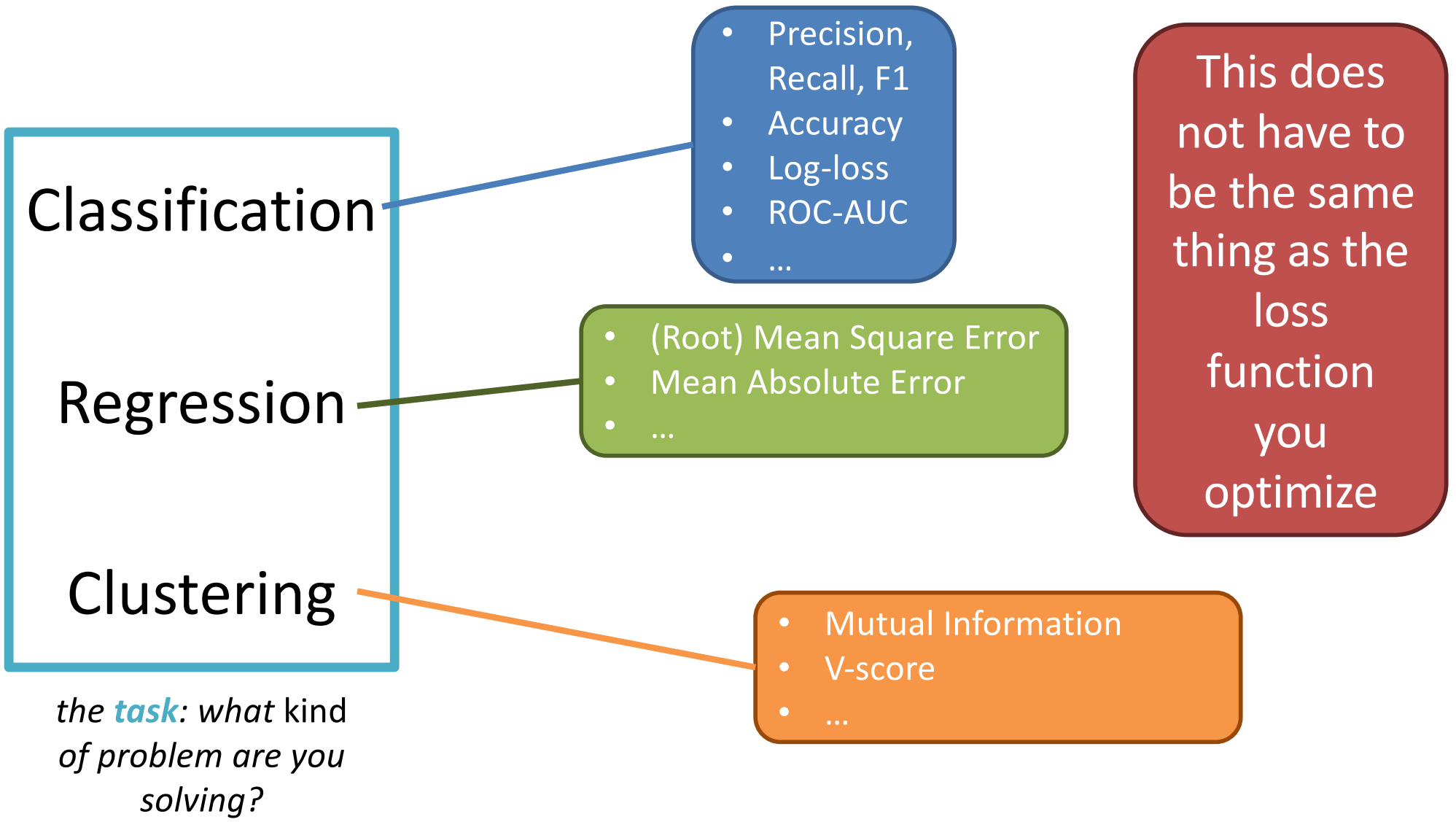
Maximum Entropy Models/ Logistic Regression

CMSC 678

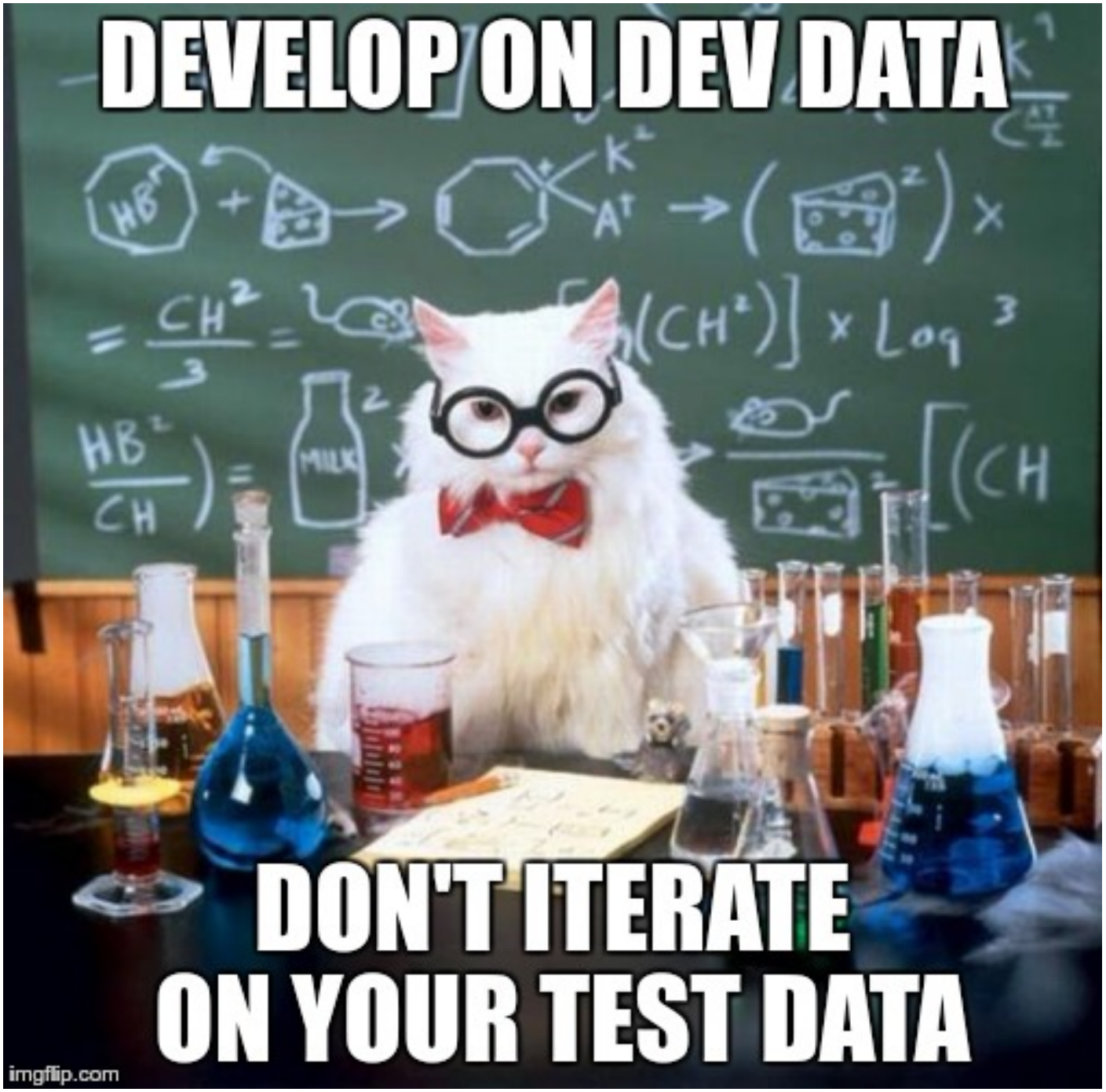
UMBC

Recap from last time…

Central Question: How Well Are We Doing?



Rule #1


DEVELOP ON DEV DATA
DON'T ITERATE
ON YOUR TEST DATA
imgflip.com

We've only developed binary classifiers so far…

Option 1: Develop a multi-class version

Option 2: Build a one-vs-all (OvA) classifier

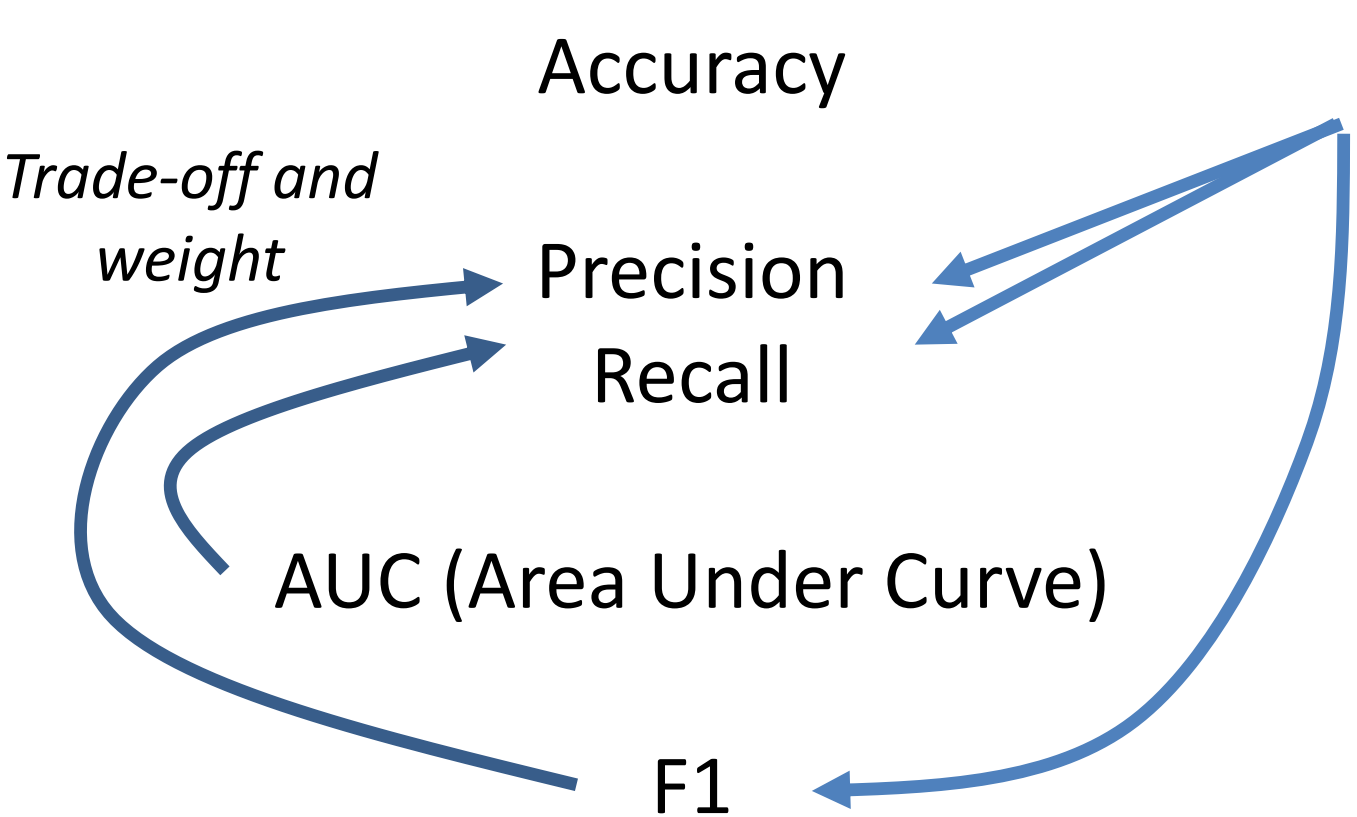
Option 3: Build an all-vs-all (AvA) classifier

(there can be others)

Which option you choose is problem-dependent:

1. Why might you want to use option 1 or options OvA/AvA?
2. What are the benefits of OvA vs. AvA?
3. What if you start with a balanced dataset, e.g., 100 instances per class?

Some Classification Metrics



Confusion Matrix

		Correct Value		
		●	○	▭
Guessed Value	●	#	#	#
	○	#	#	#
	▭	#	#	#

Outline

Log-Linear (Maximum Entropy) Models

Basic Modeling

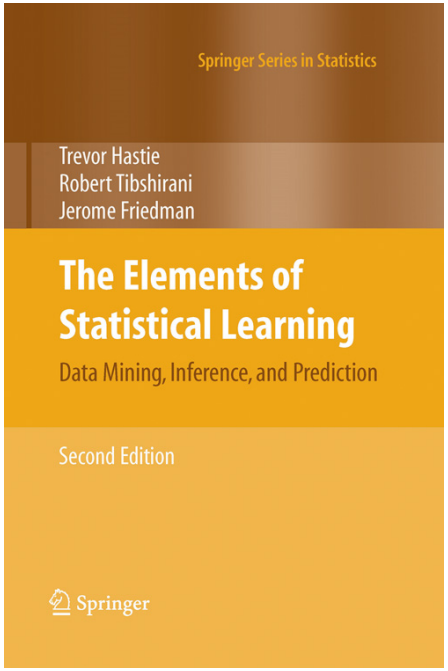
Connections to other techniques *("… by any other name…")*

Objective to optimize

Regularization

Maximum Entropy (Log-linear) Models

$$p(y \mid x) \propto \exp(\theta^T f(x, y))$$

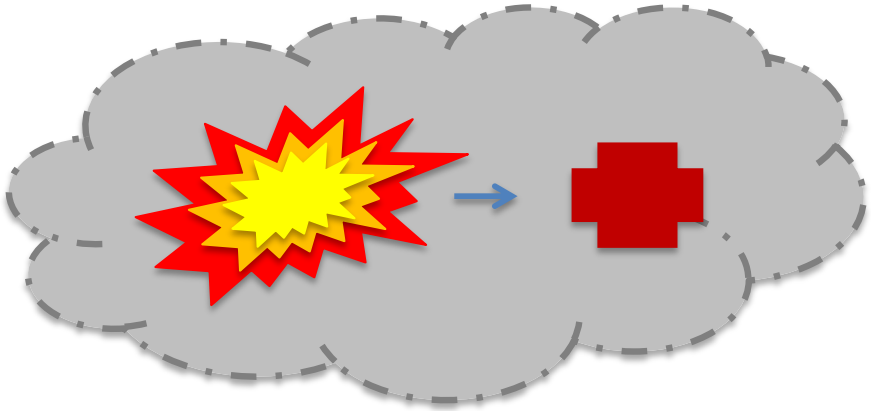


"model the posterior probabilities of the K classes via linear functions in θ, while at the same time ensuring that they sum to one and remain in [0, 1]" *~ Ch 4.4*

Document Classification

Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region.

ATTACK



Observed document

Label

Q: What *features* of this document could indicate an **ATTACK**?

Document Classification

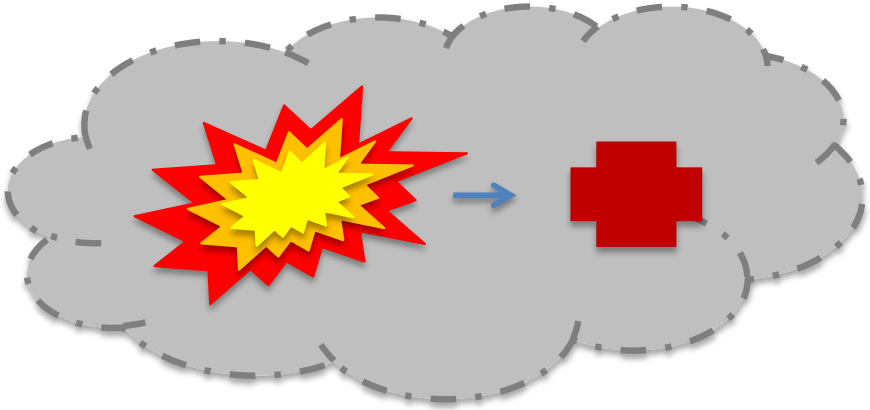
Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region.

attack

ATTACK

- **# killed:**
- **Type:**
- **Perp:**

Document Classification



Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in **Junín** department, central Peruvian mountain region.

ATTACK

there could be many relevant clues

Features

The “clues” that help our system make its decision

Apply a vector of features $f(📄, y) = (f_1(📄, y), \ldots, f_K(📄, y))$ to a given document 📄 and possible label *y*

$f_{\text{fatally shot, ATTACK}}(📄, \text{ATTACK})$

$f_{\text{seriously wounded, ATTACK}}(📄, \text{ATTACK})$

$f_{\text{Shining Path, ATTACK}}(📄, \text{ATTACK})$

$f_{\text{happy cat, ATTACK}}(📄, \text{ATTACK})$

…

Features

The “clues” that help our system make its decision

Apply a vector of features $f(📄, y) = (f_1(📄, y), \ldots, f_K(📄, y))$ to a given document 📄 and possible label *y*

$f_{\text{fatally shot, ATTACK}}(📄, \text{ATTACK})$

$f_{\text{seriously wounded, ATTACK}}(📄, \text{ATTACK})$

$f_{\text{Shining Path, ATTACK}}(📄, \text{ATTACK})$

$f_{\text{happy cat, ATTACK}}(📄, \text{ATTACK})$

…

Each feature function f_k can take any real value:

- binary
- count-based
- likelihood

Features

The "clues" that help our system make its decision

Apply a vector of features $f(📄, y) = (f_1(📄, y), \ldots, f_K(📄, y))$ to a given document 📄 and possible label *y*

$f_{\text{fatally shot, ATTACK}}(📄, \text{ATTACK})$

$f_{\text{seriously wounded, ATTACK}}(📄, \text{ATTACK})$

$f_{\text{Shining Path, ATTACK}}(📄, \text{ATTACK})$

$f_{\text{happy cat, ATTACK}}(📄, \text{ATTACK})$

…

Each feature function f_k can take any real value:

- binary
- count-based
- likelihood

Features that don't "fire" don't apply to the pair

$$f_k(📄, y) = 0$$

Features:
Score and Combine Our Possibilities

define for each key phrase/ clue...

$\theta_{\text{fatally shot, ATTACK}}$(📄, ATTACK)

$\theta_{\text{seriously wounded, ATTACK}}$(📄, ATTACK)

$\theta_{\text{Shining Path, ATTACK}}$(📄, ATTACK)

$\theta_{\text{happy cat, ATTACK}}$(📄, ATTACK)

...

Remember: each $\theta_{w,l}$*(📄,y) is actually computed as* $\theta_{w,l} * f_{w,l}$*(📄,y)*

Features:
Score and Combine Our Possibilities

… and for each label

define for each key phrase/ clue…

$\theta_{\text{fatally shot, ATTACK}}$(📄, ATTACK)

$\theta_{\text{seriously wounded, ATTACK}}$(📄, ATTACK)

$\theta_{\text{Shining Path, ATTACK}}$(📄, ATTACK)

$\theta_{\text{happy cat, ATTACK}}$(📄, ATTACK)

…

$\theta_{\text{fatally shot, TECH}}$(📄, ATTACK)

$\theta_{\text{seriously wounded, TECH}}$(📄, ATTACK)

$\theta_{\text{Shining Path, TECH}}$(📄, ATTACK)

$\theta_{\text{happy cat, TECH}}$(📄, ATTACK)

…

Remember: each $\theta_{w,l}(📄, y)$ *is actually computed as* $\theta_{w,l} * f_{w,l}(📄, y)$

Features:
Score and Combine Our Possibilities

… and for each label

define for each key phrase/ clue…

$\theta_{\text{fatally shot, ATTACK}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{seriously wounded, ATTACK}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{Shining Path, ATTACK}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{happy cat, ATTACK}}(\text{📄}, \text{ATTACK})$

…

$\theta_{\text{fatally shot, TECH}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{seriously wounded, TECH}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{Shining Path, TECH}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{happy cat, TECH}}(\text{📄}, \text{ATTACK})$

…

Remember: each $\theta_{w,l}(\text{📄}, y)$ *is actually computed as* $\theta_{w,l} * f_{w,l}(\text{📄}, y)$

Not all of these will be relevant

Features:
Score and Combine Our Possibilities

… and for each label

define for each key phrase/ clue…

$\theta_{\text{fatally shot, ATTACK}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{seriously wounded, ATTACK}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{Shining Path, ATTACK}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{happy cat, ATTACK}}(\text{📄}, \text{ATTACK})$

…

$\theta_{\text{fatally shot, TECH}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{seriously wounded, TECH}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{Shining Path, TECH}}(\text{📄}, \text{ATTACK})$

$\theta_{\text{happy cat, TECH}}(\text{📄}, \text{ATTACK})$

…

Remember: each $\theta_{w,l}(\text{📄}, y)$ *is actually computed as* $\theta_{w,l} * f_{w,l}(\text{📄}, y)$

Each of these scored features describes how "good" a particular phrase is for a given document type if the provided document document 📄 has a proposed type

Score and Combine Our Possibilities

Shortcut notation: focus only on the features that "fire"

θ_1(fatally shot, ATTACK)

θ_2(seriously wounded, ATTACK)

θ_3(Shining Path, ATTACK)

...

Weight each of these: score how "important" each feature (clue) is

Q: How many features are there?

A: As many as you want there to be (but be careful of underfitting/overfitting)

Score and Combine Our Possibilities

θ_1(fatally shot, ATTACK)

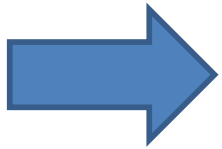
θ_2(seriously wounded, ATTACK)

θ_3(Shining Path, ATTACK)

...

Weight each of these: score how "important" each feature (clue) is

COMBINE



posterior probability of ATTACK

Scoring Our Possibilities

score(Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region . , ATTACK) =

θ_1(fatally shot, ATTACK) ✚

θ_2(seriously wounded, ATTACK) ✚

θ_3(Shining Path, ATTACK) ✚

…

our linear regression model

Maxent Modeling

$p($ ATTACK $|$ Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region . $) \propto$

SNAP(score(Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region . , ATTACK))

What function…

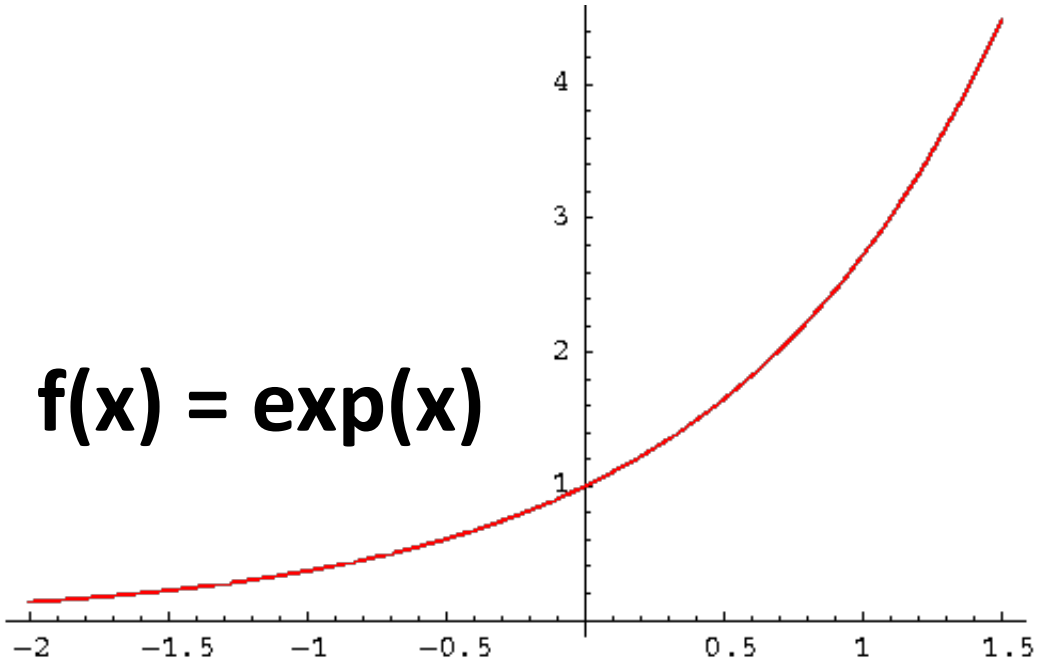
operates on any real number?

is never less than 0?

What function…

operates on any real number?

is never less than 0?



Maxent Modeling

$$p(\text{ATTACK} \mid \text{Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region .}) \propto$$

$$\exp(\text{score}(\text{Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region .}, \text{ATTACK}))$$

Maxent Modeling

$p($ ATTACK $|$ Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region . $) \propto$

$$\exp(\theta_1(\text{fatally shot, ATTACK}) + \theta_2(\text{seriously wounded, ATTACK}) + \theta_3(\text{Shining Path, ATTACK}) + \ldots)$$

this is assuming binary features, but they don't have to be

Maxent Modeling

$p($ ATTACK $|$ Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region . $) \propto$

$$\exp\Big(\begin{array}{l} \text{weight}_1 * f_1(\text{fatally shot, ATTACK}) \\ \text{weight}_2 * f_2(\text{seriously wounded, ATTACK}) \\ \text{weight}_3 * f_3(\text{Shining Path, ATTACK}) \\ \ldots \end{array} \begin{array}{l} + \\ + \\ + \end{array} \Big)$$

Maxent Modeling

$$p(\text{ATTACK} \mid \text{Three people have been fatally shot, and five people, including a mayor, were seriously wounded as a result of a Shining Path attack today against a community in Junin department, central Peruvian mountain region .}) =$$

Q: How do we define Z?

$$\frac{1}{Z} \exp(\ldots \text{tally shot, ATTACK}) + \text{weight}_2 * f_2(\text{seriously wounded, ATTACK}) + \text{weight}_3 * f_3(\text{Shining Path, ATTACK}) + \ldots))$$

Normalization for Classification

$$Z = \sum_{\text{label } y} \exp\Big(\text{weight}_1 * f_1(\text{fatally shot}, Y) + \text{weight}_2 * f_2(\text{seriously wounded}, Y) + \text{weight}_3 * f_3(\text{Shining Path}, Y) + \ldots \Big)$$

Q: What if none of our features apply?

Guiding Principle for Maximum Entropy Models

"[The log-linear estimate] is the least biased estimate possible on the given information; i.e., it is ***maximally noncommittal with regard to missing information****."*

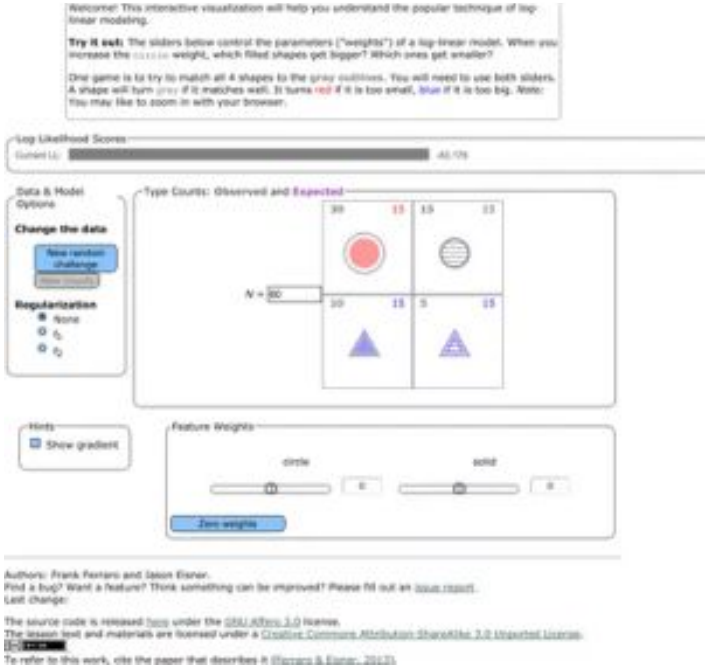
Edwin T. Jaynes, 1957

$\exp(\theta \cdot f)$ ➔

$\exp(\theta \cdot 0) = 1$

https://www.csee.umbc.edu/courses/graduate/678/spring19/loglin-tutorial

Lesson 1: Basic Feature Design



Ingredients for classification

Inject *your* knowledge into a learning system

Feature representation

Training data:
labeled examples

Model

Courtesy Hamed Pirsiavash

Ingredients for classification

Inject *your* knowledge into a learning system

Problem specific

Difficult to learn from bad ones

Feature representation

Training data: labeled examples

Model

Courtesy Hamed Pirsiavash

What features would you extract to…

distinguish a picture of me from a picture of someone else?

determine whether a sentence is grammatical or not?

distinguish cancerous cells from normal cells?

Courtesy Hamed Pirsiavash

Outline

Log-Linear (Maximum Entropy) Models

Basic Modeling

Connections to other techniques *("… by any other name…")*

Objective to optimize

Regularization

Connections to Other Techniques

Log-Linear Models

Connections to Other Techniques

Log-Linear Models

as statistical regression

(Multinomial) logistic regression

Softmax regression

"Solution" 1: A Simple Probabilistic (Linear*) Classifier

decision rule:

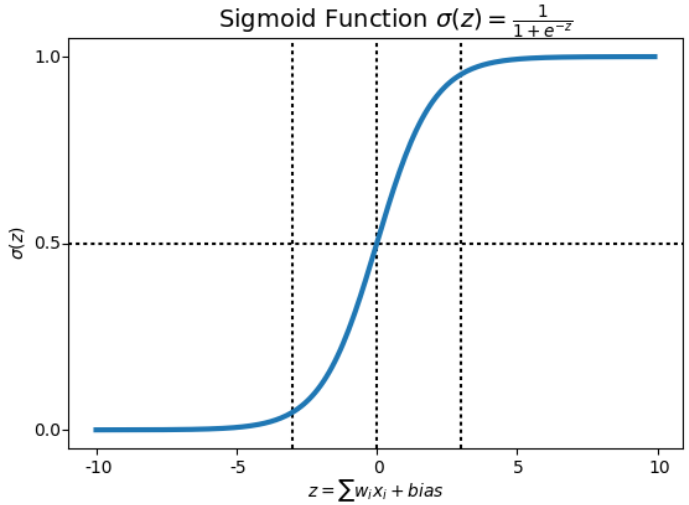
$$\hat{y}_i = \begin{cases} 0, & \sigma(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) < .5 \\ 1, & \sigma(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) \geq .5 \end{cases}$$

turn responses into probabilities

Remember from "Linear regression"

loss f... $< 0]$

minimize posterior 0-1 loss:



$$\min_{\mathbf{w}} \sum_i \mathbb{E}_{\hat{y}_i}[1[y_i p(\hat{y}_i = 1|x_i) < 0]] =$$

$$\max_{\mathbf{w}} \sum_i p(\hat{y}_i = y_i|x_i)$$

why MAP classifiers are reasonable

Plot from https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f

*linear not strictly required

Connections to Other Techniques

Log-Linear Models

as statistical regression

(Multinomial) logistic regression

Softmax regression

based in information theory

Maximum Entropy models (MaxEnt)

Connections to Other Techniques

Log-Linear Models

as statistical regression: (Multinomial) logistic regression

as statistical regression: Softmax regression

based in information theory: Maximum Entropy models (MaxEnt)

a form of: Generalized Linear Models

Generalized Linear Models

$$y = \sum_k \theta_k x_k + b$$

response *linear* wrt parameters*

the *response* can be a general (transformed) version of another response

**affine is okay*

Generalized Linear Models

$$y = \sum_k \theta_k x_k + b$$

response *linear* wrt parameters*

the *response* can be a general (transformed) version of another response

logistic regression $$\frac{\log p(x = i)}{\log p(x = K)} = \sum_k \theta_k f(x_k, i) + b$$

**affine is okay*

Connections to Other Techniques

Log-Linear Models

as statistical regression: (Multinomial) logistic regression

as statistical regression: Softmax regression

based in information theory: Maximum Entropy models (MaxEnt)

a form of: Generalized Linear Models

viewed as: Discriminative Naïve Bayes

Connections to Other Techniques

Log-Linear Models

as statistical regression	(Multinomial) logistic regression
	Softmax regression
based in information theory	Maximum Entropy models (MaxEnt)
a form of	Generalized Linear Models
viewed as	Discriminative Naïve Bayes
to be cool today :)	Very shallow (sigmoidal) neural nets

Outline

Log-Linear (Maximum Entropy) Models

Basic Modeling

Connections to other techniques *("… by any other name…")*

Objective to optimize

Regularization

Version 1: Minimize Cross Entropy Loss

loss uses y (random variable), or model's probabilities $\ell^{\text{xent}}(\overrightarrow{y^*}, p(y))$

$$\ell^{\text{xent}}(\overrightarrow{y^*}, y) = -\sum_k \overrightarrow{y^*}[k] \log p(y = k)$$

index of "1" indicates correct value

$$\begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{pmatrix}$$

one-hot vector

minimize xent loss → maximize log-likelihood (A2, Q2)

objective is convex

Version 2: Maximize (Full/Log) Likelihood

$$\prod_i p_\theta(y_i|x_i) \propto \prod_i \exp(\theta^T f(x_i, y_i))$$

These values can have very small magnitude ➔ underflow

Differentiating this product could be a pain

Version 2: Maximize Log-Likelihood

Wide range of (negative) numbers

Sums are more stable

$$\log \prod_i p_\theta(y_i|x_i) = \sum_i \log p_\theta(y_i|x_i)$$

Version 2: Maximize Log-Likelihood

Wide range of (negative) numbers

Sums are more stable

$$\log \prod_i p_\theta(y_i|x_i) = \sum_i \log p_\theta(y_i|x_i)$$

$$= \sum_i \theta^T f(x_i, y_i) - \log Z(x_i)$$

Differentiating this becomes nicer (even though Z depends on θ)

Log-Likelihood Gradient

Each component *k* is the difference between:

Log-Likelihood Gradient

Each component *k* is the difference between:

the total value of feature f_k in the training data

$$\sum_i f_k(x_i, y_i)$$

Log-Likelihood Gradient

Each component *k* is the difference between:

the total value of feature f_k in the training data

$$\sum_i f_k(x_i, y_i)$$

and

the total value the current model p_θ *thinks* it computes for feature f_k

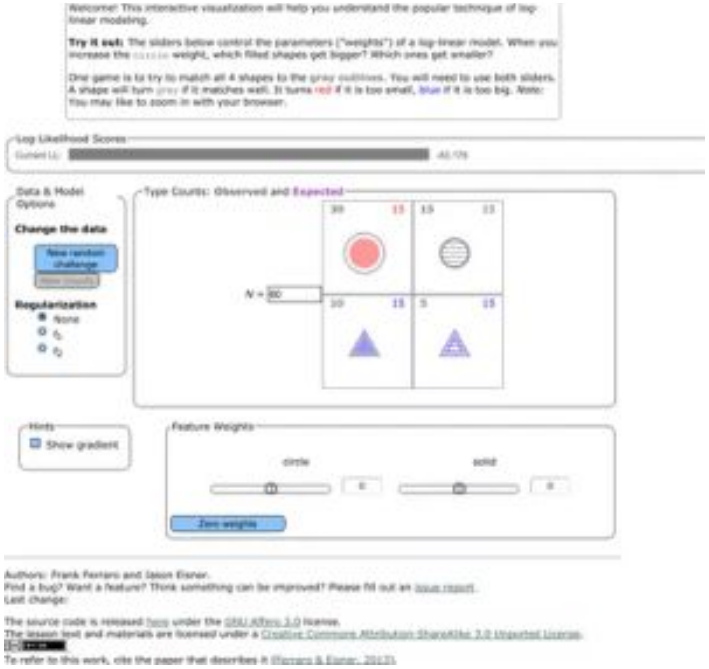
$$\sum_i \mathbb{E}_p[f(x_i, y')$$

"Moment Matching"

A1 Q4, Eq-1 (what were the feature functions)?

https://www.csee.umbc.edu/courses/graduate/678/spring19/loglin-tutorial

Lesson 6: Gradient Optimization



Log-Likelihood Gradient Derivation

$$\nabla_\theta F(\theta) = \nabla_\theta \sum_i [\theta^T f(x_i, y_i) - \log Z(x_i)]$$

Log-Likelihood Gradient Derivation

$$\nabla_\theta F(\theta) = \nabla_\theta \sum_i [\theta^T f(x_i, y_i) - \log Z(x_i)]$$

$$= \nabla_\theta \sum_i f(x_i, y_i) -$$

$$Z(x_i) = \sum_{y'} \exp(\theta \cdot f(x_i, y'))$$

Log-Likelihood Gradient Derivation

$$\nabla_\theta F(\theta) = \nabla_\theta \sum_i [\theta^T f(x_i, y_i) - \log Z(x_i)]$$

$$= \nabla_\theta \sum_i f(x_i, y_i) - \sum_i \sum_{y'} \frac{\exp\left(\theta^T f(x_i, y')\right)}{Z(x_i)} f(x_i, y')$$

scalar p(y' | x_i)

vector of functions

use the (calculus) chain rule

$$\frac{\partial}{\partial \theta} \log g(h(\theta)) = \left(\frac{\partial g}{\partial h(\theta)}\right)\left(\frac{\partial h}{\partial \theta}\right)$$

Log-Likelihood Gradient Derivation

$$\nabla_\theta F(\theta) = \nabla_\theta \sum_i [\theta^T f(x_i, y_i) - \log Z(x_i)]$$

$$= \nabla_\theta \sum_i f(x_i, y_i) - \sum_i \sum_{y'} \frac{\exp\left(\theta^T f(x_i, y')\right)}{Z(x_i)} f(x_i, y')$$

Do we want these to *fully* match?

What does it mean if they do?

What if we have missing values in our data?

Outline

Log-Linear (Maximum Entropy) Models

Basic Modeling

Connections to other techniques *("… by any other name…")*

Objective to optimize

Regularization

Weight regularization R(**w**)

Nice if *R(**w**)* is convex

Small weights regularization

$$R^{(\text{norm})}(\mathbf{w}) = \sqrt{\sum_d w_d^2} \qquad R^{(\text{sqrd})}(\mathbf{w}) = \sum_d w_d^2$$

Sparsity regularization

$$R^{(\text{count})}(\mathbf{w}) = \sum_d \mathbf{1}[|w_d| > 0] \quad \textit{not convex}$$

Family of "p-norm" regularization

$$R^{(\text{p-norm})}(\mathbf{w}) = \left(\sum_d |w_d|^p \right)^{1/p}$$

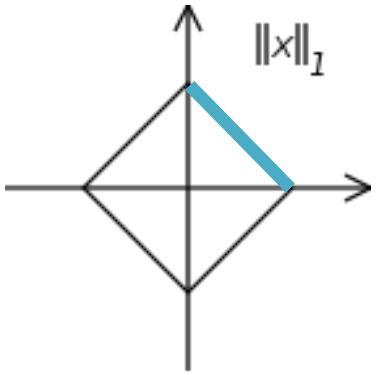
convex: $p \geq 1$

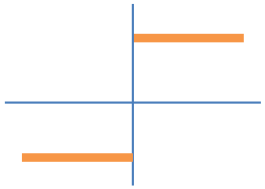
not convex: $0 \leq p < 1$

Courtesy Hamed Pirsiavash

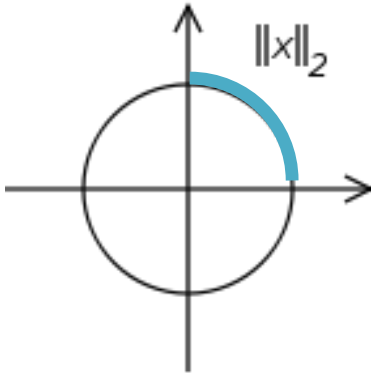
Contours of p-norms

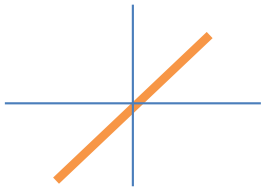
$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$



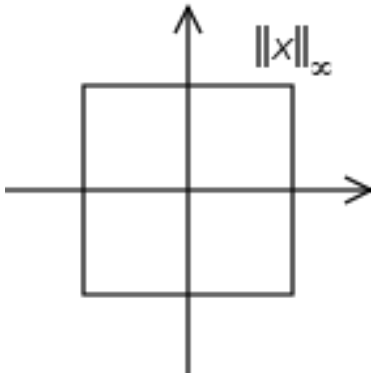


$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$$





$$\|x\|_\infty = \max_{i=1,\ldots,n} |x_i|$$



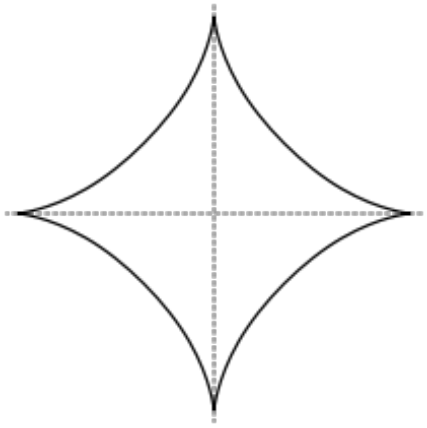
examine shape (slope) of surfaces to determine effect on the regularized parameters

Courtesy Hamed Pirsiavash

http://en.wikipedia.org/wiki/Lp_space

Contours of p-norms

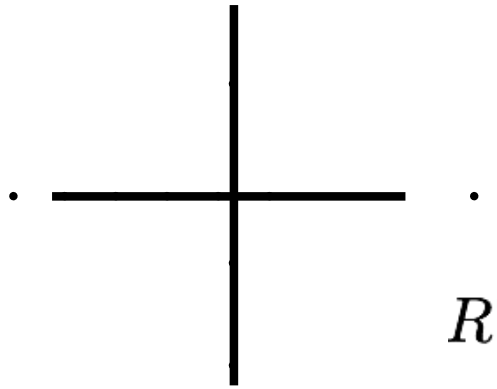
$$p = \frac{2}{3}$$



examine shape (slope) of surfaces to determine effect on the regularized parameters

Counting non-zeros:

$$p = 0$$



$$R^{(\text{count})}(\mathbf{w}) = \sum_d \mathbf{1}[|w_d| > 0]$$

Courtesy Hamed Pirsiavash

http://en.wikipedia.org/wiki/Lp_space

A Simple Regularized Linear Classifier

decision rule: $\hat{y}_i = \begin{cases} 0, & \mathbf{w^T x_i} < 0 \\ 1, & \mathbf{w^T x_i} \geq 0 \end{cases}$

loss function: $\ell = 1[y_i \mathbf{w^T x_i} < 0]$

hyperparameter

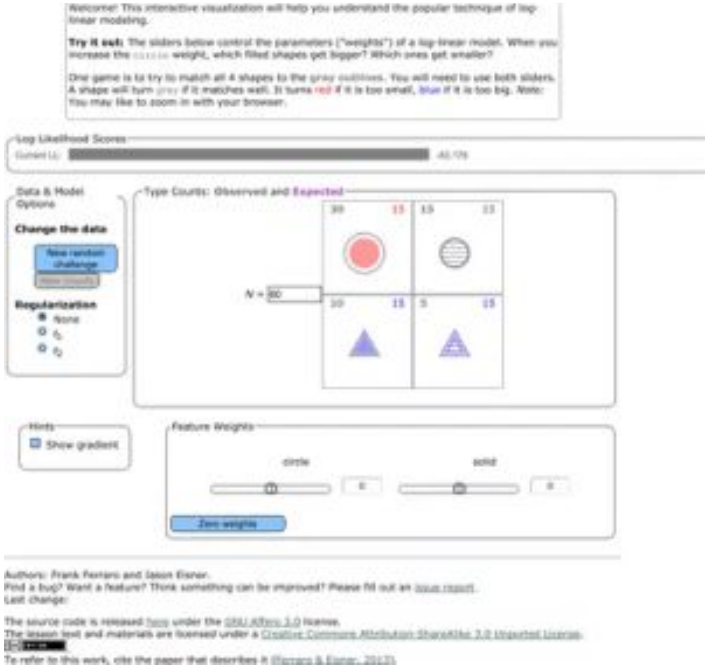
$$\min_{\mathbf{w}} \sum_n \mathbf{1}[y_n \mathbf{w}^T \mathbf{x}_n < 0] + \lambda R(\mathbf{w})$$

fewest mistakes on training

regularize toward a simpler model

https://www.csee.umbc.edu/courses/graduate/678/spring19/loglin-tutorial

Lesson 8: Regularization



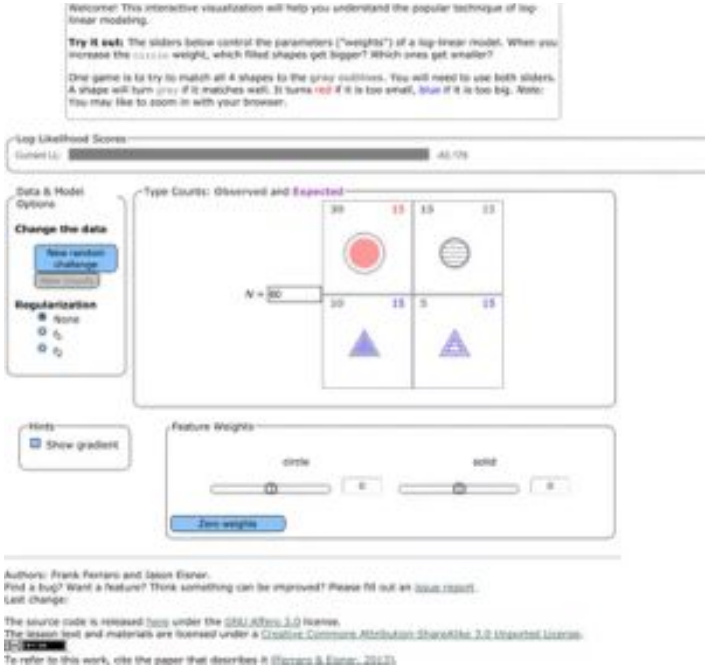
Understanding Conditioning

$$p(y \mid x) \propto \exp(\theta \cdot f(\mathrm{x}))$$

Is this a good posterior classifier? (no)

https://www.csee.umbc.edu/courses/graduate/678/spring19/loglin-tutorial

Lesson 11: Global vs. Conditional Modeling



Connections to Other Techniques

Log-Linear Models

as statistical regression	(Multinomial) logistic regression
	Softmax regression
based in information theory	Maximum Entropy models (MaxEnt)
a form of	Generalized Linear Models
viewed as	Discriminative Naïve Bayes
to be cool today :)	Very shallow (sigmoidal) neural nets

Outline

Log-Linear (Maximum Entropy) Models

Basic Modeling

Connections to other techniques *("… by any other name…")*

Objective to optimize

Regularization