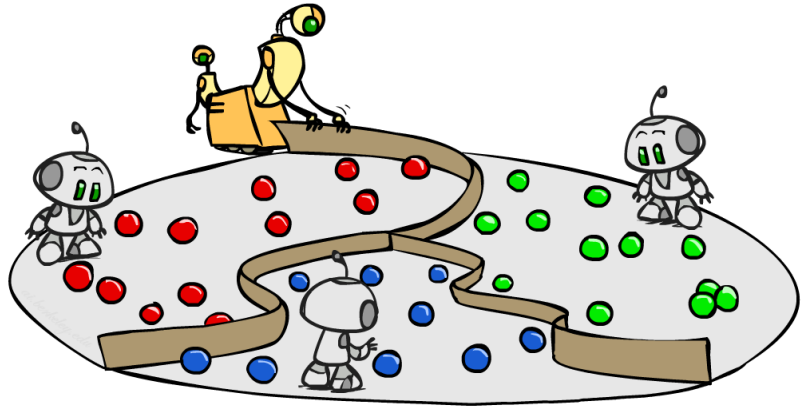
Machine Learning III: Classification

CSE 473: Introduction to Artificial Intelligence



James Weichert (jpw@cs)
cs.uw.edu/473
acknowledgements

Welcome!
What's your favorite summertime activity?
sli.do **#cse473**

Administrivia

ANNOUNCEMENTS

- **Test 2** Friday
  - One double-sided 8.5x11" note sheet allowed
  - Focus on RL and probabilistic inference
  - One question on basic ML

ASSIGNMENTS

- **Homework 3 due** tomorrow (8.6)
- **Project 4 due** next week (8.13)

Recap: Supervised Learning

FOUNDATIONS

Goal: Learn unknown **target function** f

- Input: **Training data** with *labeled* examples (x_i, y_i), where $f(x_i) = y_i$
- Output: **Hypothesis** h that is "close" to f
  - $h \approx f$ implies that h predicts well on unseen (test set) data
- Knowledge: Family of hypotheses H to choose h from
  - Linear models, logistic regression, neural networks, decision trees, non-parametric models, etc.
- Classification: Learn f with discrete output value

Gradient Descent

DEFINITION

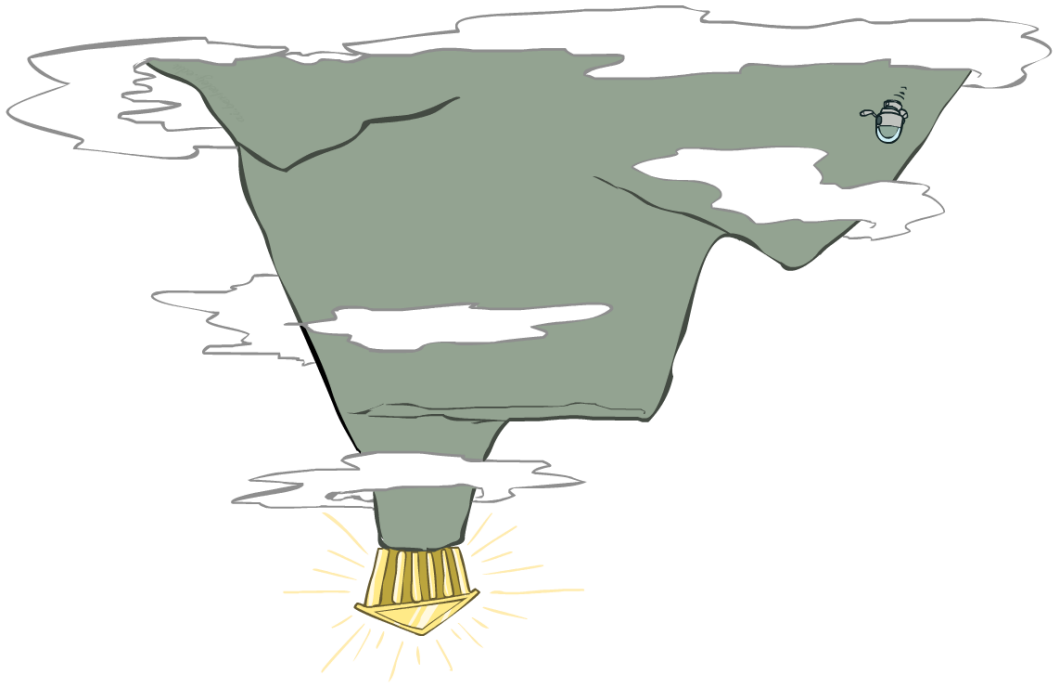
Gradient descent is a *greedy* optimization approach

- Take small steps in the direction of the steepest descent (negative gradient)

$$w \leftarrow w - \eta \frac{\partial}{\partial w} \mathcal{L}$$

  - Stop when updates become too small

- Considerations
  - How to find, calculate gradient?
  - Step size η matters a lot!
  - When to stop?



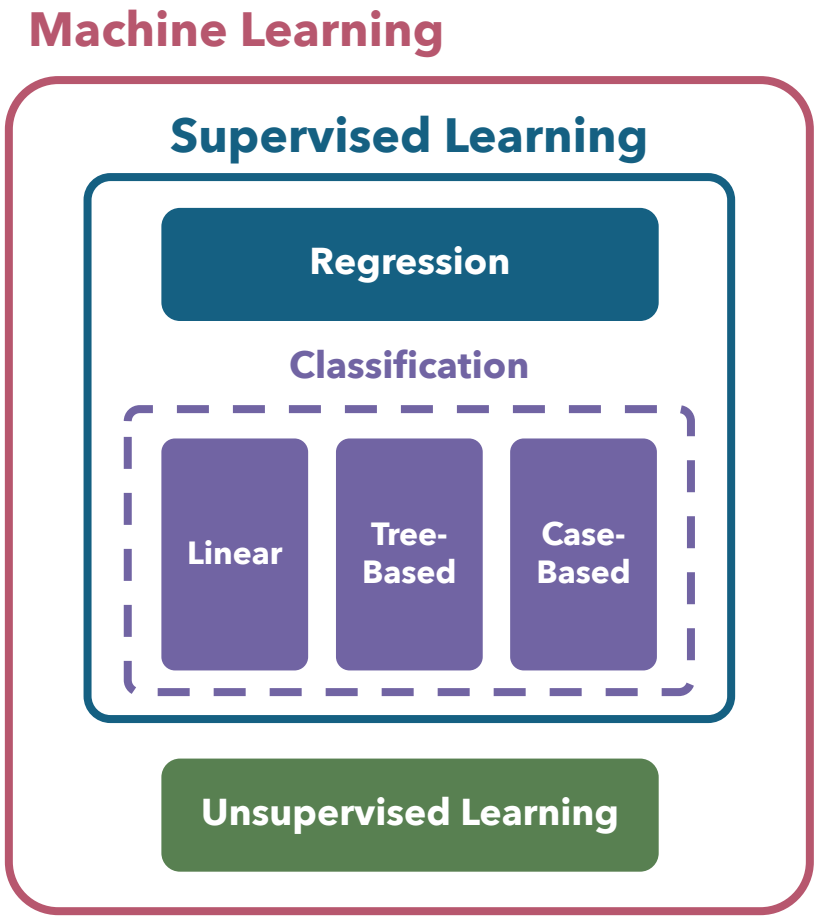
Classification Approaches

CONTEXT

Goal: Learn unknown discrete **target function** f

- 'Linear' Classifiers (linear in *weights*)
  - Perceptron
  - Logistic Regression
  - Support Vector Machines
- Tree-Based Classifiers
  - Naïve Bayes
  - Decision Tree
- Case-Based (Non-Parametric) Classifiers
  - K-Nearest Neighbors

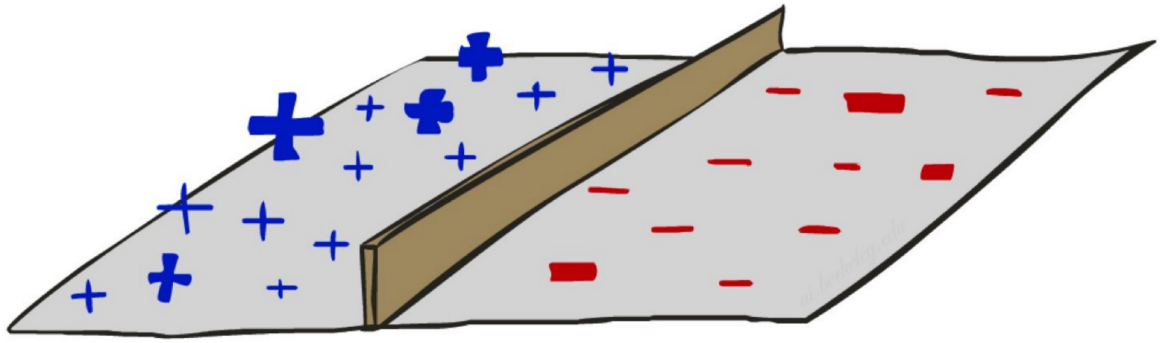
Learn likeliest hypothesis given data

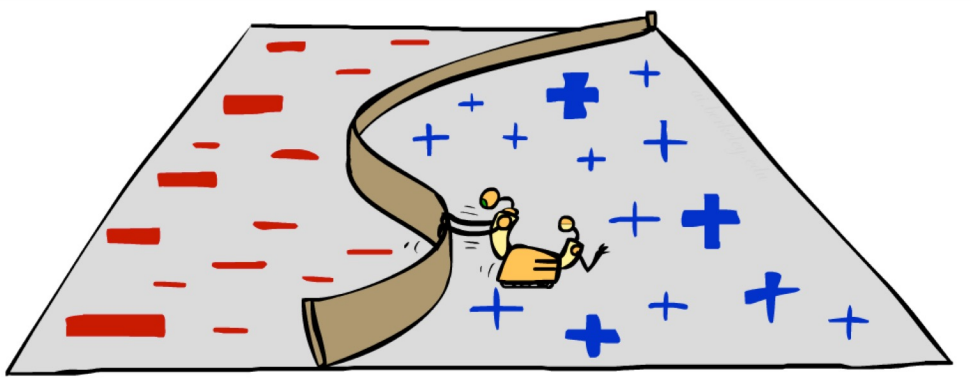


Separability

CONTEXT

What happens when classes are not (linearly) **separable**?





Logistic Regression

DEFINITION

Logistic regression is a *classification* model that fits (regresses) a sigmoid curve to approximate the class probabilities

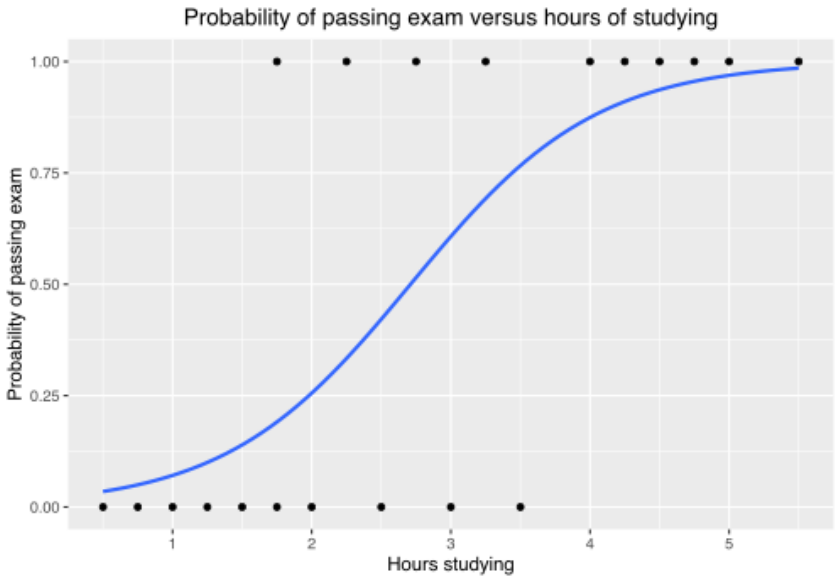
$$h_{\boldsymbol{w}}(x) = P(x+) = \sigma(w_0 + w_1 x_1) = \frac{1}{1 + e^{-(w_0 + w_1 x_1)}}$$

- $h_{\boldsymbol{w}}(x) > 0.5 \Longrightarrow$ positive
- $h_{\boldsymbol{w}}(x) \leq 0.5 \Longrightarrow$ negative

Linear in weights, nonlinear in output!

- Approach: *Maximize* joint likelihood $L_h(\boldsymbol{x}, \boldsymbol{y})$

$$L_h(x, y) = \prod_i \underbrace{h_{\boldsymbol{w}}(x_i)^{y_i}}_{y_i = 1} \cdot \underbrace{(1 - h_{\boldsymbol{w}}(x_i))^{1-y_i}}_{y_i = 0}$$



Support Vector Machines

DEFINITION

SVMs maximize the margin between the decision boundary and data points

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$$

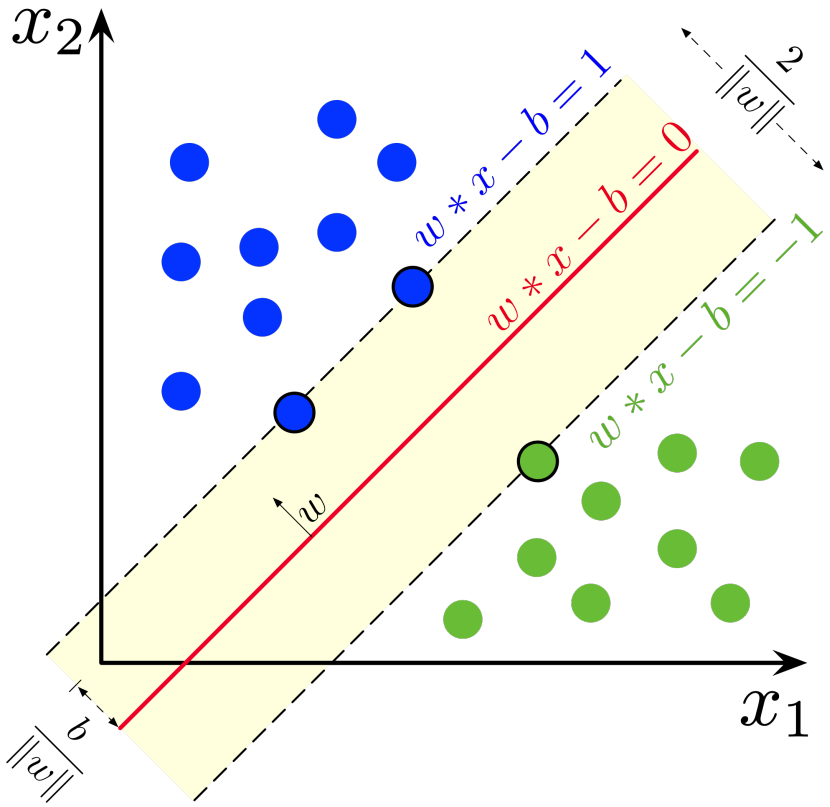
- $h_{\boldsymbol{w}}(x) > 0 \implies$ positive
- $h_{\boldsymbol{w}}(x) \leq 0 \implies$ negative
- Note: margin width is $\frac{2}{\|\boldsymbol{w}\|}$

- Approach: *Maximize* width by minimizing $\boldsymbol{w}$
  - "Hard Margin" (no misclassifications)

$$\min_{\boldsymbol{w}} \frac{1}{2} \| w \|^2 \;\; s.t. \;\; y_i(\boldsymbol{w}^\top \boldsymbol{x}_i) \geq 1 \;\; \forall i$$

  - "Soft Margin" (allow misclassifications)

$$\mathcal{L}_{\text{SM}} = \| w \|^2 + C \left[\frac{1}{n} \sum_i \max(0, 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i)) \right]$$



QUESTIONS

Questions?

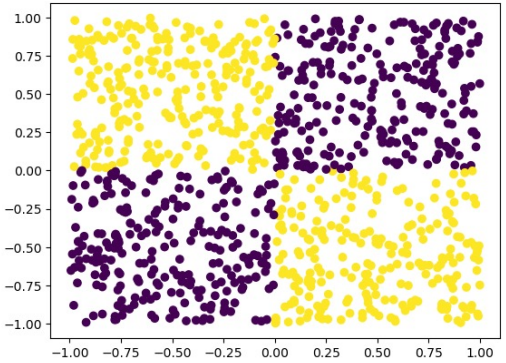
live and on sli.do #cse473

More Complicated Classifications

CONTEXT

What about more than two classes?

What if features are not independent?



XOR

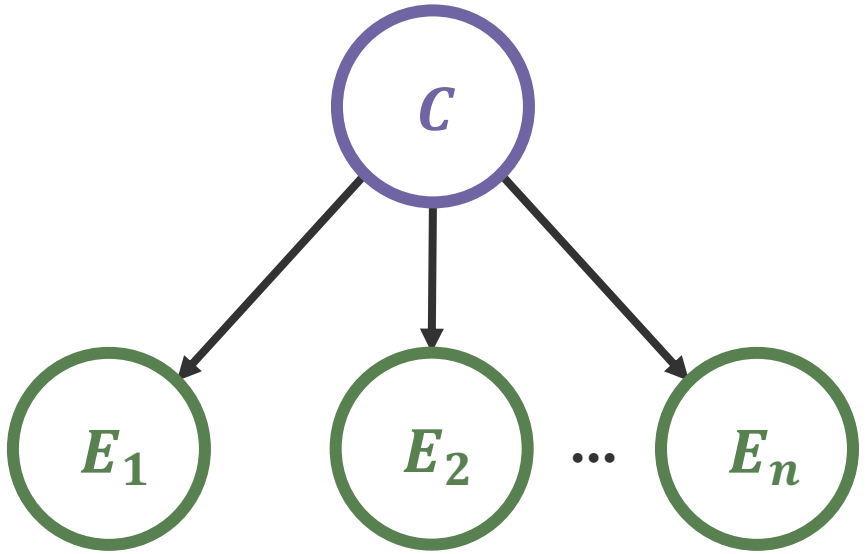
Naïve Bayes

FOUNDATIONS

Given evidence $E_1, \ldots, E_n$ what is posterior of class C?

$$P(C|E_1, \ldots, E_n) = \alpha\, P(C, E_1, \ldots, E_n) \propto P(C) \prod_i P(E_i|C)$$

"**Naïve**" because of the simplifying assumption that evidence variables are **conditionally independent** given C



- Example of **Bayesian Learning**
  - Update belief in hypothesis $h \in \mathcal{H}$ based on data D

$$P(h|D) = \alpha \cdot P(D|h) \cdot P(h)$$

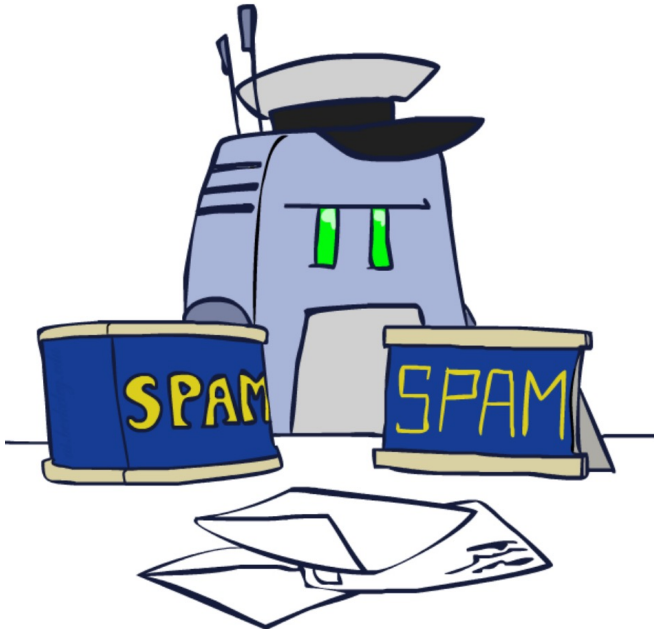
  - Predict using likelihood-weighted average over $h_k \in \mathcal{H}$

$$P(d_{N+1}|D) = \alpha \sum_k P(d_{N+1}|h_k)\, P(D|h_k) P(h_k)$$

  - Bayesian learning gives optimal predictions!
  - But need to sum over all $h_k \in \mathcal{H}$

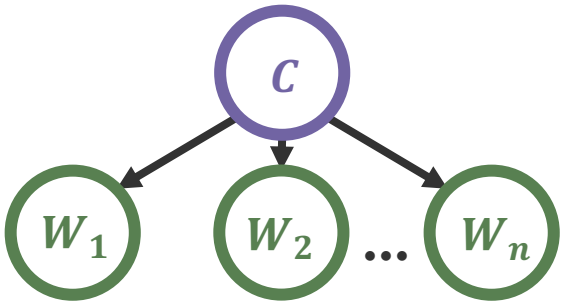
Spam Filtering

EXAMPLE



Filtering out spam emails using Naïve Bayes is easy!

- **Goal:** Classify as **spam** or **ham** (not spam)
- Setup:
  - Train with large collection of labeled example emails
  - Assume "bag-of-words" model
    - Word, regardless of position, in emails are selected randomly from likelihood $P(W_i|C)$



Dear Sir.
First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. ...

Can you meet tomorrow to discuss the project? Need to get your thoughts on a few things…

Spam or Ham?

EXAMPLE

Prior: $P(C = ham) = 0.666$

$$P(C|W_1, \dots, W_n) \propto P(C, W_1, \dots, W_n) = P(C) \prod_i P(W_i|C)$$

W	$P(W|\text{spam})$	$P(W|\text{ham})$	$P(\text{spam}, W_1, \dots, W_k)$	$P(\text{ham}, W_1, \dots, W_k)$
[PRIOR]			0.333	**0.666**

Spam or Ham? (1)

EXAMPLE

Prior: $P(C = ham) = 0.666$

$$P(C|W_1, \ldots, W_n) \propto P(C, W_1, \ldots, W_n) = P(C) \prod_i P(W_i|C)$$

W	$P(W|\text{spam})$	$P(W|\text{ham})$	$P(\text{spam}, W_1, \ldots, W_k)$	$P(\text{ham}, W_1, \ldots, W_k)$
[PRIOR]			0.333	**0.666**
James	0.00002	**0.00021**	0.046	**0.954**

Spam or Ham? (2)

EXAMPLE

Prior: $P(C = ham) = 0.666$

$$P(C|W_1, \ldots, W_n) \propto P(C, W_1, \ldots, W_n) = P(C) \prod_i P(W_i|C)$$

W	$P(W\|\text{spam})$	$P(W\|\text{ham})$	$P(\text{spam}, W_1, \ldots, W_k)$	$P(\text{ham}, W_1, \ldots, W_k)$
[PRIOR]			0.333	**0.666**
James	0.00002	**0.00021**	0.046	**0.954**
would	0.00069	**0.00084**	0.038	**0.962**

Spam or Ham? (3)

EXAMPLE

Prior: $P(C = ham) = 0.666$

$$P(C|W_1, \ldots, W_n) \propto P(C, W_1, \ldots, W_n) = P(C) \prod_i P(W_i|C)$$

W	$P(W\|\text{spam})$	$P(W\|\text{ham})$	$P(\text{spam}, W_1, \ldots, W_k)$	$P(\text{ham}, W_1, \ldots, W_k)$
[PRIOR]			0.333	**0.666**
James	0.00002	**0.00021**	0.046	**0.954**
would	0.00069	**0.00084**	0.038	**0.962**
you	**0.00881**	0.00304	0.103	**0.897**

Spam or Ham? (4)

EXAMPLE

Prior: $P(C = ham) = 0.666$

$$P(C|W_1, \ldots, W_n) \propto P(C, W_1, \ldots, W_n) = P(C) \prod_i P(W_i|C)$$

W	$P(W\|\text{spam})$	$P(W\|\text{ham})$	$P(\text{spam}, W_1, \ldots, W_k)$	$P(\text{ham}, W_1, \ldots, W_k)$
[PRIOR]			0.333	**0.666**
James	0.00002	**0.00021**	0.046	**0.954**
would	0.00069	**0.00084**	0.038	**0.962**
you	**0.00881**	0.00304	0.103	**0.897**
like	0.00086	0.00083	0.106	**0.894**

Spam or Ham? (5)

EXAMPLE

Prior: $P(C = ham) = 0.666$

$$P(C|W_1, \ldots, W_n) \propto P(C, W_1, \ldots, W_n) = P(C) \prod_i P(W_i|C)$$

W	$P(W\|\text{spam})$	$P(W\|\text{ham})$	$P(\text{spam}, W_1, \ldots, W_k)$	$P(\text{ham}, W_1, \ldots, W_k)$
[PRIOR]			0.333	**0.666**
James	0.00002	**0.00021**	0.046	**0.954**
would	0.00069	**0.00084**	0.038	**0.962**
you	**0.00881**	0.00304	0.103	**0.897**
like	0.00086	0.00083	0.106	**0.894**
to	0.01517	0.01339	0.118	**0.882**

Spam or Ham? (6)

EXAMPLE

Prior: $P(C = ham) = 0.666$

$$P(C|W_1, \ldots, W_n) \propto P(C, W_1, \ldots, W_n) = P(C) \prod_i P(W_i|C)$$

W	$P(W\|\text{spam})$	$P(W\|\text{ham})$	$P(\text{spam}, W_1, \ldots, W_k)$	$P(\text{ham}, W_1, \ldots, W_k)$
[PRIOR]			0.333	**0.666**
James	0.00002	**0.00021**	0.046	**0.954**
would	0.00069	**0.00084**	0.038	**0.962**
you	**0.00881**	0.00304	0.103	**0.897**
like	0.00086	0.00083	0.106	**0.894**
to	0.01517	0.01339	0.118	**0.882**
lose	**0.00008**	0.00002	0.349	**0.651**

Spam or Ham? (7)

EXAMPLE

Prior: $P(C = ham) = 0.666$

$$P(C|W_1, \ldots, W_n) \propto P(C, W_1, \ldots, W_n) = P(C) \prod_i P(W_i|C)$$

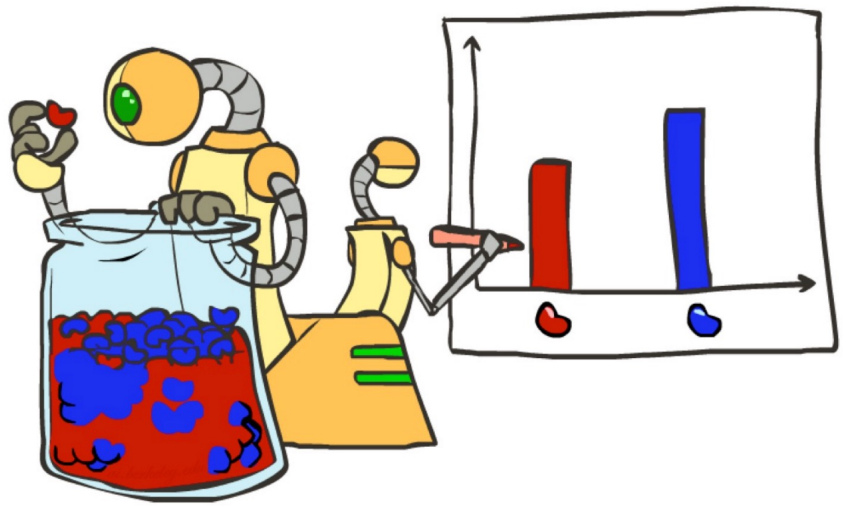
W	$P(W\|\text{spam})$	$P(W\|\text{ham})$	$P(\text{spam}, W_1, \ldots, W_k)$	$P(\text{ham}, W_1, \ldots, W_k)$
[PRIOR]			0.333	**0.666**
James	0.00002	**0.00021**	0.046	**0.954**
would	0.00069	**0.00084**	0.038	**0.962**
you	**0.00881**	0.00304	0.103	**0.897**
like	0.00086	0.00083	0.106	**0.894**
to	0.01517	0.01339	0.118	**0.882**
lose	**0.00008**	0.00002	0.349	**0.651**
weight	**0.00016**	0.00002	**0.811**	0.189

Statistical Learning

FOUNDATIONS

- Consider
  - Independent and identically distributed data $D = (x_1, y_1), \ldots (x_N, y_N)$
  - Probabilistic hypothesis $h \in \mathcal{H}$ with $P_h(y|x)$
- Maximum Likelihood $P(D|h) = \prod_j P_h(y_j|x_j)$
  - $h_{\text{ML}} = \text{argmax}_h P(D|h)$
- Maximum a Posteriori $P(D|h) \cdot P(h)$
  - $h_{\text{MAP}} = \text{argmax}_h P(h|D) = \text{argmax}_h P(D|h) \cdot P(h)$
- Bayesian Learning

$$P(y|x, D) = \sum_h P_h(y|x) \cdot P(D|h) \cdot P(h)$$



Decision Trees

FOUNDATIONS

Analogous to human approach to classification (?)

- Iteratively partition feature space based on feature values
  - One approach: partition based on what feature-cutoff pair provides the most 'information'
- Expressivity
  - May increase chance of expressing true function f, but at the cost of overfitting
  - Hypothesis space is generally *astronomical*!
- **Goal:** *Compact* tree

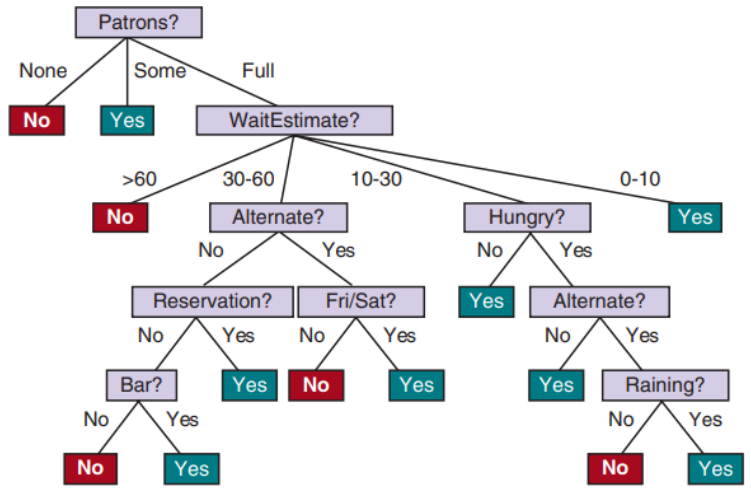


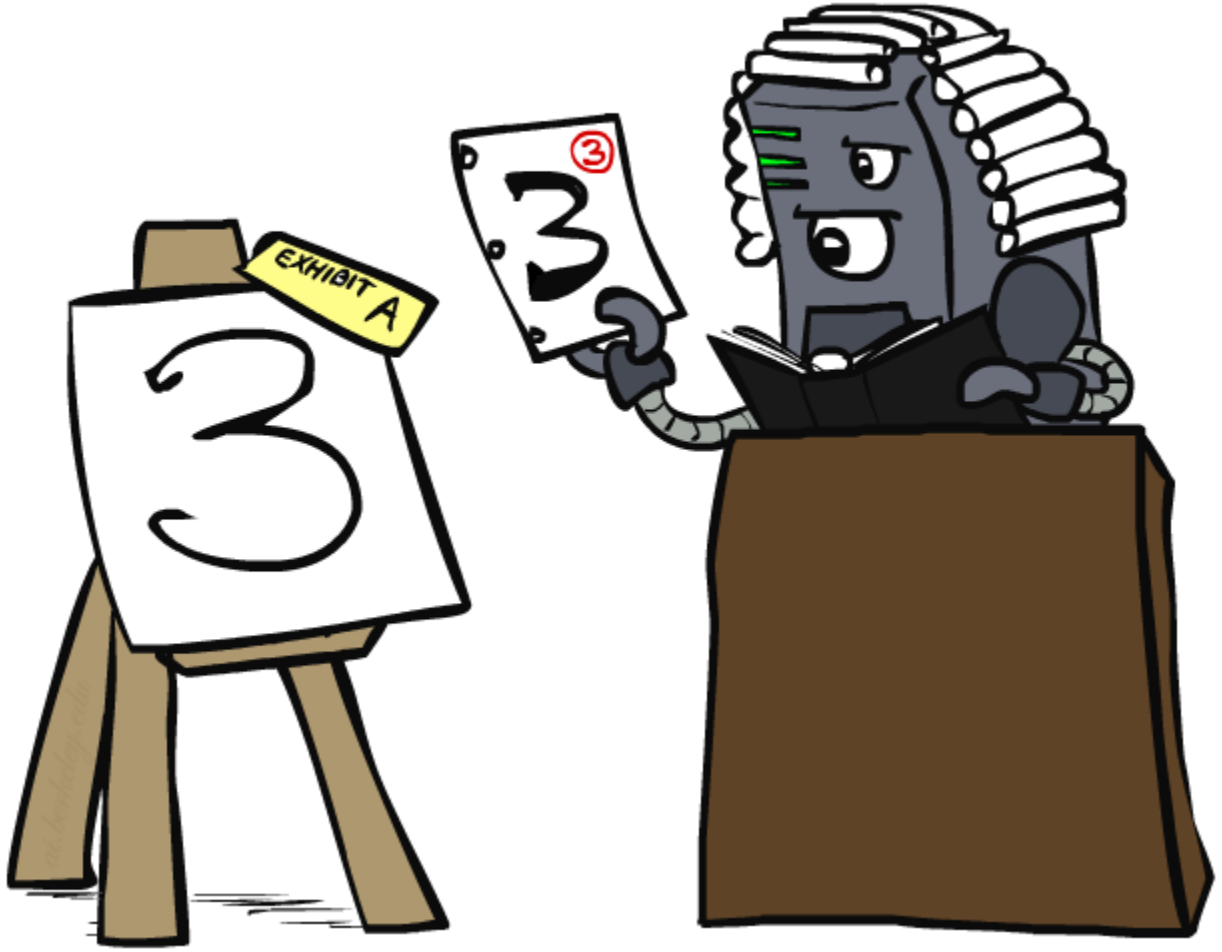
Figure 19.3 A decision tree for deciding whether to wait for a table.

Questions?

live and on sli.do #cse473

Case-Based Learning

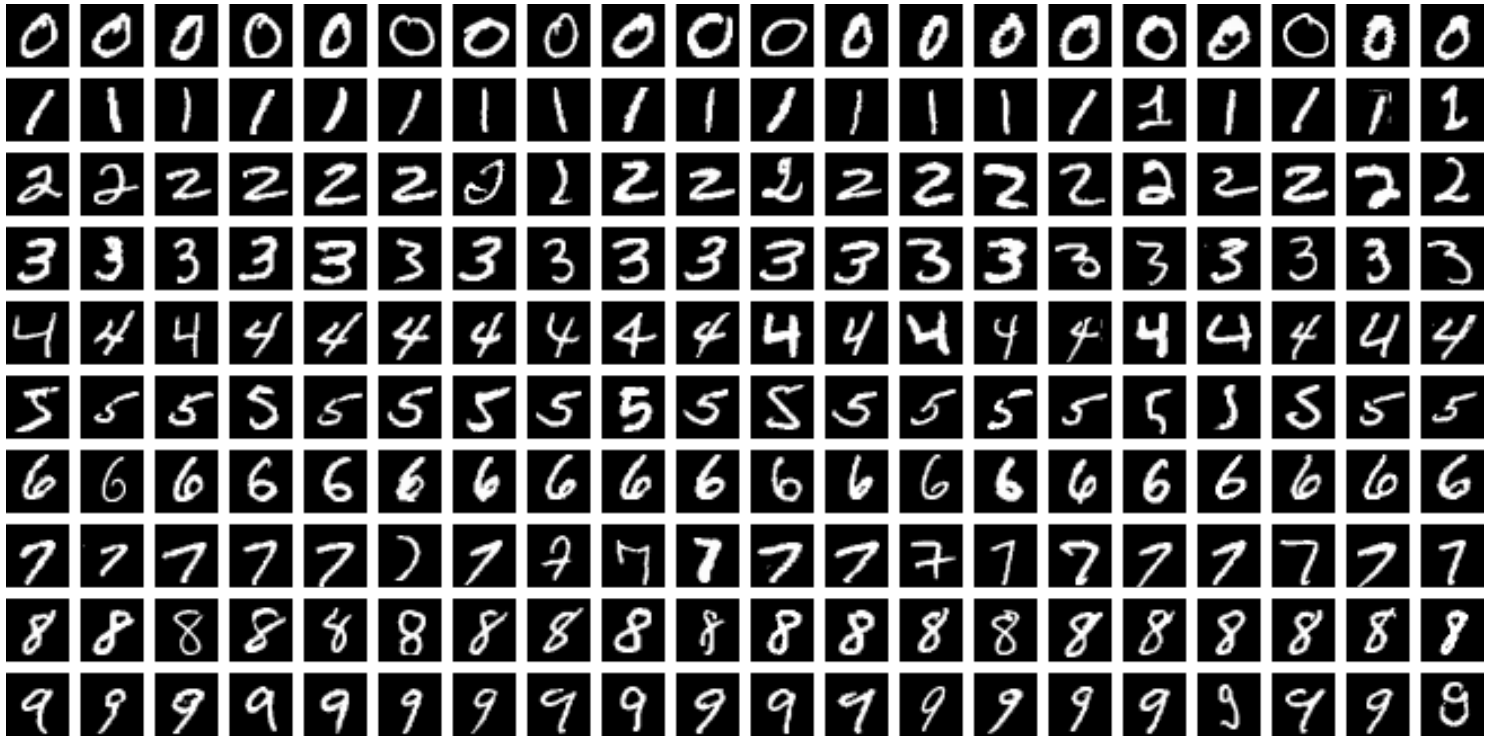
CONTEXT



Digit Classification

EXAMPLE

The MNIST handwritten digit dataset contains 70,000 28x28 images of digits



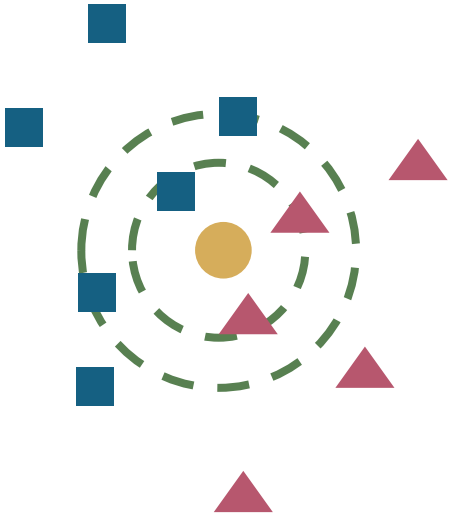
Nearest Neighbors

FOUNDATIONS

Classify based on similarity with other (labeled) instances

- Non-parametric because output depends on specific training data points

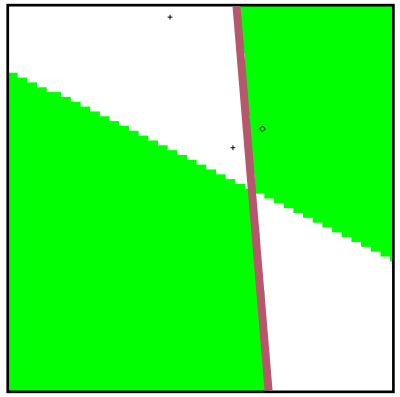
- **k-nearest neighbors** picks the majority class among the k closest neighbors in the training set
  - Training complexity is 0! (but at test time, need to compare to all neighbors)
  - Curse of dimensionality
  - How to define *similarity*?



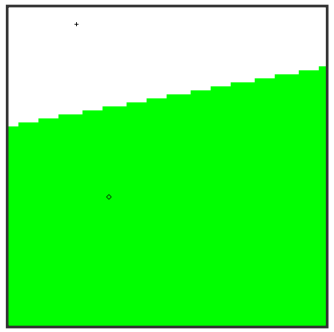
Parametric vs. Non-Parametric

DEFINITION

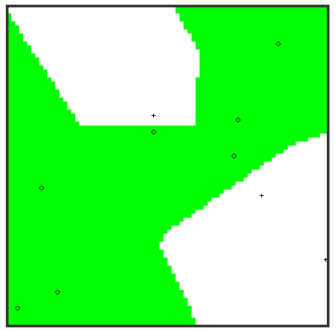
- Parametric Models
  - Fixed set of parameters, don't *depend* on data
  - More data means more learning, better accuracy
- Non-Parametric Models
  - Model output depends on specific training data
  - Complexity of the classifier increases with data
  - Better in the limit, often worse in the non-limit



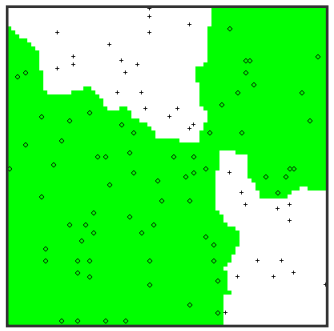
Ground Truth



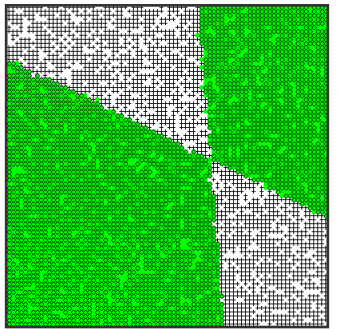
2 Examples



10 Examples



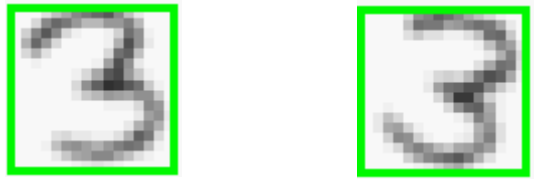
100 Examples



10000 Examples

Similarity

FOUNDATIONS

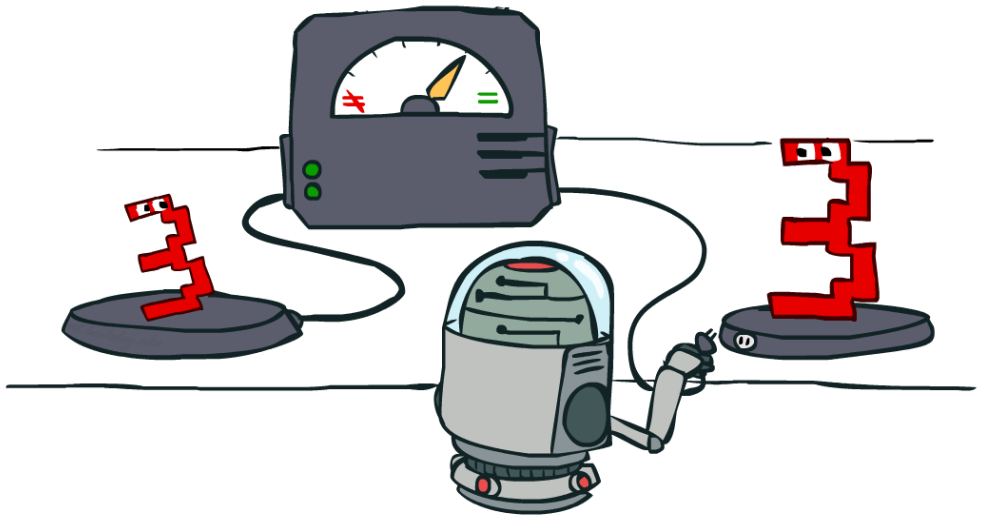


Distance is often a poor measure of similarity

- Many similarities involve dot products of features:

$$\text{sim}(x, x') = f(x) \cdot f(x') = \sum_i f_i(x) \cdot f_i(x)$$

So why not create a better similarity function?



Invariant Metrics

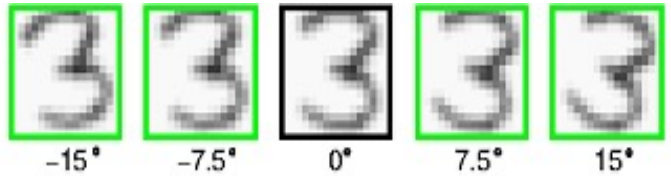
DEFINITION

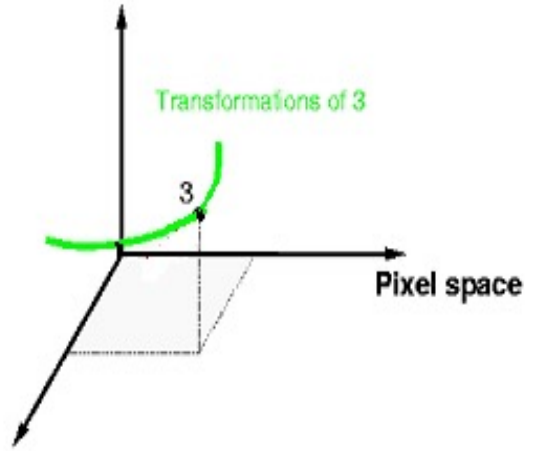
Better similarity functions for digit classification use **knowledge** about vision

- **Invariant metrics** make similarity invariant under certain transformations

- Rotation Invariant Similarity

$$\text{sim}_{\text{rotation}} = \max[\text{sim}(r(\ ), r(\ ))]$$





- Each example is now a curve in $\mathbb{R}^{256}$

QUESTIONS

Questions?

live and on sli.do #cse473

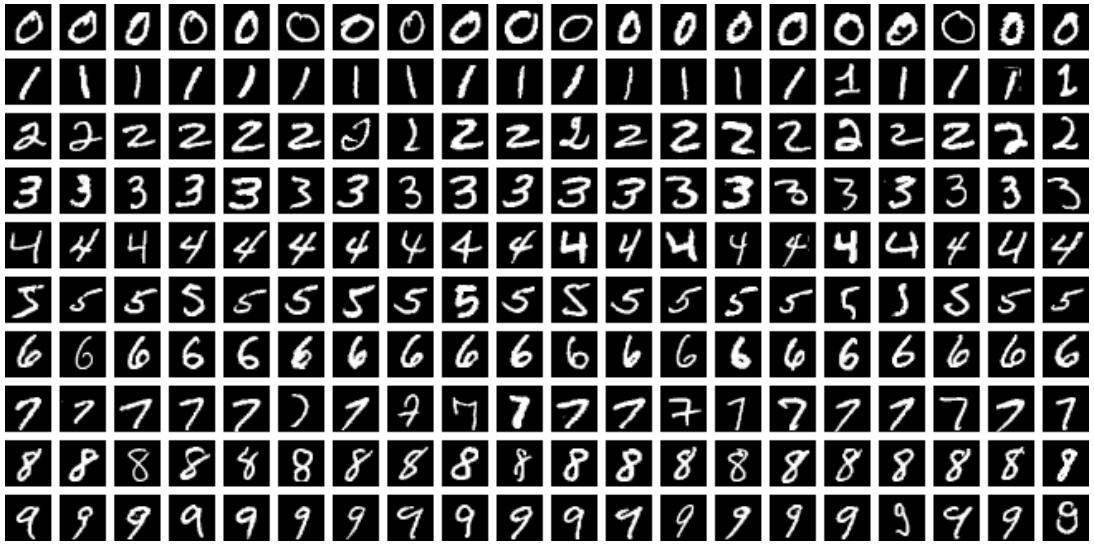
Taking Stock

CONTEXT

How do these models compare?

Model	Test Acc.	Train Time	Test Time
Logistic Reg.	92.55%	176.6s	0.01s
SVM (RBF)	96.85%	2.9s	7.74s
Decision Tree	87.58%	8.2s	0.01s
kNN (k=5)	96.88%	0.1s	3.58s
Neural Net.	?	?	?

One-shot benchmark using standard sklearn implementations (no hyperparameter tuning). Python notebook generated by Claude Code.



Benchmark experiment Inspired by Michael Nielsen

that's it for today!

SUMMARY

- Linear, tree-based, and case-based approaches to classification
- Tradeoffs in expressivity, training/inference time, interpretability

UPCOMING

- **Test 2**
- Neural Networks

REMINDERS

- Complete **Practice Problem 17**
- Do **Homework 3** (due 8.6)
- Do **Project 4** (due 8.13)