*}$$

Same strategy works for more complex metrics

Applies to more general settings



An application to recommender systems



User assigns rating to each item.

$$r_{i,j} \in [K]$$

Solve this as simultaneous (over items) multiclass classification problem i.e. multioutput classification

$$\phi(h) = \sum_{i=1}^{K} \sum_{j=1}^{K} |i - j| C_{i,j}$$

Postprocessed OrdRec

$$s_i(x) \approx p(y = i|x)$$

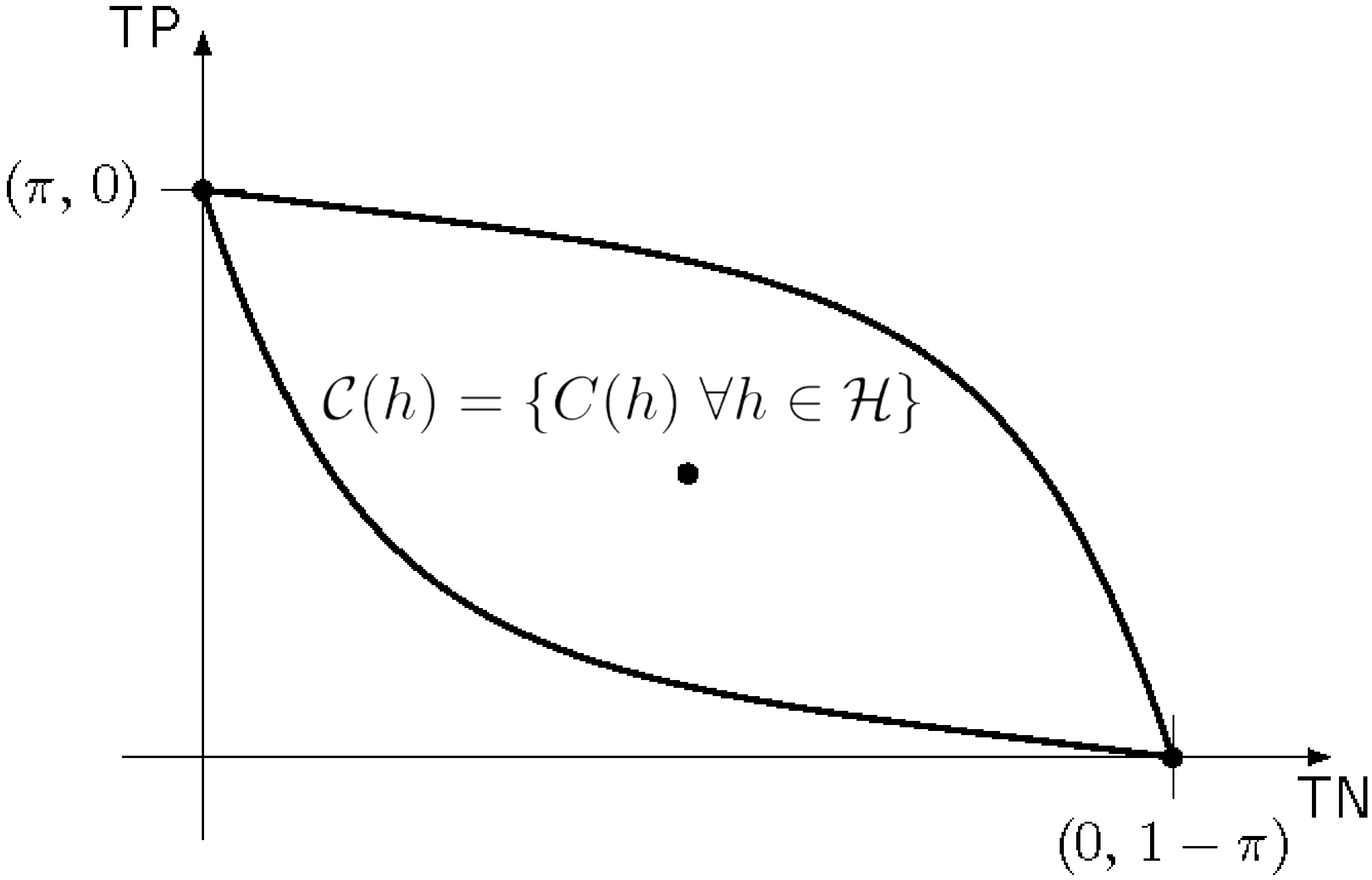
$$\operatorname*{argmax}_{i \in [K]} \mathbf{s}_i(x) \quad \operatorname*{argmax}_{i \in [K]} \mathbf{a}_i^\top \mathbf{s}(x)$$

Koren, Yehuda, and Joe Sill. "OrdRec: an ordinal model for predicting personalized item rating distributions." *Recsys* 2011.

AVERAGE	ORDREC	C-ORDREC
MICRO	0.8603±0.0010	**0.8640±0.0009**
MACRO	0.8577±0.0032	**0.8643±0.0022**
INSTANCE	0.8565±0.0014	**0.8619±0.0011**

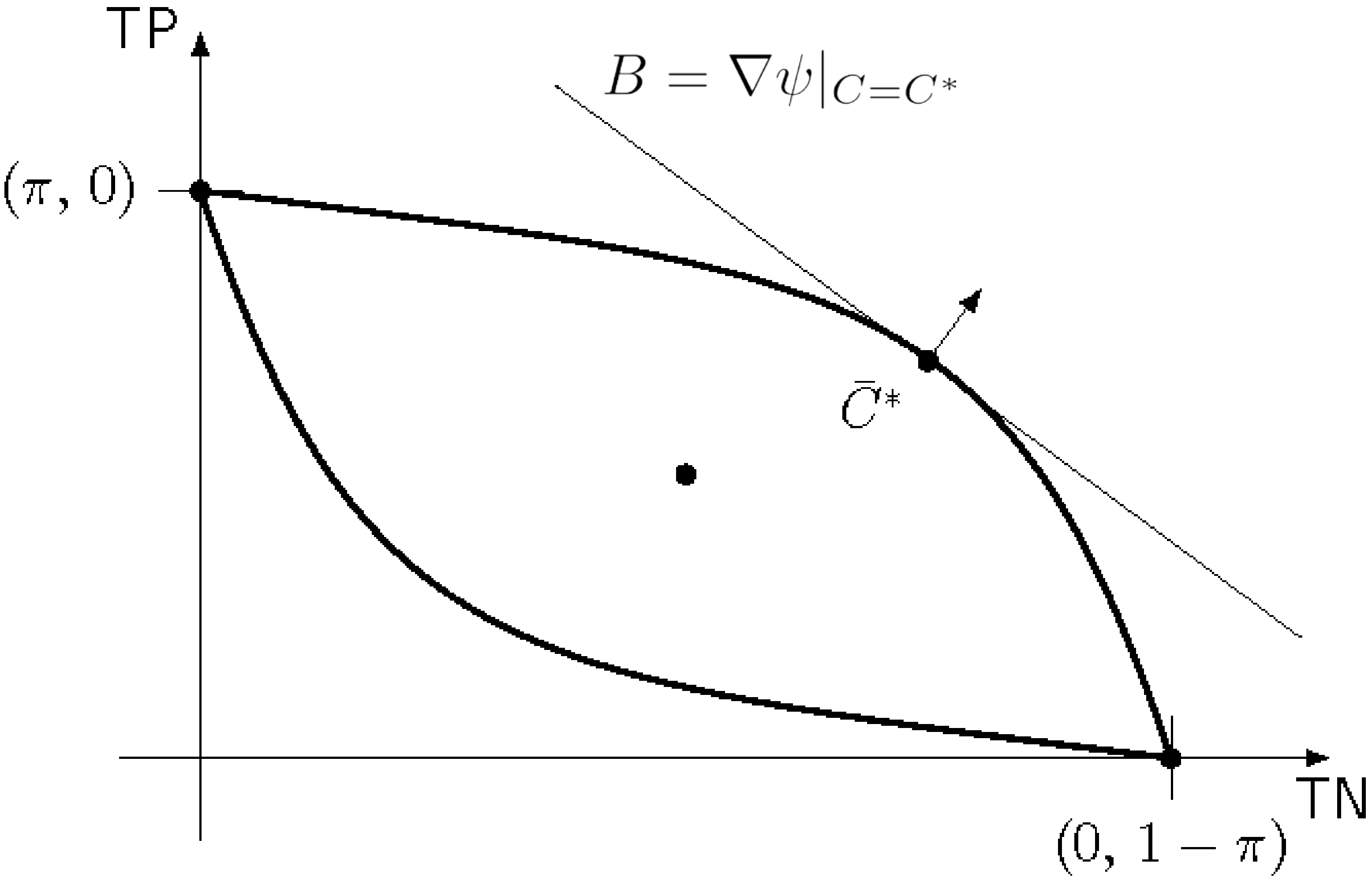
When & Why does re-weighting work?

THE GEOMETRY OF CONFUSION



$\mathrm{TP} + \mathrm{FN} = \pi,\ \mathrm{TN} + \mathrm{FP} = 1 - \pi$

- Set of feasible confusion matrices is a bounded convex set
- Optimization properties will depend on how gradient field of the metric interacts with the feasible set
- Any monotonic metric will be optimized at the boundary



- All points on the boundary are determined by the support function
- This characterization is exhaustive i.e. characterizes ALL metrics that are consistently optimizable via linear post-processing

$$s_i(x) \to p(y = i|x)$$

$$B \to \nabla\psi|_{C=C^*}$$

$$\operatorname*{argmax}_{i\in[K]} \mathbf{b}_i^\top \mathbf{s} \to h^*(x)$$

This classification strategy is consistent

Binary classification with general metrics

$$s(x) \approx p(y = i|x)$$

Logistic regression w/ MLE
Holder densities w/ kernel approx.

$$g_\delta(x) = \text{sign}\left(s(x) - \hat{\delta}\right)$$

Plug-in classifier

$$\hat{h}_n(x) = \underset{\delta \in [0,1]}{\text{argmax}}\ \phi_n\left(g_\delta\right)$$

Threshold search

$$|\phi(h^*) - \phi(\hat{h}_n)| \leq O\left(\frac{\log n}{n}\right)$$

Yan, **K.**, Zhong, Ravikumar (2018)

Which metric should you use?

THE BINARY CLASSIFICATION CASE

Recall: Lots of real world examples

$$\phi(h) = a_1\mathrm{TP} + a_2\mathrm{FP} + a_3\mathrm{FN} + a_4\mathrm{TN}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},\ \mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{FP}+\mathrm{TN}},\ \mathrm{Prec} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}},\ \mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TP}},\ \mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FN}}.$$

$$\mathrm{AM} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\pi} + \frac{\mathrm{TN}}{1-\pi}\right) = \frac{(1-\pi)\mathrm{TP} + \pi\mathrm{TN}}{2\pi(1-\pi)},\ F_\beta = \frac{(1+\beta^2)\mathrm{TP}}{(1+\beta^2)\mathrm{TP} + \beta^2\mathrm{FN} + \mathrm{FP}} = \frac{(1+\beta^2)\mathrm{TP}}{\beta^2\pi + \gamma},$$

$$\mathrm{JAC} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}+\mathrm{FP}} = \frac{\mathrm{TP}}{\pi + \mathrm{FP}} = \frac{\mathrm{TP}}{\gamma + \mathrm{FN}},\quad \mathrm{WA} = \frac{w_1\mathrm{TP} + w_2\mathrm{TN}}{w_1\mathrm{TP} + w_2\mathrm{TN} + w_3\mathrm{FP} + w_4\mathrm{FN}}.$$

Limited formal guidance

Academia:
Use the standard metric in your application area

- Accuracy
- Top-K accuracy
- F1 measure

Industry:
Hire a consultant or economist

- User survey
- A/B tests

Our Approach

Query an "expert" to determine the real-world value of a classifier i.e. the ideal evaluation metric

$$\phi(h) = ?$$

Pairwise queries

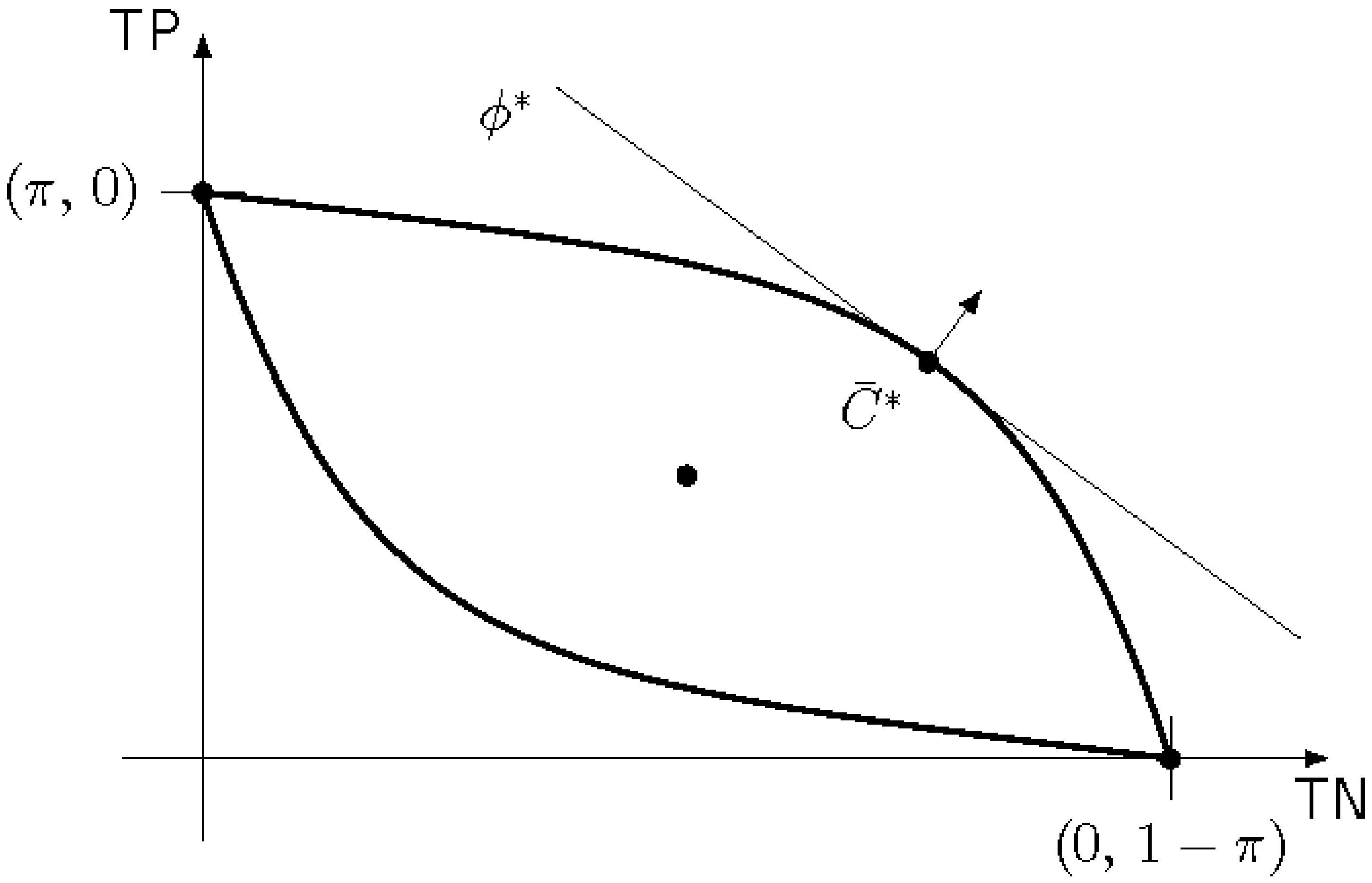
Experts give inaccurate results for value queries

More accurate results for comparison queries

$$\phi(h_1) \text{ vs. } \phi(h_2) ?$$

THE “ORACLE” CARES ABOUT WORST CASE QUERY COMPLEXITY

Speed Matters!

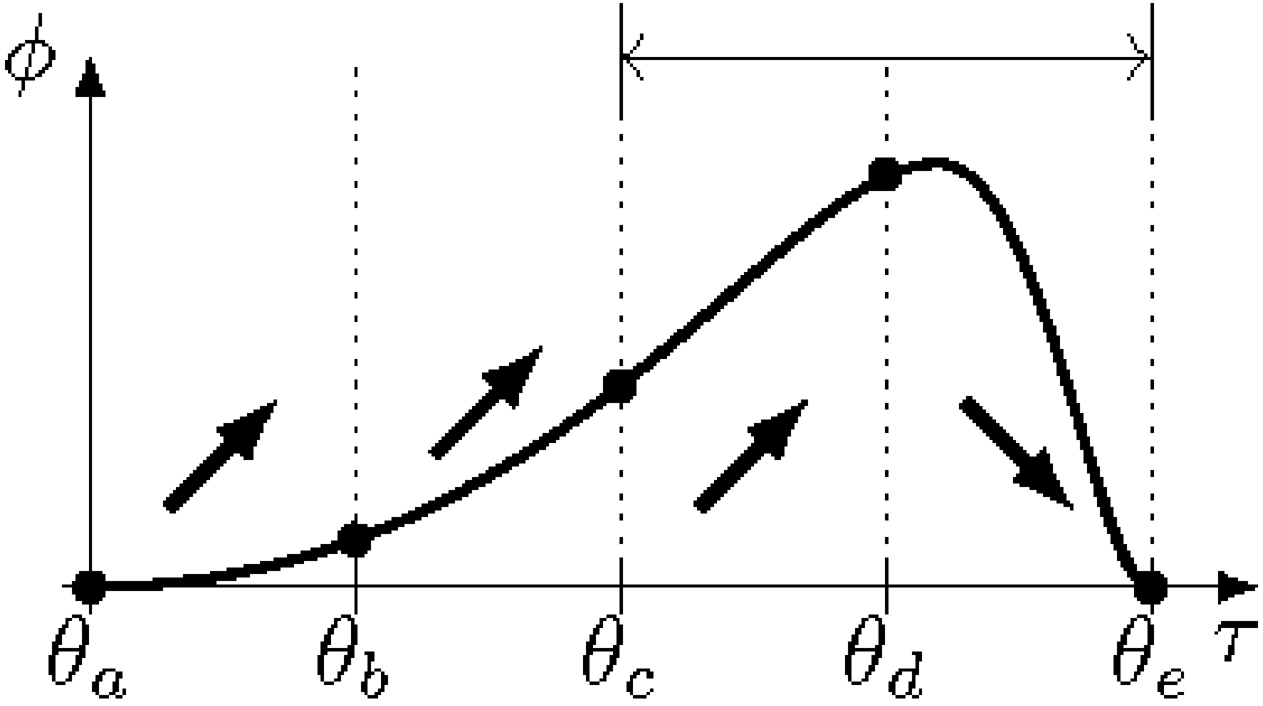


Exploiting the geometry…

- Only need to query classifiers on the boundary – since we already know optimal is within this subset
- Boundary is one-dimensional, parameterized by “angle”

Using binary search

- Under weak conditions, metric is unimodal with respect to boundary
- Thus, can simple binary search to find the optimal confusion matrix
- Simultaneously recovers gradient of the optimal metric



Guaranteed recovery with finite queries

For the linear case, when algorithm terminates, we recover

$$\phi^*(h) = a_1^* \mathrm{TP}(h) + a_2^* \mathrm{FP}(h) + a_3^* \mathrm{FN}(h) + a_4^* \mathrm{TN}(h)$$

Guaranteed to be ϵ accurate after $\mathcal{O}\left(\log(\frac{1}{\epsilon})\right)$ steps

If no additional assumptions, this matches lower bound

Stable to system noise e.g. noisy responses from the "expert"

Conclusion

Metric choice has a large impact on real-world machine learning performance.

1

Re-weighted post-processing is efficient for optimizing complex metrics.

2

Can reduce metric elicitation for binary classifiers to binary search with bounded query complexity.

Measurement is at the core of empirical research

- Extensions to other machine learning problems e.g. ranking, regression, …
- Faster elicitation using alternative query mechanisms
- Noise tolerance, robust elicitation

Thank you

QUESTIONS?

SANMI@ILLINOIS.EDU